CLOUD BASED INTERACTIVE STUDENT INFORMATION CHATBOT

BY

EGBOBAWAYE JOSEPH

PSC1707453

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCE

UNIVERSITY OF BENIN

BENIN CITY

DECEMBER, 2022

CLOUD BASED INTERACTIVE STUDENT INFORMATION CHATBOT

BY

EGBOBAWAYE JOSEPH

PSC1707453

A PROJECT SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCE

UNIVERSITY OF BENIN

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF

BACHELOR OF SCIENCE (B.Sc.) DEGREE IN

COMPUTER SCIENCE, UNIVERSITY OF BENIN, BENIN CITY

DECEMBER 2022

CERTIFICATION

This is to certify that EGBOBAWAYE JOSEPH, in the Department of Computer Science, Faculty of Physical Sciences, University of Benin, Edo State, with matriculation number PSC1707453 presented this project in partial fulfilment of the requirements for the award of the degree of Bachelor of Science (B.Sc.) in Computer Science, University of Benin.

_____ _____

Mr. E. E. OBASOHAN **Date**

Project Supervisor

APPROVAL

This project is hereby approved in partial fulfilment of the requirements of Bachelor of Science (B.Sc.) degree in Computer Science, University of Benin, Benin City.

_____

Dr. (Mrs.) A.O. Egwali

Head of Department

DEDICATION

This work is dedicated to God Almighty, for His infinite mercy and provision upon my life during the period of this course and always.

ACKNOWLEDGEMENT

My profound gratitude to God Almighty for the provision of good health, strength, knowledge and infinite mercy upon my life. I am grateful for His endless protection, love and guidance, grace and showers of blessings upon me to come thus far. My humble appreciation goes to my project supervisor, Mr. E. E. Obasohan for his guidance, support and advice throughout the duration of writing my project.

I also want to thank especially the Head of Department Prof. Mrs A.O Egwali, my distinguished lecturers who fed me with academic diet throughout my days as an undergraduate; Prof (Mrs) S. Egbokhare, Prof G.O. Ekuobase, Prof (Mrs) VVN Akwukwuma, Prof A.A. Imiavan, Dr. K.C. Ukaoha, Dr. F.O Chete, Engr. Dr. FAU Imouokhome, Dr. Mrs. V.I. Osubor, Dr. F.O. Oliya, Dr. P. Ebietomere, Dr. S.O.P Oliomogbe, Dr. (Mrs.) S. Koyenha, Mr. E. Nwelih, Dr. J.C. Obi, Mr E.C. Igodan, Dr. (Mrs) R.O. Osaseri, Mr. P.I. Imiefoh, Mr. K.O Otokiti, Mr. F. Osagie, Mr. E. Obayagbona, Miss O.L. Usiosefe. You all have diligently placed me on the right path in my career pursuit, may God almighty continually bless you in all your endeavour.

My unreserved appreciation goes to my parents; Mr. Fred Egbobawaye and Mrs. Juliet Egbobawaye; my siblings; Mrs. Omorefe Daniel John, Evang. Stanley, Imuegbe and Enoma. My uncles: Mr, Daniel Osaigbovo and family, Mr. Eddie Osadiaye and family, my aunties for their unending love, support, encouragement and prayers throughout my learning period in this university. Special thanks also to Justin Egi, and Queensly Temilade Elijah for their joint academic support and encouragement during the raining days of studies. God continually bless you both.

Finally, I want to appreciate my class reps; Godson and Tracy, my friends, colleagues as well as every member of B.Sc, Computer Science 2021/2022. You are all awesome. It has been thrilling and fun learning with and from you all. I love you all.

ABSTRACT

Chat bots typically provide a text-based user interface, allowing the user to type commands and receive text as well as text to speech response. Chat bots are usually stateful services, remembering previous commands in order to provide functionality. When chat bot technology is integrated with popular web services it can be utilized securely by an even larger audience. The college enquiry chat bot will be built using artificial algorithms that analyses user's queries and understand user's message.

This project involves using an artificial conversational entity (chatbot) to ask questions about college and receive responses. This System is a web application that answers the student's question. Students simply need to ask questions through the chatbot. Students can chat in any format; the user does not have to follow a particular format. The student can stay informed about college activities thanks to this system.

This proposed chatbot is developed using the Google Dialog flow framework. Dialog flow is a natural language understanding platform that makes it easy to design and integrate a conversational user interface into mobile app, web application, device, bot, and interactive voice response system.

Keywords: Chatbot, chatterbot, pattern matching, keyword matching, cloud-based

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER ONE

INTRODUCTION

Students in the 21st century are tech-savvy, so they expect a school or institution's communication system to be seamless, real-time, and individualized. Despite the limitations imposed by distance and time, social media has emerged as the preferred method of communication for connecting and bringing students closer together. Online chatbots can also benefit from the use of the social media platform. The need for tools that enable real-time and 24-hour responses has grown as a result of the high social media presence of educational establishments and student inquiries. An educational establishment stands to gain significantly from utilizing an online chatbot. However, the use of chatbots in schools is still a relatively new idea, and there is not much written specifically about adoption models or frameworks, particularly in Nigeria.

1.1 BACKGROUND OF THE STUDY

The university receives a large number of applications at the beginning of each new session. In order to convert these applications into enrolments, prompt communication is essential. Courses, fees, scholarships, facilities, the campus, housing, and transportation are just a few of the topics students may be curious about. Students who come to the university from places with different climates, cultures, and time zones will need help settling in. Universities may find it challenging to respond to every student due to the volume of applications and emails they receive.

In addition to returning students, prospective students must be informed of recent developments in the university environment. For instance, looking for information about the library, gym, and canteens, as well as the times of various events.

There are many reasons why a college inquiry system is needed, including. Due to the slow nature of the college website, an outsider wouldn't know where to look for a specific piece of information, making it difficult for someone outside of the college's domain to obtain information (Lalwani, Bhalotia, Pal, Bisen, & Rathod, 2018). Chatbots offer a solution to this issue by providing students with all of the essential information they require to settle in on campus.

A chatbot is a computer program that uses text to communicate with users. These programs frequently aim to pass the Turing test by convincingly simulating how a human conversation partner would act.

In his well-known 1950 article titled "Computing Machinery and Intelligence," Alan Turing proposed what is now known as the Turing test as a criterion for intelligence. (Turing, 1950) This criterion is based on a computer program's ability to impersonate a human in a written conversation with a human judge in real time to such an extent that the judge is unable to reliably distinguish between the program and a real human based solely on the content of the conversation. The popularity of Turing's proposed test sparked a lot of interest in the 1966 publication of Joseph Weizenbaum's program ELIZA, which appeared to be able to fool users into thinking they were conversing with a real person. However, Weizenbaum himself did not assert that ELIZA possessed genuine intelligence, and his paper's introduction presented it more as a debunking exercise (Weizenbaum, 1966)

The primary operation of ELIZA, which has since been copied by developers of chatbots, entails the recognition of cue words or phrases in the input and the output of corresponding pre-prepared or pre-programmed responses that can move the conversation forward in a manner that appears to be meaningful (for instance, by responding to any input that contains the word

"MOTHER" with "TELL ME MORE ABOUT YOUR FAMILY") (Weizenbaum, 1966). Despite the superficial processing involved, an illusion of comprehension is created as a result. Because human judges are so willing to give the benefit of the doubt when conversational responses can be interpreted as "intelligent," ELIZA demonstrated that such an illusion is surprisingly simple to create.

Humans' willingness to interpret computer output as truly conversational, even when it is actually based on rather simple pattern-matching, can be exploited for useful purposes, according to interface designers. As a result of the fact that the majority of people prefer to interact with programs that are akin to humans, chatbot-like techniques have the potential to be useful in interactive systems that require users to provide information—as long as that information is relatively straightforward and falls into predictable categories. As a result, online help systems that use chatbots to identify the kind of assistance that users need may offer a more "friendly" user experience than a more formal search or menu system. The use of chatbots in this way has the potential to move them from Weizenbaum's "shelf... reserved for curios" to the "genuinely useful computational methods" category.

In dialog systems, chat bots are typically utilized for a variety of practical purposes, such as user service and information acquisition. This system, which will respond to users' inquiries. Software application user interfaces can take many different forms, including command-line, graphical, web application, and even voice. Even though graphical and web-based applications are the most widely used user interfaces, there are times when an alternative interface is required. A chat bot-based interface might be what's needed because of multi-threaded complexity, concurrent connectivity, or service execution-related details. Typically, chat bots offer a text-based user interface that enables the user to type commands, receive text, and respond via text to

speech. Most of the time, chat bots are stateful services that remember previous commands (and possibly even conversations) to function. Chat bot technology has the potential to be used safely by an even larger audience when it is integrated with popular web services.

Chatbot recognize the user input as well as by using pattern matching, access information to provide a predefined acknowledgment. It is implemented using pattern comparing, in which the order of the sentence is recognized and a saved response pattern is acclimatize to the exclusive variables of the sentence. They cannot register and respond to complex questions, and are unable to perform compound activities (Dahiya, 2017). Over duration of interactions, chatbots gather enormous data that provides helpful information about common concerns students face to make significant changes to its operation.

1.2 PROBLEM STATEMENT

In the lead up of starting a new academic year, applicants are buzzing with a lot of questions and it’s undoubtedly a busy time for admissions departments responding to an influx of questions requiring fast responses. In past years, students and parents had to visit the college to enquire about details and other information about the college that is a lengthy and time-consuming process. This is also a hectic and resource wasting process for the admissions offices.

New students frequently inquire about the school's various aspects. From how to get their clearance, to which classes to register for, to just getting around campus. Even older students have inquiries about the school. The majority are unsure of where to begin in order to obtain the needed responses. A quick solution to this is to have an information chatbot. To save time, energy and resources this can now be done over the internet with chatbots

1.3 AIM AND OBJECTIVES

The aim of this research work is to develop an interactive student information retrieval chatbot that will respond and give answers to students' questions and inquiries about major aspect of the school and their studies.

The objectives are as follows:

1. To provide students and faculty a quick and easy way to have their questions answered
2. To analyse users queries and understand users message.
3. To provide an answer to the query of the user efficiently and effectively.
4. To develop a system which replies using a responsive GUI similar to a real person talking to the user.
5. To save the time of the user in finding
6. To determine the required features for the construction of the knowledge base

1.4 SIGNIFICANCE OF THE STUDY

Students of all ages can use this study to learn more about their school and department without having to go into an actual office. As a result, offices where information can be obtained are less congested. In addition, this study contributes to the alleviation of stress that students and information officers who provide necessary information experience. Information that is stored in the cloud will also be easy to access, so students can access it from anywhere with an internet connection without feeling constrained or physically exhausted.

1.5 SCOPE OF THE STUDY

The scope of this study is centred on the design and implementation of an information chatbot for the students of the University of Benin. This information chatbot will contain:

- Admission Information
- Course information
- Fees and payments
- Directions and navigation

1.6 DEFINITION OF TERMS

- Cloud: networked computing facilities providing remote data storage and processing services via the internet.
- Chatbot: also known as a smartbots, talkbot, chatterbot, Bot, IM bot, interactive agent, Conversational interface or Artificial Conversational Entity, is a computer program or an artificial intelligence which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test.
- Artificial Intelligence: Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.
- Natural language processing: Natural language processing (NLP) is a subfield of computer science information engineering and artificial intelligence concerned with the interactions between computer and human (natural) languages, in particular how to program computers to process and analyse large amounts of natural language data.
- Dialog flow: Dialogflow is a Google-owned framework that enables users to develop human-computer interaction technologies that can support Natural Language Processing
- Knowledge base: A knowledge base (KB) is a technology used to store complex structured and unstructured information used by a computer system

- CSV: A CSV (comma-separated values) file is a text file that has a specific format which allows data to be saved in a table structured format.
- FAQ: An abbreviation for frequently asked question. It is a document (as on a website) that provides answers to a list of typical questions that users might ask regarding a particular subject

CHAPTER TWO

LITERATURE REVIEW

2.1 CHATBOTS

A chatbot is an artificial intelligence program that is designed or constructed to simulate an intelligent conversional discussion. It can also be defined as human computer interaction model where computer program offers intelligent interaction between users and computer using a well-defined or everyday language. Cloud based student information chatbot system is an artificial algorithm that analyses the student queries and reply messages (Patel, Bhagora, & Singh, 2020)

Chatbots came a long time back in the second half of the 19th century with simple conversations and now turned into powerful tool for e-commerce and media brands. The chatbots are broadly classified into two types.

The first type has numerous predefined rules, conditions, streams, and breakdown conditions that are followed by triggers to respond to specific questions or instructions. The chatbot that provides the understudy with data in a specific montage is one straightforward model. One individual or customer can request any number from precedents, such as the student's name and ID number; the visit bot will then search for the appropriate response and return the result. These kinds of chatbots may be clever if the chatbot's designer is knowledgeable and can structure the chatbot based on each discussion question.

The second type makes use of recent developments like ML (machine learning) and simulated intelligence (man-made reasoning), which understand the meaning of words for all dialects and feelings without relying solely on pre-defined information. A customer might inquire, "What course is CSC499?" Additionally, the engineer's input can be conveyed by the

chatbot, which then uses that information to improve precision over time by comprehending convoluted conversations.

A chatbot's purpose is to lead a conversation that informs clients about data access via a lightweight application. Based on the information and learning methods, chatbots can be divided into two categories, which are as follows:

- Rule-based chatbots: The Standard-based chatbot functions as an intuitive chatbot that responds appropriately based on the design principles. They are also known as the FAQ bot. The software engineer provides these kinds of chatbots with a lot of information through separate responses to both questions. When a customer asks a question, it follows the software engineer's instructions and responds similarly.
- AI chatbots: Artificial Intelligence (AI) chatbots act like a fake human brain. As a modern intellectual, it employs NLP (normal language) handling strategies. It comprehends both solicitations and the setting in demand, as well as the question's purpose and the customer's sentiment. As a result, it learns how the client discussions are structured.

2.2 HISTORY OF CHATBOTS

Joseph Weizenbaum created the first chatbot in 1966. It was called Eliza, and it acted like a psychotherapist and asked important questions of the client. With scripted responses, it presented a deception of comprehension. (Weizenbaum, 1966)

In 1972, Kenneth Colby created the advanced chatbot Repel than Eliza, which could imitate a person with distrustful schizophrenia. Later on, in 1988, a project to create chatbots with artificial intelligence (also known as man-made consciousness) and human collaboration

was launched under the name Jabberwacky. In 1992, Imaginative Labs for MS-DOS develops the chatbot Dr.Sbaitso, which combines AI and is designed to display a digital voice,

```
Human:  Men are all alike.
ELIZA: In what way?
Human:  They're always bugging us about something.
ELIZA: Can you think of a special example?
Human:  Well, my boyfriend made me come here.
ELIZA: Your boyfriend made you come here?
Human:  He says I'm depressed much of the time.
ELIZA: I'm sorry to hear that you are depressed much of the time.
Human:  It's true.  I'm unhappy.
ELIZA: Do you think coming here will help you not to be unhappy?
Human:  I need some help, that much seems certain.
```

Figure 2.1 A Sample Conversation with ELIZA

Counterfeit Phonetic Web PC Element (A.L.I.C.E. chatbot) was created in 1995 using regular language preparation. This chatbot uses heuristic example to coordinate human contribution to request a discussion (Bani & Singh, 2017).

```
User:  Hello ALICE
Alice:  Hi there!
User:  How old are you?
Alice:  I am eighteen in computer years
User:  Who is your father?
Alice:  I was programmed by Dr.Richard S. Wallace
User:  so you are a software program?
ALICE: Interesting deduction.  Thanks and you are a
delightful person too.  Am I a software program?  Let
me put it this way, unknown person:the ALICE series
is the most reliable computer software ever made.  We
are for all practical purposes foolproof and
incapable of error.
User:  Wonderful!  So what's your favourite food ALICE?
Alice:  Yeah that's right!  I require only electricity
```

Figure 2.2 A Sample Conversation with ALICE

In 2001, a clever bot known as smarterchild was developed and typically distributed across SMS systems. Later, in 2006, IBM developed Watson, a program that uses machine learning and normal language handling to extract relevant information from a large amount of data.

Siri, an intelligent personal assistant chatbot that uses a common language user interface to answer questions and fulfil other requests, was released in 2010 and served as the foundation for all subsequent chatbots.

In 2012, Google launched Google Now, an application for mobile phones. (Sayed & Jain, 2016) Currently, it is the component of updates and UI changes for portable use. Later in 2015, bots gained popularity with new features like voice collaboration using language preparation calculations in the Alexa bot and Cortana bot, which can understand common voice commands and is available in a variety of languages. These bots also search the internet. Facebook launched bots for the flag-bearer stage in 2016, allowing developers to create chatbots and collaborate with Facebook customers. Customers can use a variety of chatbots, such as NIKA (for timesheets), ACEBOT (for costs), TWYLA (for client administration), QnA (for FAQ), and WIZU (for inputs), which are all available to them. (Lakshmi, Reddy, Kireeti, T.Swathi, & Ismail, 2019)

2.3 RELATED WORKS

(Griol & Molina, 2014) created a multimodal conversational agent for an improved e-learning experience that teaches children to value and protect their surroundings. A modular and scalable framework that makes it easier to create pedagogical conversational agents that can communicate with students through speech and natural language was used to accomplish this.

The Voice Extensible Markup Language was used as the standard in this work to implement interactive voice dialog for speech communication between humans and computers. As a result, a web-based interactive software with a friendly chatbot that children can use to learn about the urban environment has been created. Twelve preliminary results, on the other hand, were not confirmed.

(Lin, 2016) created a web-based platform for collecting interactions between humans and chatbots. The server is written in Python and uses the tornado framework. The HTTP protocol in JSON format was used primarily to communicate with the chatbot. However, the developed system did not provide adequate facilitation for the data annotation and evaluation.

(Sayed S. , Jain, Lokhandwala, & Barodawala, 2016) developed a hotel reservation android chatbot system using PHP, AIML, Java, and SQLite as implementation technologies. The chatbot uses AI calculation to determine whether the user's input is inappropriate, insufficient, complete, or conversational. The user is notified to enter the missing parameter if the input is insufficient, and if the input is inappropriate, the user is notified to enter the incorrect input. The chatbot engages in casual conversation with the user if the input is conversational. The user will receive a precise output if the input is complete. Due to the fact that SQLite, not MySQL, is used to store the database, the developed system does not support multi-user environments.

(Doshi, Pawar, & Shelar, 2017) proposed and developed an android-based artificial intelligence chatbot that interacts with the user through text and voice response using open source program-o. This system made use of the PHP-based open source AIML engine, which is an interpreter for the chatbot's AIML script and stores the details of the bot in a MySQL database. Additionally, it uses Android's speech recognizer API from Google to convert voice into text. The outcome demonstrates that an Android application was capable of effective voice interaction.

The chatbot is accessible only to Android users and cannot be accessed via the web; however, this application can be used to communicate with the bot via voice and text.

A college inquiry chatbot developed by (Salve, Patil, Pratik, Vishruta, & Vyankatesh, 2017) accepts text-based user input. This system uses a predefined set of rules for the system's training and made use of HTML, CSS, AIML, and SQLite for the bot's implementation after taking contributions from the client and preparing into a book recognition. The outcome demonstrates that, with the assistance of their system, the total amount of time required to complete all tasks, including a visit to the college, waiting in line, and inquiry, is reduced. However, because SQLite is used as the database management system, the developed system is unable to handle a large amount of data.

(Hussain & Sianaki, 2019) utilized machine learning to develop a chatbot inquiry system. The bot's script is written in Node.js, and the application programming interface (API) and virtual spirits are used to design it. A web hook connects the bot to a Facebook page. The API.ai virtual cloud storage server houses the bot's learning data. The system utilized integrated artificial intelligence to respond to the api.ai-provided question, and the bot was integrated into Facebook for the end user. However, the implemented system is fully trained with the system dataset but only works with the Facebook system, making it difficult for users who do not already have a Facebook account to use it.

(Lakshmi, Reddy, Kireeti, T.Swathi, & Ismail, 2019) utilized AIML to implement and design a chatbot for students. The system's voice and text inputs are suitable for answering queries thanks to the implementation of Latent Semantic Analysis (LSA) in the Python programming language. By employing natural language handling techniques and acting like a modern intellectual, the pattern matching makes the system behave like an artificial human brain.

This system is accurate up to 90% of the time and helps the user get the right answer in a shorter amount of time. However, the data it collects isn't well analysed, and it only applies to current college students and doesn't help prospective college candidates.

(Barletta, Caivano, & Nannavecchia, 2019) developed a software tool that allows students to freely upload questions. This tool can be used by any college. The chatbot uses AIML as a background to knowledge for processing the response, the system retrieves information from the AIML file database, and finally, it developed a software tool that is used by colleges to assist students in freely uploading their databases. However, the system does not make use of a standard database management system. Instead, it uses an artificial intelligence algorithm.

A human-chatbot interaction system was created by (Nie, 2020). The NLP-based tool was implemented using Python and SQL, and it takes the user-typed message as its input and applies data pre-processing to it. It resulted in an Android application that prevents users from sending participant messages that are inappropriate or inappropriate. The developed application, on the other hand, can only be downloaded from the Google Play Store and is only accessible to Android users.

(Hanisha, Hemanth, & Sirisha, 2021) developed a robust system-based intelligent conversational agent. It was put into action with Dialogflow technology, and the system's process was broken down into phases like data collection, data manipulation, data augmentation, and response generation. The system was implemented with Dialogflow, which, in contrast to having multiple webhooks, can only provide a webhook to an application

2.4 WEB APPLICATION TECHNOLOGIES

A number of technologies and libraries were investigated for the purpose of development of this project and the following were identified as the ones to be used for creating the system.

2.4.1 JavaScript

JavaScript is a scripting, client-side, programming language. It can be used to provide encase the functionality of HTML pages. It can be used for various purposes such as setting character limitations on text areas.

2.4.2 HTML and CSS

HTML is the standard mark-up language used for the development of web interfaces. It must be used in combination with CSS, which is responsible for the layout of the HTML components.

2.4.3 Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase

2.4.4 Bloc

Bloc is a state management library created and maintained by Felix Angelo. It helps developers implement the Bloc design pattern in their Flutter application. It means that a developer must know the state of an app at any time.

2.4.5 Dialogflow

Dialogflow is a Google-owned framework that enables users to develop human-computer interaction technologies that can support Natural Language Processing. Dialogflow is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems and related uses

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 METHODOLOGY

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release.

The series of releases is referred to as “increments”, with each increment providing more functionality to the customers. After the first increment, a core product is delivered, which can already be used by the customer. Based on customer feedback, a plan is developed for the next increments, and modifications are made accordingly. This process continues, with increments being delivered until the complete product is delivered.

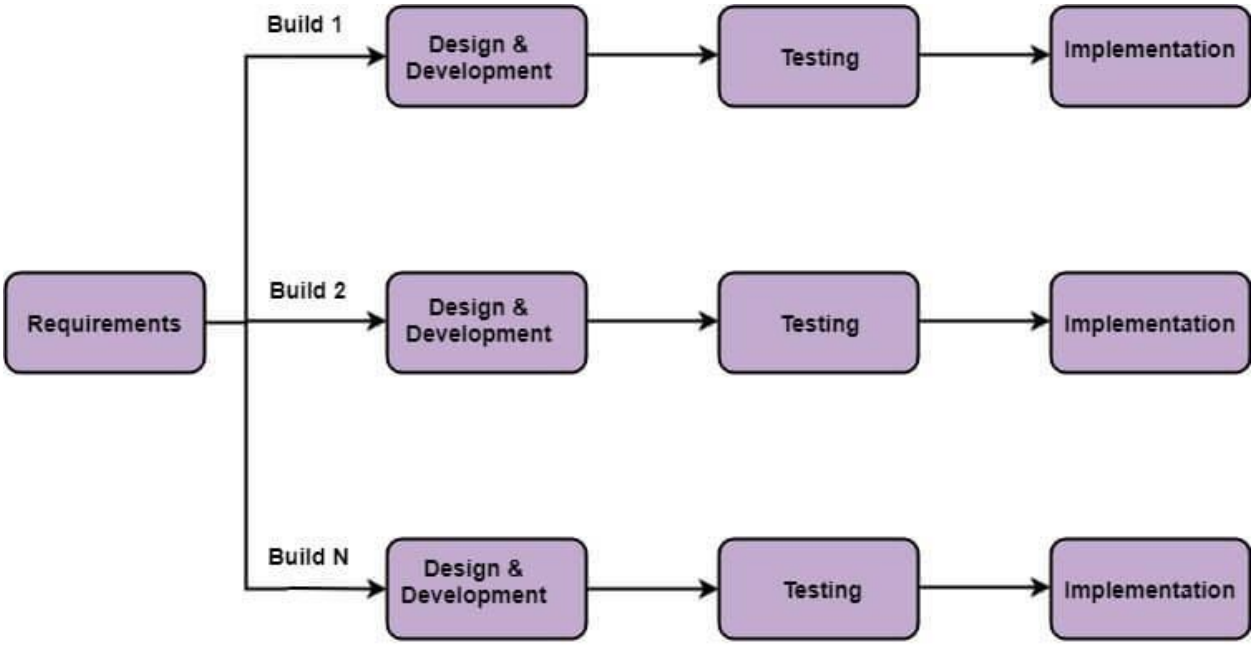


Figure 3.1 Incremental Model

3.2 SYSTEM ANALYSIS

In object-oriented analysis, the system requirements are determined, the classes are identified and the relationships among classes are identified.

3.3 REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

There are two main different types of system requirements that should be gathered by those working on software projects.

3.3.1 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product.

The functional requirements of this project are:

i. The system should allow users to chat.
ii. The system shall inform the user if an answer is not available.
iii. The system should identify the intent of a question to provide an accurate answer
iv. The chatbot should capture, read and process large amounts of data to gain insights from relevant data and to quickly solve customer problems.
v. The administrator should be able to add, update and delete questions, answers and keywords

3.3.2 Non-functional Requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioural requirements.

The non-functional requirements of this project are:

i. The system should be easy to understand

ii. The system shall maintain an easy to use interface across all functionality and for all users

The chatbot should be launched quickly.

iii. The system should run smoothly with adequate execution speed, operating at maximum performance

iv. The administrative system should be protected from unauthorized access.

v. The database should protected from attacks and unauthorized access.

vi. The interface should be protected from attacks.

vii. The system should run on a variety of operating systems that support the Java language.

viii. The system should run on a variety of hardware.

ix. Exceptions should be reported effectively to the user if they occur

3.4 SYSTEM DESIGN

A Chatbot refers to a chatting robot. It is a communication simulating computer program. It is all about the conversation with the user. The conversation with a Chatbot is very simple. It answers to the questions asked by the user. During designing a Chatbot, how does the Chatbot communicate to the user? And how will be the conversation with the user and the Chatbot is very important. The design of a Chatbot is represented using diagram:

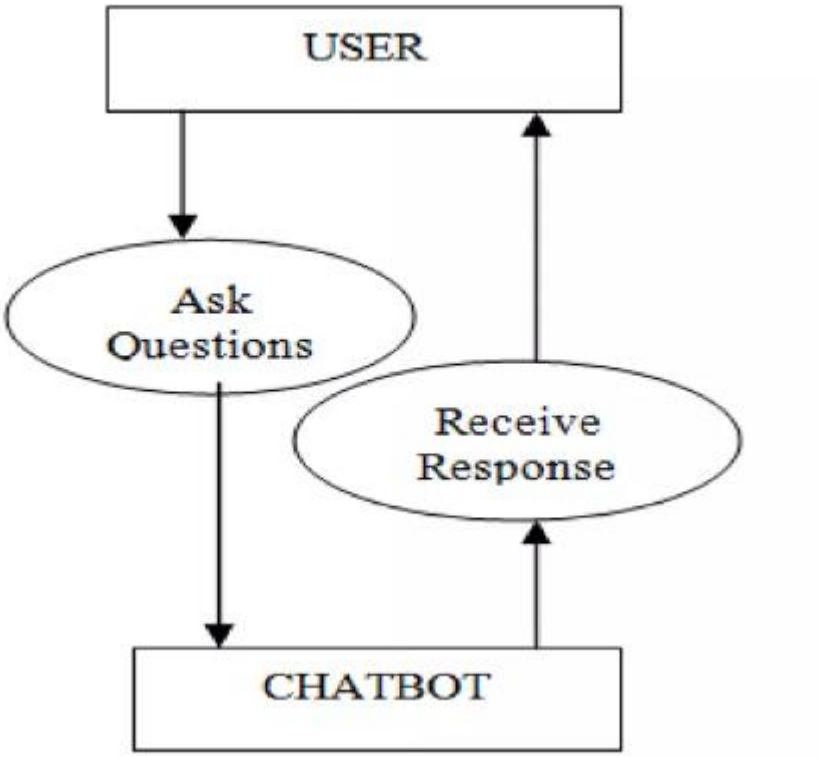


Figure 3.2 Use Case Diagram of Chatbot

Computers play a significant role in our society today. Information is provided by computers; they keep us occupied and assist us in numerous ways. A program designed to imitate intelligent text or spoken communication is known as a chatbot. However, the text-only chatbot serves is the basis for this paper. Chatbots use pattern matching to identify user input and access information to provide predefined acknowledgments. For instance, if the user asks the bot something like, "What is your name?" The chatbot will probably respond with something along the lines of "My name is Chatbot." or "You can call me Chatbot" is how the chatbot responds based on the user-provided sentence. The user receives a response based on a pattern that has already been defined when the input is entered into the database. Pattern comparing is used to

implement a chatbot, which recognizes the sentence's order and adapts a saved response pattern to the sentence's individual variables. They are unable to perform compound activities, register, or respond to complex questions.

3.5 SYSTEM MODEL

The model design is divided into two modules: Online Enquiry and Online Chatbot. The Online Enquiry contain queries related to the College and Faculty. The Online Chatbot provides responses to query regarding Returning Students, Freshers, and Applicants. Additionally, the chatbot can be questioned by users who wish to inquire about the college at the time of admission or about any aspect of the institution.

The chatbot system's process flow diagram can be seen in Figure 3.3. The Natural Language Processing (NLP) technique is applied when a query is entered, and if the question falls within the scope, the system generates a response by connecting the application and cloud server.

Additionally, the database will be checked for queries; if the query is valid, the response will be displayed. The user will be required to enter a different query if the current one is in.

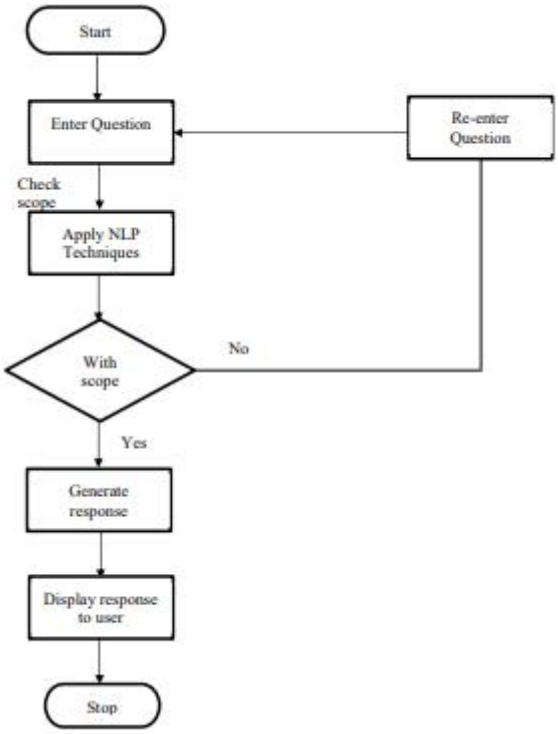


Figure 3.3 Process Flow Diagram

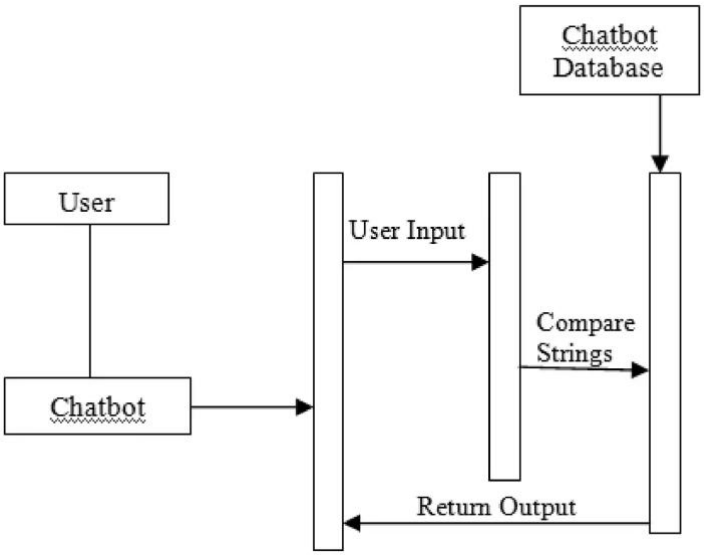


Figure 3.4 Sequence Diagram

3.6 KNOWLEDGE BASE

A knowledge base, also known as a help centre or resource centre, is a hub or useful resource meant to guide and help users in engaging with a product. Chatbots work around the framework of a Knowledge Base (KB). The success or failure in efficiently handling customer queries is heavily linked with how the KB is built and the resources available to it (Singh, 2022). Integrating the chatbots with a sturdy knowledge base for customer service is crucial to improving your overall customer experience and boosting customer satisfaction scores.

Data were sources and acquired from the department of student affairs, the school website, the school portal and prospectus of various faculties. This is to provide information required in the pre-processing stage as a preparatory level for the preparation of the chatbot to be developed. The nature of data collected are all student related data.

Figures show examples of the data that were collected.

Introduction to Computing	Introduction to Computing, CSC111, csc 111, CSC 111, csc111
Programming Essentials	Programming Essentials, CSC111, CSC 111, csc111, csc 111
Introduction to software packages	Introduction to software packages, CSC120, csc120, csc 120, CSC 120
Software Programming in Pascal	Software Programming in Pascal, csc211, csc 211, CSC211, CSC 211
Symbolic Programming in Fortran	Symbolic Programming in Fortran, CSC212, csc212, CSC 212, csc 212
Information Technology	Information Technology, csc217, csc 217, CSC 217, CSC217
Information Interfaces Technology	Information Interfaces Technology, csc237, csc 237, CSC237, CSC 237
Introduction to Data Processing	Introduction to Data Processing, csc220, csc 220, CSC220, CSC 220
Assembly Language II or C programming	Assembly Language II or C programming, csc312, csc 312, CSC312, CSC 312
Data Structures	Data Structures, csc313, csc 313, CSC313, CSC 313
Operations Research	Operations Research, csc314, csc 314, CSC314, CSC 314

Figure 3.5 Knowledge base on course information

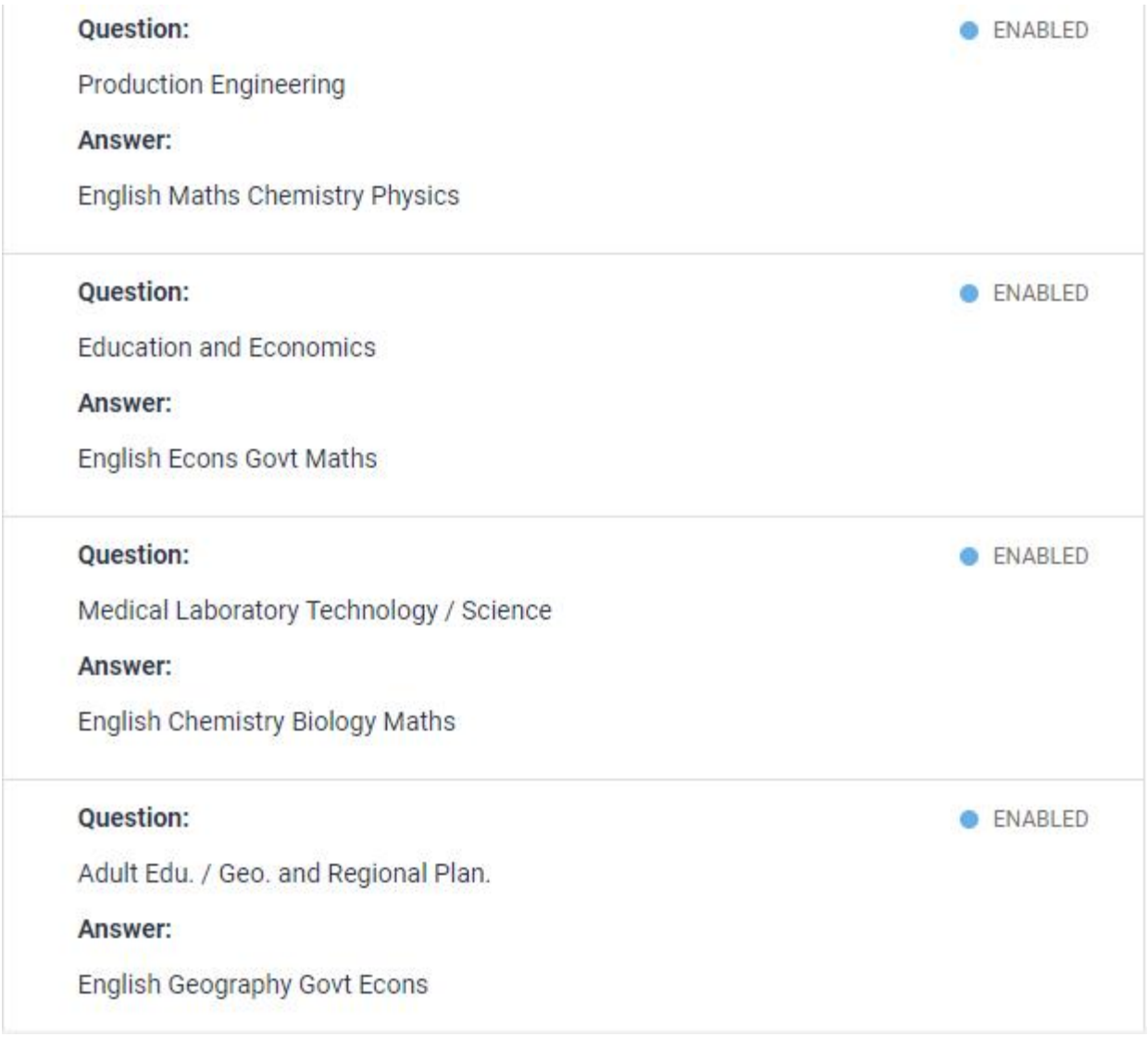


Figure 3.6 Knowledge base on jamb subject combination for different courses of study

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND TESTING

4.1 ALGORITHM OF THE CHATBOT SYSTEM

a) *Step 1*: Start

b) *Step 2*: Read the user query

c) *Step 3:* Pre-processing of the query

d) *Step 4*: Fetch the remaining keywords from the query

e) *Step 5:* Match the fetched keywords with the keywords in the knowledge base, and provide an appropriate response. The keywords will be matched using keyword matching algorithm

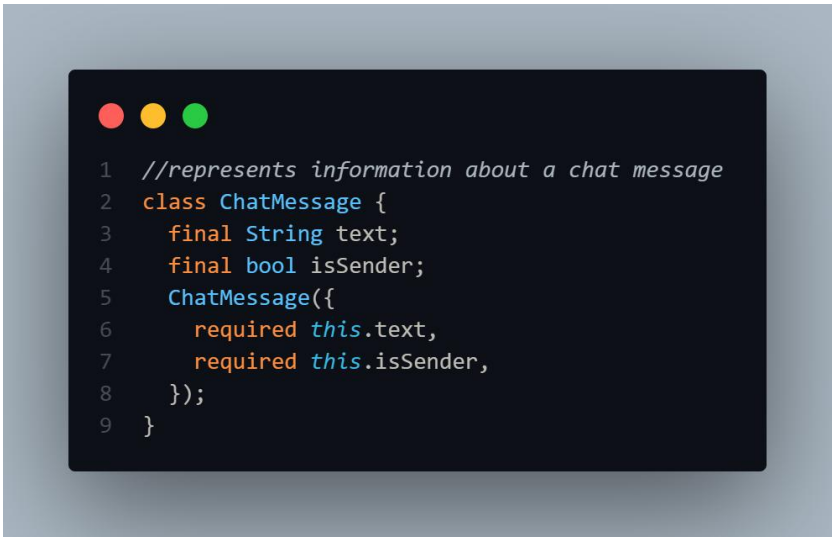
f) *Step 6:* Return the query response and output to the user

g) *Step 7*: Stop

4.2 DATA LAYER IMPLEMENTATION

4.2.1 Data Model

Our Data Model will specify distinct relationships between the data. Screens can retrieve data as needed from this in-memory database. Our data models for this app are: Chat model and Service model. The code below represents the data models used:



```dart
//represents information about a chat message
class ChatMessage {
  final String text;
  final bool isSender;
  ChatMessage({
    required this.text,
    required this.isSender,
  });
}
```

Figure 4.1 Chat Message Data Model

```dart
// represents information on the services provided by the app
class ServiceModel {
  String iconPath;
  String title;
  String description;
  ServiceModel({
    required this.iconPath,
    required this.title,
    required this.description,
  });
}
```

Figure 4.2 Service Data Model

4.2.2 Data Source

The data source of the system is made up of one classes: DialogflowService. This class is responsible for interacting with all external source, in this case, Dialogflow. It contains one method that sends the user query to Dialogflow and fetches the response and converts to our data object. The code snippet can be found in the image below:

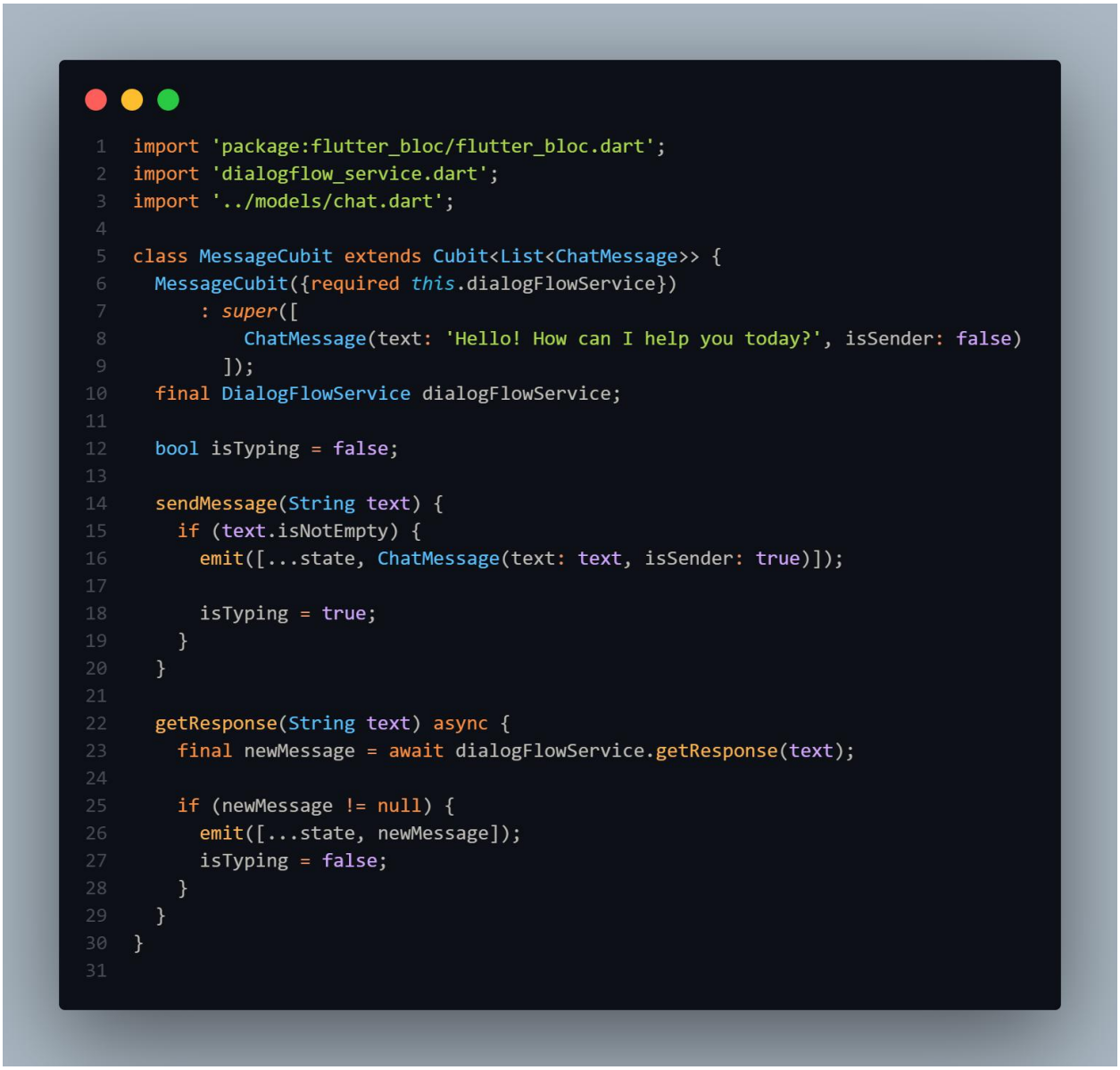
```dart
// ignore: import_of_legacy_library_into_null_safe
import 'package:flutter_dialogflow/dialogflow_v2.dart';
import 'package:student_information_chatbot/models/chat.dart';

final str = {'hi': 'our'};

class DialogFlowService {
  Future<ChatMessage?> getResponse(String query) async {
    AuthGoogle authGoogle =
        await AuthGoogle(fileJson: 'credentials.json').build();

    Dialogflow dialogflow =
        Dialogflow(authGoogle: authGoogle, language: Language.english);

    AIResponse response = await dialogflow.detectIntent(query);

    ChatMessage message = ChatMessage(
      text: response.getMessage() ??
          CardDialogflow(response.getListMessage()[0]).title,
      isSender: false,
    );
    return message;
  }
}
```

Figure 4.3 Data source code snippet

4.3 BUSINESS LOGIC LAYER IMPLEMENTATION

The business logic layer bridges the gap between the Data Access Layer and the Presentation Layer. It oversees the database's and presentation layer's communication. The business logic layer of this system is made up of one class, MessageCubit, which uses the BLOC statement management library. Every functional part of the system is implemented using an appropriate method. Each method is called from the presentation layer when this is necessary. Spring security has also been used to ensure that only authorized methods are accessible to users.

We implemented two methods: one for handling the sending of queries and the second to handle getting the response from Dialogflow. The image below shows the message cubit class.



```dart
import 'package:flutter_bloc/flutter_bloc.dart';
import 'dialogflow_service.dart';
import '../models/chat.dart';

class MessageCubit extends Cubit<List<ChatMessage>> {
  MessageCubit({required this.dialogFlowService})
      : super([
          ChatMessage(text: 'Hello! How can I help you today?', isSender: false)
        ]);
  final DialogFlowService dialogFlowService;

  bool isTyping = false;

  sendMessage(String text) {
    if (text.isNotEmpty) {
      emit([...state, ChatMessage(text: text, isSender: true)]);

      isTyping = true;
    }
  }

  getResponse(String text) async {
    final newMessage = await dialogFlowService.getResponse(text);

    if (newMessage != null) {
      emit([...state, newMessage]);
      isTyping = false;
    }
  }
}

```

Figure 4.4 Message cubit class

4.4 PRESENTATION LAYER IMPLEMENTATION

Dart and the Flutter framework were used to create the presentation layer, which aims to clearly distinguish the template from the business logic. The appropriate method is called from the message cubit whenever an action is required. The dialogflow service is created after a bloc

provider is declared in the MyApp widget in the manner depicted below. We can now call any of the authorized methods after calling the message cubit in the presentation thanks to this.

```
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return BlocProvider(
      create: (context) => MessageCubit(dialogFlowService: DialogFlowService()),
      child: MaterialApp(
        debugShowCheckedModeBanner: false,
        theme: theme,
        title: 'Joe Chatbot',
        routes: {
          '/': (context) => const LandingPage(),
          '/chat': (context) => const ChatPage()
        },
        initialRoute: '/',
      ),
    );
  }
}
```

Figure 4.5 MyApp widget showing the instantiation of the dialogflow service

4.5 USER INTERFACE

The final system includes a web interface with two screens: the landing page and the chat page. The landing page introduces the system, what information the system can provide and how to use it. This is depicted in the image below:

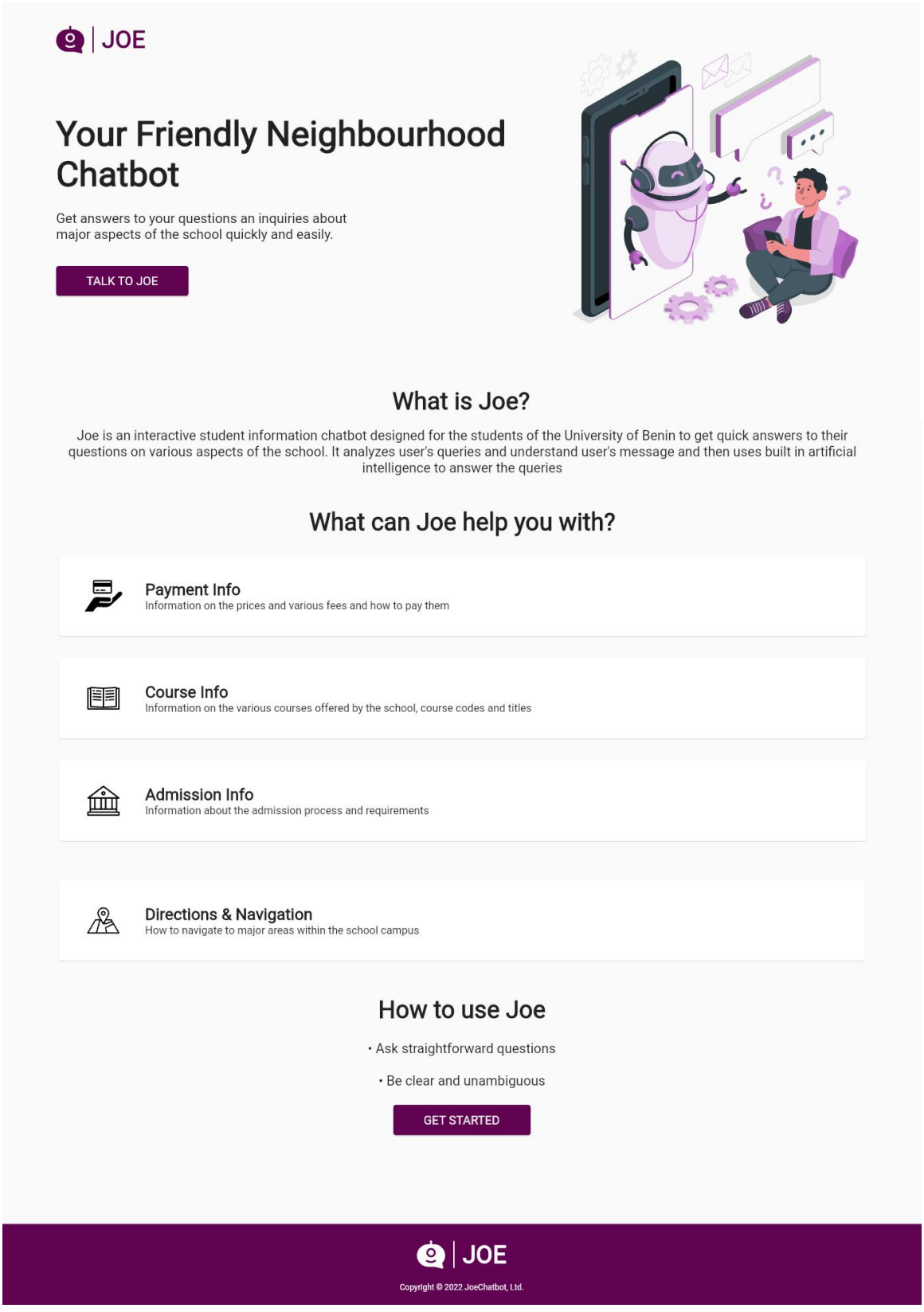


Figure 4.6 Joe landing page

The chat page is where the user can engage in conversation with the chatbot as shown below:

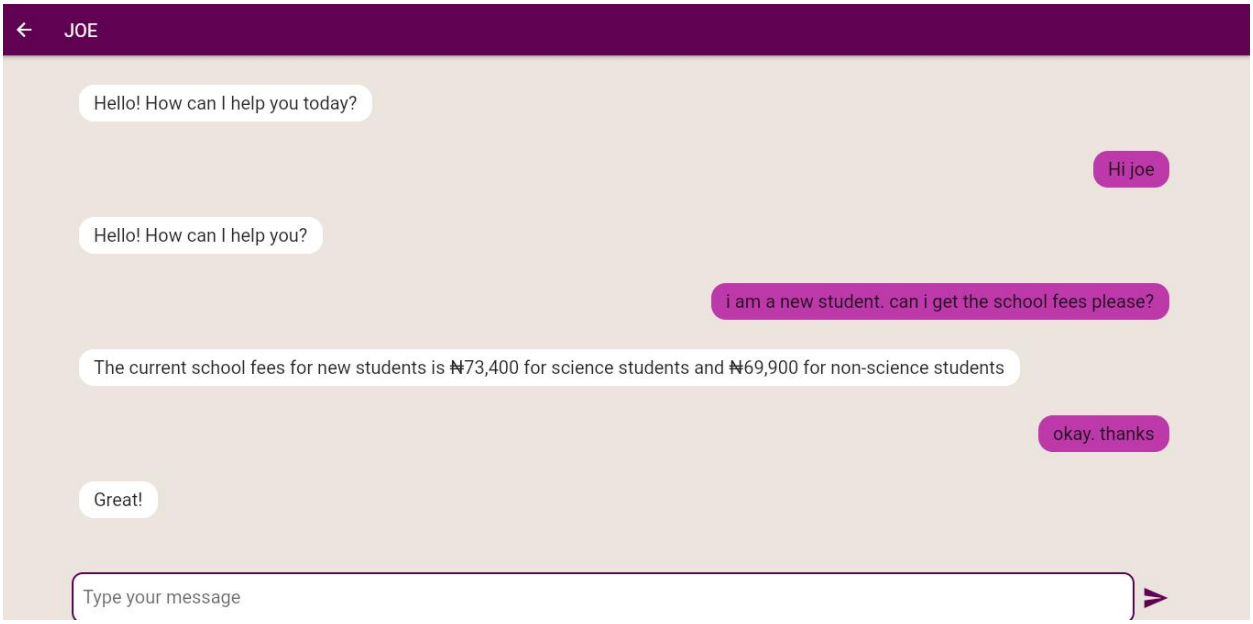


Figure 4.7 Joe chat page

CHAPTER FIVE

CONCLUSIONS AND RECONMENDATION

5.1 CONCLUSION

A chatbot is a growing trend that improves customer service at a lower cost and boosts business efficiency. The chatbot information model was developed and implemented in this study using the dart programming language, Dialogflow for the back end, and Flutter for the client-server side. For the purpose of developing the chatbot system, samples of information regarding admission, payment, and course information were gathered from the admission office, student affairs unit, school website, school web portal, and prospectus of various faculties. The developed system offers a method of communication that is as effective and efficient as communicating with friends and loved ones. In addition to information on admission, fees, and courses at the University of Benin, this study provides information on admission frequently asked questions (FAQ).Students can get answers to their questions by using the chat bot. These web-based systems allow students to ask questions at any time.

5.2 RECOMMENDATION

This work reveals that an interactive student chatbot which is information readily available is an important and needed development, thus, the following recommendations are given for implementation and deployment.

1. As a necessary development to ease stress on students, strain on staff and cut down continual expenses, colleges and tertiary institution as the University of Benin should

embrace and opt-in to develop a robust cloud based interactive chatbot to easily assist student on their journey to getting started with the school.

2. A chatbot is just as powerful as its scope. In developing a robust chatbot to handle student inquiries, colleges and universities should spend ample time in sourcing and collating required data and information to ensure that the basic and extended scope of students FAQ is covered in the scope of the chatbot.
3. A robust chatbot might be expensive to develop at the point but over time it rules out the financial implication that would have be incurred by conventional information relay process. Even more, the University of Benin can decide to monetize the chatbot by displaying targeted educational advertisement. This in turn can provides the institution with more internally generated revenue.

REFERENCES

Bani, B. S., & Singh, A. P. (2017). College Enquiry Chatbot Using A.L.I.C.E . *International Journal of New Technology and Research (IJNTR)*.

Barletta, V. S., Caivano, D., & Nannavecchia, A. a. (2019). . A Spell-Checking Web Service API for Smart City Communication Platforms. *Open Journal of Applied Sciences, 9*(12), 8-19.

Dahiya, M. (2017). A Tool of Conversation: Chatbot. *International Journal of Computer Sciences and Engineering, 5*, 158-161.

Doshi, S. V., Pawar, S. B., & Shelar, A. G. (2017). Artificial intelligence Chatbot in Android system using open source program-O. *International Journal of Advanced Research in Computer and Communication Engineering, 6*(4), 816-821.

Griol, D., & Molina, J. M. (2014). Developing multimodal conversational agents for an enhanced e-learning experience. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 3*(1), 13-26.

Hanisha, R. S., Hemanth, P. S., & Sirisha, N. R. (2021). Smart-Bot: An Innovative System Simplifying the Hectic Tasks in Cyclic Activities. *International Conference on Inventive Computation Technologies (ICICT)*, 551-556.

Hussain, S., & Sianaki, O. A. (2019). A survey on conversational agents/chatbots classification and design techniques. *International Conference on Advanced Information Networking and Applications*, 946-956.

Lakshmi, K., Reddy, Y. K., Kireeti, M., T.Swathi, & Ismail, M. (2019). Design and Implementation of Student Chat Botusing AIML and LSA. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(6), 1742-1746.

Lalwani, T., Bhalotia, S., Pal, A., Bisen, S., & Rathod, V. (2018). Implementation of a Chatbot System using AI and NLP. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 6*(3), 26-30.

Lin, L. D. (2016). A web-based platform for collection of human-chatbot interactions. *International Conference on Human Agent Interaction*, 363-366.

Nie, T. (2020). *Hand Gesture Recognition for Human robot Interaction.* Malaysia.

Patel, R., Bhagora, N., & Singh, P. a. (2020). Cloud Based Student Information Chatbot. *International Research Journal of Modernization in Engineering Technology and Science, 2*(4), 928-931.

Salve, P., Patil, V., Pratik, S., Vishruta, P., & Vyankatesh, G. a. (2017). College Enquiry Chat Bot. *International Journal on Recent and Innovation Trends in Computing and Communication, 5*(3), 463-466.

Sayed, S., & Jain, R. (2016). Android based Chat-Bot. *International Journal of Computer Applications*.

Sayed, S., Jain, R., Lokhandwala, B., & Barodawala, F. a. (2016). Android based Chat-Bot. *International Journal of Computer Applications, 137*(10), 29-32.

Singh, P. (2022). *How To Create A Knowledge Base For Your Chatbot.* IT Chronicles.

Turing, A. (1950). COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, 433-460.

Weizenbaum, J. (1966). ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM* , 36-45.