Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P); 2583-7907 (E)
Vol.12 No.2, 2023, pp.1-17
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajcst-2023.12.2.3697

Automation of Customer Support System (Chatbot) to Solve Web Based Financial and Payment Application Service

Roqib Akintunde Akinyemi[1], Wumi Ajayi[2] and Ayuba Atuman[3]
[1&3]Department of Computer Science, Lead City University, Ibadan, Nigeria
[2]Department of Computer Science, Babcock University, Ikenne, Nigeria
E-mail: roqibakinyemi2010@gmail.com, ajayiw@babcock.edu.ng, atumanson12@live.com

(Received 10 July 2023; Revised 20 July 2023; Accepted 17 August 2023; Available online 25 August 2023)

Abstract - **One of the most important features of any online service is the quality of its customer care. However, with the development of NLP tools, businesses are considering automated chatbot solutions to keep up with the increasing demand for their products and services. In view of this, the chatbot was developed using AIML java interpreter library Program AB which helps match input and output predefined in the AIML file. AIML (Artificial Intelligence Markup Language) was used to preprocess and train the bot using ready-made AIML file for FAQ questions. Also, vaadin was used to build a web UI to interact with the trained AIML bot. Finally, a google script was written to translate from any language to English for the bot to understand and send the response in the preferred language of the user. Findings showed that the response time of the bot is dependent of the network, as the design gave a score of 70%, 80%, 90% and 90% for load testing, stability, reliability testing and usability testing, respectively. Also, the bot is compatible with different operating systems, both for forward compatibility and backward compatibility having a score of 95%. The bot was able to answer customer questions, enquiries and complaints and the response time of the bot depends on the strength of the network since it is web based. Hence, the system provided a simple, cheaper, and durable customer financial and payment application service. Since chatbots cannot answer all questions, businesses that decide to implement them should ensure that they have enough protections in place against attacks and that routine requests are standardised to ensure optimal performance.**

Keywords: **Chatbot, Customer Support, Google Script, Online Service, Testing**

I. INTRODUCTION

Customer service is crucial for the user experience, particularly in the online context, where users have high expectations for prompt responses and high-quality assistance. Customer support provides assistance to users regarding product issues and is occasionally necessary for the validation of an account. Existing tools like Zendesk, Freshdesk, and Happyfox are available for organizing and structuring customer support [1].

These tools offer various features such as support level definition, agent performance evaluation, and communication channel management, among others, which are highly functional. However, automation remains a significant area where these tools are insufficient. Users frequently submit multiple tickets, each with a unique problem description, word choice, and writing style. There are a small number of common issues that account for most of these tickets, which can be fixed in a repeatable manner [1].

The field of computer science has made significant progress in hardware, software, natural language processing, and machine learning techniques. The advancement of emerging technologies has led to the evolution of chatbots from generating machine-like responses to developing human-like agents capable of forming long-term relationships with users. Y. Yun (2016) identified ELIZA and PARRY as prominent early chatbot implementations [2]. A. A. A. Weißensteiner (2018) identified Siri, Alexa, and XiaoIce as contemporary chatbots. Chatbots have numerous potential advantages [3]. They offer prompt, reliable, and consistent responses and solutions. Chatbots possess valuable attributes that render them highly beneficial across various domains, including business, e-commerce, and healthcare, leading to their widespread adoption [4].

A chatbot is an interactive virtual assistant that can hold conversations in natural language via text [5]. Chatbots simulate human-like conversations by using natural language input to interact with computer software. The importance of simulating real individuals for chatbot acceptance is evident, highlighting the crucial role of a comprehensive knowledge base. This knowledge base refers to the pre-existing set of regulations that a chatbot possesses [6]. There are three main parts to a chatbot: the interface, which allows the chatbot to communicate with users via messaging apps or website chat sessions; the intelligence, which allows the chatbot to understand and respond to questions while gaining knowledge from each interaction; and the integration, which focuses on incorporating the chatbot into existing systems and platforms [4]. Furthermore, chatbots possess both input and output capabilities. Inputs refer to the process of recording and entering data into a system, as well as providing instructions to the system. For a chatbot to operate effectively, the user's input must align with predetermined inputs to ensure accurate interpretation by the computer system.

Chatbots can be categorized based on several factors such as the type of service offered, the domain of knowledge, the approach used to generate responses, the objective, the level of permission, and the extent of human assistance [2] [4]. The level of personalization, the nature of the conversation, and the function served are just a few of the criteria that can be used to categorise chatbots. While they don't pretend to be human friends with their users, interpersonal chatbots provide useful features like making reservations and searching for frequently asked questions. Intrapersonal chatbots function within the user's personal realm and aim to comprehend the user's intentions and needs in a manner similar to human understanding [4], [6]. Chatbots that interact with other chatbots to complete a task are called "inter-agents." The chatbot's knowledge domain is the body of information it was instructed to use effectively in a conversation. While closed-domain chatbots are only able to answer queries within their own domain, open-domain chatbots can answer inquiries from a variety of other domains [7]. Response generation method classification considers the approach used for generating responses. Chatbots can be categorized into three types: rule-based, retrieval-based, or generative-based [8].

Classification based on goals involves categorizing chatbots according to their main purpose. Chatbots with a fixed source store and provide specific information to users. Chat-based or conversational chatbots are programmed to engage in conversations with users using a conversational style that closely resembles human interaction. Task-based chatbots perform specific tasks in response to user requests [4]. Additionally, chatbots can be classified in accordance with the permissions provided by the respective development platforms. Open-source platforms provide developers with access to the source code, enabling them to have full control over most implementation details [4]. Closed platforms consist exclusively of proprietary code and are commonly provided by large enterprises. Chatbots can be categorized into two types: human-mediated and fully autonomous, which is determined by the level of human involvement. A human-mediated chatbot typically includes a component that involves human computation. Automated chatbots excel in rapid information processing, but human computation offers flexibility and can address gaps resulting from algorithms.

II. STATEMENT OF THE PROBLEM

The growing number of users of web-based financial and payment application services has led to various issues such as long queues, service malfunctions (e.g., traffic, server errors, payment errors), and system overrides. Customers preferred remote resolution of transaction issues, inquiries, and complaints in their preferred language. Several chatbots have been developed for various purposes, such as financial transactions [1], [9], [10]. To the researcher's knowledge, there is a lack of exploration in the existing literature regarding the development of chatbots that can translate from any language to English and respond in the user's preferred language, particularly in the financial sector.

This paper explores the use of Artificial Intelligence (AI) in an intelligent personalized chatbot system to automate customer support tasks, such as providing support, personalized assistance, and remote inquiries in multiple languages using frequently asked questions (FAQs). By doing so, the author aims to alleviate the workload of support service agents.

III. AIM AND OBJECTIVES OF THE STUDY

This study aims to design a chatbot to solve web-based financial and payment application services. Specific objectives are to

1. Create a web page on which the chatbot was deployed using Java interpreter and spring boot project using Spring initializr.
2. Implement the design in (i), the Bot logic and build a web UI to interact with the AIML bot using vaadin.
3. Evaluate and test the performance of the designed chatbot using accuracy, scalability, and functionality.

IV. LITERATURE REVIEW

A. Conceptual Review

Chatbots simulate human-like conversations by utilizing natural language input to interact with computer software [1]. Some define it as a computer software that employs artificial intelligence to simulate human conversation. A chatbot is software that enables text-based communication using natural language [5]. Simulating real individuals is crucial for the acceptance of chatbots by users, highlighting the importance of a comprehensive knowledge base, which consists of the predefined rules that a chatbot possesses. As advertising efforts continue to proliferate across various platforms, including online and offline channels, reaching and engaging customers has become a challenging task. Chatbots have the potential to emerge as a leading method for businesses to engage with individual customers, thereby influencing a company's competitive position. The growing popularity of chatbots can be attributed to the increased use of messaging services and advancements in Artificial Intelligence [10].

V. USE OF CHATBOT

A. Customer Service

Chatbots are revolutionizing the post-sales experience of customers, thereby transforming customer care [8]. Although chatbots cannot handle all types of queries, they can effectively handle routine queries, which constitute most service requests [3], [11]. The chatbot requires approximately five weeks of training to produce high-quality output. In addition, chatbots facilitate convenient and efficient customer support by leveraging existing messaging systems commonly utilized by clients on a regular basis. Consequently, numerous users will find this mode of interaction to be comfortable, thereby enhancing convenience. Chatbots provide businesses with valuable insights into their digital customers and ensure they stay informed about emerging technologies.

In addition to their widespread use, a specific virtual agent named CLARA was developed for the purpose of tourism [12]. CLARA is not only a conference information system, but also a local tour guide. This multifunctionality offers valuable information and assistance to first-time visitors [8]. There are three main parts to the system's architecture: the client system, the web socket server, and multiple information resources. The user inputs a query through the graphical interface of the application. The system then produces a JSON message containing the query, the domain (which determines if the information is related to conferences or tourism), and optionally, the GPS coordinates if the user allows the system to access them. The algorithm searches the index for generic queries, such as "Elizabeth's," and retrieves the most similar examples. If the index does not provide an answer, the system will search for answers using either the conference search engine or the tourist search engine.

B. Tourism and Hospitality

The rise of digitalization has led many sectors to prioritize online applications and the internet. According to [13], the hotel industry reports that 75% of bookings are made through online channels, leading to a higher demand for online information provision. The integration of a hotel chatbot can assess a hotel's level of competitiveness within the market. The primary objective of a hotel chatbot is to fulfill specific marketing goals, such as revenue growth. The knowledge base of the system is limited to hotel information and tourist-related matters. It offers users details about the hotel, its surroundings, and commonly asked tourist questions. The implementation of a chatbot on a hotel's website produced notable outcomes. According to A. A. A. Weißensteiner, (2018), the average duration of chatbot conversations was 4.2 seconds, with a maximum duration of 118 seconds. Additionally, 36% of interactions consisted of only one user input, while 60% had no more than two user inputs. Hotel chatbots have shorter conversations compared to general chatbots. This is due to their limited knowledge base and the absence of small talk. Additionally, it is argued that a hotel chatbot should be considered as a supplementary feature to a hotel's website, rather than a standalone tool exclusively used for making room reservations. A. A. A. Weißensteiner (2018), found that 56% of discussions focused on inquiries about the hotel or its offerings, while 12% of discussions centered around inquiries about the chatbot [3].

C. History of Chatbot

1. ELIZA: ELIZA is considered the pioneering chatbot. Joseph Weizenbaum developed the program in 1966, which employs pattern matching and substitution techniques to simulate conversation [14]. The program was designed to simulate human conversation. ELIZA, the Chatbot, functioned by inputting user-entered words into a computer and matching them with a predetermined set of scripted responses [3]. The program employs a script that simulates the role of a psychotherapist. The script had a significant impact on natural language processing and artificial intelligence, leading to its widespread adoption in academic institutions across the country.

The primary objective of the initial chatbot development was to provide amusement and user interaction. Users often develop strong emotional attachments to computers when they interact with them as psychotherapists. This phenomenon, known as the ELIZA-effect, involves attributing human-like intelligence to chatbots. ELIZA utilizes a keyword matching approach, whereby the program examines input for matching keywords. The system generates an answer based on specified rules if matching keywords are found; otherwise, a related remark is retrieved [15]. ELIZA's understanding of the user's difficulties is limited as she primarily relies on matching their responses to pre-programmed responses.

2. PARRY: PARRY, developed in 1972 by American psychiatrist Kenneth Colby, is the subject of study [16]. The program simulated a patient exhibiting symptoms of schizophrenia. It aims to replicate the disease. The program is designed to simulate human thinking and behavior. PARRY operates through a complex framework involving assumptions, attributions, and "emotional responses" that are activated by modifying the weights assigned to verbal inputs [17]. To assess the validity of PARRY's performance, a modified version of the Turing test was employed.

3. ALICE: In 1995, the Artificial Linguistic Internet Computer Entity (ALICE) was created and placed into use. In contrast to ELIZA's crude keyword matching technique, ALICE uses AIML files, which are based on Artificial Intelligence Markup Language, to store its knowledge of English conversation patterns. AIML objects, found in the corresponding files, are made up of various classes and topics. AIML topics are identified by name and linked to additional classifications. In contrast, AIML views categories as its primary cognitive building block. Each category serves as a guideline for aligning the user's input with the desired outcome, as well as a pattern and template [3].

4. Siri: Siri, developed by Apple in 2010 for iOS, is an intelligent personal assistant and learning navigator that utilizes a natural language user interface [17]. It established the foundation for subsequent AI bots and personal assistants. The United States Patent and Trademark Office has filed a patent application outlining a novel Apple service that enables users to engage in inquiries and conversations with Siri via the Messages platform. Apple's patent outlines a Siri feature that enables users to interact with it without the need for verbal communication, similar to texting or using Facebook Messenger. This could be beneficial in various public domains. The user has the ability to respond to various forms of media, such as text, audio, images, and video, when these are shared with them. This would enhance the quality of interaction between a consumer and a digital assistant. The patent includes instances of a dialogue between Siri and a user within the Messages app, where the user poses inquiries.

VI. RECURRENT NEURAL NETWORKS RNNS

Recurrent Neural Networks (RNNs) are a specialized type of neural networks that are specifically designed for processing data sequences. These networks are essentially characterized by the presence of loops [1], [18]. Neurons possess an internal state, often denoted as Ct, which is transmitted back into the neuron during the next time step. The neuron generates a value, denoted as ht, at each timestep. However, a major concern with naive recurrent neural network (RNN) implementations is their susceptibility to the problems of vanishing and exploding gradients [1]. Multiple variations of Recurrent Neural Networks (RNNs) have been proposed as potential solutions to address this issue, utilizing diverse methodologies. Long-Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that has been widely used in various applications. LSTM units are the fundamental LSTMs, a common choice in Natural Language Processing, and are frequently employed [19]. They have achieved promising outcomes in the fields of machine translation and image captioning. Figure 1 depicts an unrolled LSTM layer. This section provides an explanation of the key components of an LSTM unit, namely the forget gate, input gate, cell state, and output gate. The input sequence x, represented as x = (x0, x1, · · · , xτ) T, and the hidden state h, represented as h = (h0, h1, · · · , hτ)T, produced by the LSTM layer, are related according to the following equations [1].

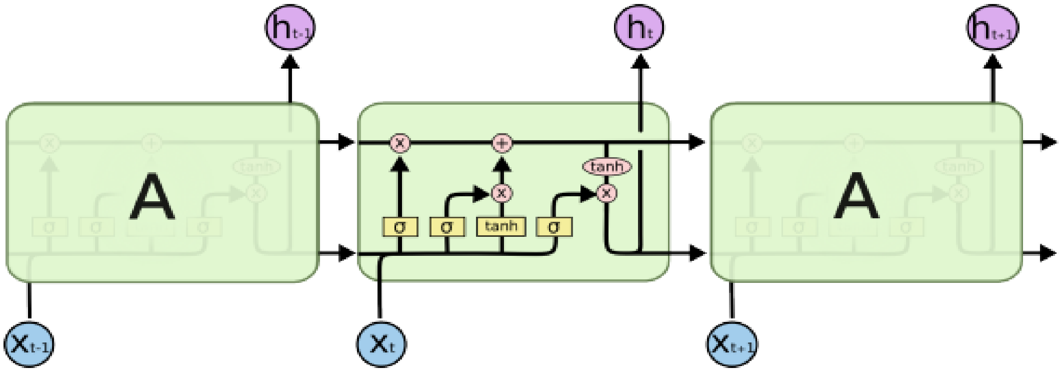


Fig. 1 Unrolled LSTM layer made of 3 Units [2]

$$f_t = \sigma(W_f \cdot[h_t{-}1, x_t] + b_f) \quad (1)$$
$$i_t = \sigma(W_i \cdot[h_t{-}1, x_t] + b_i) \quad (2)$$
$$C_t = f_tC_t{-}1 + i_t \tanh(W_C \cdot[h_t{-}1, x_t] + b_C) \quad (3)$$
$$o_t = \sigma(Wo \cdot [h_t{-}1, x_t] + b_o) \quad (4)$$
$$h_t = o_t \tanh(C_t) \quad (5)$$

Equation 1 is the forget gate's output. Intuitively, it represents which information the unit will keep or forget from the previous cell state C_{t-1} as it is directly multiplied with it. Figure 2 illustrates this process [1]. This equation calculates the forget gate (ft), which determines how much of the previous cell state (Ct-1) to retain and how much to forget. The forget gate takes the previous hidden state (ht-1) and the current input (xt) as inputs and applies a sigmoid activation function (σ) to produce values between 0 and 1.

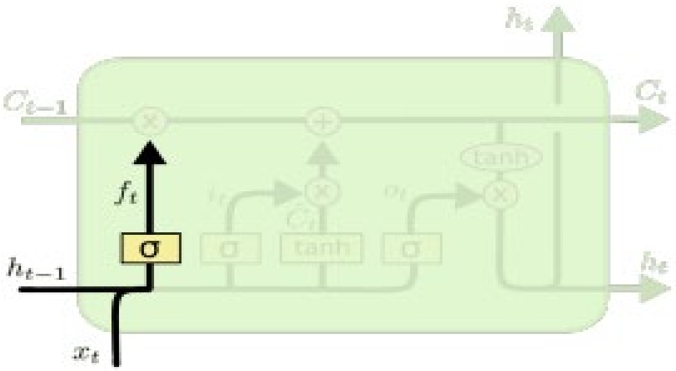


Fig. 2 LSTM Unit's Forget Gate [2]

Equation 2 represents the response of the input gate. The cell state is updated by introducing a new candidate value, C t , which is scaled by the input gate, as illustrated in Figure 3 [1]. This equation is used to compute the input gate (it), which determines the amount of new information to be incorporated into the cell state. The input gate, akin to the forget gate, receives the previous hidden state (ht-1) and the current input (xt) as inputs, which are then processed through a sigmoid activation function.

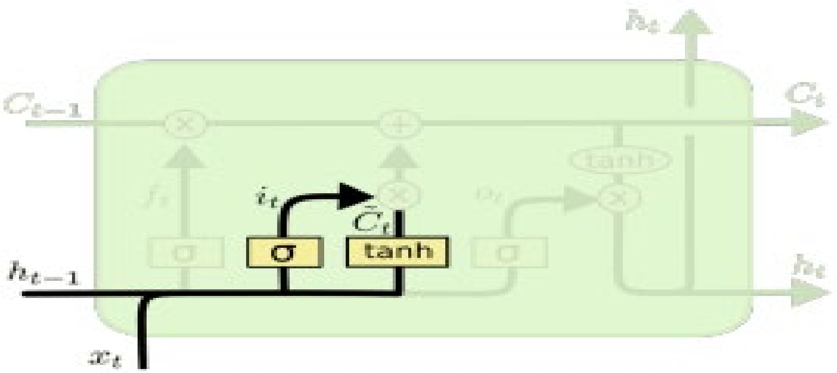


Fig. 3 LSTM Unit’s Input Gate [2]

The cell’s state is subsequently altered based on equation 3. It integrates the impacts of the input and forget gates. Figure displays the relevant relationships for this upgrade, which are indicated by underlining.

After updating the cell state, the activation of the output gate can be calculated using equation 4 to generate the hidden state ht of the unit. Figure 5 illustrates this.

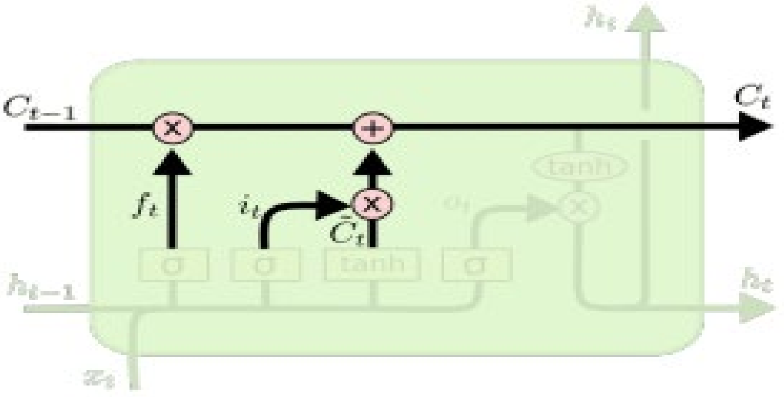


Fig. 4 LSTM Unit’s Cell State Update [2]

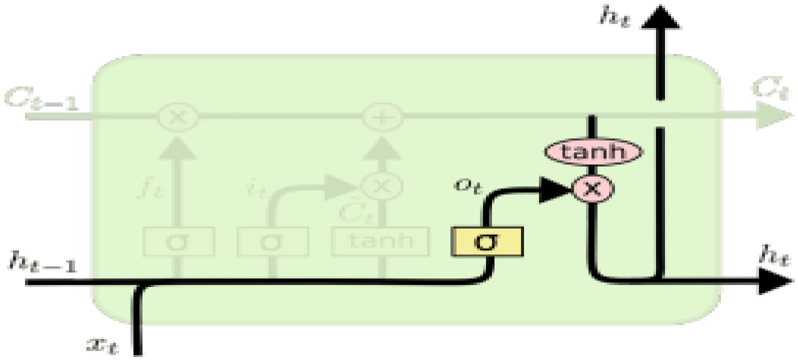


Fig. 5 LSTM Unit’s Output Gate [2]

In equations, slows the modification of the cell state (Ct) by incorporating both the previous cell state (Ct-1) and the new information (it) derived from the input gate. The utilization of a hyperbolic tangent (tanh) activation function is employed to introduce non-linearity to the merged input. In equation 4 the output gate (ot) is computed, which governs the proportion of the updated cell state (Ct) that is to be emitted as the hidden state (ht). The output gate receives the previous hidden state (ht-1) and the current input (xt) as inputs. It applies a sigmoid activation function. The last equation calculates (equation 5) the new hidden state (ht) by applying the output gate (ot) to the updated cell state (Ct) after passing it through a tanh activation function.

The utilization of the LSTM model in chatbot development allows for the processing of sequential input data, such as a conversation's word sequence, while preserving contextual information throughout the conversation. Understanding the context of user queries, generating appropriate responses, and managing complex conversational interactions are essential tasks that require careful attention. The chatbot's capacity to retain and modify information in the cell state (Ct) enables it to acquire pertinent information from previous messages and utilize it to produce responses that are more contextually appropriate and coherent. The LSTM model possesses significant capabilities in constructing chatbots that can participate in conversations that are more authentic and closely resemble human interactions.

A. Gated Recurrent Units

GRUs are a special kind of LSTMs that incorporate the cell and hidden state updates into the forget gate. GRUs require less time to train since they have fewer parameters than LSTMs [20]. [1] has demonstrated that their efficiency is on par with that of standard LSTM networks. The equations for GRUs are provided below

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (6)$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (7)$$
$$h^{\sim}{}_t = \tanh(W \cdot [r_t h_{t-1}, x_t]) \quad (8)$$
$$h_t = (1 - z_t)h_{t-1} + z_t h^{\sim}{}_t \quad (9)$$

The variables zt and rt represent the inner gates of the GRU, as depicted in Figure 6. The chatbot development process involved considering both LSTM and GRU models [21].
Equations 6-9 form an integral component of the Gated Recurrent Unit (GRU) model, an instance of a recurrent neural network (RNN) employed in the domain of natural language processing (NLP). This model finds application in various NLP tasks, such as the development of chatbots. The GRU model is a specific type of the Long Short-Term Memory (LSTM) model that has been specifically developed to effectively capture and model long-range dependencies present in sequential data.

Equation 6 computes the update gate (zt), which governs the extent to which the previous hidden state (ht−1) is preserved and updated with the new information from the current input (xt). The update gate receives the concatenation of the previous hidden state and the current input as its inputs. It then applies a sigmoid activation function (σ) to generate values within the range of 0 and 1.

Equation 7 calculates the reset gate (rt), which determines the extent to which the previous hidden state should be disregarded and the degree to which new input information should be taken into account. The reset gate, akin to the update gate, performs a sigmoid activation function on the concatenation of the previous hidden state and the current input.

Equation 8 computes the candidate activation (h˜t), which denotes the additional information that is to be incorporated into the hidden state. The proposed method involves the utilization of a reset gate (rt) to modulate the previous hidden state (ht−1) in conjunction with the current input (xt). This is followed by the application of a hyperbolic tangent (tanh) activation function, which serves to introduce non-linearity.

Equation 9 updates the hidden state (ht) by considering the update gate (zt) and the candidate activation (h˜t). The calculation involves the determination of weights by the update gate, which are then used to compute a weighted sum of the previous hidden state and the candidate activation.

The utilization of the GRU model in chatbot development enables the processing of sequential input data, such as a conversation's word sequence, and the capturing of extensive dependencies throughout the sequence. The update gate (zt) regulates the balance between retaining pertinent context from preceding states and integrating fresh information from the current input. The reset gate (rt) is responsible for determining the extent to which the previous state should be forgotten or reset prior to taking the new input into consideration. The process of candidate activation (h˜t) involves the incorporation of additional information into the hidden state. The GRU model is utilized to enhance the ability of chatbots in effectively representing sequential data and preserving pertinent context throughout extended periods. The utilization of GRU (Gated Recurrent Unit) enables chatbots to enhance their comprehension of the context of user queries, resulting in the generation of more contextually appropriate responses. Consequently, chatbots can engage in conversations with users that are more natural and meaningful.

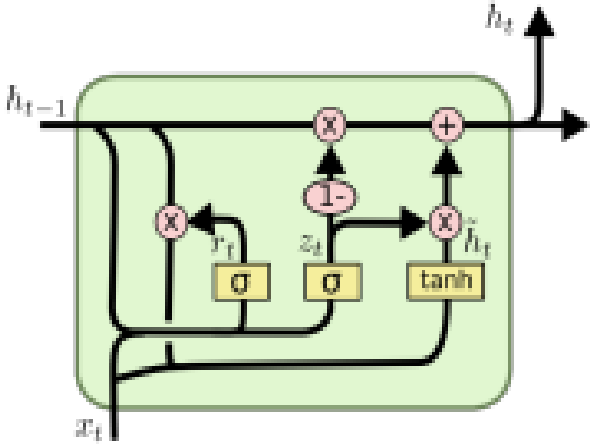


Fig. 6 Schematic View of a GRU [2]

VII. CHATBOT ARCHITECTURE

The architecture of a general chatbot typically comprises five key components: User Interface, Natural Language Understanding (NLU), Dialogue Management (DM), Backend, and Response Generation (RG) [22].

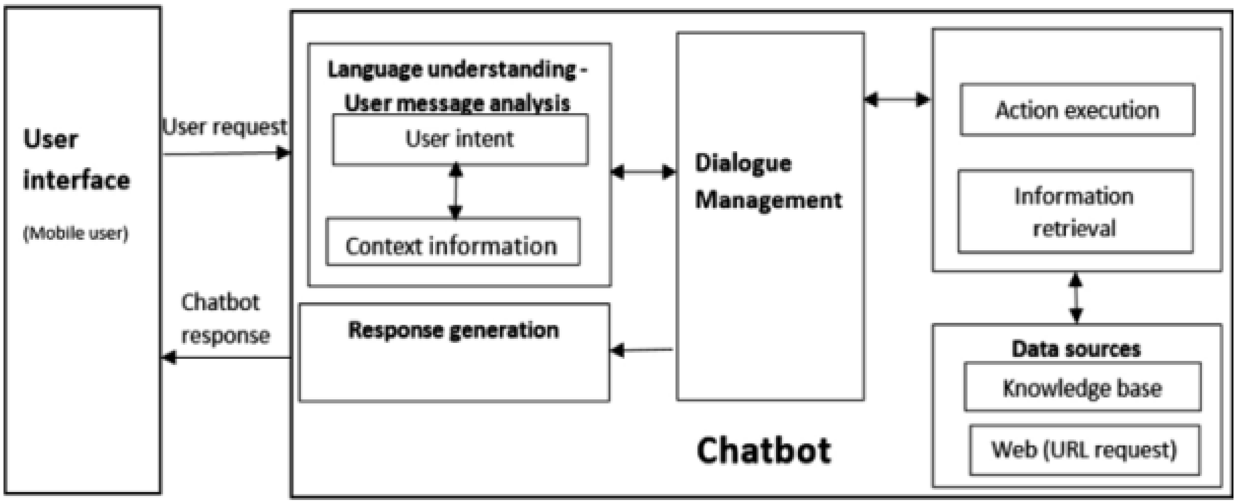


Fig. 7 General Chatbot Architecture [23]

A. User Interface

The user interface facilitates user interaction and engagement with a chatbot via messenger platforms such as Facebook Messenger, Cortana, and Slack [22]. The functioning of a chatbot begins when it receives a request from a user. Automated speech recognition (ASR) systems are employed by voice-based conversational agents to transcribe the user's spoken input into textual form [22]. A conversational agent that operates through speech utilises a text-to-speech (TTS) mechanism to transform textual responses into audible speech. The process of text-to-speech can be divided into two primary stages: text analysis and waveform generation. The process of text analysis encompasses the normalisation of textual data and the examination of phonetic and prosodic elements. On the other hand, waveform synthesis involves the careful selection of pre-recorded speech that satisfies particular criteria [22].

B. Natural Language Understanding

Upon receiving the user's request, the system employs the Natural Language Understanding (NLU) component to extract pertinent information from the input and produce a representation of its meaning, which can be utilised in subsequent stages [23]. Natural Language Understanding (NLU) typically encompasses three fundamental tasks, namely the categorization of dialogue acts, the classification of intent, and the filling of slots [23]. The process of dialogue act classification entails the identification of the user's input's purpose, achieved by categorising it into a distinct dialogue act type. The verbal expression can be categorised as a query, declaration, proposition, or alternative form of communicative action. To effectively respond to the user's request, it is important to comprehend the current dialogue act. Intent classification is the process of determining the primary objective of the user. Intentions typically vary depending on the specific domain. For example, a request may pertain to tasks such as ordering meals, making hotel reservations, or obtaining weather forecasts. In the domain of hotel reservations, an agent's objective may involve making, canceling, or modifying a reservation. Similarly, in the domain of food ordering, an agent's intention may involve placing, querying, or modifying an order [23]. Slot filling is the concluding stage of the natural language understanding (NLU) process. The agent extracts additional relevant details to fully comprehend the user's request, in conjunction with the dialogue act and the user's purpose. Multiple methodologies exist for slot filling.

1. *Regular Expression:* Text pattern description tool. Patterns consist of sequences of characters, where each character can be a regular character or a metacharacter with a specific meaning [23].

2. *Tokenization* involves segmenting a text into discrete units, which can be words, punctuation marks, or numerical values. Tokenizing an English text is typically straightforward due to the presence of white space between words. There are instances where the boundaries between words or phrases are unclear. Special cases in language usage encompass contracted elements [23].

3. *Text normalization* is the procedure of converting text into a standardized format. Word normalization refers to the identification of words that have equivalent meanings but differ in their spellings, such as G.R.A. and GRA. Case folding refers to the conversion of text into lowercase letters.

4. The Bag of Words commonly referred to as the vector space model, is recognized as a straightforward method for text analysis. We calculate the occurrence rate of individual words while disregarding syntactic structure and word arrangement. Text normalization and stop word removal are effective techniques for achieving this goal [23]. The bag-of-words technique is insufficient when a document's interpretation relies on linguistic factors like grammar or word order [24].

5. Latent Semantic Analysis (LSA), also known as Latent Semantic Indexing, does not incorporate linguistic knowledge. LSA, or Latent Semantic Analysis, focuses on the underlying meaning of words rather than the words themselves. It identifies and groups together keywords that commonly appear in similar contexts [23]. To assess word similarity, a matrix is constructed where each row corresponds to a term, each column corresponds to a document, and each cell denotes the frequency of the word in the document [23]. The matrix undergoes a transformation through the application of Singular Value Decomposition in order to reduce its dimensionality.

6. Part-of-speech (POS) Tagging: One aspect of the procedure entails assigning a tag to each word in a given text, indicating its syntactic function, such as its categorization as a verb, noun, pronoun, conjunction, and other relevant classifications. The utilisation of Part-of-Speech (POS) tagging proves to be advantageous in resolving the ambiguity of terms that possess multiple interpretations within sentences [23].

7. Dialogue Management: The Dialogue Management component manages information from other components and controls the context of discussions and activities of the chatbot [24]. The DM module collaborates with other modules and encompasses the logical components of the speech application. This highlights the importance of the chatbot architecture and the need for thoughtful and intentional design choices.

8. Backend Chatbots: The necessary information is obtained from the Backend to fulfill the tasks, and subsequently transmitted to the Dialogue Management Component and the Response Generation Component [25]. The rules for rule-based chatbots should be stored in a Knowledge Base (KB). To enhance the resilience of the chatbot, it is advisable to have a substantial and comprehensive set of rules in the Knowledge Base. A chatbot can use a relational database (RDB) to retrieve past conversations. Considering prior information enhances the chatbot's consistency, precision, and dependability in interactions. The development of the knowledge base (KB) is essential and necessary, although it can be a labor-intensive and stressful process due to the manual work involved. To address this challenge, developers proposed a technique for automatically generating a new knowledge base (KB) from an existing chatbot's KB.

9. Response Generation: After collecting the necessary information, the conversation system analyzes the content and determines the most effective way to communicate the response [23]. The primary function of the Response Generation component is to generate user-friendly responses. The RG pipeline consists of five steps: signal analysis, data interpretation, document planning, microplanning, and implementation [24].

C. Empirical Study

E. Adamopoulou (2020), presents a practical case study of a chatbot solution implemented in the GAMING1 organization [26]. The author elucidated the theoretical basis of the employed tactics. The paper discusses the utilization of gated recurrent unit neural networks and other deep learning techniques. Next, a compilation of the challenges addressed by the chatbot is documented. The author subsequently introduces and elucidates a scalable software architecture for the chatbot. This study presents various neural network structures for user intent classification and includes models for requesting a human operator. Gated recurrent units were found to be the most effective for classification, while simpler models showed satisfactory performance for human operator requesters.

S. Natale (2021), conducted a study on the advancement of chatbots in higher education [27]. This study examines the advantages of incorporating chatbots through the Facebook Page of an academic program. This paper also provides an explanation of the process of developing a chatbot using Google Dialog flow as the natural language understanding (NLU) platform and Facebook Messenger as the interface platform. A total of 807 sentences from 125 individuals were extracted from the chat logs, serving as a representative sample. These sentences allowed for the identification of 33 distinct intents. The proposed chatbot achieves high performance with precision, recall, and F1-score values of 0.98, 0.888, and 0.897, respectively. Chatbots have effectively addressed concerns for both page managers and end users.

N. Tavichaiyuth (2021), employ Google Colab and the Internet of Things in their innovative approach to Arabic Chatbot [28]. This study introduces a novel approach for developing efficient Arabic chatbots by leveraging the Internet of Things (IoT). An experiment was conducted to develop and deploy an Arabic chatbot for an IOT-based data center using Google Colab and the Python Chatterbot module.

Y. M. Mohialden (2021), investigated the impact of social virtual presence agents and a content-based product recommendation system on customers' intention to make online purchases [29]. This study aims to investigate the impact of two different types of chatbot interactions on users: one with a fully pre-recorded computer-generated agent and the other with a pre-recorded human agent. Additionally, the study seeks to understand how the presence of a personalized content-based product recommendation system can affect consumers' intention to make purchases in an online shopping context. The primary data were analyzed using a PLS-SEM model on an online platform that simulated a virtual store experience. The Human Social Virtual Presence Agent has a stronger positive impact on Intellectual stimulation and Hedonic Benefits compared to a computer-generated personification Agent when assisting shoppers. This phenomenon can be attributed to the emotional disturbance experienced by participants when exposed to Agent's computer-generated images and sounds. Furthermore, the inclusion of a recommendation system has a positive impact on customers' purchase intent when compared to the absence of such a system. This study highlights the significance of social contacts, particularly those facilitated by human interaction, in influencing customers' purchasing decisions. Additionally, it examines the impact of recommendation systems on customers' buying intentions.

F. P. D. Sousa (2020) created an Intelligent Chatbot based on a Question-and-Answer System utilizing the RNN-LSTM Model [30]. The chatbot is designed to store data collected through a question-and-answer system, which can later be used in the Python application. This application will use the Cornell Movie Dialog Corpus, which is a dataset consisting of a large collection of fictional dialogues extracted from movie scripts and accompanied by extensive metadata. The Python chatbot application utilizes various models, with a specific focus on the LSTM model. The chatbot program using the LSTM model produces accurate metrics and a dataset that corresponds to the user's input in the chatbot dialogue box. The selection of models for a program depends on data that can impact its performance. The program's objective is to determine the level of accuracy it can achieve, which plays a crucial role in model selection. The LSTM model integrated into the chatbot suggests that Parameter Pair 1 (size layer 512, num layers 2, embedded size 256, learning rate 0.001, batch size 32, epoch 20) from File 3 is the most suitable parameter pair. This pair achieves an average accuracy of 0.994869.

P. Anki, (2021), developed and deployed a Kurdish language chatbot using the Chatfuel platform [31]. This study presents the development and implementation of a chatbot aimed at facilitating text-based internet discussions for Kurdish speakers as a substitute for direct communication with human operators. The software agent utilizes the Chatfuel platform to implement natural language processing. Chatfuel utilizes artificial intelligence to engage in human-like conversations through voice commands or text messages. The chatbot is evaluated using an electronic tourist guide designed to aid tourists visiting religious sites in the Barzanja mountain village in Iraqi Kurdistan. The case study utilizes 300 questions and corresponding responses. The study involved 100 participants. The participant inquires, and the bot provides a response if it identifies the question. If the question is not recognized, the bot offers a default answer and provides instructions on how to properly utilize the system. The data from these experiments is collected and analyzed to identify challenges related to the Kurdish language. Developing software agents for parsing Kurdish texts poses multiple challenges. Kurdish texts have not yet been subjected to natural language processing (NLP) techniques. The presence of Kurdish typeface disorder, along with the absence of standardized keyboards and writing styles, poses challenges in the processing of Kurdish text. Furthermore, the Kurdish language encompasses various dialects that exhibit unique writing systems.

H. K. Ahmed (2021) successfully devised and executed the creation of xiaoice, a chatbot that demonstrates social empathy [32]. This article explores the evolution of Microsoft XiaoIce, a widely acclaimed social chatbot. XiaoIce is an artificial intelligence companion that has been developed with the objective of fulfilling human requirements for communication, affection, and social inclusion through the establishment of an emotional bond. The system design considered both intellectual quotient (IQ) and emotional quotient (EQ) in its development. The approach to human-machine social chat was conceptualised as a decision-making procedure, as opposed to employing Markov Decision Processes. The primary objective of optimising XiaoIce was to improve long-term user engagement, as quantified by the anticipated Conversation-turns Per Session (CPS). This paper presents a comprehensive overview of the system architecture and its primary constituents, namely the dialogue manager, core chat, skills, and an empathic computing module. During our comprehensive deliberations, we present evidence of XiaoIce's aptitude in accurately discerning human emotions and mental states, comprehending user intentions, and proficiently addressing user inquiries. Since its inception in 2014, XiaoIce has successfully interacted with a significant number of active users, surpassing 660 million. Moreover, it has managed to establish enduring relationships with a considerable portion of its user base. Based on an analysis of extensive online logs, it has been determined that XiaoIce exhibits an average Characters Per Second (CPS) of 23. This surpasses the performance of both other chatbots and human conversations in terms of speed and efficiency.

L. Zhou, (2020), created an IntelliBot. The researchers examined the response generation strategies employed by current chatbots and identified their shortcomings in terms of effectively engaging in a conversation with a user [33]. The platform also provides a specialized chatbot called IntelliBot, which is a dialogue-based system that generates responses using multiple methodologies. IntelliBot has acquired domain-specific knowledge through training on two datasets: the Cornell movie dialogue dataset and a custom-built insurance dataset. IntelliBot's performance was subsequently validated and compared to three other chatbots documented in existing literature: RootyAI, ChatterBot, and DeepQA. The results demonstrate that IntelliBot excels in user engagement and delivering comprehensive responses in the insurance field.

M. Nuruzzaman, (2020), developed a cognitive chatbot for personalized context-based customer service (72) [34]. The effectiveness of the chatbots was established. This study focuses on providing personalized contextual customer support through cognitive chatbots. The quantitative research approach was used to analyze primary data obtained from 300 B2B corporate respondents. This study addresses the limited existing research on chatbots and their potential for improving customer service. The results suggest that customers highly prioritize real-time information on the reliability and availability of products and services. Customers benefit from automated responses to frequently asked questions, which enhances their overall experience by providing a seamless and efficient resolution to recurring concerns. This study makes significant theoretical contributions to the existing literature on chatbots by combining two models into a simplified one. It also demonstrates that trust plays a role in the decision to use intelligent chatbots, which in turn promotes automation.

VIII. METHODOLOGY

A. Research Approach

The methods involved in developing the chatbot for the customer support system, including the chatbot's architecture, individual components, and the data that passes through the chatbot system, will be detailed in detail in this chapter. Furthermore, the minimum hardware requirements and software needs will be covered.

B. Requirement Specification

1. Hardware Minimum Requirements: The minimum hardware requirements pertain to the physical features of the machine required to run the chatbot. The following are the features: at least 250 GB HDD, 4 GB RAM, and an Intel Pentium Dual-Core processor.

2. Software Requirements: These are the computer programmes and procedures needed to put the chatbot into action. The tools used include,

i. Spring boot: to implement the bot logic.
ii. Vaadin: to build a web UI to interact with the AIML bot.
iii. AIML (Artificial Intelligence Markup Language): was used to train the bot.
iv. Detect Language java library was used for interpretation to any language of choice.

C. System Design

1. Chatbot Interface

Java is a programming language that is not tied to any specific platform. Java can be executed on platforms that support a Java interpreter. The process involves translating the high-level program into assembly language, which is also known as machine language. The AIML Java interpreter library is a software tool that allows for the interpretation of AIML (Artificial Intelligence Markup Language) in the Java programming language. Program AB was utilized to design the interface.

2. Web UI

A Web user interface, also known as a Web app, enables users to interact with content or software hosted on a remote server using a Web browser.The web user interface (UI) was developed using the "Vaadin" framework to facilitate interaction with the AIML bot. Vaadin is a Java-based open-source platform designed for developing web applications. The software facilitates two distinct programming paradigms: server-side and client-side. The web user interface (UI) for this research project was developed using Vaadin, which facilitated interaction with the AIML bot. Vaadin is a Java-based open-source platform used for the development of web applications.

3. Development of the Chatbot with AIML

The main aim of the AIML language is to enhance the efficiency of the dialogue modelling process. This facilitates the utilisation of the stimulus-response mechanism. In addition, AIML can be classified as a markup language that utilises XML as its underlying structure. AIML functions by instantiating a class object that is responsible for representing conversational patterns. AIML has emerged as the predominant language for chatbot development due to its notable attributes such as its straightforwardness, accessibility for learning purposes, simplicity of integration, and the wide availability of pre-existing AIML sets. The system functions as a rudimentary word pattern-matching mechanism that generates a response based on a provided query. The AIML robot employs AIML files to generate responses by leveraging the correlation between user queries and stored knowledge.

D. Research Method

1. Data Collection: The messages dataset was extracted from already made AIML file for FAQ questions. Artificial Intelligence Markup Language, or AIML, is an XML-based language used to program natural language software agents, such as chatbots. The already made questions are shown below using bank account, Nigeria Immigration Service, FIRS and custom duty question:

2. Bank Account: "Can I Open An Account While Outside The Country?", "Is It Possible To Open An Account In The Country And Operate The Account While Out Of The Country?", "How Can I Open A Salary Account?", "How Long Does It Take Before My Account Becomes Inactive?", Can A Savings Account Become Dormant?, How Long Does It Take To Open An Account In Banks?, What Are The Requirements For A Change Of Address in Banks?.

3. Nigerian Immigration Services: “Will I have to start the process for my passport or visa in the bank?”, Will I get my Visa or Passport once payment is made in the bank?

4. Federal Inland Revenue Service Payments: “What is TIN?”, Can payment be made in any Bank’s branch? “What documents required for payment?” “How can a new taxpayer get the TIN? Must a taxpayer be assessed before payment?”

5. Customs Duty Payments: “Can payment be made at any bank?, What are the documentation required for payment? What is SGD Assessment Notice?”, “Who issued the SGD Assessment Notice?”

E. Preprocessing

The following are the preprocessing steps in the order in which they will be performed:

1. Reduce the case of all characters.
2. Remove superfluous punctuation characters (e.g., “!?,ˆ ...)
3. Tokenize the texts by using words as tokens. A word is defined in this context as a contiguous sequence of characters enclosed in white space.
4. Compute the vocabulary of the texts (i.e., the set of tokens used).
5. Replace each token in the sequences with its vocabulary index.
6. Reduce the length of the sequences to a fixed value of 100 tokens.

F. Training

AIML, an acronym for Artificial Intelligence Markup Language, was employed for the development, training, and monitoring of the conversational advancements of the bot. Additionally, the platform offers a webhook functionality that enables the deployment and integration of the chatbot on the user interface. The API includes a feature that allows for the integration of external data to enhance the bot’s knowledge base. The external data is contained within the customer support datasets, which can be accessed through the admin login portal and DBMS. The bot undergoes extensive supervised learning during its training process, which involves closely monitoring its conversational progress.

G. Implementation of Software

Java was chosen as the primary programming language for the implementation of the chatbot in this research. Java, a language known for its ease of learning, was considered suitable for this particular scenario. The scikit-learn library is a significant tool in this project, although it may be considered minor in comparison to other libraries [35]. In this context, it is utilized for various purposes, such as dividing a dataset into train, validation, and test sets.

H. Generating Responses

The responses were generated using a retrieval-based approach. This feature allows the agent’s maintainers to prevent the agent from making offensive statements to the user. The Action Planner module retrieves responses and contains the logic for handling user intents. The system may communicate with the backend to confirm the user’s account status and check if there were any delays in the most recent withdrawal to an international bank account.

I. Agent Environment

The agent environment consists of multiple discrete components. One type of dynamic input that the agent may receive at any time is user messages. The user’s text is stored as a string along with metadata that includes a pointer to the conversation structure, the message’s timestamp, the platform used, and other relevant details. The chatbot can engage with users to gather more information or acknowledge that their request has been received and addressed. Figure 9 illustrates the visual representation of the environment in which the agent operates.

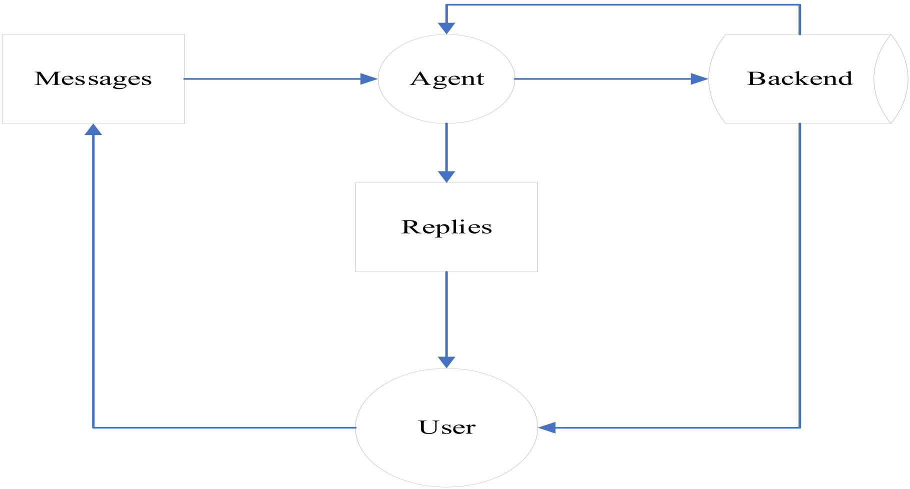


Fig. 9 Agent Environment Model

J. Performance Evaluation of the Developed System

The developed system was evaluated using the manual software testing technique by a test engineer [36]. Hence the following test were done to evaluate the performance of the system.

1. Performance and Scalability Testing (Response Time, Stability, Load).
2. Usability Testing (Easy to understand, Easy to access, Faster to Access, Effective Navigation).
3. Compatibility and Reliability Testing (Software, Hardware. Network, Mobile).

IX. RESULTS AND DISCUSSION OF FINDINGS

The chatbot was developed using a Java interpreter. AIML java interpreter library Program AB was added which main function is to match input and output predefined in the AIML file (Al-Ghadhban & Al-Twairesh, 2020). Below is the snippet code for the Java interpreter used.

```
<category>
<pattern>CAN I OPEN AN ACCOUNT WHILE
OUTSIDE THE COUNTRY</pattern>
<template>You can open an account online here while
out of the country if you already have a Bank
Verification Number (BVN). If you do not have a BVN,
you will still receive an account number, but the
account will remain restricted until you complete the
required documentation.</template>
</category>
```

This would lead to the following dialogue;

User: *Can I open an account while outside the country*?

Bot : *You can open an account online here while out of the country if you already have a bank verification number (BVN). if you do not have a BVN, you will still receive an account number, but the account will remain restricted until you complete the required documentation.*

A. Result of Spring Boot Project

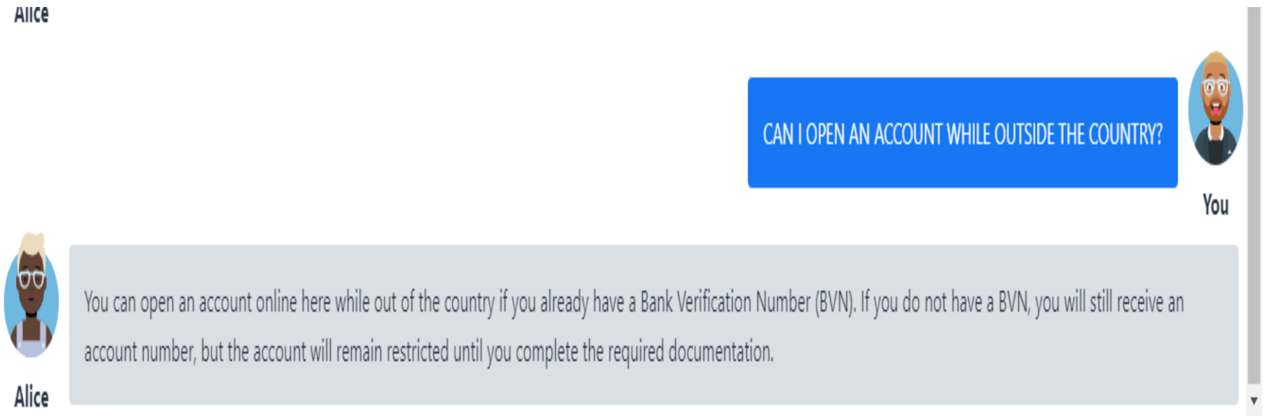
The spring boot project was created using Spring initlializr and implementing the Bot logic. Spring Initializr is a web-based tool used to generate the structure of the Spring Boot Project [37]. Below is the snippet code for the Java interpreter used.

```
import
org.springframework.boot.builder.SpringApplicationBuilder;
import
org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
public class ServletInitializer extends
SpringBootServletInitializer {
  @Override
  protected SpringApplicationBuilder
configure(SpringApplicationBuilder application) {
      return application.sources(Application.class);
  }
```

B. Result on AIML file for FAQ Questions

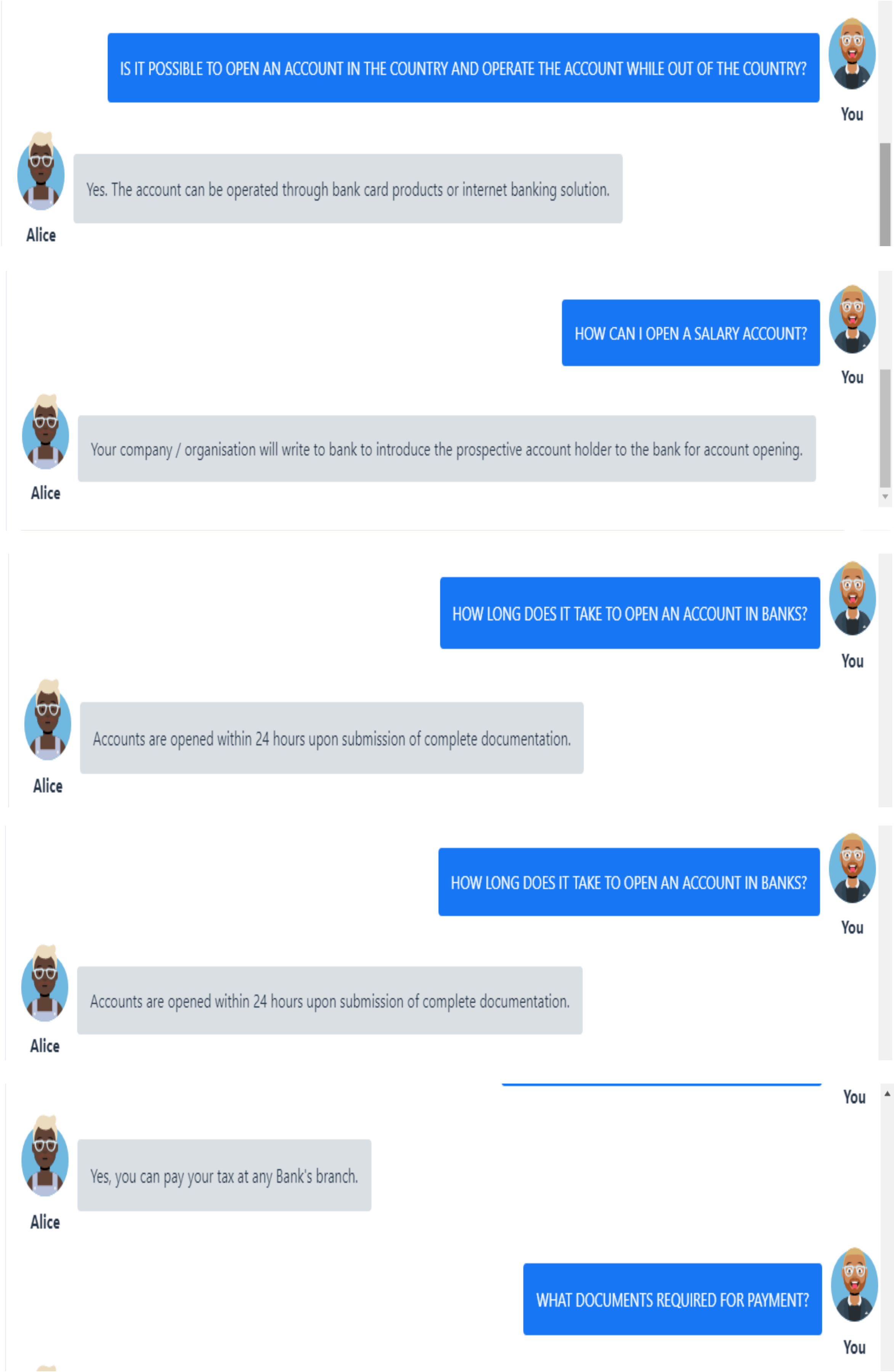
An AIML file comprises the Artificial Intelligence Markup Language (AIML), which serves the purpose of defining the personality of a chat robot, commonly referred to as a chatbot. A chatbot is typically equipped with a set of AIML files that dictate its conversational structure and imbue the robot with a distinct personality. The following code snippet represents the Java interpreter utilized.

```
category>
   <pattern>HOW LONG DOES IT TAKE BEFORE MY
ACCOUNT BECOMES INACTIVE</pattern>
   <template>Current accounts become dormant after 6
months of inactivity.</template>
  </category>
```



CAN I OPEN AN ACCOUNT WHILE OUTSIDE THE COUNTRY?

You can open an account online here while out of the country if you already have a Bank Verification Number (BVN). If you do not have a BVN, you will still receive an account number, but the account will remain restricted until you complete the required documentation.



IS IT POSSIBLE TO OPEN AN ACCOUNT IN THE COUNTRY AND OPERATE THE ACCOUNT WHILE OUT OF THE COUNTRY?

You

Yes. The account can be operated through bank card products or internet banking solution.

Alice

HOW CAN I OPEN A SALARY ACCOUNT?

You

Your company / organisation will write to bank to introduce the prospective account holder to the bank for account opening.

Alice

HOW LONG DOES IT TAKE TO OPEN AN ACCOUNT IN BANKS?

You

Accounts are opened within 24 hours upon submission of complete documentation.

Alice

HOW LONG DOES IT TAKE TO OPEN AN ACCOUNT IN BANKS?

You

Accounts are opened within 24 hours upon submission of complete documentation.

Alice

You

Yes, you can pay your tax at any Bank's branch.

Alice

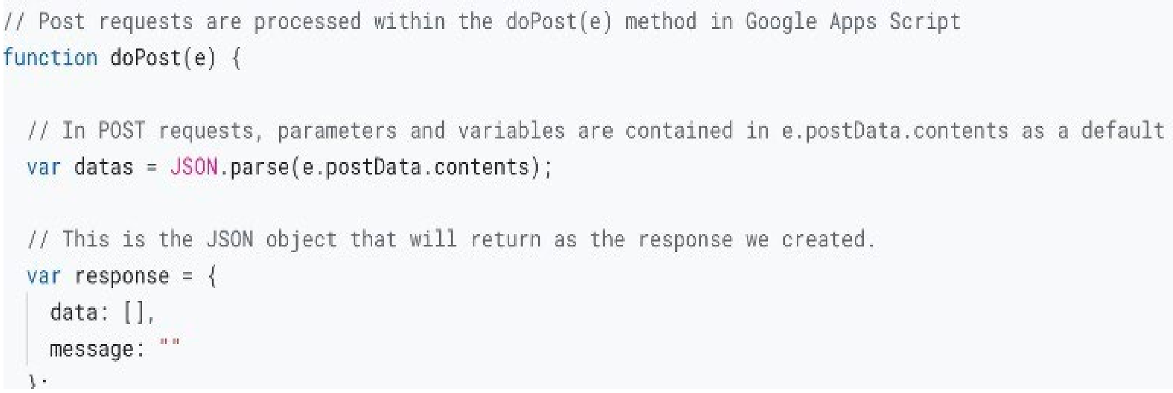
WHAT DOCUMENTS REQUIRED FOR PAYMENT?

You

Fig. 10 Snapshot of the Result of FAQ questions and Chatbot Responses

C. Result on Translation from any Language to English

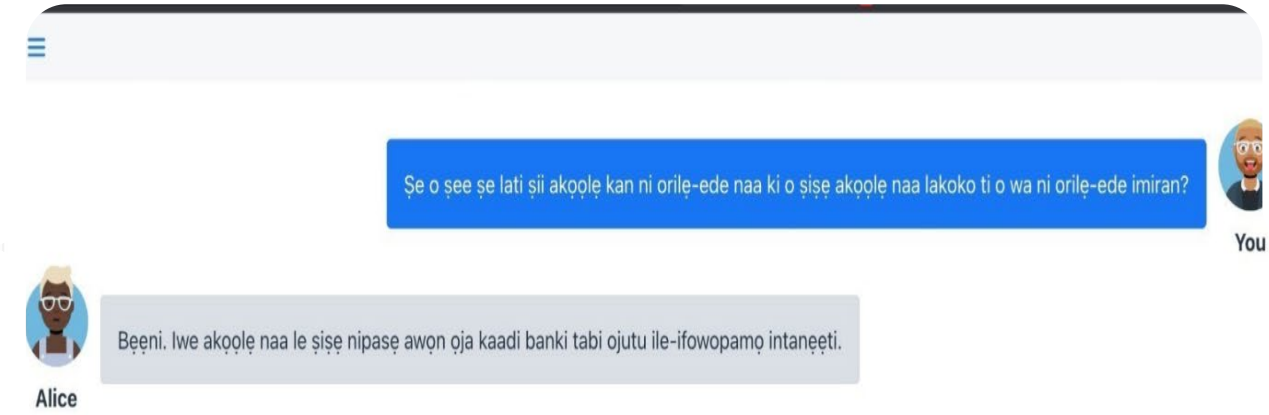
Google script was written to translate from any language to English for the bot to understand and send the response in the preferred language of the user. For this research, the Bot was tested to translate to Yoruba language as shown in figure. Below is the snippet code for the google script.



```
// Post requests are processed within the doPost(e) method in Google Apps Script
function doPost(e) {

  // In POST requests, parameters and variables are contained in e.postData.contents as a default
  var datas = JSON.parse(e.postData.contents);

  // This is the JSON object that will return as the response we created.
  var response = {
    data: [],
    message: ""
  };
```



Ṣe o ṣee ṣe lati ṣii akọọlẹ kan ni orilẹ-ede naa ki o ṣiṣẹ akọọlẹ naa lakoko ti o wa ni orilẹ-ede imiran?

Bẹẹni. Iwe akọọlẹ naa le ṣiṣẹ nipasẹ awọn ọja kaadi banki tabi ojutu ile-ifowopamọ intanẹẹti.

Fig. 11 Snapshot of the Result of Response in the Preferred Language of Use

D. Result of Performance Evaluation of the Developed System

TABLE I PERFORMANCE EVALUATION TABLE

Sl. No.	**Test Type**		**Score (%)**
1	Performance	i. Response Time ii. Stability iii. Load iv. Reliability	90 80 70 90
2	Usability	i. Easy to Understand ii. Easy to Access iii. Faster to Access iv. Effective Navigation	95 90 90 85
3	Compatibility	i. Software ii. Hardware iii. Network iv. Mobile	95 90 80 90
4	Scalability	i. Throughput ii. Memory Usage iii. CPU Usage iv. Network Usage	80 5 0 0

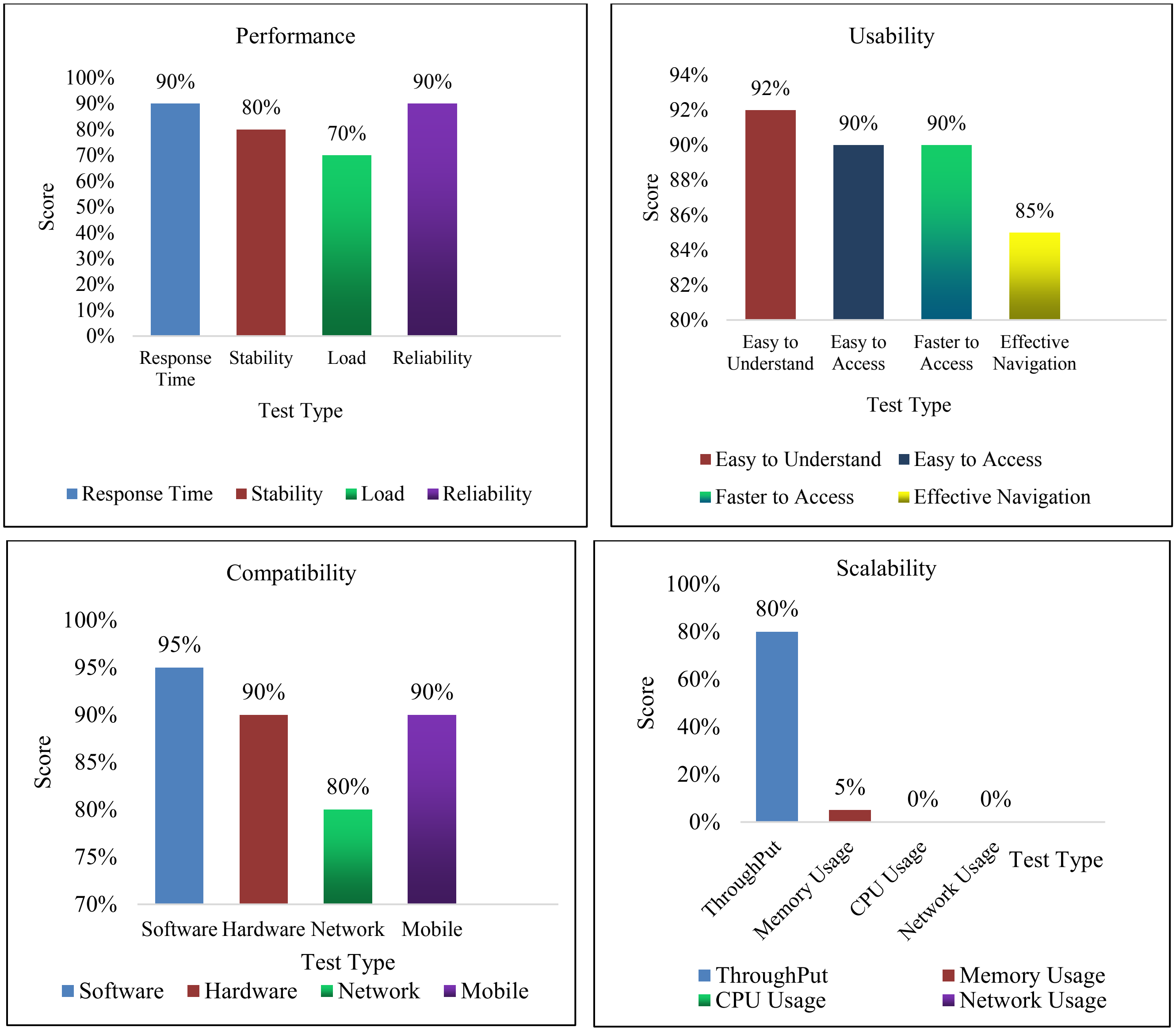


Fig. 12 Performance Evaluation Chart

E. Discussions of Findings

The chatbot was developed using AIML java interpreter library Program AB which helps match input and output predefined in the AIML file. This would lead to the following dialogue shown in Figure 10. The spring boot project was created using Spring initlializr and implementing the Bot logic. AIML(Artificial Intelligence Markup Language) was used to preprocess and train the bot using ready-made AIML file for FAQ questions.

Also, vaadin was used to build a web UI to interact with the trained AIML bot. Finally, a google script was written to translate from any language to English for the bot to understand and send the response in the preferred language of the user. The speed of the bot network however depends on the network used. Any failure in the network may delay the response time of the system.

X. RECOMMENDATIONS

Based on the findings from this study, the following recommendations were made:

i. It is recommended to standardise routine queries in order to achieve a high level of performance. It is unlikely that chatbots will possess the capability to address all inquiries, particularly within the domains of financial and payment application services.

ii. Standardising routine queries is crucial in order to ensure a consistently high level of performance, as Chatbots may not possess the capability to resolve all queries, particularly within the context of financial and payment application services.

iii. It is additionally recommended that conducting usability testing is imperative to ensure the utility of a chatbot, as the design of chatbots must be of high quality in order to gain user acceptance.

XI. SUGGESTIONS FOR FURTHER RESEARCH

The following are the suggestions for further research:

1. Additional languages could be acquired and developed to expand the reach and accessibility to a broader range of users.
2. A potential alternative iteration of the chatbot could be conceptualised, featuring a distinct dynamic framework. This variant would benefit from enhanced responsiveness through the inclusion of additional small-talk capabilities and the establishment of more personalised and direct interactions with customers.

XII. CONCLUSION

The bot was able to answer customer questions, enquiries and complaints and the response time of the bot depends on the strength of the network since it is web based. The system had a good response time, scalability, and reliability. Hence, the system provided a simple, cheaper and durable customer financial and payment application service.

REFERENCES

[1] F. Peters, Master thesis: Design and implementation of a chatbot in the context of customer support, 2018.

[2] Y. Yun, D. Ma, and M. Yang, “Human-computer interaction-based decision support system with applications in Data Mining,” Future Generation Computer Systems, Vol. 114, pp. 285-89, 2021. DOI: 10.1016/j.future.2020.07.048.

[3] A. A. A. Weißensteiner, “Chatbots as an approach for a faster enquiry handling process in the service industry,” *Signature*, Vol. 12, No. 04, 2018. DOI: 10.3030/811429.

[4] S. Schanke, G. Burtch, and G. Ray, “Estimating the impact of ‘humanizing’ customer service chatbots,” Information Systems Research, Vol. 32, No. 3, pp. 736-751, 2021. DOI: 10.1287/isre.2021.1015.

[5] A. Truong *et al.,* “Towards Automated Machine Learning: Evaluation and comparison of AUTOML approaches and Tools,” *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI),* 2019. DOI: 10.1109/ictai.2019.00209.

[6] J. Hill, W. R. Ford and I. G. Farreras, “Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations,” *Computers in human behavior*, Vol. 49, pp. 245-250, 2015.

[7] W. Maroengsit, T. Piyakulpinyo, K. Phonyiam, S. Pongnumkul, P. Chaovalit, and T. Theeramunkong, “A survey on evaluation methods for chatbots,” in *In Proceedings of the 2019 7th International conference on information and education technology*, pp. 111-119, Mar. 2019.

[8] M. H. Huang and R. T. Rust, “Engaged to a robot? the role of AI in Service,” *Journal of Service Research*, Vol. 24, No. 1, pp. 30-41, 2020. DOI:10.1177/1094670520902266.

[9] A. Følstad and M. Skjuve, “Chatbots for customer service: user experience and motivation,” in *In Proceedings of the 1st international conference on conversational user interfaces*, pp. 1-9, Aug. 2019.

[10] T. Schachner, R. Keller, and F. V. Wangenheim, “Artificial Intelligence-based conversational agents for chronic conditions: Systematic Literature Review,” *Journal of Medical Internet Research*, Vol. 22, No. 9, 2020. DOI: 10.2196/20701.

[11] C. Rutschi and J. Dibbern, “Towards an understanding of scaling the software robot implementation,” *Progress in IS*, pp. 453-466, 2020. DOI: 10.1007/978-3-030-45819-5_18.

[12] R. Richer, N. Zhao, B. M. Eskofier, and J. A. Paradiso, “Exploring smart agents for the interaction with multimodal mediated environments,” *Multimodal Technologies and Interaction*, Vol. 4, No. 2, pp. 27, 2020. DOI: 10.3390/mti4020027.

[13] R. Law, R. Leung, A. Lo, D. Leung, and L. H. Fong, “Distribution channel in hospitality and tourism,” *International Journal of Contemporary Hospitality Management*, Vol. 27, No. 3, pp. 431-452, 2015. DOI: 10.1108/ijchm-11-2013-0498.

[14] A. Przegalinska, L. Ciechanowski, A. Stroz, P. Gloor, and G. Mazurek, “In bot we trust: A new methodology of Chatbot Performance Measures,” *Business Horizons*, Vol. 62, No. 6, pp. 785-797, 2019. DOI: 10.1016/j.bushor.2019.08.005.

[15] T. Nemoto and T. Fujimoto, “A classification and analysis focusing on attempts to give a computer a personality: A technological history of Chatbots as Simple Artificial Intelligence,” *Innovations in Applied Informatics and Media Engineering*, pp. 59-70, 2023. DOI: 10.1007/978-3-031-30769-0_6.

[16] S. Singh and H. K. Thakur, “Survey of various AI chatbots based on technology used,” *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020. DOI: 10.1109/icrito48877.2020.9197943.

[17] S. Fernandes *et al.*, “Survey on various Conversational Systems,” *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020. DOI: 10.1109/ic-etite47903.2020.126.

[18] S. J. Shaikh, “Artificially Intelligent, Interactive, and Assistive Machines: A Definitional Framework for Intelligent Assistants,” *International Journal of Human-Computer Interaction,* Vol. 39, No. 4, pp. 776-789, 2023.

[19] P. Kumari and D. Toshniwal, “Long short term memory-convolutional neural network based deep hybrid approach for solar irradiance forecasting,” *Applied Energy*, Vol. 295, pp. 117061, 2021. DOI: 10.1016/j.apenergy.2021.117061.

[20] Le, Ho, Lee, and Jung, “Application of long short-term memory (LSTM) neural network for flood forecasting,” *Water*, Vol. 11, No. 7, pp. 1387, 2019. DOI: 10.3390/w11071387.

[21] W. Wang, “Running programs,” *Absolute Beginners Guide to Computing*, pp. 29-42, 2016. DOI: 10.1007/978-1-4842-2289-8_3.

[22] N. A. Al-Madi, K. A. Maria, M. A. Al-Madi, M. A. Alia, and E. A. Maria, “An intelligent Arabic chatbot system proposed framework,” *2021 International Conference on Information Technology (ICIT)*, 2021. DOI: 10.1109/icit52682.2021.9491699.

[23] X. Huang and A. CIS, *Chatbot: Design, Architecutre, And Applications,* 2021.

[24] Y. Zhang *et al.*, “Life after speech recognition: Fuzzing semantic misinterpretation for voice assistant applications,” *Proceedings 2019 Network and Distributed System Security Symposium*, 2019. DOI: 10.14722/ndss.2019.23525.

[25] M. McTear, Z. Callejas, and D. Griol, “Conversational interfaces: Devices, wearables, virtual agents, and Robots,” *The Conversational Interface*, pp. 283-308, 2016. DOI: 10.1007/978-3-319-32967-3_13.

[26] E. Adamopoulou and L. Moussiades, “An overview of chatbot technology,” *IFIP Advances in Information and Communication Technology*, pp. 373-383, 2020. DOI: 10.1007/978-3-030-49186-4_31.

[27] S. Natale, “Introduction,” *Deceitful Media*, pp. 1-15, 2021. DOI: 10.1093/oso/9780190080365.003.0001.

[28] N. Tavichaiyuth and E. Rattagan, “Developing chatbots in higher education: A case study of academic program chatbot in Thailand,” 2021.

[29] Y. M. Mohialden, M. T. Younis, and N. M. Hussien, “A novel approach to Arabic Chabot, utilizing Google Colab and the internet of things: A case study at a computer center,” *Webology*, Vol. 18, No. 2, pp. 946-954, 2021. DOI: 10.14704/web/v18i2/web18365.

[30] F. P. D. Sousa, *The impact of social virtual presence agents and content-based product recommendation system on on-line customer purchase intention* (Master’s thesis). repositorio.iscte-iul.pt, 2020.

[31] P. Anki, A. Bustamam, H. S. Al-Ash, and D. Sarwinda, “Intelligent chatbot adapted from question and answer system using RNN-LSTM model,” *Journal of Physics: Conference Series*, Vol. 1844, No. 1, pp. 012001, 2021. DOI:10.1088/1742-6596/1844/1/012001.

[32] H. K. Ahmed and J. Ali Hussein, “Design and implementation of a chatbot for Kurdish language speakers using Chatfuel platform,” *Kurdistan Journal of Applied Research*, pp. 117-135, 2021. DOI: 10.24017/science.2020.2.10.

[33] L. Zhou, J. Gao, D. Li, and H. Y. Shum, “The design and implementation of Xiaoice, an empathetic social chatbot,” *Computational Linguistics*, Vol. 46, No. 1, pp. 53-93, 2020. DOI: 10.1162/coli_a_00368.

[34] M. Nuruzzaman and O. K. Hussain, “IntelliBot: A dialogue-based chatbot for the insurance industry,” *Knowledge-Based Systems*, Vol. 196, pp. 105810, 2020. DOI: 10.1016/j.knosys.2020.105810.

[35] R. K. Behera, P. K. Bala, and A. Ray, “Cognitive chatbot for personalised contextual customer service: Behind the scene and beyond the hype,” *Information Systems Frontiers*, 2021. DOI: 10.1007/s10796-021-10168-y.

[36] F. Gutierrez, “Spring Boot,” *Spring Cloud Data Flow*, pp. 9-31, 2020. DOI: 10.1007/978-1-4842-1239-4_2.

[37] R. A. Badru, A. A. Waheed, O. A. Akinmoluwa, and O. R. Obayemi, “Generation of surveillance networked nodes for oil pipelines’ theft,” *International Journal of Recent Engineering Science*, Vol. 8, No. 5, pp. 21-26, 2021. DOI: 10.14445/23497157/ijres-v8i5p104.

[38] D. Al-Ghadhban and N. Al-Twairesh, “Nabiha: An Arabic dialect chatbot,” *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 3, 2020. DOI: 10.14569/ijacsa.2020.0110357.