IMPLEMENTATION OF A VOICE-ENABLED BIBLE CHATBOT USING SPEECH-TO-TEXT RECOGNITION

BY

AYENI OLUWATOBI IFEOLUWA

(18CG024726)

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

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

JULY, 2022

CERTIFICATION

I hereby certify that this project work titled IMPLEMENTATION OF A VOICE ENABLED BIBLE CHATBOT USING SPEECH TO TEXT RECOGNITION is a bona fide work carried out by Ayeni Oluwatobi Ifeoluwa (18CG024726) submitted to the Department of Computer and Information Sciences, College of Science and Technology, Covenant University, Ota.

|  |  |
| --- | --- |
| **Professor Jelili Oyelade** |  |
| Supervisor | **Signature and Date** |
|  |  |
| **Professor Olufunke O. Oladipupo** |  |
| Head of Department | **Signature and Date** |

DEDICATION

I dedicate this work to my late grandmother. She started this academic journey with me and was always there for me every step of the way. May her precious soul continue to rest in peace.

ACKNOWLEDGEMENT

The uttermost acknowledgement goes to God almighty for his unending love, grace, kindness, favour, and mercy all through the course of working on this project. To him alone be all the glory.

Acknowledgement also goes to my parents, Mr and Mrs Ayeni. I would like to say a big thank you for your constant support, words of encouragement, and love shown towards me. I am beyond grateful for all you have done for me. I pray the Lord almighty continues to bless and replenish your effort. Also, to my siblings, Seun and Tosin, I am very grateful for your constant support.

I would also like to thank my Godparents, Pastor and Mrs Olaifa, for their never-ending love and support shown towards me whenever I need it. I pray the Lord God almighty continues to bless you abundantly and make you a source of blessing like you always are to me.

To my supervisor, Professor Oyelade Jelili, I would like to thank you for your constant effort and guidance in ensuring this project was a huge success. May the good Lord continue to bless you, sir.

I would also like to thank Dr Azubuike Ezenwoke for his contribution, constant support, and guidance made available to me. I pray the Lord God almighty continues to bless you. Also, a big thank you to my uncle Opeoluwa Ademulegun for his support all through the phase of this project. God bless you, sir.

I wish to sincerely thank all my friends who helped make this successful project one in one way or the other. God bless you all.

Finally, I would like to thank the Computer and Information Science Department. I would like to start with the Head of Department, Professor Olufunke O. Oladipupo, my level advisers, Dr Onyeka Emebo and Dr Oluranti Jonathan, and the entire faculty and staff in the department. Through them, I have gained a very vast knowledge of this field. God bless you all.

TABLE OF CONTENT

[CERTIFICATION i](#_Toc109421127)

[DEDICATION ii](#_Toc109421128)

[ACKNOWLEDGEMENT iii](#_Toc109421129)

[TABLE OF CONTENT iv](#_Toc109421130)

[LIST OF TABLES viii](#_Toc109421131)

[LIST OF FIGURES ix](#_Toc109421132)

[ABSTRACT xi](#_Toc109421133)

[Chapter One: INTRODUCTION 1](#_Toc109421134)

[1.1. Background information 1](#_Toc109421136)

[1.2. Motivation 3](#_Toc109421137)

[1.3. Statement of problem 3](#_Toc109421138)

[1.4. Aims and objectives of study 3](#_Toc109421139)

[1.4.1. OBJECTIVES 4](#_Toc109421140)

[1.5. Methodology 4](#_Toc109421141)

[1.6. Significance of study 5](#_Toc109421142)

[1.7. Limitations of study 6](#_Toc109421143)

[1.8. Project organization 6](#_Toc109421144)

[Chapter Two: Literature Review 7](#_Toc109421145)

[2.1. OVERVIEW OF CHATBOTS 7](#_Toc109421147)

[2.1.1. BRIEF HISTORY OF CHATBOTS 7](#_Toc109421148)

[2.1.2. TYPES OF CHATBOTS 7](#_Toc109421149)

[2.2. BIBLE STUDY OVERVIEW 10](#_Toc109421150)

[2.2.1. BIBLE STUDY IN A DIGITAL AGE 11](#_Toc109421151)

[2.2.2. SCRIPTURE RECOMMENDATION 12](#_Toc109421152)

[2.3. SPEECH-TO-TEXT RECOGNITION 12](#_Toc109421153)

[2.3.1. INTRODUCTION TO SPEECH RECOGNITION 12](#_Toc109421154)

[2.3.2. BASIC SPEECH RECOGNITION MODEL 13](#_Toc109421155)

[2.3.3. SPEECH-TO-TEXT CONVERSION METHODS 15](#_Toc109421156)

[2.3.4. DEVELOPING AUTOMATIC SPEECH RECOGNITION 17](#_Toc109421157)

[2.4. HUMAN-COMPUTER INTERACTION 18](#_Toc109421158)

[2.5. PYTHON 18](#_Toc109421159)

[2.5.1. PYTHON IN NATURAL LANGUAGE PROCESSING 19](#_Toc109421160)

[2.5.2. PYTHON IN SPEECH-TO-TEXT RECOGNITION 19](#_Toc109421161)

[2.6. REVIEW OF EXISTING SYSTEMS 20](#_Toc109421162)

[2.6.1. YOU VERSION BIBLE APPLICATION 20](#_Toc109421163)

[2.6.2. BLUE LETTER BIBLE APPLICATION 21](#_Toc109421164)

[2.6.3. OLIVE TREE BIBLE SOFTWARE 21](#_Toc109421165)

[2.6.4. ACCORDANCE BIBLE APPLICATION 23](#_Toc109421166)

[CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN 25](#_Toc109421167)

[3.1. INTRODUCTION 25](#_Toc109421169)

[3.2. SYSTEM ARCHITECTURE 25](#_Toc109421170)

[3.3. SYSTEM REQUIREMENTS 26](#_Toc109421171)

[3.3.1. FUNCTIONAL REQUIREMENT 27](#_Toc109421172)

[3.3.2. NON-FUNCTIONAL REQUIREMENT 27](#_Toc109421173)

[3.4. SYSTEM DESIGN 28](#_Toc109421174)

[3.4.1. PHYSICAL DESIGN 28](#_Toc109421175)

[3.4.2. INPUT DESIGN 28](#_Toc109421176)

[3.4.3. OUTPUT DESIGN 28](#_Toc109421177)

[3.4.4. LOGICAL DESIGN 29](#_Toc109421178)

[3.5. USE CASE DIAGRAM 29](#_Toc109421179)

[3.6. ACTIVITY DIAGRAM 30](#_Toc109421180)

[3.7. SEQUENCE DIAGRAM 33](#_Toc109421181)

[CHAPTER FOUR: IMPLEMENTATION 35](#_Toc109421182)

[4.1. INTRODUCTION 35](#_Toc109421184)

[4.2. SYSTEM REQUIREMENTS 35](#_Toc109421185)

[4.3. SYSTEM INTERFACE 36](#_Toc109421186)

[4.3.1. PYTHON FLASK ENVIRONMENT 37](#_Toc109421187)

[4.3.2. HISTORY PAGE INTERFACE 38](#_Toc109421188)

[4.4. INTELLIGENT MODULE 39](#_Toc109421189)

[4.4.1. INPUT COLLECTION 39](#_Toc109421190)

[4.4.2. INPUT AND OUTPUT PROCESSING 40](#_Toc109421191)

[4.4.3. OUTPUT GENERATION 41](#_Toc109421192)

[4.5. SYSTEM EVALUATION 42](#_Toc109421193)

[CHAPTER FIVE 48](#_Toc109421194)

[SUMMARY, RECOMMENDATIONS, AND CONCLUSION 48](#_Toc109421195)

[5.1. SUMMARY 48](#_Toc109421196)

[5.2. RECOMMENDATIONS 48](#_Toc109421197)

[5.3. CONCLUSION 49](#_Toc109421198)

[REFERENCES 50](#_Toc109421199)

LIST OF TABLES

[Table 2.1:Comparative Review of Existing Bible Apps 25](#_Toc106291058)

[Table 4.1: Software requirements 36](#_Toc106291059)

[Table 4.2: Hardware Deployment Requirement 37](#_Toc106291060)

[Table 4.3: Software Deployment Requirements 37](#_Toc106291061)

LIST OF FIGURES

[Figure 2.1: Speech Recognition model 13](#_Toc106290767)

[Figure 2.2:Linear Predictive Coding Process 14](#_Toc106290768)

[Figure 2.3:SpeechToText System Flow 17](#_Toc106290769)

[Figure 2.4:YouVersion BibleApp 21](#_Toc106290770)

[Figure 2.5:Blue Letter BibleApp 22](#_Toc106290771)

[Figure 2.6:Olive Tree Bible Software 23](#_Toc106290772)

[Figure 2.7:Accordance BibleApp 24](#_Toc106290773)

[Figure 3.1:Chatbot Architecture 27](#_Toc106290774)

[Figure 3.2:Use Case Diagram for Chatbot 31](#_Toc106290775)

[Figure 3.3:Activity Diagram for Bible Study 32](#_Toc106290776)

[Figure 3.4:Activity Diagram for Bible verse search 33](#_Toc106290777)

[Figure 3.5:Sequence diagram for Bible study 34](#_Toc106290778)

[Figure 3.6:Sequene diagram for Bible verse search 35](#_Toc106290779)

[Figure 4.1:Flask web framework welcome page 39](#_Toc106290780)

[Figure 4.2:Flask web framework History Page 40](#_Toc106290781)

[Figure 4.3:Bar chart showing responses to question 1 44](#_Toc106290782)

[Figure 4.4:Bar chart showing responses to question 2 44](#_Toc106290783)

[Figure 4.5:Bar chart showing responses to question 3 45](#_Toc106290784)

[Figure 4.6:Bar chart showing responses to question 4 45](#_Toc106290785)

[Figure 4.7:Bar chart showing responses to question 5 46](#_Toc106290786)

[Figure 4.8:Bar chart showing responses to question 6 46](#_Toc106290787)

[Figure 4.9: Bar chart showing responses to question 7 47](#_Toc106290788)

[Figure 4.10:Bar chart showing responses to question 8 47](#_Toc106290789)

[Figure 4.11:Bar chart showing responses to question 9 48](#_Toc106290790)

[Figure 4.12:Bar chart showing responses to question 10 48](#_Toc106290791)

[Figure 4.13:Bar chart showing responses to question 11 49](#_Toc106290792)

ABSTRACT

Integrating technological means into the world of religion has a wide variety of opportunities to thrive, provide ease for individuals, and solely aimed at positively affecting such individuals. A chatbot is an aspect of Artificial Intelligence that simulate conversations with users in natural language. This project aims to develop a chatbot for bible study by implementing a voice-enabled chatbot using speech-to-text recognition technology. In order to achieve the aims, the objectives include surveying existing technology, choosing the best architecture, designing an appropriate model for the system, implementing the system using the appropriate language, and evaluating the system after implementation. The methodologies used to attain the objectives include studying and observing existing systems. Using UML (Unified Modeling Language) diagrams to show the system's architecture, use python programming languages for implementation and validate the effectiveness of the proposed system.

The system's significance is an effective recommendation of scriptures to users by the chatbot to individuals based on privacy assurance. Also, it will help bring individuals more conversant with the Bible, as the study aims to impact the user positively with ease because it is from their comfort zone. It can be accessed anywhere and anytime. However, this project can achieve its set out aim. The ease of access to more information can be improved by introducing a database for offline purposes and training the bot with a larger context.

Chapter One

# Introduction

## Background information

Technology has made daily living more accessible by implementing various technological means and innovations. The benefit depends on what they offer to the end-user, one of which is chatbots. Chatbots are artificial intelligence software that simulates conversations with a user. These simulations can understand human language, process it, and interact with humans while performing specific tasks configured to assign. Chatbots are mainly built based on NLP (Natural Language Processing).

Natural Language Processing is a branch of Artificial Intelligence that enables computers to understand texts and audible conversations the same way humans understand them (Eleni Adamopoulou & Lefteris Moussiades,2020). They combine computational rule-based human language modelling with machine learning or deep learning models. With all these models, computers can understand and process human language in text or voice data. The NLP model also has specific data points to give out answers.

Chatbot implementation occurs in various ways to meet the needs of individuals or entities based on artificial intelligence. For example, most banks across the globe have found more accessible ways to attend to the needs of their customers through the implementation of user-friendly and interactive chatbots attending to virtually all the needs of the customer. The model is developed based on the day-to-day operations of the bank and the requirements identified by the banks.

With this same knowledge, after exploring the beautiful capabilities of implementing a chatbot. Benefits in the religious practice entail helping individuals grow in their various religions. From electronic scriptures to robot priests, different faiths have absorbed new ideas from the world of technology to enhance mainstream religious practices. For example, Muslims can download apps such as Muslim Pro. This app helps to full daily prayer timetables, notify the individual for both their sunrise and sunset and recommends and tells them times for their fast and all sorts of other functions.

Applying artificial intelligence to religious practices at large can be enormous, hence the reason for focusing on the Bible. (Robert Filback and Stephen Krashen,2016), proved that having close personal relationships, regular reading, and Bible study helps inspire individuals. It also serves as the best source to get advice and wisdom for most issues we might face. Also, most individuals around the society find it difficult to study the Bible due to laziness, or some get bored reading the hard copy. (Michael & Jeremiah, 2016), carried out a case study between women and men aged 19 and 68. It aims to determine the adverse effect of adequate and regular bible study. This research aimed to show how it tends to impact the individual's life either positively or negatively. They concluded that those who read the Bible enjoy it more and tend to make better or wiser decisions with a regular time of constant study, making the impact positive. Hence, with the integration of artificial intelligence, studying the Bible will be more conversational and exciting for individuals of such calibre.

Since technology has become more powerful, more accessible, and more widespread recently, the application has become pivotal in enhancing competitiveness. Hence enabling development, and bringing progress to all levels of society, bible scholars, pastors, and the religious society in general. The benefits of this implementation help through the moral building of individuals' spirit and character formation. The individual will need to be IT-compliant and adopt the same given, helping lift Africa to a place of relevance(Efiong, 2016).

The correlation between technology and religion has started to develop over the years. Algorithms and bots help in what we see, the kind of information we interact with, and the content we interact with regularly. According to (Hugh Davies, 2019), many Christians use various information technology means to advance their faith. Implementing a voice-enabled bible chatbot will help build up or stir the mind and dedication of the users on the right path.

## Motivation

(Nishad & Mohammed, 2020), they proposed with visible evidence that in the nearest future, there will be human-like machine intelligence that cuts across all forms of tasks and activities humans carry out nowadays. Now in alliance with what was postulated by the researchers here, implementation of this project not only aligns with this but also will help increase the merging of information Technology into the religious world.

Many believe there is no room for technology in the religious world, hence one of the motives behind this project. Implementing this model will help close the gap of easier access and on-the-go personalization to religious content. Technology will help in building the faith of others which was already made evident according to (Hugh Davies, 2019). Here he discussed the need for technology aligning with religious doctrines to help individuals follow their faith with the best possible ease.

## Statement of problem

In the wave of recent technological advancements, many people require automated methods and ways to access the Bible instead of reading manually, hence the need for implementing this model.

Many applications like this have been designed and implemented in recent years. For example, the YOU version bible app is a top-rated bible app used by millions worldwide. Make use of the functionalities in the app manually, and scripture recommendation is mainly made based on the application's algorithm.

Hence, the implementation of this model aims to solve the problem of individual religious growth within the society and community at large. The chatbot will be an easy go-to for help and give recommendations based on inputs and conversations held with the bot.

## Aims and objectives of study

With the implementation of this project, the aim is to develop a chatbot for bible study using a voice-enabled chatbot using speech-to-text recognition technology.

### OBJECTIVES

To achieve these aims, the objectives for implementing this project are.

1. To survey existing technology and choose the best means to approach the system.
2. To choose and design the best model and architecture suitable for the system.
3. To implement the system using the best suitable programming language needed.
4. To evaluate the system after implementation.

## Methodology

The methodology states how to achieve the objectives mentioned above.

Based on the first objective, "To survey existing technology and choose the best means to approach the system." There will be a literature review on existing systems like this or relating to this project. Then compare them side by side, looking at existing flaws in the already existing ones, and choosing the best methods based on existing findings to approach the project.

Based on the second objective stating, "To choose and design the best model and architecture most suitable for the system." The best design is selected to approach this project and under this is where we have the model, algorithm, and architecture. There are various architectures already in place, depending on the method applied. There will be a review and research on existing architecture and how to apply it to existing systems. The UML model and design tools entail the necessary actions to model and design this system. The system will use models to illustrate its processes using diagrams such as use cases, Activity diagrams, and Sequence diagrams.

Based on the third objective stating, "To implement the system using the best suitable programming language needed." Implementing the system can use various programming languages, but the best language will go for the implementation. During this, a review will take place comparing the components, libraries, and functions the languages offer and the speed at which they process the code. Hence, the python language is the right choice to write down the backend code and the flask web framework for the front end.

Based on the fourth objective stating, "To evaluate the system after implementation." Every system, upon completion, must be evaluated or tested to ensure its functionality level. To accomplish this, evaluate the system based on the features' usability and output accuracy when access to a function is available. Get user feedback through a questionnaire using google forms and analytics.

## Significance of study

This project uses Artificial Intelligence and human-to-computer interaction principles. It helps to give out the best information based on already set rules for the system to follow. This proposed implementation of the chatbot is a big delve into the future. Nowadays, technology has become more advanced daily, and everyone is looking for the best and easier way to perform their daily activities. Some of the paramount importance are as follows.

1. Effective recommendation of scriptures to users by the chatbot based on privacy assurance: Most of the time, individuals like to share their issues or concerns, but due to this privacy factor, they do not feel comfortable doing so. With the implementation of this chatbot, the issue of another party getting to know the conversations is of no significance as the chatbot will be the only party to listen and recommend the best scriptures and motivational books to help that situation.
2. Reduces physical contact to the minimum: In times of the world's pandemic, this project's implementation will help reduce the need for individuals to meet one-on-one with individuals to discuss. The chatbot will be an alternative means for individuals to converse.
3. It will help improve the user's connection to God more.
4. It helps bring individuals closer and study the Bible much more.

Asides from the stated significance of this project, this will help to bring individuals more conversant with the Bible. The study aims to impact the user positively with ease because it is from their comfort zone. It can access anywhere and anytime.

## Limitations of study

As in all projects, risks and limitations will always be due to the project's time, quality, scope, and financing. One of the limitations is that the chatbot will not understand the human context. The chatbot understands only information that applies to the rules set. During a conversation with the chatbot, it might not be able to provide appropriate information or outputs if specific specified keywords are missing. If utilized improperly, it can misinterpret users' intentions.

## Project organization

This project consists of five chapters. The first chapter entails the introduction to the project. Background information, the problem statement, the study's goals and objectives, the methodology utilized to complete the project, the study's importance, its limits, and the project organization are all included in the first chapter. The second chapter is the literature review, highlighting a definitional overview of the literature and relevant terms alongside existing systems. The third chapter is the system analysis, and the design entails the methodology and procedures used to achieve this project. The fourth chapter discusses the analysis and results of the outcome. In contrast, the last chapter covers the conclusion, summary of findings made, and relevant recommendations.

Chapter Two

# Literature Review

## OVERVIEW OF CHATBOTS

This section of the chapter will cover the basic overview of a chatbot and all it entails.

### BRIEF HISTORY OF CHATBOTS

In 1950, a computer scientist named Alan Turing proposed the Turing Test to determine. It was as an outcome of this test that chatbots became widely known. The first-ever built chatbot is Eliza, developed in 1996. Its sole purpose was to act as a psychologist giving feedback to the user's inputs in the form of questions. It made use of a simple sequence matching and a format-based response technique. Its dialoguing ability was not the best, but it was sufficient to disorient individuals when they individual is unfamiliar with how a computer operates. It gives the individual the drive to develop other chatbots(Ayanouz et al., 2020).

In 1995 the chatbot, namely ALICE, was developed and is famously known for winning the Loebnr prize in 2000, 2001, and 2004. Its intelligence depends on artificial intelligence and relies on a straightforward pattern-matching learning algorithm—a language for Markup. (Adamopoulou & Moussiades, 2020).

The development of chatbots over time as a result of improvements in technology and business procedures led to the development of virtual assistants like Apple's Siri, Microsoft CORTANA, Amazon's Alexa, and others.

### TYPES OF CHATBOTS

Due to the vast and various uses of bots, different types of frameworks belong to different chatbots. Based on the knowledge of the chatbots, functions, goals, objectives, method of processing inputs and generating responses, and either human-assisted or the build method.

The knowledge of the chatbots considers the amount of knowledge the bot can access or the number of data sets it trains. Under this, we have.

1. **Open Domain chatbots**: this type talks about general topics and responds appropriately, according to the area and proximity of its knowledge.
2. **Closed domain chatbots:** these types focus on a particular area of knowledge and are not capable of responding to questions outside this area.

The functions classification of chatbots considers the closeness of the chatbot to the user, the number of transactions, and the tasks the bot performs. Under this category, we have.

1. **Interpersonal chatbots:** These types provide services relating to the domain of communication. E.g., restaurant bookings, FAQ bots. They collect information and process it to pass them on to the user. They are not obliged to be friendly, have a personality, or remember information about the user (Mahmoud & Kumar, 2020).
2. **Intrapersonal chatbots:** They mainly operate in the user's personal space. They can comprehend the user just like a human.
3. **Inter-agent chatbots:** While some chatbots need ways to communicate with others, others will become omnipresent. Alexa-Cortana interaction is one illustration of this.

There are also classifications based on their goals and the chatbots' primary goal to achieve. Some of them are.

1. **Informative chatbots** give users access to information it stores or are available from a specific source, such as FAQ chatbots.
2. **Chat-based/Conversational chatbots** talk and have conversations with the user, like we human beings. The main goal is to respond accordingly and accurately based on their inputs through the discussions.
3. **Task-based chatbots:** They perform a specific task they program to achieve. They are intelligent in asking for needed information, understanding the input, and recommending accordingly.

There are also classifications based on how they process the input received and generate a response to the inputs. This explanation entails how the information received is processed and how the outputs would be derived. There are three models used to ensure the bot generates a proper response, and they are.

1. **Rule-based model chatbots:** They respond to the user following a predetermined set of rules, understanding the input text's verbal form without producing any. The programmer explicitly sets the instructions and information, and the bot arranges and displays using traditional patterns. The bot can respond to more queries if the rule database is more extensive than before. However, most existing research on these topics only considers the most recent input message when studying response selection for discussions with one turn. Chatbots that are more like humans can choose from various responses, look back on earlier conversations, and respond to users in the most appropriate way**.** (Lim et al., 2020).
2. **Generative model chatbots:** They generate answers in the best way possible for the others. They can use both current and previous user messages to create a response to the user's input. They are more oriented to think like humans and use machine learning models and deep learning techniques.
3. **Retrieval-based model chatbots** provide additional versatility as they employ APIs to analyze and search the accessible materials. They employ keyword matching, machine learning, or deep learning strategies to find the best response. They retrieve some responses the user already created from a guide before applying the paired approach to the response selection.

We have a classification that considers the number of functions determined by the intervention of the individual in their components. Human computation offers greater flexibility and robustness. However, it is still slower than a machine, making it difficult to scale to more user demands. (Mydyti & Kadriu, 2021).

Chatbots also have a category according to the platform's permissions for development. The media could be open source or closed source.

The open-source platforms allow the programmer to have the most say and operations in most aspects of the implementation. The closed source platforms are safer and more secure to implement the bots. Some would say it has a significant disadvantage due to its lack of openness. It is highly recommended not to use for implementing small-scale chatbots. It is more appropriate for large organizations, with the data that the companies will collect(Adam et al., 2021).

As we can see, we cannot say the application of chatbots can belong to one area. Still, these categories exist and can be applied depending on the project and the proportions of functions the individual wants the capability to be.

## BIBLE STUDY OVERVIEW

Bible study is applying a set of diverse disciplines to the study of the Jewish and Christian scriptures, the Bible. It is an in-depth look at the message of the whole Bible, including its original context, language, and overall mission and purpose(Efiong, 2016).

The critical purpose of bible study is to enable individuals to make an informed choice and a genuine response to the Bible. (Efiong, 2016)

Applying this to a larger society helps to guide humans' consciences in the light of the scriptures towards a fruitful living and service to humanity and God. Outlined below are some of the aims behind the regular study of the scriptures.

1. **Spiritual and character formation:** The goal here is to help individuals grow in Christian faith towards a holistic lifestyle and wholesome interpersonal abilities as a sign of one's dedication to Christian discipleship. (Efiong, 2016)
2. **Christian worldview/ Intellectual formation:** Regular Bible study equips one with a solid knowledge base, the capacity for critical thought, and the ability to engage in respectful dialogue about opposing viewpoints. Along with exhibiting acquaintance with a wide range of topic material in the humanities, social sciences, and natural sciences, it also assists in demonstrating growing critical thinking skills.
3. **Preparation for service in the world:** It helps demonstrate effective and selfless service in both members and helps people get ready for effective service in the world, especially in their chosen field of study. Additionally, show them how to apply their self-awareness and the Bible to their direction in life.

### BIBLE STUDY IN A DIGITAL AGE

The way people study the Bible has changed dramatically since the invention of the printing press in the year 1440 when the first Bible was published and made widely accessible. Previously, there was no such thing as a printed copy of the Bible to learn. From here, we can say the technological advancement of the Bible began and the ways they can be studied. Over the years, more technological advancements came along the way. They changed how individuals study the scriptures, one of which we could use Oral Roberts. For instance, he did more than create Marshall McLuhan's electronic network when he initially placed his palm over a camera lens. He urged spectators to connect by spiritually touching their televisions. (Davies, 2019).

Beyond these means, the field of religious gadgetry to help stir up the faith of individuals exploded through the twentieth century with devices such as Digital Tibetan prayer wheels, Buddha chant Boxes, Scientology E-matters, and Christian audio bibles.

The Buddha phone came into existence in 2007 on the streets of Shenzhen, originally named the Shaolin phone. The handset offered all the standard functionality a phone should. Still, the Symbian S60 operating system also featured a virtual prayer room where Buddha and various Bodhisattva can worship while on the go. These other religions began to embrace the fantastic notion of enabling a feature that would fully assist Christians worldwide. It will help them study the Bible and even alert them when they have last visited or left the area on the phone.

Another excellent example of means of bible study in the digital age is the indestructible Bible. Since religion is something to ponder on, as Jeffrey Mahan explains, examining faith mediation is crucial to comprehending religion itself (Davies, 2019). They are audio bible books designed to serve as the main spreaders of the Christian message even in harsh environments. The "military bible stick," a multifunctional mp3 player aimed at the military duty people, the "omega audio bible," a solar-powered audio testament, and "the proclaimer," a hand-cranked audio player, are a few examples. All these were implemented in line with new technological methods to make studying the word more accessible and reachable.

These tools function in situations of conflicting and competing beliefs; thus, their durability as useful things secure their longevity and promote the religion they represent. (Davies, 2019).

(Hannah Stevens, 2020), she took on the research of finding out the effects of Information Technology on religious practices. Hence, she concluded that a vast 70% read the scripture online, and 56% of these young adults have searched online for more technologically inclined ways to carry out their religious practices.

Hence, Bible study in a digital age is a growing trend and is viable across various areas regarding Information Technology.

### SCRIPTURE RECOMMENDATION

According to Collins Dictionary, scripture refers to writings regarded as holy in a particular religion. Based on the intent of this model, the focus is the Bible in Christianity. Scripture recommendation is one of the subdivisions of or significant areas of bible study. Various scriptures in the Bible are applicable across multiple ideologies based on what the situation may presume or entail. Each scripture has a message or understanding concerning a particular term and should only apply to that area (Michael & Jeremiah, 2016).

Therefore, recommending a scripture must be done in line with the context of the individual. Hence implementation of this model comes into play. A chatbot that uses artificial intelligence to understand the user's purpose and provide scriptural recommendations based on the user's dialogue is what this project uses.

## SPEECH-TO-TEXT RECOGNITION

This section will entail a detailed review of speech-to-text recognition and its function.

### INTRODUCTION TO SPEECH RECOGNITION

According to (National Center for Technology Innovation), speech recognition is a technology that can recognize speech, allowing the user's voice to serve as the primary interface between the user and the computer. The program can identify words and expressions in a loud conversation and translate them into a format whereby the machine can read and understand them appropriately. A speech recognition system can be categorized based on the following parameters.

1. **Speaker:** There are different types of speakers with different voice configurations. Hence, various models exist for a particular collection or type of speaker.
2. **Vocal sound:** In voice recognition, the way the speaker delivers his or her audio material is also very important. Different models can identify unique or distinct expressions with pauses in between.
3. **Vocabulary:** Words are essential in establishing the overall performance and measurement of the system, particularly in terms of size and complexity.

### BASIC SPEECH RECOGNITION MODEL

Some standard steps should be followed for every speech recognition system, as shown in figure 2.1.



**Figure 2.1: Speech Recognition model**

1. **Pre-processing:** The speech signals are converted into other digital signals to be processed later. The digital signals are then moved to first-order filters to flatten the signs insubstantially hence why the signal energy rises at a higher level.
2. **Feature Extraction:** Various sets of expressions that sync with the speech signals gather as one. Evaluate the features by processing the acoustic waveforms. The goal is to provide a reliable representation of the input signal by adding a succession of pertinent data. Using various feature extraction techniques, however, the following are a few of them that this project uses:

* **Linear Predictive Coding (LPC):** Approximate speech samples are a linear combination of speech samples from the past. Block digital signals into frames of N samples. Then each sample frame is spaced to minimize the discontinuation of signs. Each framed window is then autocorrelated, after which the LPC analysis converts each frame of autocorrelations into an LPC parameter set (Suta et al., 2020). Figure 2.2 shows the linear predictive coding process.



**Figure 2.2:Linear Predictive Coding Process**

* **Mel-Frequency Cestrum Co-efficient (MFCC):** This process is a sophisticated technique that uses an individual's hearing perception system. It applies specific steps to a signal received through a specific input, such as packaging the speech waveform to remove interference. The Mel filter bank algorithm projected against the Mel spectrum to replicate human hearing, harmonic analysis to convert each frame from a time domain to a frequency domain, and windows it to minimize movement discrepancies.
* **Dynamic Time Warping:** This algorithm is very versatile. One of the significant uses is computing the similarity between double series that could differ in speed based on the dynamic programming method. The aim is to align two feature vector sequences iteratively until an optimal match is affirmative.

1. **Acoustic Models:** This is the cardinal part of automated speech recognition. The process establishes a possible similarity or correlation between auditory information and phonology. This process's primary goal is a relationship between the actual speech units and auditory observations.

**Language Models:** Train this artificial intelligence model to predict the next word or words in a text based on the preceding terms. The model is part of the technology that predicts the next word the individual wants to type on a mobile device, allowing faster message completion. It distinguishes between words and phrases with similar sounds.

**Pattern Classification:** This examines similarities between unknown patterns and good references to established practices. Patterns recognize speech after the system goes through training at the testing point. In order to achieve Pattern matching, use the following methods:

1. **Template Based Approach:** This method includes storing a reference library of distinct speech patterns to represent dictionary words. Compare the uttered word and match it with the reference dictionary to identify it.
2. **Knowledge-Based Approach:** This technique automatically trains the system to produce a set of production rules from samples using a distinctive collection of voice attributes.
3. **Neural Network-Based Approach:** Use this for solving extremely complicated recognition tasks. The general overview is to compile and integrate knowledge from various sources concerning the problem.
4. **Statistical Based Approach:** This involves using training methods to model the differences in speech statistically.

### SPEECH-TO-TEXT CONVERSION METHODS

Speech-to-text conversion is the process of turning spoken words into written texts. It is similar to speech recognition; however, speech understanding is more complex. They combine various methods for their efforts while using the same voice recognition stages and concepts. Here are a few of the conversion techniques that this project will use.

* **Hidden Markov Model (HMM):** According to (Ayushi et al.,2018), This is a statistical model used in voice recognition. Speech signals can be considered approximations or short-time stationary signals. Real-time speech-to-text conversion is helpful for those who use mobile devices. It is conditional on the following factors:
* **Recognition accuracy:** Recognition compares different test or speech patterns to each class reference pattern and assesses the degree of similarity between them. It should be completely self-contained and unaffected by the speaker.
* **Recognition speed:** Users become irritated if the system takes too long to recognize speech and the system loses its value. Signals go through three stages:
* **Pre-processing:** Transform speech signals into speech frames, which produce a single sample and reduce noise.

1. **HMM, training:** Create a pattern representing a class's characteristics by contrasting one or more speech sounds from the same category.
2. **HMM, recognition** entails calculating a similarity score between several unknown test patterns and each sound class reference.
3. **Artificial Neural Network Classier (ANN) based Cuckoo Search Optimization:** Better Recognition, communication, and noise filtering are all benefits of using this. Accomplish this process by following a three-step procedure.

* Pre-processing speech signals is a vital part of speech recognition. The aim is to eliminate avoidable waveforms of the movement.
* Extract Mel Frequency Cep-strum Coefficients (MFCC) and Linear Predictive Coding Coefficients (LPCC) from the speech signal.
* Classification: As a classifier, use an artificial neural network. The neural network in question, according to (Ayushi et al., 2018), is a three-layered classifier with n input nodes, one hidden node, and k output nodes. The input layer uses two inputs: the Mel Frequency Cep-strum Coefficients and Linear Predictive Coding Coefficients features. The features are inputs that networks get trained with and produce output concerning them. Figure 2.3 shows the speech-to-text flow of a system.



**Figure 2.3:SpeechToText System Flow**

### DEVELOPING AUTOMATIC SPEECH RECOGNITION

According to (Ghai & Singh 2012), the following steps have been observed and recognized as appropriate in developing a general automatic speech recognizer. The steps include.

1. Collect the speech through a one-way direction and silent microphone.
2. Secondly, parameterize signals using an appropriate feature extraction technique.
3. Acoustic analysis: According to (Paul Boersma, 2013), acoustic analysis is an increasingly needed skill in linguistic analysis. This analysis is the process of training waveforms and converting each waveform into a series of coefficient vectors. This method will help notice less significant and articulate speech patterns depending on the wavelengths (Elsherbeny et al., 2022).
4. Model definition: A prototype of Hidden Markov Models defines each element of the various task vocabulary.
5. Model Training: Each Hidden Markov Model instance is initialized and goes through training with the already trained data.
6. Task definition: The grammar required for the speech recognizer is defined.
7. Recognition: Recognize input speech signals unknown at this stage.

## HUMAN-COMPUTER INTERACTION

According to Wikipedia, human-computer interaction is the research in the design and applications of computer technology, which main aim is the operationality of the interface between the system and user. They frequently happen when carrying out a straightforward automated task, referring to a configuration in which people and computer-related equipment share a functional area. (J. Wang et al., 2021).

According to (Zhang et al., 2002), as non-rational humans, we have several possibilities to connect with various technology in non-rational or bounded-rational ways for diverse goals. In every potential instance of a human-technology conversation, they should contain cognitive, emotive, and affective aspects.

Human-Computer interaction is a broad and diverse system. The system is connected by the growing concept of usability and the continuous advancement and push towards valuing human activities and experience as the primary motivation in technology(B et al., 2017).

## PYTHON

One of the most widely used high-level programming languages is python. Python is an interpretable object-oriented interactive programming language. It contains dynamic typing and binding, modules, classes, exceptions, autonomous memory management, and high-level data structures like lists, tuples, sets, and associative arrays (sometimes referred to as dictionaries). (Dhruv et al., 2021)

Parallel computing systems utilize this function. It offers a relatively simple coding syntax while remaining a sophisticated programming language.

Python offers several advantages over other languages, such as libraries that cuts code by one-third for programmers. Because of this, python has surpassed all other languages in machine learning. (Dhruv et al., 2021; Shanmugapriya & M., 2020)

### PYTHON IN NATURAL LANGUAGE PROCESSING

Python is an excellent high-level programming language for programmers because of its clean, simple, and explicit rules. Python is an excellent choice for developing this chatbot because it has much third-party expansion library support. The libraries make it versatile in machine learning, artificial intelligence, natural language processing, and other fields. (Mishra, 2019)

The Natural Language Toolkit (NLTK) is a popular python library used mainly for natural language processing. Natural Language Toolkit is a python package that can swiftly and efficiently process Natural Language text. The library originates from the University of Pennsylvania as a natural language processing research and teaching tool. (M. Wang & Hu, 2021)

Natural Language Toolkit also includes an active discussion forum, as well as an easy-to-use interface with over 50 corpora and lexical resources like WordNet, as well as a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength Natural Language Processing libraries.

It can perform secondary processing of the processing results in conjunction with python's standard solid library and other third-party libraries. It gives a good foundation for working with complicated text. Because the Natural Language Toolkit is not a standard python library, the programmer has to download it before use.

### PYTHON IN SPEECH-TO-TEXT RECOGNITION

To implement speech-to-text recognition, use a python library named SpeechPy. SpeechPy is a python library that includes speech preparation techniques, speech features, and necessary post-processing procedures. It includes the most often used speech features, such as Mel-Frequency Cestrum's and filter bank energies, and the log-energy of filter banks. (Torfi, 2018)

The package aims to give academics an essential tool for extracting and processing speech features in applications like Automatic Speech Recognition and Speaker Verification. SpeechPy meets the requirements needed. It includes the most often utilized speech elements and the most important pre and post-processing processes. (Dhruv et al., 2021)

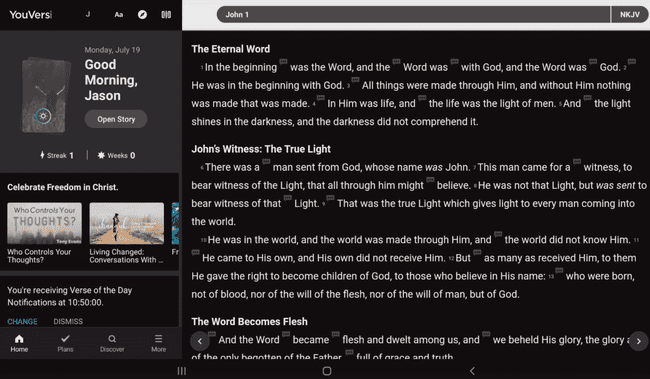
The software is available for free as open-source software. SpeechPy is well-known for continuous integration for instant error checking and change validity.

## REVIEW OF EXISTING SYSTEMS

This section will be discussing a review of some bible applications already existing and are in mass circulation. There are various mobile bible applications available for use across several platforms. This study will be looking at a few of such applications.

### YOU VERSION BIBLE APPLICATION

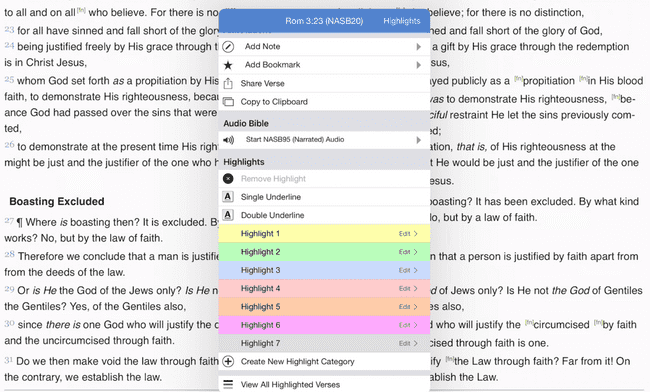
You version is an online and mobile application founded by Bobby Gruenewald and Life church in 2008. It is available across the android, windows, and iOS platforms. According to YOUVERSION, in 2020, the application will feature 2062 bible versions in 1372 languages. The application uses an algorithm that recommends scripture based on the already set principles and rules for the application. The algorithm also helps guide scriptures based on user experience.



**Figure 2.4:YouVersion BibleApp**

### BLUE LETTER BIBLE APPLICATION

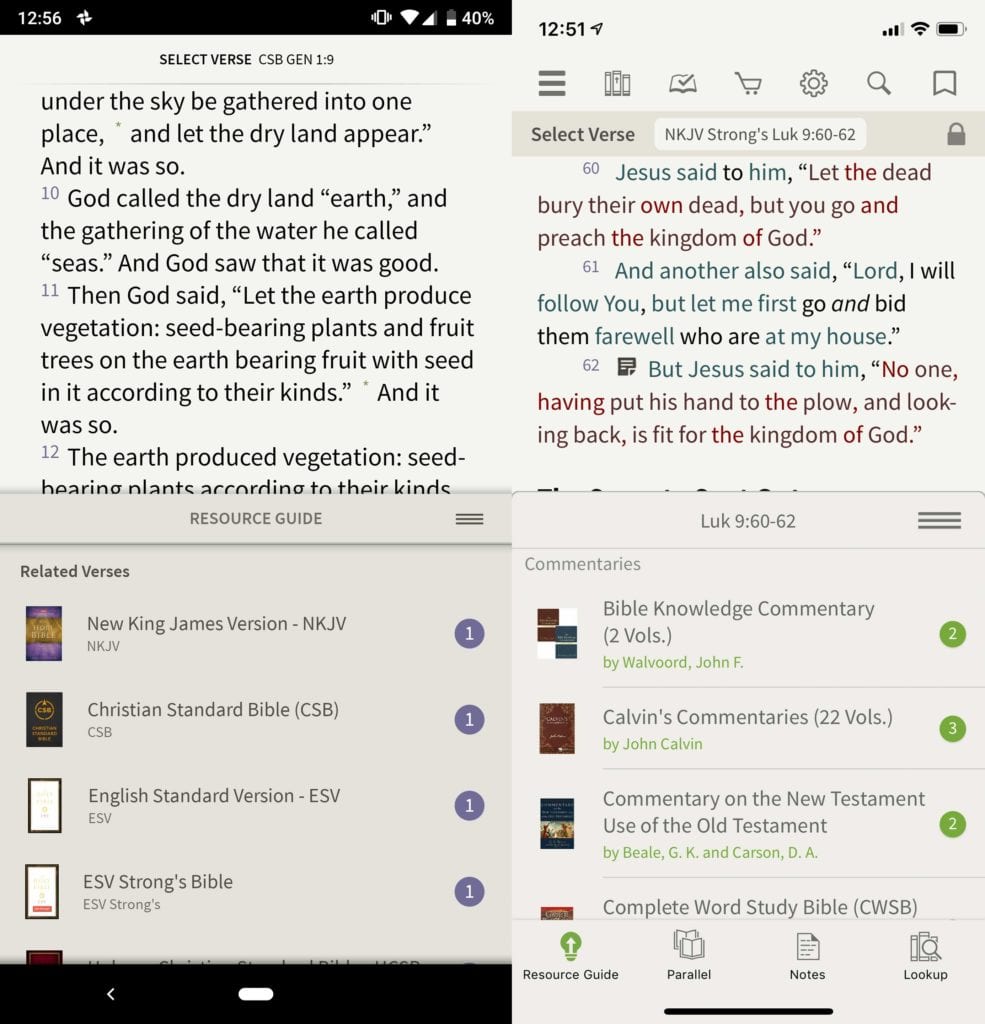
The 1995-founded Blue Letter Bible App is a study tool that claims to do reading, researching, and studying the Bible simply and pleasurable. Strong concordances for the entire Bible are supported, along with 12 English Bible translations with integrated study resources, including "Gesenius Lexicon" for the Old Testament and "Thayer's Lexicon" for the New Testament. The application is free to download and has no ads or in-app purchases. The application is available for download only on the iOS and Android platforms. One of the unique features of this app is the ability to provide a way to take notes, either on a specific verse or just a general note, with another command to make various notebooks.



**Figure 2.5:Blue Letter BibleApp**

### OLIVE TREE BIBLE SOFTWARE

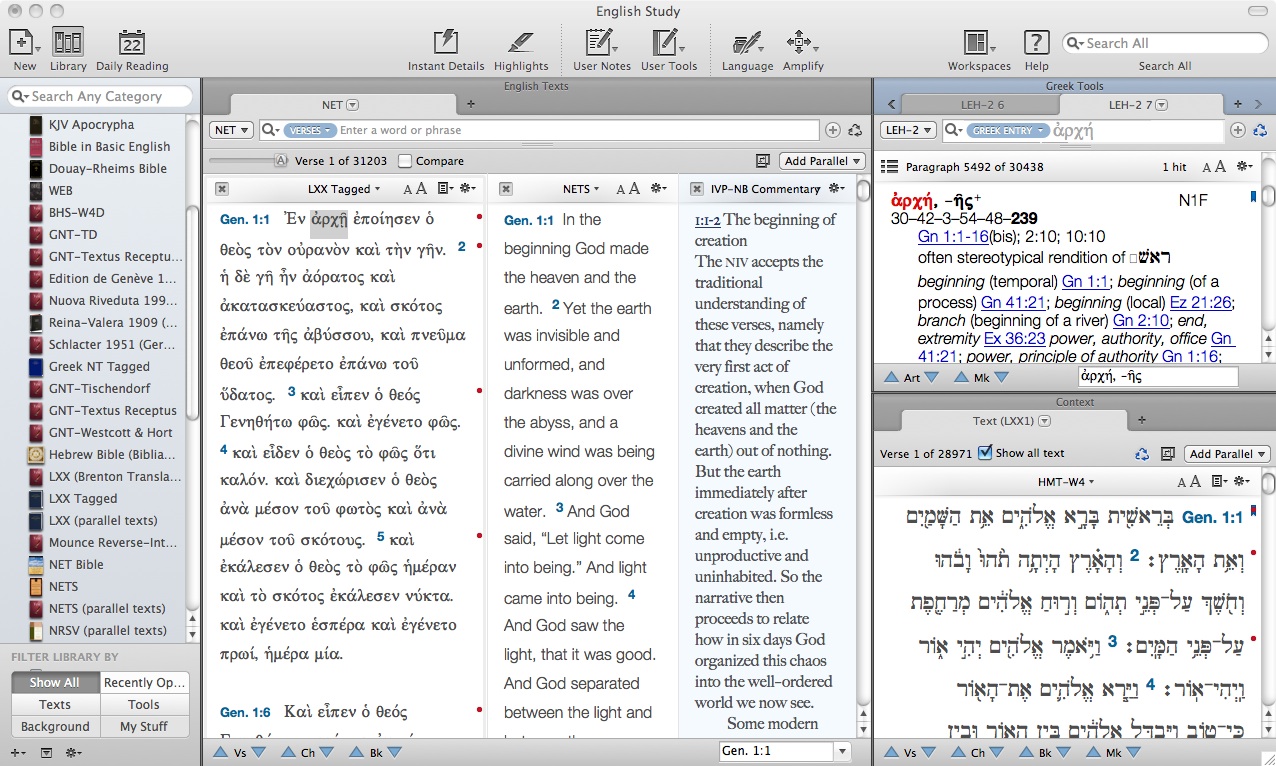
Olive tree bible software is an application famous for being a very friendly and easy user interface program. The application provides supporting resources with a special unique feature of highlighting a word, bringing out its contrasting relation to other scriptures, and explaining it further. It can also recommend scriptures, spot search, and offer them in various languages. The app's algorithm design is in a way it recommends as the user goes away, where it recommends based on the user's session over a period. Available on the iOS, Windows, and Android platforms.



**Figure 2.6:Olive Tree Bible Software**

### ACCORDANCE BIBLE APPLICATION

Accordance is a mobile bible study program initially written exclusively for the Mac OS. However, it is now available on the iOS, windows, and android platforms developed by the Oaktree software group. It originated in the year 1988. One unique feature of this app is its speed for searching for scriptures and lookups. The application offers in-app purchases for some of its features, which might be a down part for some users. The system's algorithm is based on the user's experience and is customized based on such parameters. It also has other functions like spot search and scripture recommendation. This application does not have in-note functionalities unless the user embeds external note-taking applications into the app.



**Figure 2.7:Accordance BibleApp**

**Table 2.1:Comparative Review of Existing Bible Apps**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BIBLE APPLICATIONS | DOMAIN | PAID/FREE | SCRIPTURE RECOMMENDATION | SCRIPTURE SEARCH | BIBLE STUDY | SPEECH RECOGNITION |
| You Version | Android, Windows & iOS | FREE | yes | yes | yes | no |
| Blue Letter Bible | Android, Windows & iOS | FREE | no | yes | yes | no |
| Olive Tree Bible | Android, Windows & iOS | FREE | yes | yes | yes | no |
| Accordance Bible | Android, Windows & iOS | PAID | no | yes | no | no |

CHAPTER THREE

# SYSTEM ANALYSIS AND DESIGN

## INTRODUCTION

This chapter discusses the study and design for the implementation of the application "Intelligent Bible Chatbot." This chapter will include the system analysis and design methodology, also known as a Systems Development Life Cycle (SDLC), which will entail the development and ongoing maintenance processes. [ (Otero, 2018)]

The methodology, requirements, and model used in the system design will be discussed in further detail in this chapter, i*ncluding* using UML diagrams such as use cases, activity diagrams, sequence diagrams, and entity relationship diagrams.

The design process starts by creating an analysis. This project entails a thorough examination of present systems and the identification of possible research difficulties, as well as the development of a new and improved system. The system design specifications are the next phase. Without the proper collection of the needs, the system would not function.

## SYSTEM ARCHITECTURE

The system architecture is a conceptual framework that specifies how a system is structured, the behaviour, and the system's additional capabilities. The system architecture helps assist the logic and understanding behind the system's design and model usage. (Tamrakar & Wani, 2021)

The architecture helps to graphically display the system's components, visualizing how to design the function together to accomplish the overall system or model.

Based on the implementation of this project, the architecture of this chatbot includes the following layers:



**Figure 3.1:Chatbot Architecture**

1. Presenters: This layer oversees making a channel (messenger, web, telegram) call to display text, a button, an image, or a video.
2. Flow: This layer executes the chatbot's operations and logic. It uses data, connects with web services, and displays information repositories throughout the development.
3. Quick Replies: This layer manages the rapid responses from the channel and uses functions provided by the flow to carry out various bot operations.
4. Post backs: This layer gets user input from the flow module or repository to activate actions.
5. Repositories: The most typical procedures for obtaining data and templates from our database or online services using the layer's templates.
6. Templates are the acts or intents to which the bot will respond. They are frequently defined using natural language processing (NLP) and include some form of data validation.

## SYSTEM REQUIREMENTS

Requirements for software have two categories, namely, functional and non-functional requirements. (Demirel & Das, 2018). The explanation for the categories as mentioned earlier will go into detail below.

### FUNCTIONAL REQUIREMENT

When gathering requirements for a system, the functional requirements inform the user of the system, what the system can do, and how the system behaves concerning the system's functionality. The functional requirements of the system are as follows:

1. The user will be able to search for bible scriptures in the Bible.
2. The system will be capable of recommending verses in the Bible to users based on keywords during conversations.
3. The system will be able to have conversations with the user via speech-to-text recognition.

The system will be able to detect and give out an error response if it is unable to comprehend inputs or is unable to find a keyword outside the scope configured to the system.

### NON-FUNCTIONAL REQUIREMENT

Non-functional requirements do not require the user, particularly with the system's service. They are the quality characteristic of a software system, such as security, availability, or performance. Explaining the capabilities of the system as well as its limitations. (Demirel & Das, 2018)

These requirements differ depending on the numerous software properties (product requirements), the desires of the company creating the program (organizational requirements), or because of outside sources.

The non-functional requirement for this system includes:

1. Usability Requirement: The user of this system should be able to use the bot with a straightforward user interface. The bot's voice should be clear enough for the user to comprehend and use the language the user will comprehend.
2. Responsiveness Requirement: The system should be able to respond to the user's input. Suppose any external interruption occurs during the usage. In that case, it should return to the same state before the interruption occurred.
3. Performance Requirement: When calling the bot or conversing with the user, it should not take long to respond. Instead, it should take a brief time (a matter of seconds) before generating a response.
4. Security Requirement: The bot should be secure from malicious attacks, and users' information should be kept safe.

## SYSTEM DESIGN

According to (Charnley et al., 2011), knowing the aspects that drive the entire system design gives the designers the knowledge they need to operate more successfully. Designing the system is done by collecting requirements. There are three significant specifications: Physical design, Logical design, and Conceptual design.

### PHYSICAL DESIGN

Physical design is one of the specifications in system design; it has two types, namely input design, and output design

### INPUT DESIGN

Input mechanisms make it easier to enter data into a computer system, whether highly structured data like objects and tasks or unstructured data like comments. The establishment of data entry screens and other forms on which users can type or write data are examples of input design.

The purpose of input design for the chatbot is to have clear and cut information from the user for the system to process quickly. The input design will use a microphone embedded in the system to get input from the user.

### OUTPUT DESIGN

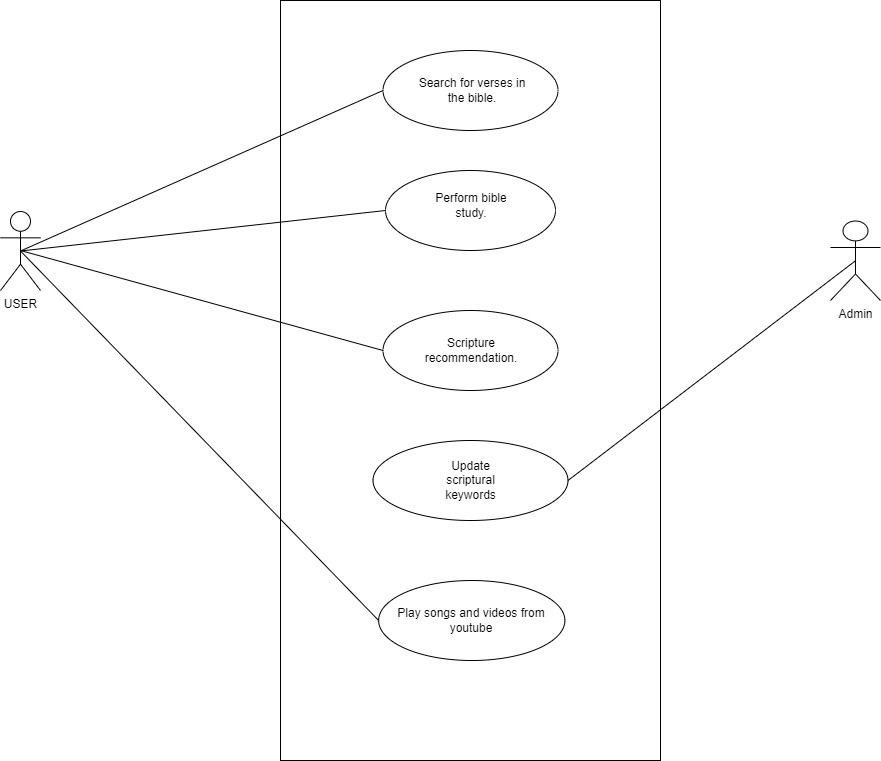
The outputs mechanism's purpose in system design is to show users information in a way they can grasp accurately at first sight. The output design essential principles represent how to use outputs and make them more accessible for the users to understand. In the development of the chatbot and apt user interface, the significance of the simple display of information is of paramount importance.

### LOGICAL DESIGN

The term "logical design" describes an abstract illustration of a system's inputs, outputs, and data flows. Achieve logical design by modelling, which involves creating a graphical representation of the actual system. The logical design includes the concept of business entities and relationships, which explains operations without specifying how to process them. Choose the physical specifics as these conceptual models combine into physical models during the design phase.

## USE CASE DIAGRAM

A use case is a type of UML (Unified Modelling Language) that shows the graphical representation of a user's possible interaction with a system. It is employed to monitor system requirements. It can also be relevant in communicating with many project stakeholders, including owners, system developers, and potential users. Generate test cases using use cases for the criteria of system acceptance. (Klimek & Szwed, 2010)



**Figure 3.2:Use Case Diagram for Chatbot**

## ACTIVITY DIAGRAM

An activity diagram is a UML (Unified Modelling Language) used to represent the system's process flow and dynamic characteristics. It emphasizes the actions taken in a use case instance of an object and the words that go into implementing an operation (method). (Bhattacharjeem & Shyamasundar, 2009)

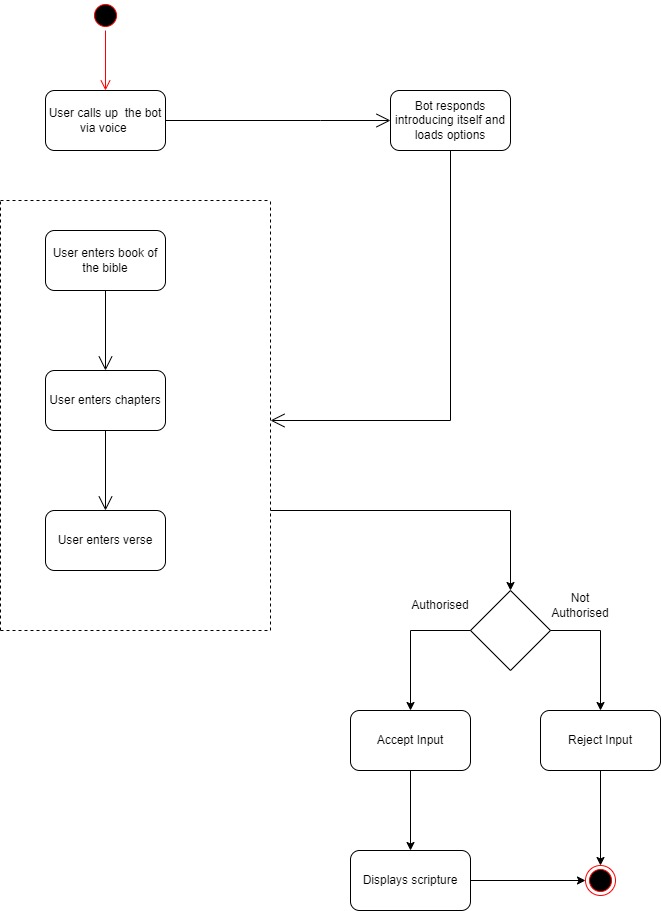
The used shape types and notations include.

1. Rounded rectangles represent actions they perform.
2. The diamonds represent decisions.
3. The black circle represents the start (initial state) of the workflow.
4. The encircled black circle represents the end of the flow (final state)

Diagram

Description automatically generated

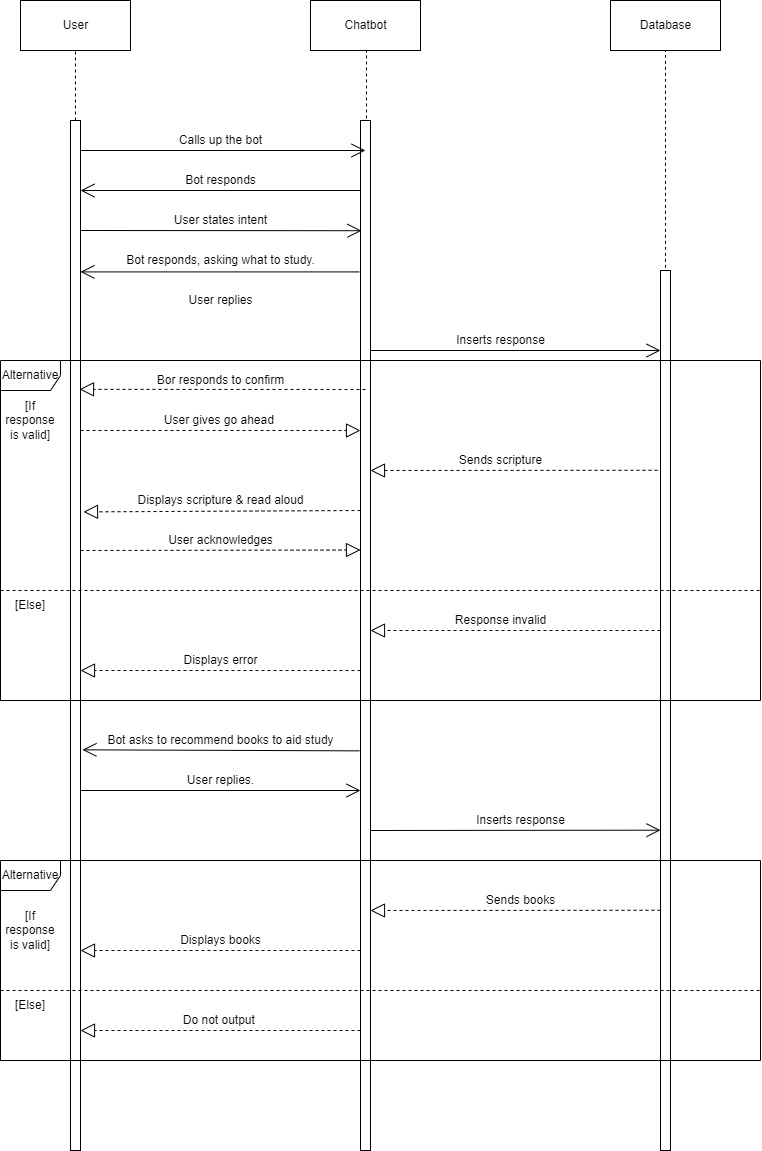
**Figure 3.3:Activity Diagram for Bible Study**



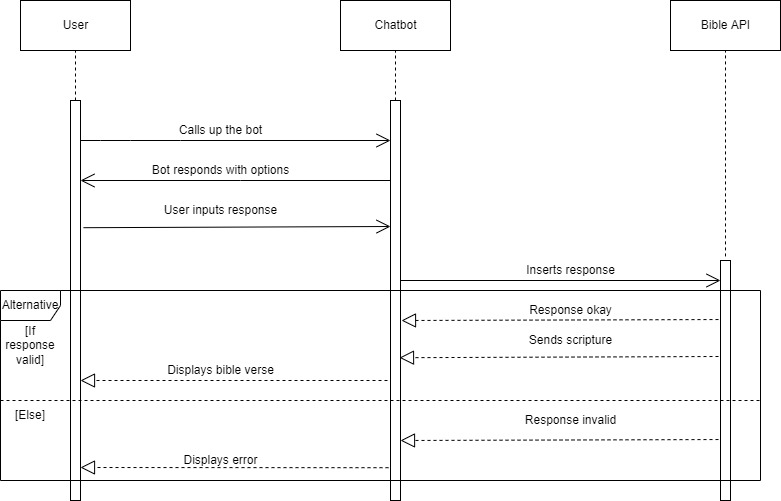
**Figure 3.4:Activity Diagram for Bible verse search**

## SEQUENCE DIAGRAM

A sequence diagram demonstrates how a group of objects interact and in what order a type of UML (Unified Modelling Language). A sequence diagram focuses on the chronological arrangement, sequential order, or transmission order of communications. (Song, 2013)



**Figure 3.5:Sequence diagram for Bible study**



**Figure 3.6:Sequene diagram for Bible verse search**

CHAPTER FOUR

# IMPLEMENTATION

## INTRODUCTION

This chapter entails in-depth details about the system, hardware and software requirements, and the interaction flow with screenshots of how the bot conversation flows for each function. The chapter will also contain various modules and other programming concepts to actualize the system.

## SYSTEM REQUIREMENTS

According to [cite here], system requirements are the configurations a system must have for a hardware or software program to function smoothly and effectively. During the development of this system, the following software, hardware, and software deployment requirements are helpful to ensure a successful implementation, as displayed in Table 4.1, Table 4.2, and Table 4.3, respectively.

**Table 4.1: Software requirements**

|  |  |
| --- | --- |
| Requirements | Software |
| Operating System | Windows 11 |
| Database | GetBible API |
| Development tool/IDE | Pycharm |
| Programming language | Python |
| Programming language version | Python 3.9 |

**Table 4.2: Hardware Deployment Requirement**

|  |
| --- |
| 8GB RAM |
| 1TB Hard drive |
| Intel® Core ™ i7-7500U Processor |
| 2.70Ghz |

**Table 4.3: Software Deployment Requirements**

|  |
| --- |
| Python3.9 |
| Speech Recognition |
| Python Text to Speech |
| Python whatkit |
| Command |
| PyAudio |

## SYSTEM INTERFACE

A system interface serves as that connection point that helps facilitate communication and interaction between two or more, often unrelated, independent systems.

Due to the nature of this project, there are two main classes of users which include:

1. USER: The user here is the individual that calls up the bot to perform the actions that the programmer assigns to the bot. It is also essential to specify here that this bot requires an internet connection, so the user should have an internet connection before using the chatbot.
2. Administrator: The administrator here is the person that handles maintaining the operations of the chatbot to ensure the bot continues to serve its executions properly. The administrator of this chatbot is the only person that can access or control any modification to the answer flow during chats or use of the bot.

### PYTHON FLASK ENVIRONMENT

Flask framework creates a local host when the code is run and executed. Flask is a very minimal light web framework that gives valuable tools and features that help make web frameworks easier on python. Select the framework due to its flexibility and ability to create a web app using just one python file.

Due to the nature of this system's implementation, after creating the local host. The local host directs the user to a web page that will serve as a user interface for the chatbot. Accessing any function notifies the user by saying "listening" to show that the conversation with the chatbot is in session.

Figure 4.1 shows the front interface of the flask environment local server. It entails a brief overview of the bot and its functions.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 4.1:Flask web framework welcome page**

### HISTORY PAGE INTERFACE

After the entire session with the bot, the page redirects the user to another page which displays the main conversation history with the bot in text form for each function the user picks. It displays the history of conversations held with the bot, showing the user input and the bot's output. The flask environment also hosts this interface through a local host.

Figure 4.2 shows the history page after the session with the bot is over for all functions.

Graphical user interface, text, application

Description automatically generated

**Figure 4.2:Flask web framework History Page**

## INTELLIGENT MODULE

This section shows various modules this bot requires during this chatbot's implementation. The system consists of multiple modules that are independent of each other. Modules such as Input Collection, Input Processing, Output Processing, and Output generation have different functions.

### INPUT COLLECTION

This module of the system's implementation handles the user input that the user inputs into the bot. Firstly before using the bot to collect input from the user, the bot has been assigned a name called "Daniel." Before the bot starts, the user addresses it by name or by prefixing it with greeting words like "hey" or "hello."

The chatbot implemented in this project is a bot that uses speech-to-text recognition. Hence, the user uses his/her voice to call the bot or enter any command or input.

To achieve this, implement the speech recognition package in python. The main aim of this package is to be able to recognize the user's speech. To do this, create a source using that package. The source used was that of the computer's microphone connected to the speech recognition package. Through this, the user can input speech into the bot.

The speech recognition package also processes the user's input that the system collects. This package helps to display whatever the user communicates to the bot to be sure the user has inputted it correctly.

### INPUT AND OUTPUT PROCESSING

This system module handles the user input processing to determine the appropriate output. The design of this chatbot is to process various inputs according to the functions it can perform and output various responses. How to achieve the implementations of the various functions will be detailed below.

**BIBLE STUDY FUNCTION**

The python command package is an excellent tool to achieve this. This package aims to view input given by the user as a command to trigger a response and goes through the predefined catalogue to determine which appropriate output that input warrants. Assign various keywords to this function to trigger a response by mentioning a particular keyword or that keyword in a phrase.

**SCRIPTURE SEARCH**

Restrict the user's input to three pieces: the Bible book, the chapter, and the verse. Use the python command package here to collect input from the user for processing and triggering a response.

The processing of getting the information needed, in this case, the bible verse, was used through a bible API (Application Programming Interface). In implementing this chatbot, the API serves as the database for fetching the required information. The API contains the raw data needed to relay the information from the website hosting the API to the chatbot. The API used for this project is called get bible API and requires a good internet connection.

Implement the python Requests package to access the API functions. The function of this package is to allow python to send HTTP (Hyper Text Transfer Protocol) requests. It has a JSON method already for a response when the HTTP request process is complete. This request helps return a response from that source with all the data with such a response. This package connects the API to the internet and request data as needed during the chatbot's implementation.

To access the Bible API's books, chapters, and verses, use the python JSON (JavaScript Object Notation) package. The package helps to store and exchange data in text form. This package was imported into the code and connected to the Bible API to extract data needed from the Bible API to generate appropriate output according to the user's input.

Made use of flask python package because it has a built-in JSONify function to handle serializing the data gotten from the Bible API. It is a lightweight web framework.

**PLAY MEDIA CONTENT FROM YOUTUBE**

To achieve this once the bot has recognized the user's intent or input through the command python library for processing and output response. The Python Pywhatkit package was used to give access to youtube, process the user's intent to youtube, and matches it for proper execution. Then the user is redirected to the youtube page displaying the user's request.

### OUTPUT GENERATION

This implementation module handles the output generation by displaying the response to the user's intent after processing. After processing the user's intent, the bot collects responses from the code source. For initial conversations with the user and the bible study function, there is already a predefined set of responses the bot will respond to based on the user intent. The voice used for response purposes in this implementation is the google supported assistant voice.

For the bible study function and everyday conversations, utilize the google text to speech recognition library to respond to the user in the form of speech format. This library's purpose is to convert text-based details to speech form by creating a function to assign that to it. Choose this library because it has been configured based on this region and gives the voice an African voice, making it more user-friendly for the users. The Python audio library, which offers python bindings for port audio or a cross-platform audio input and output library, was also utilized to enable voice customization. Without this library, the bot cannot respond via speech.

The python text to a speech library function is for the scripture search function—data required to derive a response is derivative from an API. So whatever data API gives is in the form of the text form, thereby making the use of the python text to speech library to generate a response in speech form.

Use the print feature to display the output in text form so that people can view a preview of the conversations held for that period.

## SYSTEM EVALUATION

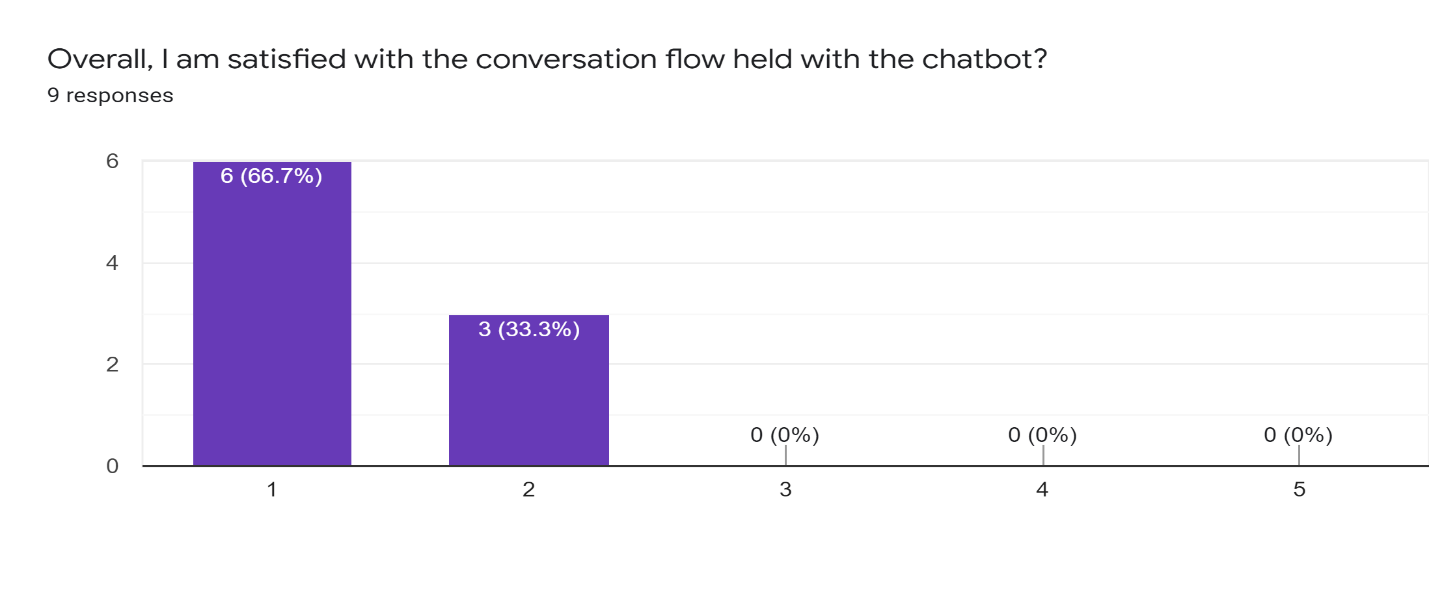
Nine individuals were to evaluate the system, 4 of whom were UI/UX experts. Everyone made use of the bot and tested all the functions laid out for the bot to do.

Last but not least, a questionnaire was given to those people with links to Google Forms to access the form and record their reviews of the bot. An explanation of the question and a summary of the results are below.

Use the average of the shortest path, the inverse of the number of edges, and the Euclidean distance to get the mean of the metrics we employed. The closer the couple, the lower the value.

1. Overall, is the user satisfied with the conversation flow held with the chatbot?

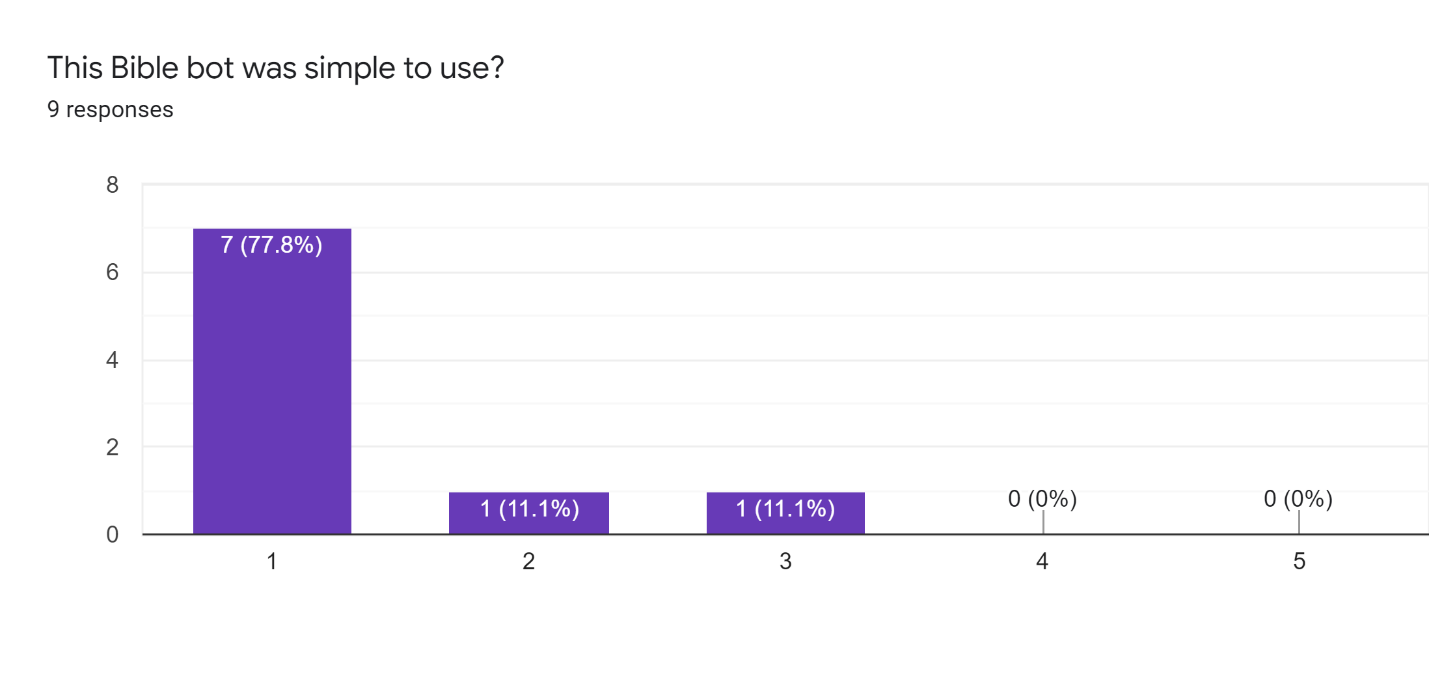
Ask the volunteers to score overall happiness in line with the flow of talks on a scale of 1 to 5, with most users selecting one. Figure 4.3 illustrates the distribution of the responses with a bar chart.



**Figure 4.3:Bar chart showing responses to question 1**

1. The Bible bot was simple to use?

This question assumes that the user found the bot easy to use without stress. Figure 4.4 illustrates the distribution of the responses with a bar chart.



**Figure 4.4:Bar chart showing responses to question 2**

1. This bot performed its functions flawlessly.

This question is to evaluate how well the bot performed its functions. Figure 4.5 illustrates the distribution of responses with a bar chart.

Chart

Description automatically generated

**Figure 4.5:Bar chart showing responses to question 3**

1. The user was able to find the scriptures he or she searched.

This question aims to get the user's feedback on the bot's accuracy in displaying the desired scripture. Figure 4.6 illustrates the distribution of responses with a bar chart.

Chart, bar chart

Description automatically generated

**Figure 4.6:Bar chart showing responses to question 4**

1. The user was comfortable using the bot.

This question aims to determine the extent to which the user performs operations within the bot with ease and simplicity. Figure 4.7 depicts the distribution of responses with a bar chart.

Chart, bar chart

Description automatically generated

**Figure 4.7:Bar chart showing responses to question 5**

1. It was easy to get used to the mode of operation.

This question aims to determine how easy it was for the users to get accustomed to the operations of the bot. Figure 4.8 illustrates the distribution of responses with a bar chart.

Chart, bar chart

Description automatically generated

**Figure 4.8:Bar chart showing responses to question 6**

1. The user could easily interact with the bot using voice.

This question intends to determine if the mode of communication with a bot through speech is efficient enough for the user. Figure 4.9 illustrates the distribution of responses with a bar chart.

Chart, bar chart

Description automatically generated

**Figure 4.9: Bar chart showing responses to question 7**

1. The user is delighted with the rate at which the bot responds.

This question aims to determine the rate at which the bot responds—in other words, determining if it takes little or no time to respond. Figure 4.10 illustrates the distribution of responses with a bar chart.

Chart, bar chart

Description automatically generated

**Figure 4.10:Bar chart showing responses to question 8**

1. The bot can understand the user's speech easily.

This question aims to determine if when the user communicates with the bot, the bot can pick up the right words spoken by the user. Figure 4.11 illustrates the distribution of responses with a bar chart.

Chart, bar chart, histogram

Description automatically generated

**Figure 4.11:Bar chart showing responses to question 9**

1. The user is satisfied with the voice the bot uses for the response.

This question aims to determine if the user is satisfied with the type of voice the bot uses to communicate. Figure 4.12 illustrates the distribution of responses with a bar chart.

Chart

Description automatically generated

**Figure 4.12:Bar chart showing responses to question 10**

1. The bot is satisfied with the user interface.

This question aims to determine if the user is satisfied with the overall interface of the bot during usage. Figure 4.13 illustrates the distribution of responses with a bar chart.

Chart, bar chart

Description automatically generated

**Figure 4.13:Bar chart showing responses to question 11**

CHAPTER FIVE

# SUMMARY, RECOMMENDATIONS, AND CONCLUSION

## SUMMARY

The chatbot developed in this project applies speech recognition and text to speech recognition to help users perform bible study and search for scriptures in the Bible through a voice command.

The integrated development environment used to develop the chatbot is PyCharm. On this backend, the bot is developed with python 3.9, while the front end uses python in conjunction with a framework named Flask.

The primary module in this chatbot includes the input and output collection and processing module (this entails the function processing also). Also, the conjunction module combines all the functions into one python file to connect to the framework. The python packages that played a vital role in the conclusion of this project include google text to speech, python audio, python command, Flask, and python speech recognition.

This project provides scriptures to users based on the conversation with the bot to aid bible study. It can also search for scriptures in the Bible. It also helps eradicate the manual process of communicating with the bot through text when the user could communicate using the user's voice instead.

## RECOMMENDATIONS

For this project, the scripture recommendation was drawn from the code directly. An improvement on this can be to create a database for offline purposes. Furthermore, a script is in a particular format, so the code will just read from it and derive its information. An administrator could oversee updating the script from the database.

Another recommendation will be to train the bot to get used to regular conversations and respond faster to inputs. This training helps make the conversation with the bot faster and improves the user experience.

## CONCLUSION

This project's primary goal is to make scriptural content easily accessible to users using just a voice command, with a focus on the Christian religion.

This project has improved the existing applications by providing such functions with a voice command. For most applications related to religion, the main issue is the manual method of scrolling through and searching for content they need. The user inputs the command and receives the information based on the context of the bot when using the audio command, in any case.

This project has undergone testing, been shown to run without a hitch, and served the intended function.

REFERENCES

Adam, M., Wessel, M., & Benlian, A. (2021). AI-based chatbots in customer service and their effects on user compliance. *Electronic Markets*, *31*(2). https://doi.org/10.1007/s12525-020-00414-7

Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, *2*. https://doi.org/10.1016/j.mlwa.2020.100006

Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020, March 31). A Smart Chatbot Architecture-based NLP and Machine Learning for Health Care Assistants. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*. https://doi.org/10.1145/3386723.3387897

B, K. M., Nagami, T., Sugase, Y., & Takemura, A. (2017). *Human-Computer Interaction. User Interface Design, Development, and Multimodality*. *10271*(October), 593–608. https://doi.org/10.1007/978-3-319-58071-5

Bhattacharjeem, A. K., & Shyamasundar, R. K. (2009). Activity diagrams: A formal framework to model business processes and code generation. *Journal of Object Technology*, *8*(1), 189–220. https://doi.org/10.5381/jot.2009.8.1.a3

Charnley, F., Lemon, M., & Evans, S. (2011). Exploring the process of whole system design. *Design Studies*, *32*(2), 156–179. https://doi.org/10.1016/j.destud.2010.08.002

Davies, H. (2019). Religious Devices: A Survey of Technologies of Worship. *Communication +1*, *7*(2), 5. https://doi.org/10.7275/2f6f-w963

Demirel, S. T., & Das, R. (2018). Software requirement analysis: Research challenges and technical approaches. *6th International Symposium on Digital Forensic and Security, ISDFS 2018 - Proceeding*, *2018-Janua*(March), 1–6. https://doi.org/10.1109/ISDFS.2018.8355322

Dhruv, A. J., Patel, R., & Doshi, N. (2021). *Python: The Most Advanced Programming Language for Computer Science Applications*., 292–299. https://doi.org/10.5220/0010307902920299

Efiong, J. E. (2016). Information Technology in Biblical Studies: Gearing Towards Sustainable Development in Africa. *Proceeding of Nigerian Association of Biblical Studies (NABIS) Conference, June 2016 at Ajayi Crowther University, Oyo, Oyo State, Nigeria.* , *September*.

Elsherbeny, M., Baz, H., & Afsah, O. (2022). Acoustic characteristics of voice and speech in Arabic-speaking stuttering children. *The Egyptian Journal of Otolaryngology*, *38*(1). https://doi.org/10.1186/s43163-021-00192-9

Ghai, W., & Singh, N. (2012). Literature Review on Automatic Speech Recognition. *International Journal of Computer Applications*, *41*(8), 42–50. https://doi.org/10.5120/5565-7646

Griesinger, D. H. (1993). Acoustic analysis. *The Journal of the Acoustical Society of America*, *93*(4), 2256–2256. https://doi.org/10.1121/1.406647

Klimek, R., & Szwed, P. (2010). Formal Analysis Of Use Case Diagrams. *Computer Science*, *11*(1), 115–115. https://doi.org/10.7494/csci.2010.11.0.115

Lim, Y., Lim, J., & Cho, N. (2020). An Experimental Comparison of the Usability of Rule-based and Natural Language Processing-based Chatbots. *Asia Pacific Journal of Information Systems*, *30*(4). https://doi.org/10.14329/APJIS.2020.30.4.832

Mahmoud, M., & Kumar, R. (2020). A Review on Chatbot Design and Implementation Techniques. *International Journal of Engineering and Technology*, *7*(April).

Michael, T., & Jeremiah, O. (2016). *Impact of Inductive Bible Study Method on Christians ; a Case Study of Nigeria Fellowship of Evangelical Students, University of Agriculture Makurdi*. *September*, 278–291.

Mishra, P. (2019). Natural Language Processing Using PyTorch. *PyTorch Recipes*, *8*(5), 165–178. https://doi.org/10.1007/978-1-4842-4258-2\_7

Mydyti, H., & Kadriu, A. (2021). The impact of chatbots in driving digital transformation. *International Journal of E-Services and Mobile Applications*, *13*(4). https://doi.org/10.4018/IJESMA.2021100106

Otero, A. R. (2018). System Development Life Cycle. *Information Technology Control and Audit*, *5*(4), 201–236. https://doi.org/10.1201/9780429465000-8

Shanmugapriya, P., & M., H. (2020). Development of Chatterbot using python. *International Journal of Computer Applications*, *176*(21), 18–20. https://doi.org/10.5120/ijca2020920184

Song, I. (2013). *Developing Sequence Diagrams in UML*. *March*. https://doi.org/10.1007/3-540-45581-7

Suta, P., Lan, X., Wu, B., Mongkolnam, P., & Chan, J. H. (2020). An overview of machine learning in chatbots. *International Journal of Mechanical Engineering and Robotics Research*, *9*(4). https://doi.org/10.18178/ijmerr.9.4.502-510

Tamrakar, R., & Wani, N. (2021). *Design and Development of CHATBOT : A Review*. *May*.

Torfi, A. (2018). SpeechPy - A Library for Speech Processing and Recognition. *Journal of Open Source Software*, *3*(27), 749. https://doi.org/10.21105/joss.00749

Wang, J., Cheng, R., Liu, M., & Liao, P. C. (2021). Research trends of human-computer interaction studies in construction hazard recognition: A bibliometric review. *Sensors*, *21*(18). https://doi.org/10.3390/s21186172

Wang, M., & Hu, F. (2021). The Natural Language ToolKit library application for python natural language processing in corpus research. *Theory and Practice in Language Studies*, *11*(9), 1041–1049. https://doi.org/10.17507/tpls.1109.09

Zhang, P., Benbasat, I., Carey, J., Davis, F., Galletta, D. F., & Strong, D. (2002). Human-Computer Interaction Research in the MIS Discipline. *Communications of the Association for Information Systems*, *9*(20), 334–355.