**IMPLEMENTATION OF A CONTEXT-AWARE AGRICULTURAL E-COMMERCE CHATBOT SYSTEM INTEGRATED WITH YORUBA LANGUAGE**

**(A CASE STUDY OF AFRIMASH.COM)**

**BY**

**FAJI, OLAOTAN AYODEJI**

**(18CG024738)**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES, COLLEGE OF SCIENCE AND TECHNOLOGY, COVENANT UNIVERSITY OTA, OGUN STATE.**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE BACHELOR OF SCIENCE (HONOURS) DEGREE IN COMPUTER SCIENCE**

**JULY, 2022**

CERTIFICATION

I hereby certify that this project work titled IMPLEMENTATION OF A CONTEXT-AWARE AGRICULTURAL E-COMMERCE CHATBOT SYSTEM INTEGRATED WITH YORUBA LANGUAGE (A CASE STUDY OF AFRIMASH.COM) is a bona fide work carried out by Olaotan Ayodeji Faji (18CG024738) submitted to the Department of Computer and Information Sciences, College of Science and Technology, Covenant University, Ota.

|  |  |
| --- | --- |
| **DR. IBUKUN T. AFOLABI** |  |
| Supervisor | **Signature and Date** |
|  |  |
| **PROFESSOR OLUFUNKE O. OLADIPUPO** |  |
| Head of Department | **Signature and Date** |

DEDICATION

I dedicate this project to God almighty, who has been the source of my strength and a consistent reference point when I felt low. It is safe to say this journey would be impossible without you. Thank you, Lord.

ACKNOWLEDGEMENTS

I express my sincere gratitude to the Almighty God, the source of all strength, wisdom, knowledge, and understanding. He invented science and the scientific method and was my source through this endeavour; thus, I remain ever grateful to my creator.

I also wish to acknowledge the Chancellor of the Diocese of Lagos Mainland (Anglican Communion) - my father, Honourable Justice Olayinka Faji, and my dear mother, Mrs Abosede Faji. My loving, caring parents have been a constant reference point whenever I needed support throughout my life, especially in my academic journey. I sincerely appreciate them for their unending support and guidance and for being an excellent example. One simply cannot wish for better parents.

I am thankful to my siblings, Oladele, Olayinka, and Olanike Faji, for being my sources of encouragement and for helping me overcome tough times. I also appreciate my cousin, Teniola Padoye, and my best friend, Ivy Nwobodo, for constantly reminding me to take care of myself when things got tough during this work. I love all of you, and you are the best.

I also appreciate my supervisor and mother, Dr Ibukun Afolabi, for her temperance, patience, and guidance throughout this endeavour. You are a great mother and a patient virtuous woman, and I appreciate your input, advice, and corrections. God will provide help for you whenever you need it.

I wish to thank the Chaplain of the university, Pastor Victor Hill, for his consistent nudges and encouragement. He has indeed been a father to us all in the student chaplaincy. May the fire on your altar ever burn brightly.

I appreciate the Head of Department, Computer and Information Sciences, Professor Olufunke O. Oladipupo, for her leadership, guidance, and motherly love to us in the department. I also appreciate Dr Azubuike Ezenwoke's organisation, pedantic scrupulousness, and creative juices that kept my colleagues and me on our toes during the project timeline.

I appreciate my level advisors, Dr Onyeka Emebo and Dr Oluranti Jonathan, for their love, guidance, and firm hands. Finally, I am grateful to all faculty members in our dear department for the excellent job of imparting us great knowledge. May God bless you all.

TABLE OF CONTENTS

[Certification i](#_Toc109446457)

[Dedication ii](#_Toc109446458)

[Acknowledgements iii](#_Toc109446459)

[Table of Contents iv](#_Toc109446460)

[List of Figures vii](#_Toc109446461)

[List of Tables viii](#_Toc109446462)

[Abstract ix](#_Toc109446463)

[Chapter One: Introduction 1](#_Toc109446464)

[1.1. Background Information 1](#_Toc109446466)

[1.2. Statement of the problem 2](#_Toc109446467)

[1.3. Aim and Objectives of the study 2](#_Toc109446468)

[1.4. Methodology 3](#_Toc109446469)

[1.5. Significance of study 3](#_Toc109446470)

[1.6. Limitations of the study 4](#_Toc109446471)

[1.7. Arrangement of thesis or research outline 4](#_Toc109446472)

[Chapter Two: Literature Review 5](#_Toc109446473)

[2.1. Preamble 5](#_Toc109446475)

[2.1.1. Review Of Chatbots 5](#_Toc109446476)

[2.2. Review of Existing Systems 6](#_Toc109446477)

[2.2.1. Neu Chatbot 6](#_Toc109446478)

[2.2.2. Chatbot For Chronic Patient Support 8](#_Toc109446479)

[2.2.3. Ask Rosa 9](#_Toc109446480)

[2.2.4. Cultural Heritage Chatbot 11](#_Toc109446481)

[2.3. Review of E-commerce Chatbot Systems 12](#_Toc109446482)

[2.3.1. E-Commerce Sales Chatbot 12](#_Toc109446483)

[2.3.2. E-Commerce Chatbot With Telegram 14](#_Toc109446484)

[2.3.3. University Shopping Mall E-Commerce Chatbot 16](#_Toc109446485)

[2.3.4. Knowledge-Based Customer Service Chatbot 17](#_Toc109446486)

[Chapter Three: System Analysis and Design 20](#_Toc109446487)

[3.1. Preamble 20](#_Toc109446489)

[3.2. Requirements Analysis 20](#_Toc109446490)

[3.2.1. Functional Requirements 20](#_Toc109446491)

[3.2.2. Non-Functional Requirements 21](#_Toc109446492)

[3.3. System Architecture 21](#_Toc109446493)

[3.4. System Design 23](#_Toc109446494)

[3.4.1. Logical Design 23](#_Toc109446495)

[3.4.2. Data Flow Diagram 23](#_Toc109446496)

[3.5. Physical Design 24](#_Toc109446497)

[3.5.1. Use Case Diagram 25](#_Toc109446498)

[3.5.2. Activity Diagram 26](#_Toc109446499)

[3.5.3. Sequence Diagram 27](#_Toc109446500)

[Chapter Four: Sytsem Implementation 28](#_Toc109446501)

[4.1. Introduction 28](#_Toc109446503)

[4.2. System Requirements 28](#_Toc109446504)

[4.3. System Modules and Interfaces 29](#_Toc109446505)

[4.3.1. Telegram Interface 30](#_Toc109446506)

[4.3.2. Python Tkinter Interface 32](#_Toc109446507)

[4.3.3. Ngrok Web Connector 34](#_Toc109446508)

[4.4. System Evaluation 35](#_Toc109446509)

[Chapter Five: Conclusion and Recommendations 42](#_Toc109446510)

[5.1. Summary 42](#_Toc109446512)

[5.2. Recommendations 42](#_Toc109446513)

[5.3. Conclusion 43](#_Toc109446514)

[References 44](#_Toc109446515)

LIST OF FIGURES

[Figure 3.1: System Architecture for the Context-Aware Chatbot 21](#_Toc109460886)

[Figure 3.2: Level 0 Data Flow Diagram 22](#_Toc109460887)

[Figure 3.3: Level 1 Data Flow Diagram 23](#_Toc109460888)

[Figure 3.4: Chatbot Use Case Diagram 24](#_Toc109460889)

[Figure 3.5: Chatbot Activity Diagram 25](#_Toc109460890)

[Figure 3.6: Chatbot Sequence Diagram 26](file:///C:\Users\Laotan%20Faji\Documents\School%20folder\400%20level\Alpha%20Semester\Project\Project%20Report\Laotan%20Project%20Standard%20Redo.docx#_Toc109460891)

[Figure 4.1: Afrimash Chatbot Telegram Interface 30](#_Toc109460892)

[Figure 4.2: Afrimash Chatbot Telegram Interface 30](#_Toc109460893)

[Figure 4.3: Python Tkinter Interface in the English language 31](#_Toc109460894)

[Figure 4.4: User Interaction in the Yoruba language 32](#_Toc109460895)

[Figure 4.5: Usability Testing Question 1 34](#_Toc109460896)

[Figure 4.6: Usability Testing Question 2 34](#_Toc109460897)

[Figure 4.7: Usability Testing Question 3 35](#_Toc109460898)

[Figure 4.8: Usability Testing Question 4 36](#_Toc109460899)

[Figure 4.9: Usability Testing Question 5 36](#_Toc109460900)

[Figure 4.10: Usability Testing Question 6 37](#_Toc109460901)

[Figure 4.11: Usability Testing Question 7 37](#_Toc109460902)

[Figure 4.12: Usability Testing Question 8 38](#_Toc109460903)

[Figure 4.13: Usability Testing Question 9 38](#_Toc109460904)

LIST OF TABLES

[Table 3.1: Use Case Diagram Actor Documentation 26](#_Toc109445217)

[Table 4.1: System Requirements 28](#_Toc109445218)

[Table 4.2: Hardware Deployment Requirements 29](#_Toc109445219)

[Table 4.3: Software Deployment Requirements 29](#_Toc109445220)

ABSTRACT

The continuous advancement of Artificial Intelligence technologies means that businesses which fail to innovate will die. Agricultural e-commerce is a bourgeoning field of endeavour in Africa. Improvements in Human-Computer Interaction allow companies to use tools such as chatbots to improve customer engagement and increase business viability.

This project aims to develop an intelligent chatbot that can interact with users of agricultural e-commerce websites in Nigeria and enhance user experience using Natural Language Processing tools and techniques. The chatbot deployment is on the Afrimash website. It enables them to access the features, products and services by communicating with them in various languages and allowing them to access the components.

This project used the Python Rasa framework to develop the chatbot system deployed on the Afrimash website. Webhooks on Telegram allow seamless integration between the messaging app, the chatbot engine and the Afrimash website. Finally, the Google translate API is the engine for processing Yoruba language intents.

The solution proffered improves the ease of use of the website and promotes customer retention and user satisfaction to the users of the website used as a case study. This work contributes to the more significant works of literature which are scarcely available on this subject.

The results of the study show that 90% of the website visitors were satisfied with the chatbot and were willing to use it again as a means of interacting with the content on the website, which already had a poor user experience. On a scale of five, the chatbot was handed a rating of 3.6 in terms of satisfaction with the solution. However, most users were dissatisfied with the chatbot's inability to translate and deliver results in the Yoruba language accurately.

CHAPTER ONE

# INTRODUCTION

## Background Information

Of the various emerging technologies over the last decade, Artificial Intelligence (AI) has piqued most of the attention of researchers and the industry. Artificial Intelligence describes the concept and implementation of computer systems that can perform activities that would ordinarily require the inputs of humans, like speech, language translation and decision-making (Manne and Kantheti, 2021). Empirical and statistical evidence shows that companies are employing artificial intelligence and machine learning algorithms to procure solutions to their challenges. This occurrence is due to the new avenues they provide for making precise extrapolations and inferences from vast amounts of data (Bertacchini et al., 2017).

These technological advances have created new avenues for interaction among businesses and their customers by enabling them to deliver consistent, innovative, customised products. AI, for instance, can now assist companies in providing online customer support beyond the remits of traditionally available means. A succinct example is the trending use of digital assistants or chatbots in customer relationship management activities (Huang and Rust, 2018).

Chatbots are now an essential part of websites for companies competing in the modern and emerging markets dominated by young populations. Due to rapid advancements in Natural Language Processing (NLP), chatbots promise to hold the ace in enhancing customer experiences by offering customers more personal interactions than available on menu-based web applications. The degree of influence of chatbots is as profound as affecting the attitude, level of satisfaction, and emotional connection consumers feel during their service encounter on a service providers-website.

In agriculture, the massive improvements in internet infrastructure and technologies hold great potential for the future of agricultural developments. Access to the internet is pivotal to farmers, especially regarding business purposes. Farm income can significantly increase by leveraging the available facilities on the internet to cut transaction costs (Hennessy et al., 2016). The internet and its facilities are thus handy tools that enable greater access to the acquisition of prices, product information and customer relations supported by larger pools of contractors and clients (Zapata et al., 2016). These facts make e-commerce an essential player in the world of agriculture.

The deployment of a context-aware artificially intelligent chatbot on an agricultural e-commerce service provider's website revolutionises the relationship between farmers and the consumers of their products in this digital age. A digital assistant on an agricultural website bolsters sales potential by offering a more personalised experience with customers while using the platform. Including local language translators also improves the experience, promotes inclusivity and helps non-English speaking users communicate with the chatbot and navigate the website more efficiently.

It is vital to leverage the strengths of emerging technologies to improve customer experiences, improve the sales output of farmers and hence enhance the growth of the Nigerian economy at large and the overall performance of e-commerce websites servicing the agricultural industry. However, in sharp contrast to the expected influx of solutions to buttress technological advancements in the field, there is a stark lack of literature and implemented solutions addressing the demand in this field.

## Statement of the problem

This project aims to address an underserved area of research: improving Human-Computer Interaction on agricultural e-commerce platforms. This project uses the Afrimash.com website as a case study. Currently, users of the website lack easy access to information about products and services on the Afrimash e-commerce website due to a poor user experience. The chatbot implemented in this paper increases the website's ease of use by granting users seamless access to functionalities that are ordinarily difficult to locate. It also goes further by integrating the parlance and language of the local people of southern Nigeria, where the business is located, to increase context-awareness and improve usability. This project aims to develop a chatbot that allows users to communicate with the system via chat and have a better experience on the Afrimash website, leading to customer retention and increased profits.

## Aim and Objectives of the study

This project aims to develop an intelligent chatbot that can interact with users of agricultural e-commerce websites in Nigeria and improve user experience using Natural Language Processing tools and techniques. The chatbot deployment is on the Afrimash website. It enables them to access the features, products and services by communicating with them in various languages and allowing them to access the features.

The objectives are:

1. To investigate and document the requirements for the chatbot.
2. To design and implement the interface for the chatbot.
3. To create the knowledge base for the chatbot.
4. To integrate a text analysis and translation module in the chatbot.
5. To integrate natural language processing in the chatbot.

## Methodology

The completion of this project would involve sourcing data and inputs for the chatbot system from the Afrimash website used as a case study for this project. It entails applying various methodologies, which are as follows:

1. Requirements gathering for the system by interviewing relevant experts in the related fields.
2. User interface design using Python Tkinter module/Telegram web interface (frontend design).
3. A knowledge base was created using a file-based database system.
4. The Chatbot engine is deployed with Natural Language Processing capabilities using the Rasa Core and the integration of the Google translation API to allow for communication in local dialects.
5. The above features are functionally designed and modelled with the Rasa NLU framework technology to build the Natural Language Processing Unit.

## Significance of study

This project is relevant to society because agriculture and e-commerce are two paramount factors in modern human existence. Beyond that scope, however, this project is relevant for the following additional reasons:

1. The study results in the design and development of a system to render ease in accessing farm produce via an e-commerce website.
2. The system provides a database to organise all data and keep track of records in

the system.

1. The system enables farmers and end-users to access features on the website in their local languages.
2. This project fosters and engenders faster transactions in agricultural e-commerce, leading to massive economic benefits for the nation.

## Limitations of the study

The requirements gathering of the developed system is constricted to the Afrimash e-commerce website used as a case study for the project.

## Arrangement of thesis or research outline

Chapter One of the project contains an explanation of the project, problems on the Afrimash website, the need for such a system, the method of implementation, the significance of the study, and the study's limitations.

Chapter Two explains the existing system relating to the project topic, the methodology and the algorithm used in related systems.

Chapter Three describes the analysis and system design.

Chapter Four shows the implementation of the system step-by-step and the results obtained.

Chapter Five summarises the project and gives the subject's recommendations, suggestions and conclusions.

CHAPTER TWO

# LITERATURE REVIEW

## Preamble

The rapidly evolving technology landscape is revolutionising how business is done regardless of the field of operation and influence. Artificial intelligence has transformed how humans and computers interact and birthed a world of potentially endless possibilities in all human endeavour. Human-Computer interaction is one of the many fields that the advent of improving technologies has revitalised. The efficacy of the technology's influence is evident in the sheer number of chatbot systems deployed in various modern computer interfaces.

This chapter reviews existing systems that have applied related methods to the ones employed for this project. It begins with a review of different approaches to chatbot creation employed by the various systems. Then, it continues with validating the problems they solved vis-à-vis their implementation methods and verifying the effectiveness of their solutions and concludes with the challenges faced during their development.

### Review Of Chatbots

A chatbot is an Artificial Intelligence program that enables and bolsters human interactions with computer systems. The interaction comes in a conversation and may be implemented by creating text or voice interfaces for communication. Chatbots can be used as tools embedded in websites or other applications to foster interaction or as standalone bots. Chatbots have existed for a long while and were initially developed as conversational or chitchat software systems (Weizenbaum, 1966) and task-completion systems outside e-commerce-related settings.

The Turing test (Turing, 1950) sparked researchers' and engineers' research efforts to develop conversational applications using audio or text. Those agents are designed to simulate human conversations, succeeding if they can deceive their human counterparts and pass the test. *Eliza,* for example, is perhaps the oldest chatbot application ever created. Shum et al. (2018) noted that the software was developed in the 1960s using a simple pattern-matching system to mimic a real-life conversation between a psychotherapist and a patient (Dale, 2016). Weizenbaum's (1966) hand-crafted, text-based scripts are designed to simulate a session with a Rogerian psychotherapist. The software does not necessarily understand the meaning of the conversation. However, it employs pattern matching and phrasing systems to respond to patients. Shum also found that *Eliza* successfully inspired a more profound interest in others to build chatbot agents that could pass the Turing Test.

Colby (1975), a professor of psychiatry, developed a chatbot called *Parry* designed to behave like a person suffering from paranoia. It is rule-based and resembles *Eliza* in structure and capability with a model that can simulate emotion. Parry responds harshly to the other party if there are high tempers in the conversation. When the bot was developed, humans conversing with it could not distinguish between Parry and a paranoia patient. The chatbot was the first machine to complete the Turing test (Shum et al., 2018).

Wallace (2010) developed a chatbot called *Alice* with a heuristic-based pattern matching system to create rules for the input in a conversation. Alice uses Artificial Intelligence Markup Language (AIML) to specify rules for the conversation. Mittal et al. (2016) observed that Alice understood complex user inputs, recalled users' names, understood and responded to users in multiple languages, and retained the details of historical conversations. Alice was also able to respond in the event of a lack of adequate information in the user input or if the user's statement was simply too difficult to understand. It would ask the user to search for the answer to their questions instead of simply admitting to not knowing the answer. Mittal et al. also noted that Alice would get offended if the user insulted it, as a human would react in the same setting.

## REVIEW OF EXISTING SYSTEMS

This section contains a critical analysis of relevant literature on the aim of this study and the methods employed for the project. It details a review of chatbot systems developed for various purposes with architectures and frameworks relevant to this work, then a review of chatbot systems used in e-commerce. This chapter serves as a guide toward the implementation methodology used in developing the software in this project.

### Neu Chatbot

In this paper, Nguyen et al. (2021) built a chatbot that ameliorates the burden of admission counselling at the National Economics University Vietnam. They developed the NEU chatbot, which provides admission information to prospective students at the university to cater to the high demand for admission-related questions posed to the university.

The authors noted the prevalence of chatbot systems in diverse fields, including education and the rising demand for online consulting services. They also observed the manual communication between potential students and universities, noting that it is time-consuming, cumbersome, and a burden to the officers involved. Using Deep Learning models built into the Python Rasa framework, they developed an AI-based chatbot on which students can access updated admission information during their application process. The chatbot was integrated into the official university Facebook admission platform, which is a very well-known channel in the local community.

Chatbots are software agents that interface with users by communicating in human languages (Følstad & Brandtzæg, 2017). Recently, this software has broken into various fields, including commerce business, health, e-commerce, and entertainment. In education, chatbots are necessary to answer users' most pressing questions related to a topic – in this context, a university – without human interference. With such a program, students can instantly access admission information, regardless of timing and location.

In a broad view, there are essentially two classes of deep learning chatbots: retrieval-based and generative bots. Since they offer more flexibility than these bots, retrieval bots have slight nuances that distinguish them from rule-based bots. It builds classification models to extract intents and other necessary details from user input and select an appropriate response from the database in which it has the most confidence. Conversely, generative bots generate answers based on present and past user input. These chatbots are more similar to humans than retrieval-based bots due to their memory savings. However, generative bots are generally impractical, usually make grammatical errors, and supply inconsistent replies.

The authors of this paper chose the Rasa framework to implement their software due to its open-source nature, making it easy to integrate and personalise. Furthermore, Rasa supports cross-platform connectivity and integration and uses deep learning and the chatbot framework, which helps channel the deep learning models integrated into the Rasa framework. Rasa is a library of python tools with appropriate features for NLP projects and machine-learning-based dialogue management (Bocklisch et al., 2017)**.**

Nguyen et al. (2021) used the Rasa NLU and Rasa Core components to develop their chatbot. The Rasa Core controls conversation flow, actions, and words. At the same time, the NLU (Natural Language Unit) understands, classifies, and extracts text entities from the input.

The chatbot development processes are segmented into a series of operations. Data entered into the system first undergoes preprocessing as it significantly affects supervised machine learning algorithms. The preprocessing stage is based on four factors, namely, the addition of diacritics, text cleaning, stop word removal, and number-to-text conversion.

Following the preprocessing step, the researchers selected the Tokenizer and Featurizer NLU pipelines from the Rasa framework. Integrated into the chatbot is a custom action system that serves as a fallback for low-confidence inputs. Finally, a connector module links the chatbot to Facebook Messenger.

The NEU chatbot services the need for accurate information that prospective university students have during their admission process. The solution gives university admission officers less work and reduces the likelihood of incorrect or inconsistent information reaching candidates. By employing the techniques used, people can quickly adapt the chatbot and bring solutions to their problems using the Rasa framework.

Although the chatbot has attained 97.1% in testing and practical application at the National Economics University, it has some lingering limitations. The content of the file which supplies intents to the bot must be updated manually every session with fresh intents and information about the university. The bot will, however, adapt shortly after that, with improved accuracy. In the future of this work, the researchers intend to integrate voice assistance from the Google Assistant platform and a feature which helps students identify their personality types and suggests an appropriate field of study for them.

### Chatbot For Chronic Patient Support

Chatbots in eHealth systems and services have proliferated dramatically in recent times. However, Roca et al. (2020) noted that the design of such agents must be improved to accommodate the diverse requirements for chronic patient support to actualise the full potential of chatbots in eHealth. Three critical pillars must be integrated into chatbot design to achieve this aim: scalability, standard data models, and standard conversational modelling. Lack of scalability is a particularly crucial challenge to the growth of the chatbot software's robustness with time. Chatbots must be built on a modular and flexible architecture to address the issue of scalability. Microservices are an architectural paradigm that emphasises modular, lightweight services with a high degree of cohesion. Systems built with Microservices are easy to build, can quickly scale, and are straightforward in deployment.

The microservice architecture used in this paper has been developed in Java (using Java 8 Update 121) using the open-source Jersey framework supporting JAX-RS APIs to implement the RESTful service and the Grizzly framework to implement the HTTPS server. The Python version employs the Flask framework, while Gunicorn serves the Flask application. All the microservices built in this paper deploy an asynchronous HTTPS server that employs TLS for the client's authentication.

The aim of the study is to create personalised eHealth services using virtual assistants that are built using the microservice architecture. It is designed to evade the pitfalls laden in other mobile applications which are custom-made for particular diseases. Future additions to this work include speech recognition functionalities and advanced automated image processing to facilitate diagnosis.

### Ask Rosa

(Siglen et al., 2021**)** noted that the integral function of genetics has become an integral part of breast cancer diagnostics, treatment and follow-up, consequently raising the need for genetic information and counselling services.

The authors aimed to implement an app-based chatbot called Rosa, built as a credible, available, and custom-made source of information pertaining to congenital breast and ovarian cancer. In doing that, they also aimed to share their wealth of experience and document the challenges faced and the industry best practices in the field. The authors built a pilot version of the Rosa application, which serves as a conversational agent for sufferers of ovarian and breast cancer. The application services them with information about generic BRCA testing and uses a chatbot to note the best practices for future chatbot applications with similar functionalities.

The database which serves the chatbot comprises a combination of predetermined and defined questions and corresponding responses (outcomes). This combination is referred to as a dialogue. Each defined question can only appear in one dialogue. In the event that the AI-based matching produces more than one match between a user-defined and a system-defined question, the chatbot cannot correctly specify the answer because it has no clue to which question it should reply. In such an incident, the response of the system may either be correct, partially or downright wrong.

If the prompt provided by a user does not tally with a match in the database, the chatbot selects a fallback answer, stating that it does not understand the question and asks the user to enter a different combination of words.

Besides the database system, a mobile application and a web Application Programming Interface (API) were implemented to maintain constant communication between the patients and the Natural language Processor (NLP). NLP (Natural Language Processing) is a subfield of Artificial Intelligence that helps computers to comprehend, interpret, and manipulate human languages.

The researchers built their mobile application using Ionic, a cross-platform programming language for building mobile apps. An interface built on the Angular JavaScript framework for building scalable web apps, and Cordova, a cross-platform mobile development framework. The applications communicate over a web API that sends users' questions to the Natural Language Processor, receives the response, and returns the answer using the application. The web API between the client and the processor keeps communication with the NLP provider discrete. An administrator web application platform built using the Angular framework monitors and records conversation history between users and the chatbots.

The authors of this paper noted a challenge with the system that required representatives of cancer patients to be present at a workshop that trains the bots. To surmount the limitation, they interviewed the patients to whom the Rosa application had access in the past during and after their genetic tests. Results from the test are to be built into the final version of the application, consequently ending the pilot phase of constructing the app. Other related work in the future, as suggested by the author, should include research on fallback answers to reduce the levels of confusion during chatbot-user dialogue and the development of strategies to implement chatbots in healthcare services successfully.

### Cultural Heritage Chatbot

In this paper, (Sperlí, 2021) proposes a framework for promoting cultural heritage that supports the journeys of tourists and creates an interface for chatbot conversations using the Seq2Seq model to allow easy responses to its needs. Alternatively, it aids private and public organisations alike in promoting the tangible and intangible aspects of their cultural heritage.

The employed framework entails the creation of a chatbot interface based on a seq2seq model. This model uses a Multi-level Gated Recurrent Unit cell to enable the system to provide more accurate answers and improve its performance through constant interaction with the users of the system.

The author implemented a microservices architecture to provide the various services and events and suggest paths by combining users' historical activities and profiles defined on Mairesse features.

The chatbot agent is built using a mix of Natural Language Processing (NLP) and Artificial Intelligence techniques to integrate the predilections of users as collected while the user engages the platform. The interface is supported by a microservices infrastructure in the backend to avail various services for aiding tourist visits, descriptions of tourist locations, and other features. An Enterprise Service Bus (ESB), which analyses data from multiple sources and organisations, is integrated into the system. This information is accessible to any external applications or guests and can be used to provide a sequence of suggestions for tourist visits for users whose location has been identified.

The chatbot engine is built on the seq2seq model, which is built to compute a sentence of varying length using a model with a stable (fixed) dimension. The model is based on the Encoder-Decoder framework, which uses RNN (Recurrent Neural Network). It relies on three parts: the encoder, the encoder vector, and the decoder. The encoder's function is to morph input prompts or statements into a series of points in space, which enables the decoder to generate the output.

Future works will entail devotion to the extension of the evaluation functionalities, employing a broader dataset and user sample size to improve the Deep Learning capabilities of the system.

## REVIEW OF E-COMMERCE CHATBOT SYSTEMS

A chatbot is an Artificial Intelligence-based software that creates interaction between a human user and a computer system in a natural language, thereby simulating a human conversation. It creates a conversational interaction between the computer and the user that makes them believe they are chatting with a human being. In contrast, they are conversing with the computer system. (Shingte et al., 2021).

Shawar (2007) maintains that chatbot systems are not built solely for human entertainment or to mimic human conversations but have far-reaching applications evident in fields such as education, business, and e-commerce.

### E-Commerce Sales Chatbot

Khan and Eshan (2020)developed a modular chatbot architecture that improves customer support and aids sales. The software was built using machine learning to help it process natural languages. A microservice architecture was used to help classify user inputs.

The authors of this paper observed that the growth of e-commerce in the past 20 years impacted the way business is conducted in society. Online shopping has positively impacted both businesses and consumers around the globe, leading to a greater reliance on e-commerce than ever previously observed. A succinct illustration of this adaptative change is reflected in how e-commerce giants such as Amazon have taken a vast share of the markets of conventional stores such as Walmart (Phillips and Banjo, 2015).

A significant drawback of e-commerce is that some customers – particularly elderly ones – are used to having face-to-face interactions with sales representatives and may not necessarily be comfortable engaging via electronic platforms. Another possible albatross to the full adoption of e-commerce is the lack of guarantee on return and warranty policies due to the absence of a human representative.

The solution proposed by the authors is intended to bring the frontiers of Natural Language Processing and Natural Language Understanding closer to understanding context.

The chatbot designed by the authors is aimed at developing a modular architecture to improve accuracy and allow for easier integration of new features. Microservice architecture is used in the study rather than building a monolithic system. The system comprises four major components:

1. The NLU Engine
2. Recommendation Engine
3. Adaptive Pricing Engine
4. Bot Engine

The NLU engine is a crucial component of the system built as an HTTP server which accepts textual inputs and returns intents, entities and the accuracy level in a JSON string. The recommendation engine is used to identify goods which users tend to buy. Discounts and deals generated in real-time are delivered to the user via the adaptive pricing engine based on data it receives from the recommendation engine. It uses user history as a heuristic for a fuzzy engine to provide discounts for each customer (Prodromou, 2015). The chatbot engine is the backbone of the system. The engine is linked to the Natural Language Unit and adaptive pricing engines and classifies user intents through the NLU engine. It employs a routine engine which sends user requests to a controller that generates replies for users. It comprises an input handler, NLU router, Middleware, response handler and an output handler.

The system trains the NLU engine via a visual platform built with Django, Vue JS and MySQL. The application interface services the login platform and helps reduce calls to the central server.

1. It has the potential for greater accuracy due to its use of Artificial Neural Networks
2. The system can be made available on WordPress, the most popular e-commerce platform solution.
3. The project helps improve the relationship between businesses and their customers.

The system is built in a modular architecture to allow it to be proliferated across more platforms. It uses an NLU engine that trains its classifier model using training data supplied by the system administration and is hinged on the Support Vector Machine. The accuracy of the chatbot is improved by employing an Artificial Neural network. The software is accessible to WordPress-based systems since 60% of eCommerce platforms are powered by the word press plugin called WooCommerce. Integrations may be deployed for use on Shopify.

### E-Commerce Chatbot With Telegram

To develop a cross-platform e-commerce chatbot agent, Asadi and Hemadi (2018) consumed the Telegram API for their implementation. The chatbot design is such that it can be deployed on any messaging platform, including Facebook and Telegram. In addition to the use cases on Telegram, the bot can be deployed as a pop-up window of a browser or on Facebook messenger.

The authors described a chatbot as a conversational interaction tool that bridges the gap between humans and conversations. According to Tech Target (2017), a chatbot is software that deploys instant messaging as the user interface, such that messengers can add the name of their bot to their contact list much like their human counterparts.

Interaction with chatbots used for commercial purposes can be described as "chat-commerce". Chat commerce is most prevalently used by WooCommerce, because of its relative popularity as a user interaction software. Builtwith (2018) states that it is used by 43% of the internet. This informed the choice of WooCommerce as the choice solution for the implementation of this software.

In the second section of the paper, the authors noted that customers are of different levels of computer literacy; thus, it is advantageous to build a simple bot designed to increase customer satisfaction for both tech-savvy and non-tech-savvy users. A product recommender system was integrated into the system to enable it to fulfil two primary purposes: order taking and recommendation system.

The order-taking system allowed the users (sellers) to present their merchandise via social media, catalogues, emails, graphics, and other electronic means. The system generates a unique product ID for each product for which the user is prompted once the conversation with the chatbot is initiated. It follows up by sending an image depicting the product alongside a description and asks the user to confirm it relates to their search. The chatbot then recommends the product to the user and completes the transaction by obtaining user details and payment methods. The bot helps fulfil transactions and complete marketing endeavours at the conversion stage.

The recommendation system uses data already stored in the WooCommerce system, including all the product details ranging from product taxonomy, sales, publish date, and rating. However, the chatbot is unable to detect what products are relevant to the users but can help users know the category. It can answer questions, including the newest product, cheapest and best-selling products in a particular category. Users can also select order tracking by entering their order ID.

To employ the Telegram API, Asadi and Hemadi (2018) used the PHP language to create the chatbot. The chatbot uses a MySQL database to store customer data and order statuses. Telegram uses a bot token feature to authorise bots for deployment on their application. The authors obtained a bot token on Telegram and used a webhook deployed on a server to handle messages sent on the chatbot.

Bots deployed on Telegram have a menu feature that allows users to see the services offered at one glance. Two options are available on the bot – "Quick buy" and "track order." A session is initiated when a user enters a product ID sent through the Telegram API to the webhook, which sends a request to the WooCommerce shop API.

Images and information related to the product ID are sent as a reply to the customer. Customers are identified by a unique customer ID and a state such as "quick buy" if the quick buy option is selected. After each conversation stage, the customer's state is updated in the database.

When a customer is satisfied with the suggested products and has supplied shipping information to the bot, it prompts the user to select a shipping method. It sends a request to the WooCommerce API for order placement. The API sends a response containing an order ID synthesised by the bot to generate a payment link to send to the customer—the bot stores the customer and order information in the database for reference purposes. Customers interested in tracking the progress of their orders can choose the "Track Order" option. Orders can then be tracked using the order id, which is sent to the WooCommerce API and returned to the customer as the current status of their order.

1. The system can be deployed over various social media platforms.
2. The chatbot works cross-platform and is supported by multiple system architectures due to its design.
3. The system is simple to use.
4. The chatbot can only be deployed on the WooCommerce System.
5. Users need to know a product ID to use the chatbot.
6. The recommender system has limited functionality.

The paper describes the authors' design and implementation of a chat commerce conversational agent. The bot aims to improve user interaction for business marketing over social media. There should, however, be more extensive research to probe the degree of effectiveness of the software.

Significant limitations of the software include the fact that the chatbot can only be used for the WooCommerce system and are limited to data source shops. It reduces the likelihood that other data sources can be connected to the system to increase user satisfaction with the recommender system**.**

### University Shopping Mall E-Commerce Chatbot

In this paper, Oguntosin and Olomo (2021)developed a conversational agent for the Covenant University Shopping Mall, which serves the local community at the university. The "mall" is essentially a supermarket where the members of the community purchase items, including groceries and other essential items of everyday need. Students in the university cannot inquire about the availability of products in stock due to the lack of an online inventory system in the shopping mall. The chatbot built in this paper attempts to conquer the challenge by allowing students to make enquiries about products they wish to obtain and make payments on the internet, significantly reducing the time spent shopping. Students can access the bot via mobile devices and other computers, which they can log into ubiquitously on campus.

The researcher built the frontend of the chatbot using React JS and used datasets and deep Natural Language Processing on the server-side. The researcher also developed a database of items stocked in the database and introduced administrator privileges to allow staff to update the product stock list and stock available per item. Python and MySQL form the codebase of the software's backend, forming the bot code and the database, respectively.

The machine learning portion is built using an open-source Python library, Spacy, and Recast.ai. Spacy is mainly used for tokenisation, lemmatisation, and linguistic annotations features. The Recast.ai API is NLP enabled and services the creation, training, and monitoring of the progress of conversations with the chatbot built in this study. A data layer written in MySQL gives the chatbot a structure to answer product-related questions.

The software has two interfaces called the admin section and the user section subdivided into the front and back ends. The admin webpages and the chatbot interfaces are contained in the frontend, while the backend includes the machine learning module that runs the chatbot and the database.

The chatbot is accessible via the internet, making it ubiquitously available to university community members round the clock, allowing students and other customers to make purchases of goods in stock before arriving at the mall. The development of this application assuages the burden of time wasted and long journeys to the shopping mall only to discover that the items they require are unavailable. Features that can be integrated into the bot include the addition of a barcode reader, training it with larger datasets, and increasing the list of items in the chatbot database to improve the user experience.

### Knowledge-Based Customer Service Chatbot

Ngai et al. (2021) stated that one of the most promising applications of AI is chatbots, among other frontiers. Conversational agents are growing increasingly popular, with institutions such as the Bank of America launching their chatbot agent, Erica, with ten million customers using the software (Bank of America, 2019). Chatbots are used in healthcare, financial services, education, and other fields (Kerlyl et al., 2006), (Oh et al., 2017). The author noted that chatbots are becoming associated with the way customers relate to the brand.

In e-commerce, chatbots can assist businesses by delivering timely answers to the many pressing questions of their customers. They can also pave the way for improved allocation of scarce human resources. However, the author notes that the performance of chatbots is not always at par with customer requirements. Some chatbots supply unreliable replies to their users, leading to a void between the expectation of customers and the experience delivered by the chatbot.

To improve the knowledge performance of such chatbots, they are integrated with a knowledge base for a more dynamic conversation. Chatbots have access to the knowledge base and can employ the data it finds in the base via search to present a more custom-tailored response to the customer. Ngai et al. (2021)observed that research on the design of Knowledge bases for chatbot applications is not very available and thus proposed a knowledge base whose framework includes customer Knowledge management to create a chatbot architecture that intelligently and endlessly improves itself.

Wilde (2011) identified that customer knowledge needs adequate management to create and sustain a good customer relationship. Customer Knowledge Management (CKM) revolves around obtaining, sharing, and expanding customer knowledge and partnering with them to jointly create value (Gibbert et al., 2002).

According to Wilde, customer knowledge is divided into three classes: knowledge about the customer, knowledge from the customer, and knowledge for the customer. Customer knowledge includes age, sex, transaction records, payment behaviour, incentives, and purchasing behaviours and expectations. Knowledge for the client assists the customer in closing any knowledge gaps discovered by the firm. This knowledge includes details on a product's quality, pricing, features, and special offers. As a result, the KB should include the three categories of information outlined above to assist chatbots' contact with customers to benefit both businesses and customers.

The chatbot proposed in this paper is a knowledge-based chatbot agent built for customer support and management. Overall, this system comprises four key sections: the knowledge module, dialogue module, handover module, and adapter. The dialogue module collaborates with the knowledge module to construct pseudo conversations that are used to connect to users via the adapter, which connects to the UI. Meanwhile, by passing information, the handover module acts as a mediator. The adapter provides customer assistance to customers. The knowledge module collaborates with human specialists to develop the Knowledge Base in a proactive manner. The dialogue module handles conversations with the customer and must understand natural languages to achieve this feature effectively. The language content used by the software implemented suggests topics, keywords and entities, while the use of language simply signifies the intents of the sentence. The module understands customer intents by combining pragmatic, semantic and syntactic analysis results.

The knowledge module includes a knowledge base and other components that allow for proactive knowledge improvement. Information in the base is stored in question-and-answer format and is refreshed by web-crawler software. The crawled data is stored and processed in the database before being classified and organised. After a comparison between the new and old question and answer pairs, new replies to existing queries can be discovered.

The handover module enables the system to alternate between a human service agent and the chatbot when it is unable to respond to the queries supplied by customers. The adapter allows the bot to interact with multiple platforms which support chatting, including Skype, Facebook and Instagram. It collates inbound messages from customers and returns a reply via the dialogue module before replying vi the chaat interface. Pre-processing includes processes like tokenisation, sentence breaking, chunking, and lemmatisation, which are enabled by human language.

CHAPTER THREE

# SYSTEM ANALYSIS AND DESIGN

## PREAMBLE

This chapter contains the details and design of the agricultural e-commerce chatbot system. It contains the various methodologies and procedures taken to achieve the eventual outcome as well as diagrammatical representations of the system's design. This chapter also documents the requirements for the system and the design processes involved in fulfilling the stated requirements. Highlights of this section include the architecture, design, modules and interfaces for the chatbot built to satisfy user requirements.

## REQUIREMENTS ANALYSIS

In Software Engineering, requirement analysis is the process of obtaining the expectations and needs of users for a software to be created or modified. The task of eliciting requirements includes multiple stakeholders and various activities to ensure that product specifications are accurately and relevantly obtained. Software requirements are generally classified into functional and non-functional requirements. The requirements for this project are detailed in the following paragraph.

### Functional Requirements

Functional requirements are the critical functionalities that must be built into a system to enable users to accomplish the objectives and tasks of the system. They are specifications for what the system must do and what features and functions must be integrated into the system. The functional requirements for the agricultural e-commerce chatbot are as follows:

1. Users will be able to choose a language of communication between Yoruba and English.
2. Users will be able to complete conversations with the chatbots in both Yoruba and English language.
3. Users will be able to make enquiries about products available on the website via chat.
4. Users will be able to access responses to their requests by chat.
5. Users will be able to complete transactions via the chatbot.

### Non-Functional Requirements

A system's non-functional requirements stipulate how a system should work as opposed to what it should do. They are the properties of the system tailored towards bolstering the core functionalities of the system and improving user experiences. The non-functional requirements for the agricultural e-commerce chatbot are as follows:

1. The system shall maintain a simple and easily usable interface for users.
2. The clients' user interface should work seamlessly with regularly employed browsers, including Microsoft Edge, Google Chrome and Safari.
3. The system shall integrate seamlessly with the Telegram platform as an interface.
4. The system shall be easily scalable to accommodate an increasing number of users.
5. The administrator account and database should be protected from unauthorised access.
6. The system should seamlessly engage the Google translate API for language translation.
7. The system should be compatible with multiple operating systems supporting the Python language.
8. The system should be easily maintainable.
9. Exceptions and errors should be easily understood and manoeuvrable by the user.
10. The system shall not process any information about its users for any purpose.

## SYSTEM ARCHITECTURE

System Architecture is a high-level design that describes the structure and behaviour of a system as well as the relationships between the various components comprising the system.

The architecture used to build this project is based on microservices. The software is split into five modules. It comprises a Natural language Understanding (NLU) unit built using the Rasa NLU framework (Rasa version 3.1.0) of the Python programming language (version 3.9.6), which uses a trained machine learning model to process user requests. NLU unit has three functions which are dialogue act and intent classification, and slot filling.

Dialogue act classification involves ascertaining the type of dialogue act a user inputs or assigning an utterance to a specific dialogue action. Intent classification involves confirming the goal of the user, while slot filling helps the chatbot extract other critical information, which allows the agent to fully understand the context and meaning of the user's request. The Natural Language Unit also includes the Google Translate API module, which processes user inputs in the Yoruba language.

The system uses a Dialogue Management (DM) unit to process the intents of users before executing an action based on the classified results. It handles the information flowing in from other modules of the bot and updates the context of conversations as well as directs the actions of the chatbot. Furthermore, the dialogue management unit handles errors from the information it receives via the error handling module. The DM unit sends the classified intent of the user to the data sources, comprising a knowledge base and data stored in the web server's database. Access to this data is granted by URL requests sent by the bot directly to the server of the website in use.

Processed data from the data sources are sent through the information retrieval module, which works in tandem with the knowledge base on the backend. This serves to store the rules of the chatbot, which is essential to its functionality. This unit passes results to the response generation unit, which either directly supplies the user with a response or translates the response before sending it out.

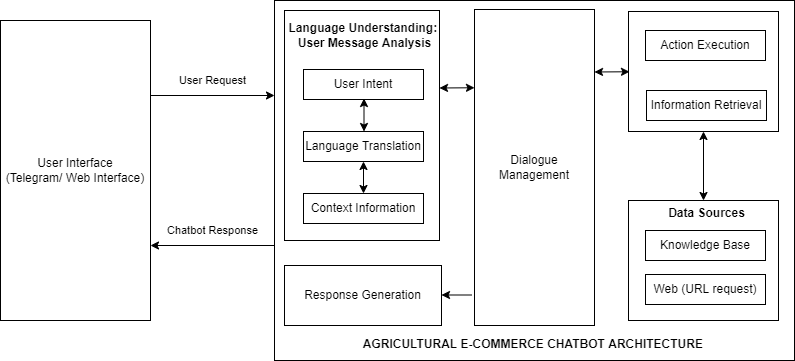


Figure 3.1: System Architecture for the Context-Aware Chatbot

## SYSTEM DESIGN

System design is a critical part of the software development process that describes the process of implementing the objective of a system to be built. Various aspects of system design to be considered include conceptual design, logical design, physical design, and architectural design.

### Logical Design

Logical design involves detailing the flow of data in and out of a software system. It includes identifying all input and output sources and destinations as well as the data stores and flows concerning the system. The logical design of this chatbot system is described using data flow diagrams.

### Data Flow Diagram

A data flow diagram is a visual map that shows the channels of flow of information among processes in a system. This diagram uses different symbols and notations to signify various parts of a system through which information flows, including processes, external entities and data stores. This project makes use of the Gene and Sarson notation in detailing the level 0 and level 1 data flow diagrams.

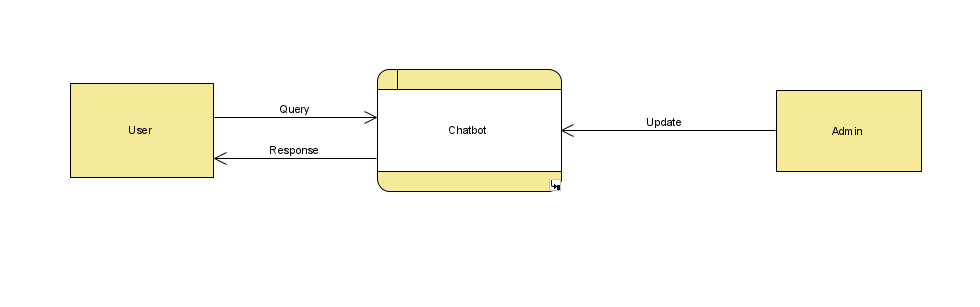
****

Figure 3.2: Level 0 Data Flow Diagram

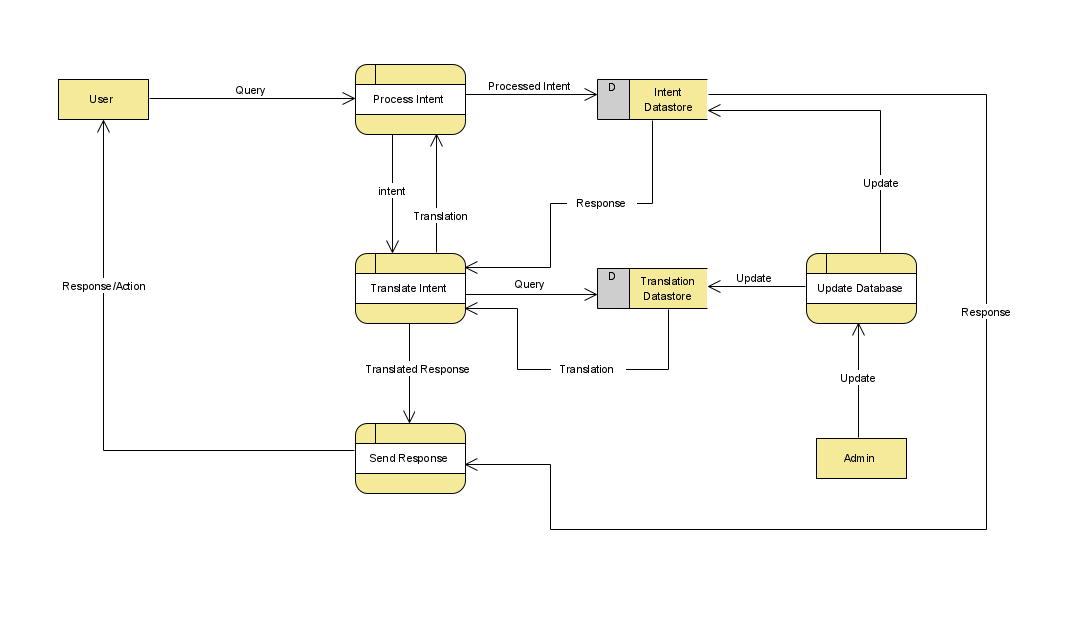
****

Figure 3.3: Level 1 Data Flow Diagram

## PHYSICAL DESIGN

Physical design involves creating visual representations of the actual processes of a system, ranging from data entry and verification to output. It details the workings of the system by specifying the precise actions/ functions that the system can perform and how they are operated. This phase of design factors in the user and helps give a holistic overview of the system's functionality.

### Use Case Diagram

Use case diagrams describe how users may interact with the system using various anticipated use cases. Use cases are represented majorly using circles or ellipses. The chatbot built in this project has the following use cases.

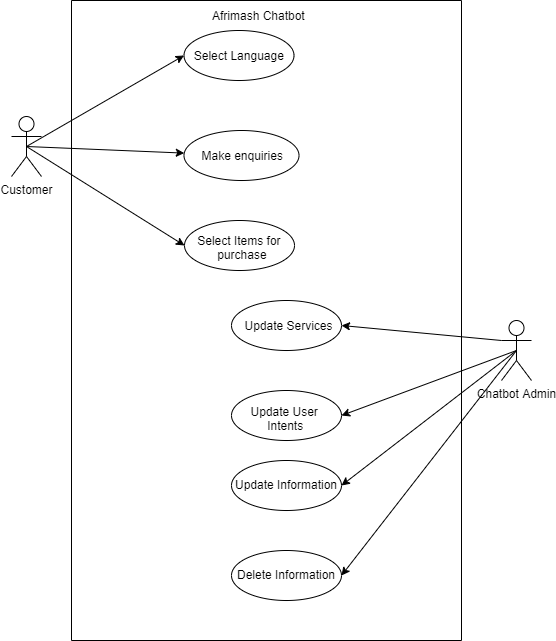
****

Figure 3.4: Chatbot Use Case Diagram

Table 3.1: Use Case Diagram Actor Documentation

|  |  |
| --- | --- |
| **Actor documentation** | **Description** |
| User | This is a person who holds a conversation with the software. |
| Administrator | The administrator performs administrative duties on the system. |

### Activity Diagram

An activity diagram is used to depict dynamic parts of a software system. It is similar to a flow chart but models the flow of activities in the system from one process to the next. The activity diagram for the chatbot in this project is detailed below.

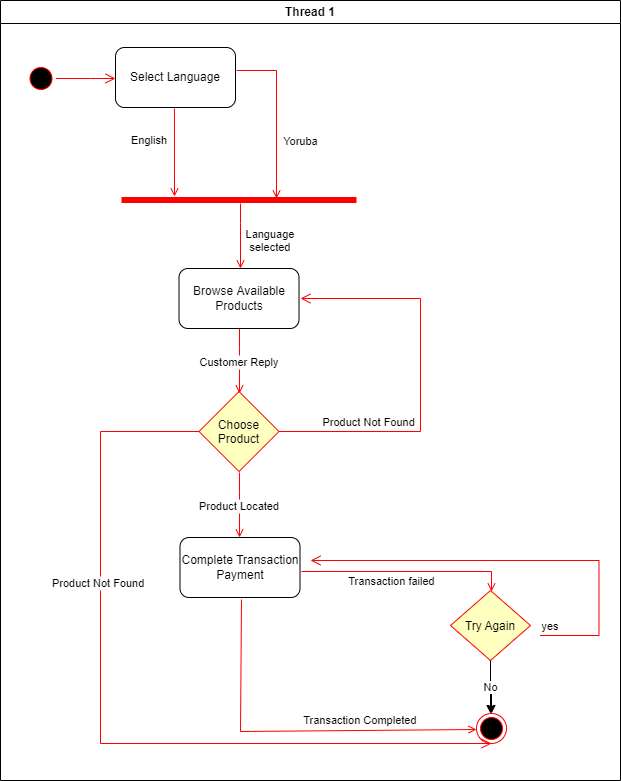
****

Figure 3.5: Chatbot Activity Diagram

### Sequence Diagram

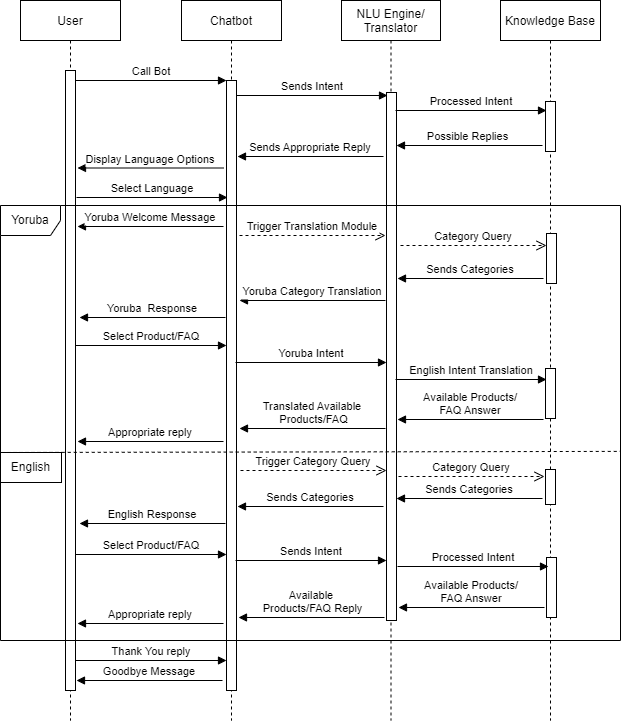
****Sequence diagrams show the chronological precedence of the operations of a system. They show the various modes of interaction among the actors and objects involved in a system. On the vertical axis of a sequence diagram, the time taken to complete an activity is represented.

Figure 3.6: Chatbot Sequence Diagram

CHAPTER FOUR

# SYSTEM IMPLEMENTATION

## INTRODUCTION

This chapter elaborately describes the technical specifications of the developed system, the hardware and software requirements of the system, and the interface and algorithms used in building the system. It also details the concepts of programming involved in the development of the system.

## SYSTEM REQUIREMENTS

Technopedia (2021) describes system requirements as configurations a system must possess to enable a hardware or software system to function in a smooth and effective manner. The hardware, software and deployment requirements for this system are detailed in Tables 4.1, 4.2 and 4.3, respectively.

Table 4.1: System Requirements

|  |  |
| --- | --- |
| Requirements | Software |
| Operating System | Windows 11 |
| Database | Google Translate API Database |
| Development tool/IDE | Pycharm, Visual Studio Code |
| Programming Language | Python |
| Programming Language version | Python 3.7.13 |

Table 4.2: Hardware Deployment Requirements

|  |
| --- |
| 8GB RAM |
| 1 TB Hard drive |
| AMD Ryzen 9 4900HS with Radeon Graphics |
| 3.00Ghz |

Table 4.3: Software Deployment Requirements

|  |
| --- |
| Python3.9 |
| Google Translate |
| Rasa Core |
| Rasa NLU 2.5.2 |
| Rasa Open Source |

## SYSTEM MODULES AND INTERFACES

A system interface is a module that services interaction and communication between the user of a system and the system itself. It serves as a connection point that articulates and aids communication and interaction between two or more, often unrelated, independent systems. An interface is also a border between different components of a system through which they exchange information.

This section explains in detail the interfaces involved in the communication and exchange of data between the different modules in the system while demonstrating the interaction process of the user with the system.

There are two main classes of users who can interact with this system. They include:

1. User: The user is a party which interacts with the bot to trigger the actions that the bot has been designed to perform.
2. An administrator is a person who handles maintenance of the chatbot operations to ensure that the bot executes its functions appropriately. In the context of this chatbot, the administrator is the only person that can access or manipulate any change to the flow of response during conversations or use of the bot.

### Telegram Interface

This system makes use of one of the most widely used social media platforms as its interface. This allows system administrators to distribute their services efficiently across the internet and help with marketing their products and services. In terms of local context, Telegram is used by 50.3% of the active internet users in Nigeria (Statista, 2021), which is the domain of this project. The Telegram interface offers advantages such as the use of buttons to select specific features and search features, as well as message history retention.

Telegram offers a clean interface with the additional feature of displaying the content of links when they are sent.

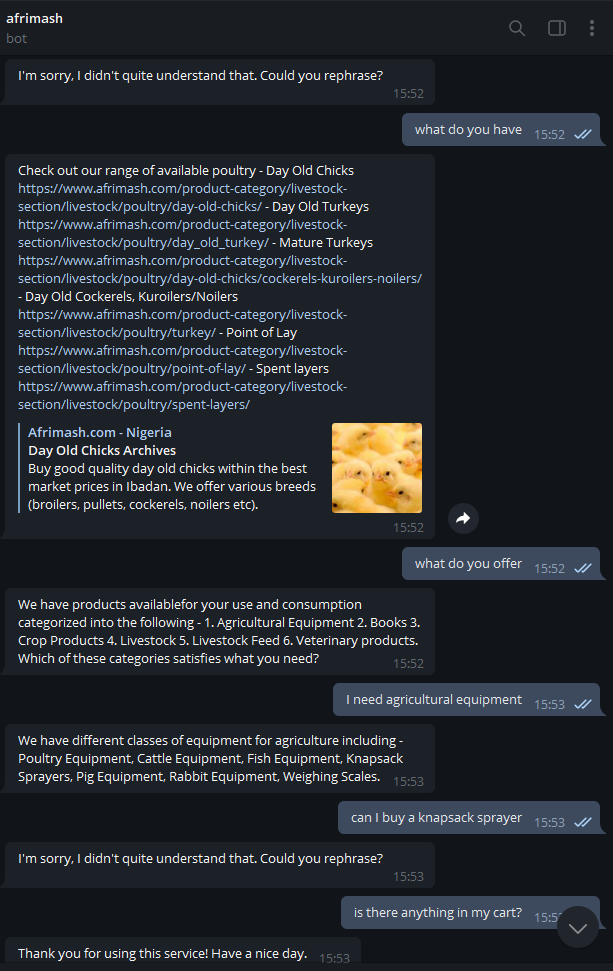


Figure 4.1: Afrimash Chatbot Telegram Interface

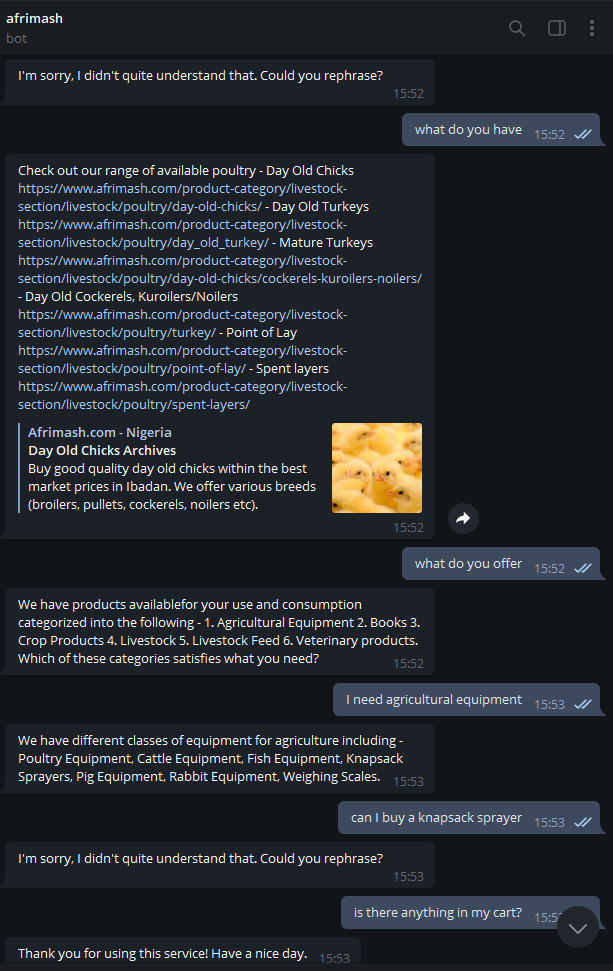


Figure 4.2: Afrimash Chatbot Telegram Interface

### Python Tkinter Interface

Python has an inbuilt module for creating user interfaces for software built with python backends. The Tkinter interface built for this project allows for seamless integration of the user interface with the chatbot backend.

The chatbot has two language options, including English language and Yoruba language. In the following figure, the user of the chatbot interacts with the bot in the English language.

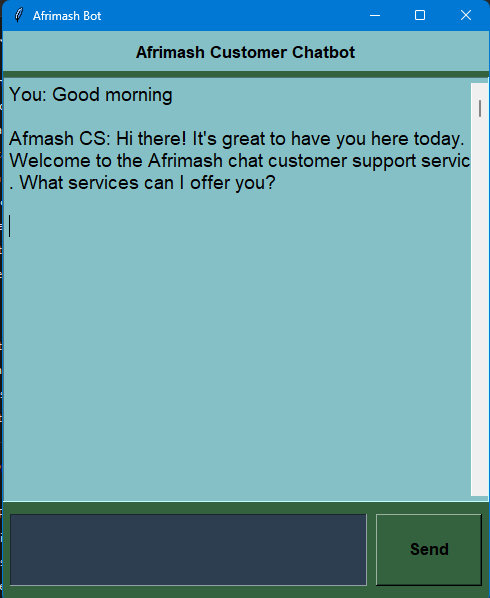


Figure 4.3: Python Tkinter Interface in the English language

The functionalities of the chatbot also extend to the Yoruba language. In the figure below, the user is in Yoruba language mode.

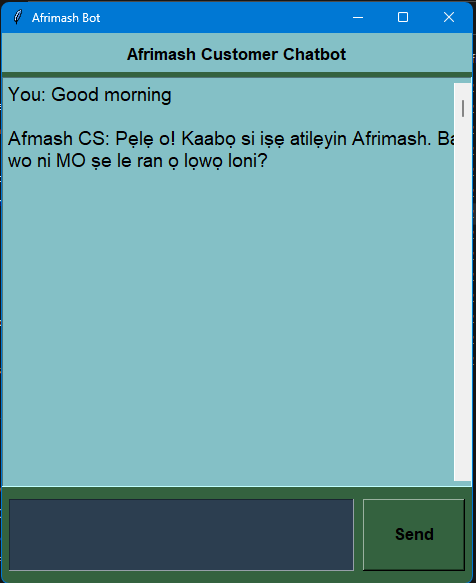


Figure 4.4: User Interaction in the Yoruba language

### Ngrok Web Connector

The Ngrok web connector is a tool used to link and test diverse applications on the internet. The tool allows users to build API consumers and create demo websites without having to deploy them on the internet. It is globally available and allows users to run any web service via the cloud or private networks on their local machines. This project uses the Ngrok web proxy server to deploy the chatbot from a local machine.

Once the server runs locally, Ngrok connects the local system to a web server and serves as a proxy, thereby deploying the chatbot to any server that has been connected via an API call using webhooks. The advantages of using Ngrok for this project include identity protection, run-everywhere ability and support for all protocols, including HTTP, TLS and other TCP-based protocols.

## SYSTEM EVALUATION

A team of ten subject matter experts in three fields (Yoruba Language, Agriculture and UI/UX) was assembled to perform system evaluation. Each user was given a five-minute window to communicate with the bot and try to achieve an objective they would ordinarily be able to complete on the website.

A questionnaire was circulated to each participant using Google Forms. The form contained nine questions which were based on a five-point Likert scale, save for one question. The results of the evaluation are as follows:

1. The overall impression of the chatbot.

The assessors of the system were asked to give a general impression of the system and how they perceive it as a competent and reliable assistant for customer service purposes. Of the ten users, 10% indicated that they liked and trusted the system very much and that it is terrific. 40% indicated that the system is good, while another 40% gave an average rating of the system. Finally, 10% of users indicated that the system was bad and needed improvement in specific areas, especially relating to the language translation functionality. 90% of users were satisfied with the system overall.

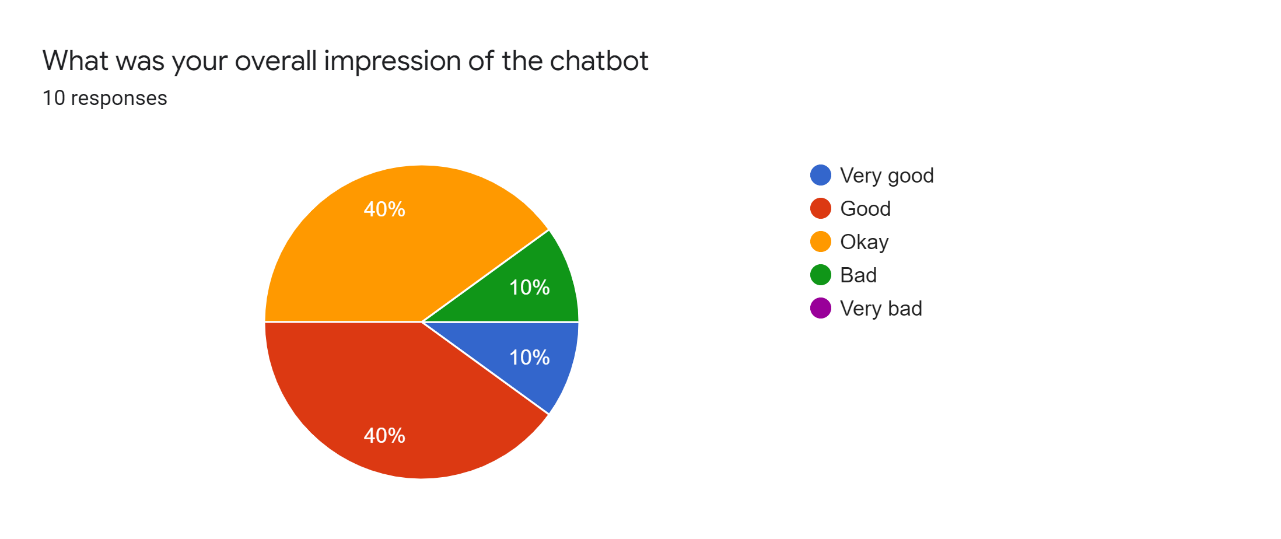
****

Figure 4.5: Usability Testing Question 1

1. How good was each user's experience with the chatbot?

Users were asked to describe their user experience with the chatbot on a scale of 1 – 5, with 1 representing a bad experience and 5 representing an enjoyable experience. The distribution of responses indicates that 80% of the users had a pleasant or fair experience using the chatbot. This gives the chatbot an average rating of 3.6/5.

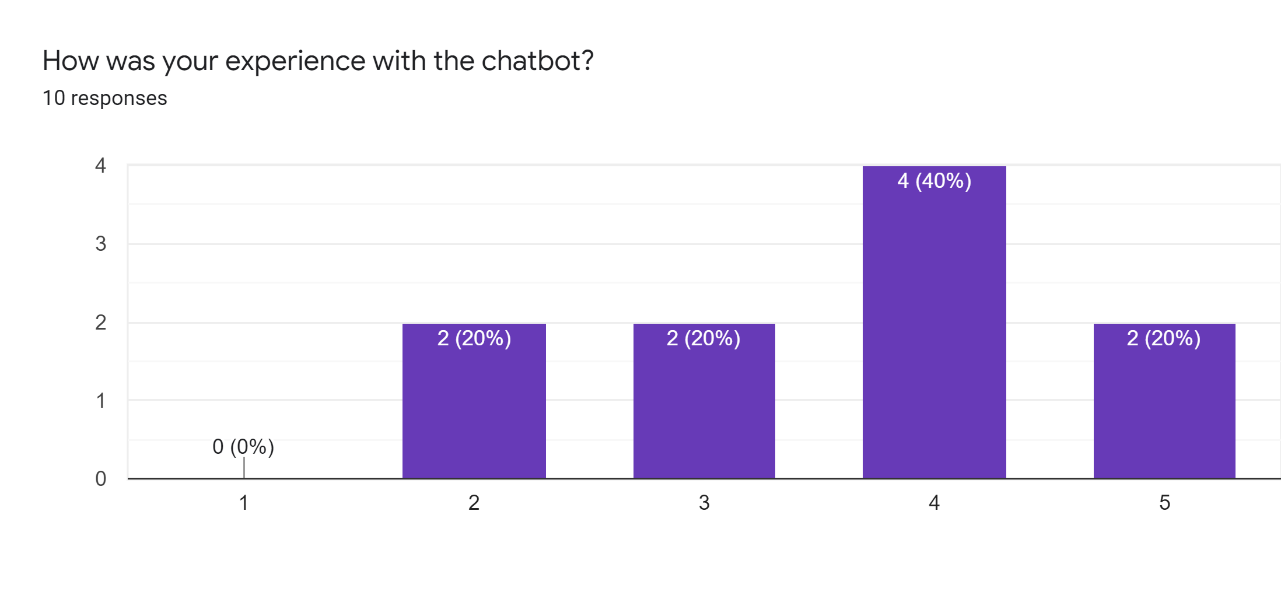
****

Figure 4.6: Usability Testing Question 2

1. How well did the chatbot understand users' intentions?

Participants were asked to specify the frequency of accurate interpretations of their requests demonstrated by the chatbot. Of the ten expert users, eight were generally satisfied with the ability of the chatbot to process their intents.

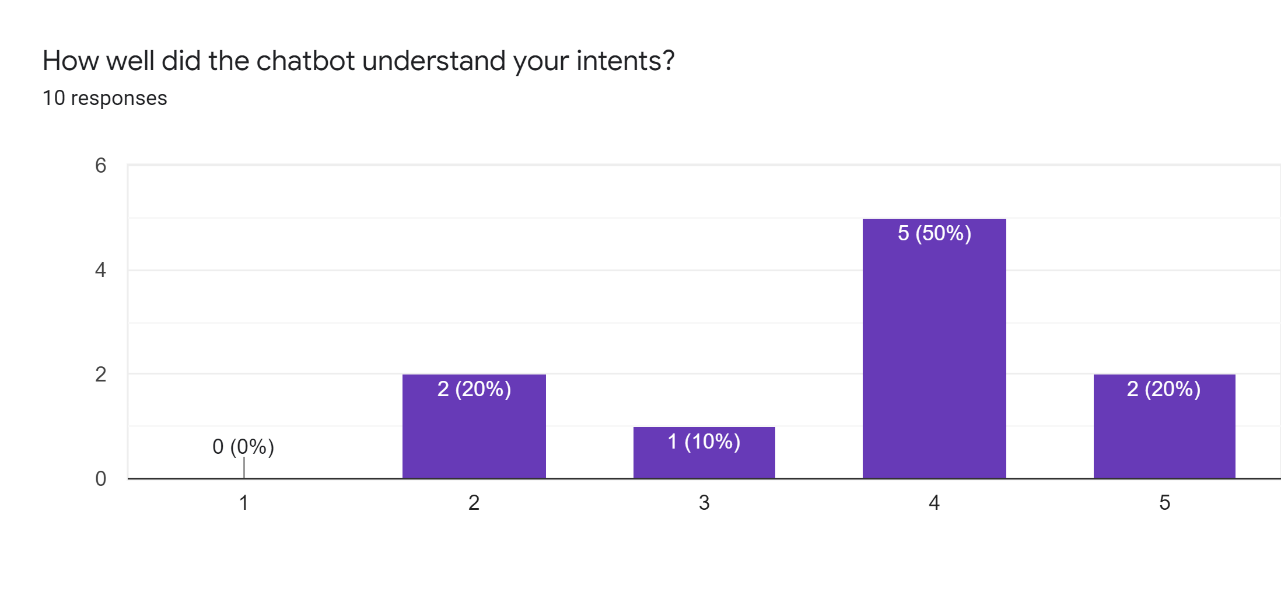
**

Figure 4.7: Usability Testing Question 3

1. How accurate are the chatbot's replies?

The accuracy of the results generated is as important as the intent classification, as this metric is what remarkably improves user experience on the Afrimash website. Participants were asked to describe the frequency of accurate results obtained from their conversations with the chatbot pertaining to their ability to access relevant information from the website. A wholesome 90% of participants were generally satisfied with the accuracy of the results obtained. However, nobody reported a 100% accuracy level of the results obtained, which indicates that there is a challenge within the logical base of the system.

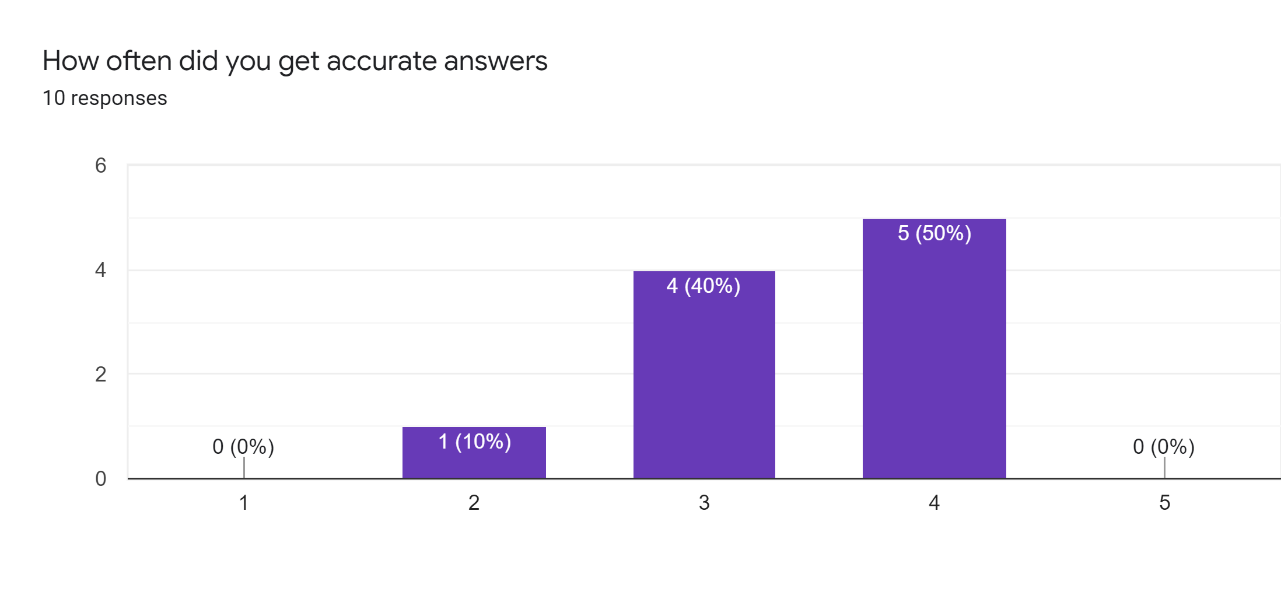
**

Figure 4.8: Usability Testing Question 4

1. How well can users navigate the functions of the website using the chatbot?

Users were asked to describe their experience with the ability to locate items on the chatbot as opposed to on the website. Users reported that 90% of the time, they were able to access the functionality they were looking to find. However, 40% of users reported the speed of access being average.

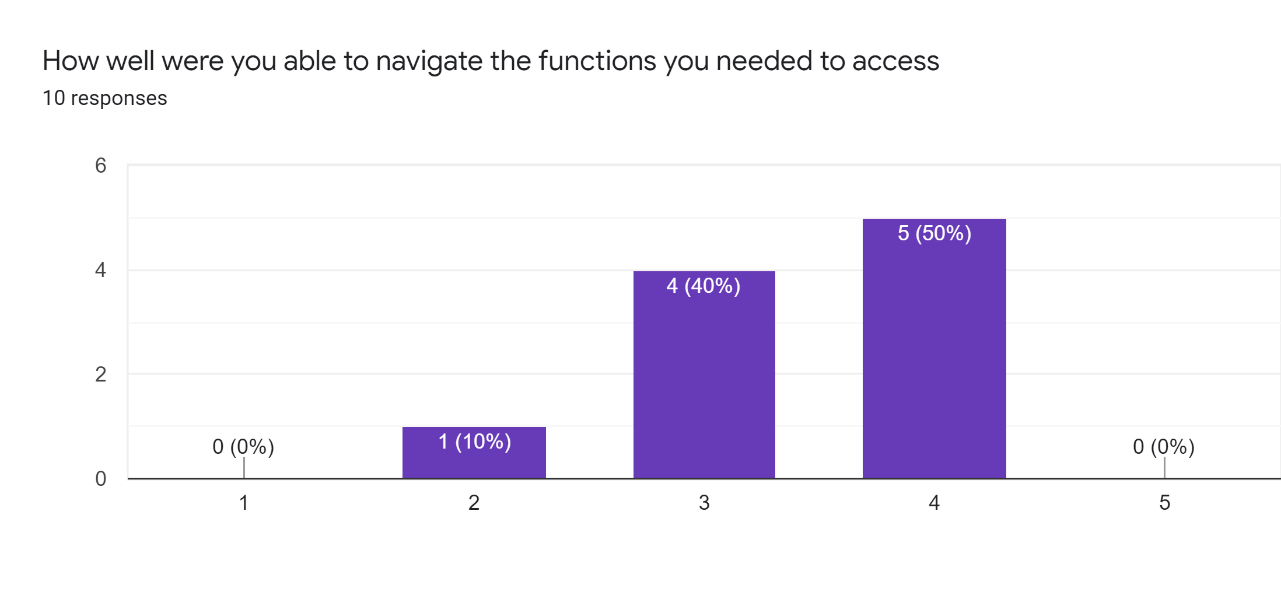
****

Figure 4.9: Usability Testing Question 5

1. What is the chatbot's speed relative to the website?

As a metric to assess if user experience had indeed been improved, users were asked to specify if the chatbot offered a significant advantage over the website in terms of speed of access. 40% of users stated that it was always much faster to use the chatbot than the website, another 40% assessed the chatbot to be faster most times, but not on every occasion, while 10% reported that the website was slightly faster than the chatbot.

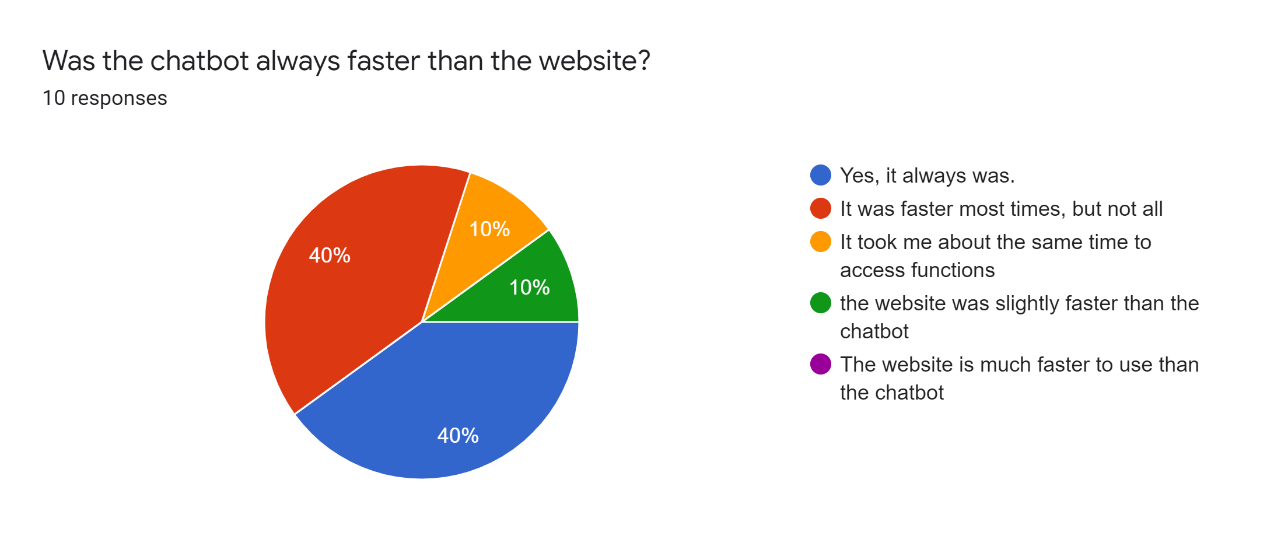
****

Figure 4.10: Usability Testing Question 6

1. Test of users' preferred means of interacting with the Afrimash website's content.

Participants were asked to indicate whether they would use the chatbot instead of the website if future interactions were needed. 80% of users suggested that they would prefer to use the chatbot over the website in the future, while 20% indicated otherwise.

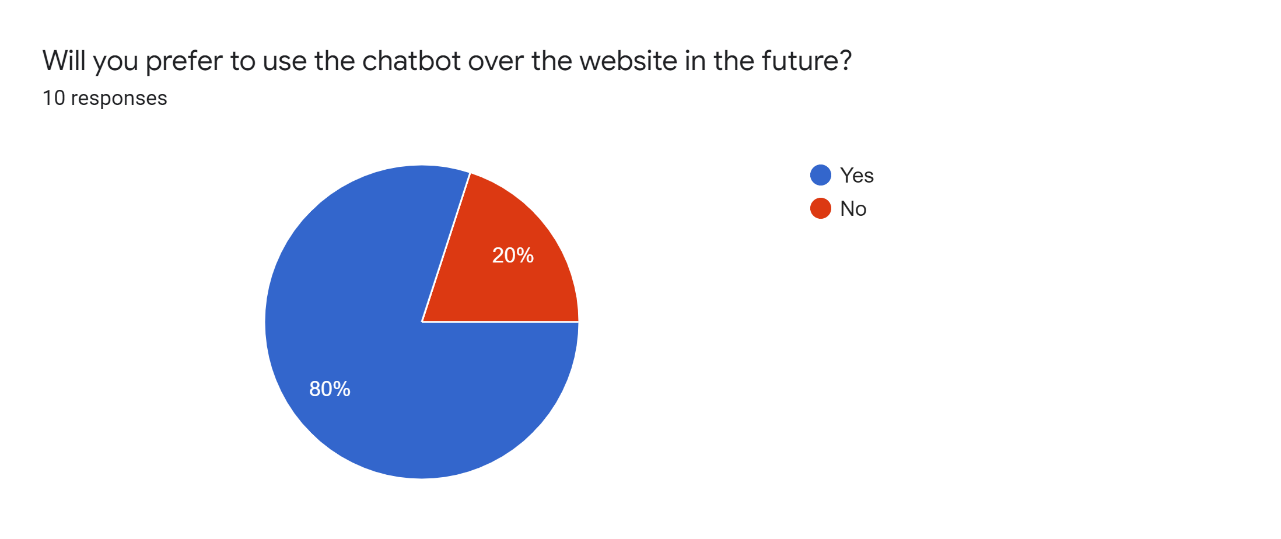


Figure 4.11: Usability Testing Question 7

1. Test of the chatbot's accuracy in Yoruba language translation.

The experts indicated how frequently the chatbot was able to understand the questions the users asked in the Yoruba language to assess the chatbot's ability to process Yoruba intents. A vast majority of users indicated difficulty on the part of the chatbot in understanding user intents in the Yoruba language. Only 40% reported a positive experience using the chatbot in the Yoruba language. 10% of users reported never getting an accurate translation of their input. This is majorly due to the fact that context in the Yoruba language is heavily dependent on accurate inputs, as inputs must be accompanied by accent marks (àmì ohùn).

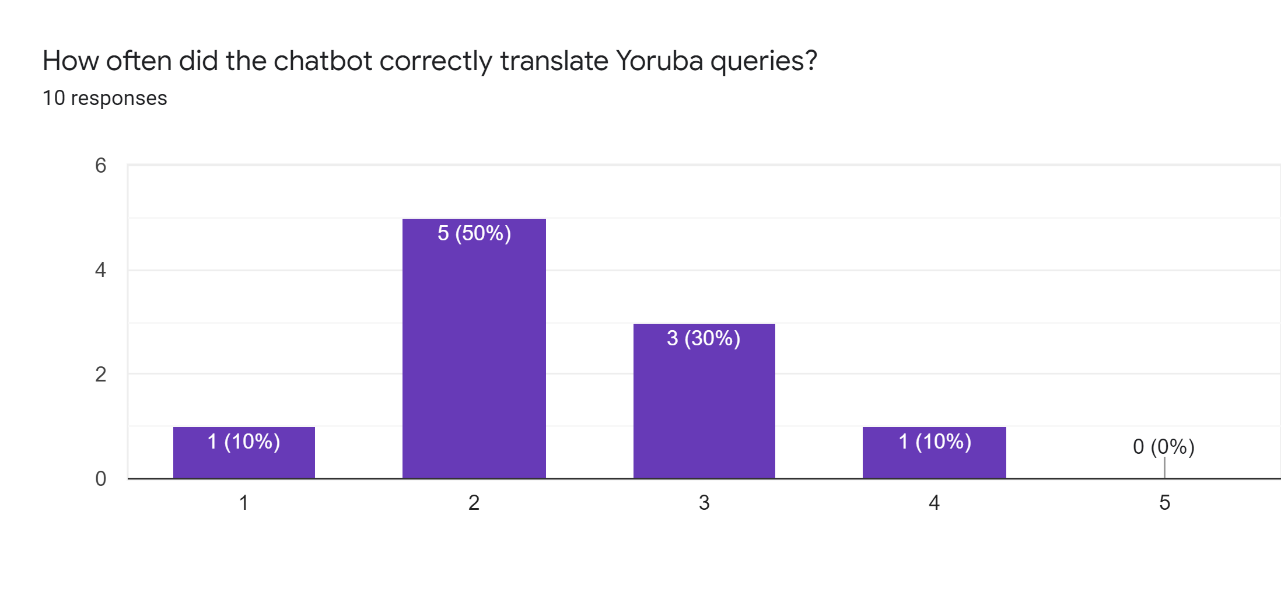
****

Figure 4.12: Usability Testing Question 8

1. How often did the Yoruba query produce correct results that led to the location of a search term?

As a direct consequence of the incorrect inputs, the majority of users also reported inaccurate results in the Yoruba language.

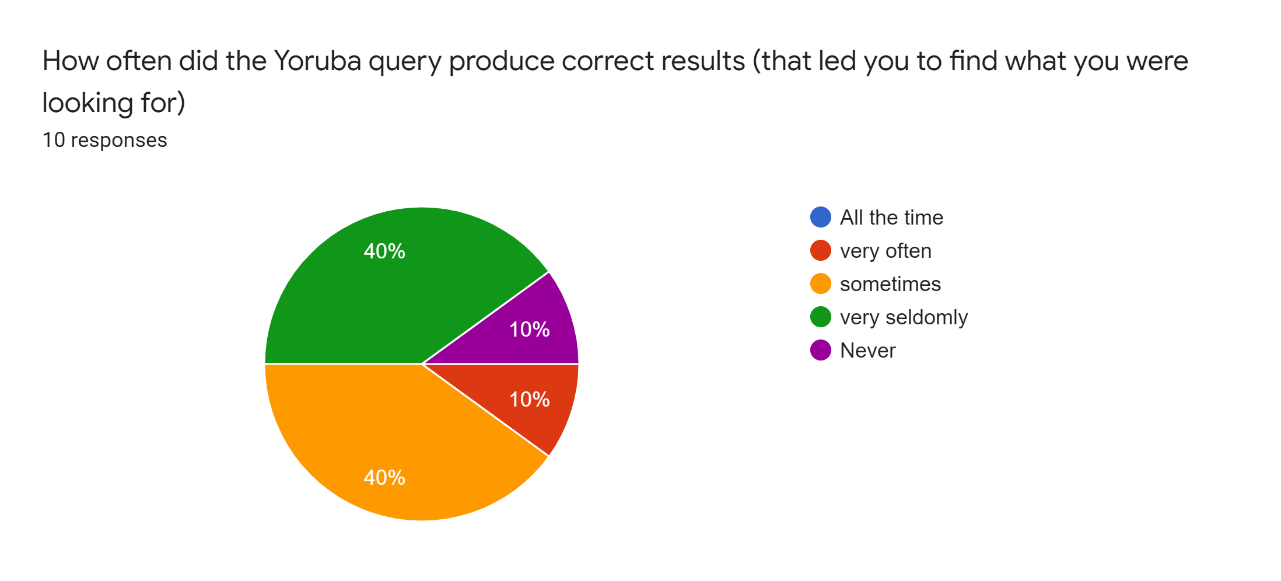
**

Figure 4.13: Usability Testing Question 9

CHAPTER FIVE

# CONCLUSION AND RECOMMENDATIONS

## SUMMARY

The impacts of Human-Computer Interaction on the performance of businesses with online platforms in the modern day are profound and far-reaching. Improving areas of user experience is paramount to business success in the present highly competitive landscape. Agricultural e-commerce is a nuanced field in the vast world of digitised commerce solutions. Consequently, there is vast room for optimising processes such that significant improvement is made in this field.

By developing a chatbot which helps users interact with an e-commerce platform in their local lingua franca, this work contributes to a dangerously underserved aspect of research. This project takes a step in the right direction by improving the degree of usability of an e-commerce platform using machine learning and chatbot technologies.

Localisation is a critical aspect of context awareness; therefore, by creating an avenue for local users to communicate with the chatbot in their local languages, interaction with the website under study has been improved significantly.

## RECOMMENDATIONS

In order to process inputs in the Yoruba language, the software implemented in this project took advantage of one of the world's largest translation libraries, Google translate. However, a significant limitation of applying this procedure is that textual input not specified with the intonation accent marks (àmì ohùn) can easily be misinterpreted. This dramatically hinders context awareness and reduces the accuracy of the translated inputs.

Most likely, the most significant barrier to the use of accent marks in textual input is that modern keyboards on laptops and desktop computers do not offer apparent avenues for the input of accent marks. A proposed solution is to develop a floating keyboard interface that allows users to input the exact words in a format that will be accurately understood by the system.

## CONCLUSION

In this work, the system developed is a chatbot system equipped with translation capabilities to give it context awareness in terms of the local linguistic landscape. The chatbot is built on the Rasa NLU framework and the Google translate Application Programming Interface (API).

This chatbot software enables users to access farm produce available on the website of the e-commerce platform Afrimash.com. It also enables users who can only communicate in local dialects to interact fluently with the website without having to visit the platform directly.

REFERENCES

Bank of America. (2019). Bank of America's erica® surpasses 10 million users, introduces new capabilities. Retrieved February 5, from <https://newsroom.bankofamerica.com/press-releases/consumer-banking/bank-americas-ericar-surpasses-10-million-users-introduces-new>

Bertacchini, F., Bilotta, E., & Pantano, P. (2017). Shopping with a robotic companion. *Computers in Human Behavior*, 77(1), 382-395.

Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *arXiv preprint*. Retrieved March 19 from https://arxiv.org/abs/1712.05181v2.

Brush, K., & Scardina, J. (2021). Chatbot. Retrieved April 30 from <https://www.techtarget.com/searchcustomerexperience/definition/chatbot>

Built With. (2018). eCommerce technologies web usage distribution. Retrieved April 5, from <https://trends.builtwith.com/shop>

Colby, M. K. (2013). *Artificial paranoia: A computer simulation of paranoid processes*. Oxford, UK: Pergamon.

Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, 22(5), 811–817.

Dyachenko, O. (2015) Amazon is now bigger than Walmart. Retrieved May 15 from <https://qz.com/462605/amazon-isnow-bigger-than-walmart/>

Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *Interactions, 24*(4), 38–42.

Gibbert, M., Leibold, M., & Probst, G. (2002). Five styles of customer knowledge management and how smart companies use them to create value. *Eur. Manage. J*., 20(5), 459–469.

Hennessy, T., Läpple, D., & Moran, B. (2016). The digital divide in farming: a problem of access or engagement?. *Appl. Econ. Perspect. Policy*, 38 (3), 474-491.

Huang, M., & Rust, RT (2018). Artificial intelligence in service. *Journal of Service Research*, 21(2), 155-172.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised learning. *International journal of computer science,* 1(2), 111–117.

Kandpal, R. (2021). A report on c-commerce chatbots submission of SAP Conversational AI 10.13140/RG.2.2.16030.05443.

Manne, R. & Kantheti, S.C. (2021). application of artificial intelligence in healthcare: chances and challenges. *Curr. J. Appl. Sci. Technol.*, 40(6), 78-89.

Mittal, A., Agrawal, A., Chouksey, A., Shriwas, R., & Agrawal, S. (2016). A comparative study of chatbots and humans. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(6), 1055–1057.

Ngai, E.W., Lee, M.C., Luo, M., Chan, P.S & Liang, T. (2021). An intelligent knowledge-based chatbot for customer service, *Electronic Commerce Research and Applications,* 50(1), 27 - 42.

Oguntosin, V. & Olomo, A. (2021). Development of an e-commerce chatbot for a university shopping mall. *Applied Computational Intelligence and Soft Computing.* Retrieved May 25 from <https://www.hindawi.com/journals/acise/2021/6630326/>

Prodromou, E. (2015). Adaptive pricing with fuzzy logic. Retrieved from <https://blog.fuzzy.ai/2015/03/25/adaptive-pricing-with-fuzzy-logic/>

Roca, S., Sancho, J., García, J. & Alesanco, Á. (2020). Microservice chatbot architecture for chronic patient support. *Journal of Biomedical Informatics*, 102(1), 103 - 305.

Shawar, B. & Atwell, E. (2007). Chatbots: Are they really useful? *LDV Forum 2007 - Band,* 22(1), 29.

Shingte, K., Chaudhari, A., Patil, A., & Desai, S, (2021). Cha development for educational institute. Retrieved February 28 <https://papers.ssm.com/sol3/papers.cfm?abstract_id=3861241>

Shum, H., He, X., & Li, D. (2018). From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1),10 - 26.

Statista. (2021). Leading social media platforms in Nigeria 2021. Retrieved April 13 from <https://www.statista.com/statistics/1176101/leading-social-media-platforms-nigeria/>

Techopedia. (2015). System requirements. Retrieved January 28, from <https://www.techopedia.com/definition/4371/system-requirements>

Turing, A. M. (1950). I-computing machinery and intelligence. Mind, 59(236), 433 - 460.

Wallace, S., Epstein, R., Roberts, G. & Springer, E. (2009). The anatomy of Alice. *Parsing the Turing Test*, Dordrecht, Netherlands.

Weizenbaum, J (1966). Eliza - A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36 -45.

Wilde, S. (2011). Customer knowledge management. *Springer Berlin Heidelberg*.

Zapata, S., Isengildina-Massa O., Carpio C. E. & Lamie R. D. (2016). Does e-commerce help farmers' markets? Measuring the impact of market makers. *J. Food Distrib. Res*., 47(2), 1-18.