NIPES

Journal of Science and Technology Research

Journal homepage: www.nipesjournals.org.ng

A Conversational Artificial Intelligence Chatbot to Deliver Telehealth Information on Covid-19

Fidelis Odinma Chete and Princess Oviawe

Department of Computer Science, University of Benin, Benin City

Article Info

***Keywords*:**
Telemedicine, Covid-19, Coronavirus, Artificial Intelligence (AI), Bot

Received 22 March 2022
Revised 08 August 2022
Accepted 10 August 2022
Available online 2 Sept 2022

Crossref NIPES JOURNALS

https://doi.org/10.37933/nipes/4.3.2022.12

https://nipesjournals.org.ng

Abstract

In recent times, vaccines and medical aid to solving the Corona Virus pandemic or COVID-19 have seen significant success, but the dissemination of correct information and timely response is still lacking especially for special communities such as universities and secondary schools as well as remote and rural areas. This problem could be solved if members of such communities could talk to a chatbot as would health personnel and the chatbot in return give a correct response and even provide additional information. The study proposed a conversational Artificial Intelligence (AI)-based application or bot called "Covid bot" on Microsoft Bot Composer for delivering telehealth information to a student specific demographic.

1. Introduction

A "bot" (a short form of "robot") is any automated software that responds to incoming information or data, using pre-determined rules and/or artificial intelligence to select a response [1]. Chatbots refers to a software application that engages in a conversation with a human being, using natural language and artificial intelligence; and since some of the chatbots are built using a text interface to a VoiceXML interpreter, a functionality that's often available for testing dialog designs [2], the word "chatbot" is most often used with applications that dialogue via text. Bots are automated hardware or software machines that are powered by advances in Artificial Intelligence (AI) technologies [3]. The performance of AI tasks such as Automatic Speech Recognition ASR), Natural Language Understanding (NLU), Text to Speech Synthesis (TTS), amongst others, has improved due to recent developments in machine learning algorithms, such as deep learning and deep reinforcement learning [3], thus resulting in the acceleration of humankind's journey towards the technological singularity [3]. In addition, from Facebook Messenger to Google Assistant, and email bots, these new conversational applications are revolutionizing the way people interact with software [4]. Medical personnel daily spend their day scheduling appointments and answering routine questions of patients. Such repetitive jobs can be easily done with a chatbot application. Additionally, patient feedback assessments are also important for collecting user responses to maintain good patient flow. In 2019 to 2020, a global novel pandemic hit the world with Corona Virus also known as COVID-19, this pandemic outbreak started in Wuhan a town in China to the rest of the world, and is still raging on to date. During the first hit of the virus, several countries experienced what was called a

"Total lockdown". This lockdown resulted in everyone staying indoors to reduce the spread of the virus, during this period access to timely health information and response was lacking. Even with the increased number of information posted on social networks such as Twitter and Facebook, there posed a challenge of filtering the right information and getting answers to specific queries on the symptoms and spread of the Coronavirus. The issue of language and accessibility barrier also posed a challenge as most information online was written in English excluding other languages and was mostly accessible through text (i.e. written blog posts) this information did not make room for text-to-speech technology to accommodate persons living with disability, this thereby marginalized a section of the society and prevented them from accessing such information easily.

This study developed a chatbot solution for delivering quick telehealth information for COVID-19 queries. The chatbot developed aims to solve the issues of timely response to queries on coronavirus, give access to information as well as accommodate users with disability by providing accessible technology like text-to-speech. The objective of this study is to develop an interface for students and other individuals alike to get answers to queries by interacting with a chatbot and getting quick responses. The proposed system is, however, in the form of a prototype.

2. Related work

Cleverbot, a chatterbot web application, described by [5], uses core natural language processing and fuzzy logic in Artificial Intelligence (AI) to strike conversations with humans. Blender, so named for its ability to merge multiple conversational skills at once, was built from upon 9.4 billion parameters and trained using 1.5 billion examples of conversation, making it so large that it had to be broken up into pieces in order to handle larger sets of data [6]. Dialo GPT project, introduced by [7], provided a foundation for building versatile open-domain chatbots that could deliver engaging and natural conversational responses across a variety of conversational topics, tasks, and information requests, without resorting to heavy hand-crafting [6]. XiaoIce, a chatbot, described by [8], is the most popular social chatbot in the world, and is uniquely designed as an AI companion by forming an emotional connection to satisfy the human need for communication, affection, and social belonging. The chat experience is based on Markov Decision Processes (MDPs) which optimize XiaoIce for long-term user engagement [6]. Mitsuku, described by [9] as a web-based human-like chatbot, based on the Artificial Intelligence Markup Language (AIML) technology and viewed as a "virtual friend", can answer questions, play games, and do tricks at the user's request, and is also capable of basic reasoning [6]. Melody, based on advanced deep learning and Natural Language Processing (NLP) technologies is a conversational bot that can give highly-customized and situation-appropriate responses to a patient's query [6]. Meena, fancifully referred to by Google as the "neural conversational model", is a chatbot that learns to respond sensibly to a given conversational context. Its main aim is to address the critical flaw in chatbots of them not making sense [6].

3. Methodology

This research used a bot-building framework that would help in building a responsive chatbot. The framework used is Microsoft Bot Framework. The web and mobile design were hosted using Azure cloud computing services. The developed chatbot simulated a human conversation to provide Computer Science students with quick answers to COVID-19 complaints, utilizing both open-ended and close-ended questions to deliver a more personalized experience. The students were able to ask questions related to Corona Virus and its current spread in natural language which they were comfortable with, such as: getting the latest information on COVID-19, identifying student symptoms, asking open-ended questions, and referring students to a COVID-19 healthcare center. The chatbot then analyzed the user's questions and generated appropriate responses based on the conversational context.

4. System analysis and design

4.1 Proposed System

The proposed system, a chatbot, simulates human conversation and assist students with their questions/queries. Students would be able to ask questions related to COVID-19 and its measures in a natural language they are comfortable with. The chatbot will identify and understand what the users' queries and generate appropriate responses. In order to make the application intuitive/friendly, it was built using a Framework with both coding and low-code capabilities, the Microsoft Bot Framework Composer. Fig 1 shows the sequence diagram of conversational chatbot.

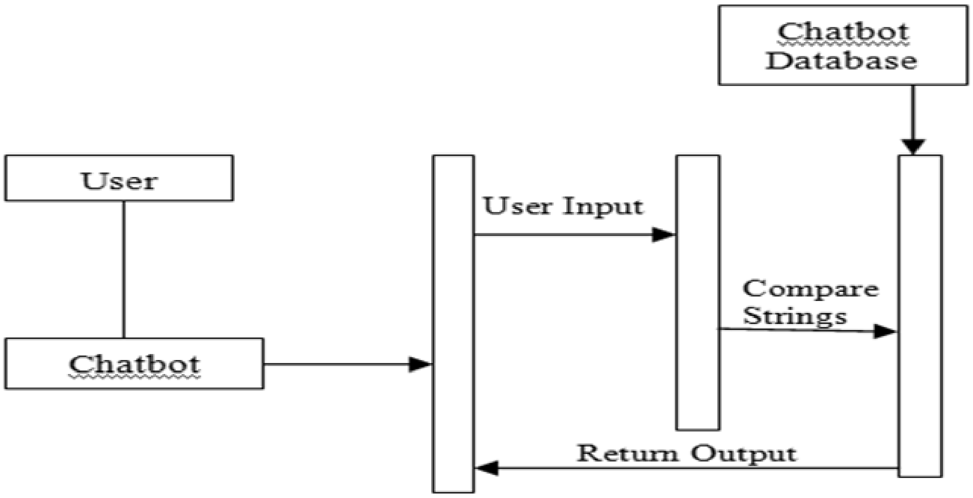


Fig 1 Sequence diagram for conversational chatbot

4.2 Messaging Platform Used for the Design

The selected messaging platform for "Covid bot" is any web browser. The Covid bot was built to be hosted on the cloud using Azure cloud computing services. This medium was chosen because it is easily accessible by every smartphone, accommodate user traffic, and scale efficiently while keeping information exchange secure.

4.3 Conversational Design

The application used Microsoft Bot Composer to create a conversational chat system to hold a conversation with users' by understanding a natural language. The Microsoft tools and services available to create a bot are listed as follows:

Microsoft Azure: As the Microsoft cloud computing ecosystem, Azure contains a wide array of services. It is the company's enterprise platform that is similar to Google Cloud (https://azure.microsoft.com/en-us/overview/what-is-azure/). It is an online portal that allows users access to access and manage cloud services and resources (which include: storing and transforming data) provided by Microsoft (https://www.simplilearn.com/tutorials/azure-tutorial/what-is-azure)

Bot Framework: This is the actual chatbot development framework and is a matrix of tools that connects the bots built to other platforms and services (https://roi4cio.com/catalog/de/product/microsoft-bot-framework). It is a comprehensive framework for building enterprise-grade conversational AI experiences (http://.botframework.com)

Bot Builder: This is the primary toolkit used to define the attributes of the bot. It is an open source product and includes Software Development Kits (SDKs) for different programming languages, debugging and visualization tools, Command Line Interface (CLI) tools, and web chat. The Bot Builder SDK enables users to build bots that support different types of interactions (http://github.microsoft. com).

Azure Bot Service: A component of Microsoft Azure, this service provides a comprehensive environment for building the bot, in addition to testing, hosting, and deployment around the web. It is a managed bot development service that helps users seamlessly via popular channels (http:// azure.microsoft.com).

Azure Cognitive Services: Also, part of Microsoft Azure cognitive services, this service contains tools that make a bot "smart". The Azure AI engine is what enables a bot to have natural language understanding (NLU) and machine learning capabilities. They are cloud-based artificial intelligence (AI) services that help users build cognitive intelligence into their applications (http:// azure.microsoft.com. Fig 2 shows the architecture of the bot framework on Azure.

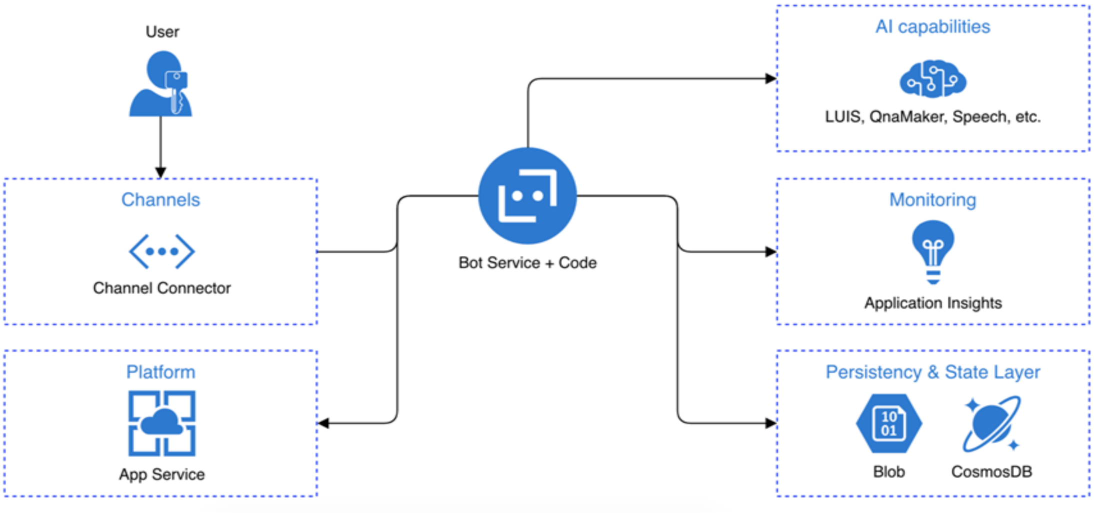


Fig 2 Bot framework architecture on Azure (Source: https://clemenssiebler.com/microsoft-bot-framework-v4-explained-javascript/)

When a user chats with the Covid bot application, it enabled users to choose their conversations by picking a "from an options tab" (using a menu approach) in order to structure the line of queries and a respective response from the bot application. The Microsoft Bot Composer provided a high-level dialogue flow to identify user queries from the menu options and then maps them to the responses that it has been trained with.

4.4 System Development Methodology

The system development methodology adopted for the development of this system is the waterfall model. The waterfall methodology breaks down a project into sequential phases wherein the output of the previous stage is the input of the next stage.

4.4.1 Activity Diagram

These diagrams, often considered as behavior diagrams, describe what must happen in the system being modeled. In the Covid Bot the activity diagrams model the development process in three phases viz:

4.4.1.1 Bot Greeting

The bot begins every chat session with an introductory text and a greeting. It also prompts the user to use the trigger phrase in order to start a dialog. Fig 3 shows the activity diagram for the chatbot greeting. Fig 3 depicts the Bot Greeting Activity Diagram.

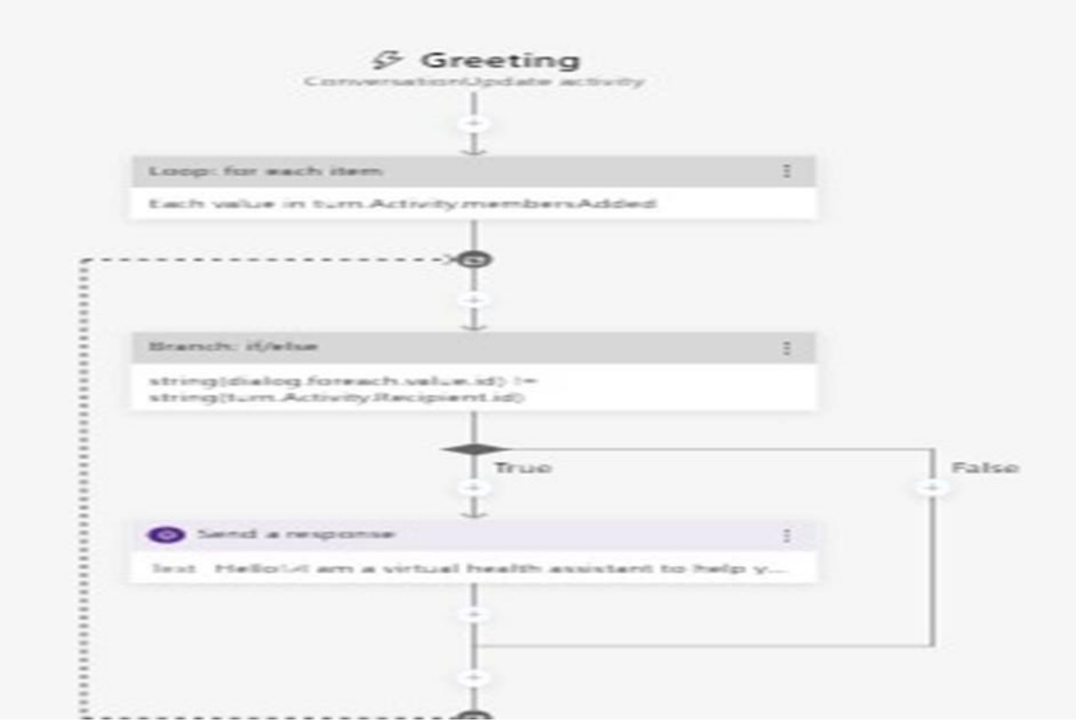
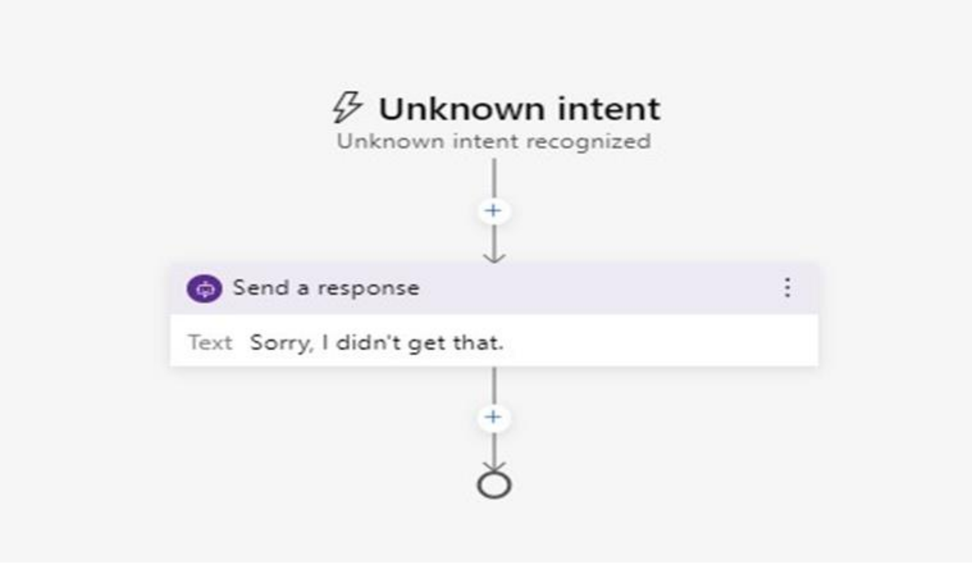

Greeting
ConversationUpdate activity
Loop: for each item
Each value in turn.Activity.membersAdded
Branch: if/else
string(dialog.foreach.value.id) !=
string(turn.Activity.Recipient.id)
True
False
Send a response
Text Hello!<I am a virtual health assistant to help y...

Fig 3 Bot Greeting activity diagram

4.4.1.2 Unknown Intent

For every word used by the user that is not recognized as a saved trigger phrase, the bot would give a response in order to alert the user of the words it currently understands. Fig 4 shows the activity diagram for an unknown intent

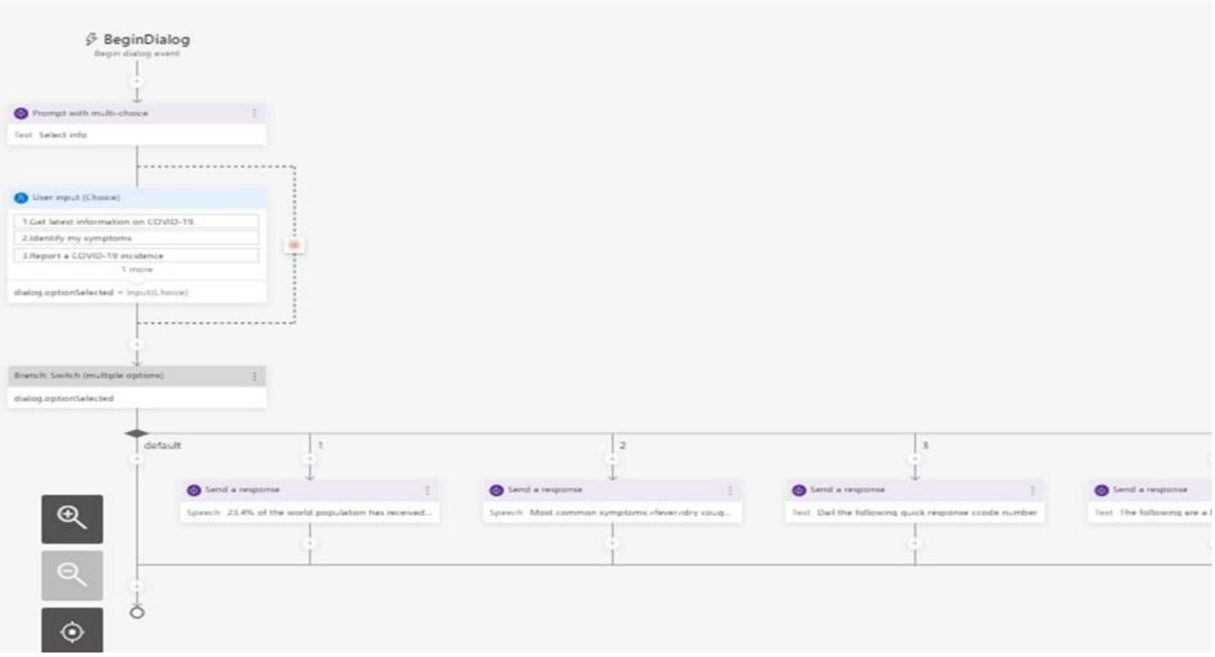


Text Sorry, I didn't get that.

Fig 4. Unknown Intent activity diagram

4.4.1.2. Begin Dialog

A dialog is the beginning of a chat session with a user. A dialog Chat session starts with the "Bot Greeting", then introduces the trigger word and finally proceeds to display close ended questions for the user to choose from. Fig 5 shows the activity diagram of the Begin dialog.



Text Select info

1.Get latest information on COVID-19

2.Identify my symptoms

3.Report a COVID-19 incidence

dialog.optionSelected = Input(Choice)

dialog.optionSelected

Speech 23.4% of the world population has received...

Speech Most common symptoms:-fever,-dry coug...

Text Dail the following quick response ccode number

Text The following are a l

Fig 5. Begin dialog activity diagram

4.4.2 Use case Diagram

This is a pictorial representation of a user's possible actions and interactions with a system. The covid bot is concerned with four major actions as follows: getting latest information on Corona virus, reporting an incidence, getting information on identifying symptoms and getting information on the closest Corona virus relief centers. Fig 6 shows the activity use case diagram for the covid bot.

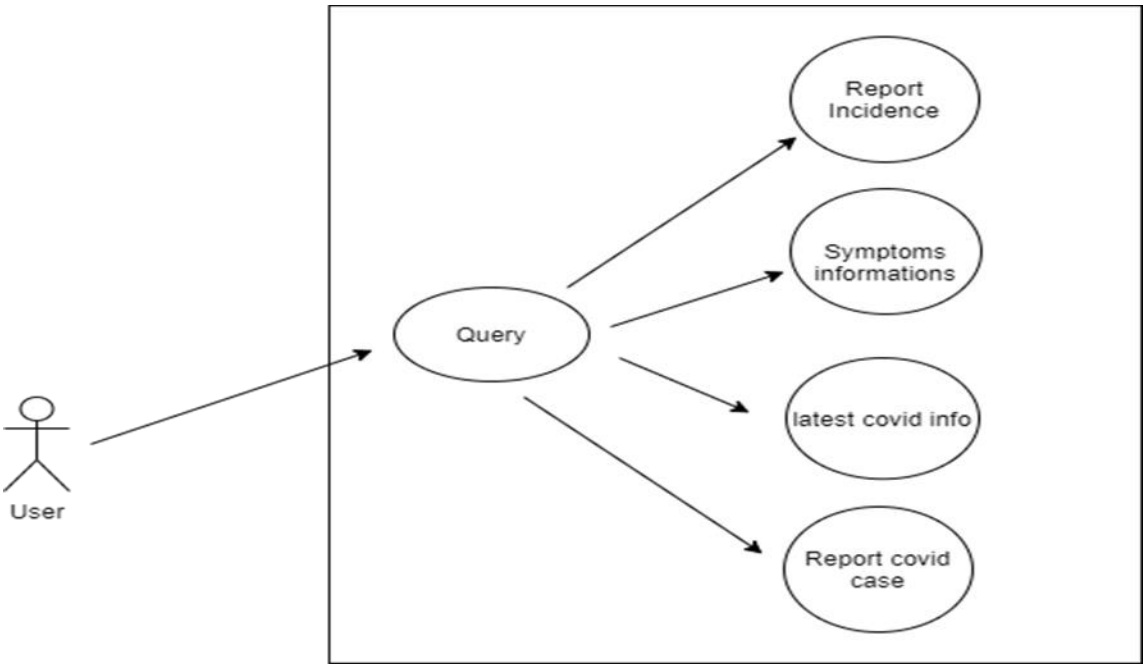


Fig 6. Use case diagram of covid bot

5. System implementation

System implementation is the process of realizing a new system design. It is this concept that converts the theoretical concepts and design into a working system. The development of the chatbot was done using Microsoft Bot Composer and Microsoft Azure cloud computing services.

a. **Hardware Requirements**

The minimum hardware or physical computer or mobile resources requirements of the chatbot are as follows: (i) 1.50 GHz Quad core processor (for Personal Computers) (ii) web browser with JavaScript enabled (iii) 1GB RAM or higher (iv) 500MB or higher for storage (v) Network devices and Internet Connection

b. **Software Requirements**

The minimum software requirements to provide optimal functioning of the software are listed as follows: (i) Operating System (windows 10 or higher, Mac OS, Android 8 or higher, IOS) (ii) web browser (Google Chrome, Safari).

5.1. Microsoft Bot Framework Composer

Microsoft Bot Framework Composer, built on the Bot Framework Software Development Kit (SDK), is an open-source Integrated Development Environment (IDE) for developers to author, test sand manages conversational experiences (http://docs.microsoft.com). The framework is a powerful visual authoring canvas that enables dialog, language-understanding models, question and answer maker knowledge bases and language generation responses to be authored from within one canvas (https://docs.microsoft.com/en-us/composer/introduction). It, in addition, enables these experiences to be extended with codes for more complex tasks such as system integration (https://docs.microsoft.com/en-us/composer/introduction).

5.2. Implementation

With the dialog, chat functionality and interface design finalized, the next step is putting all of it together for the chatbot. The waterfall model was used for the development of the application, because each iterative prototype is developed only after the end of the previous prototype. The development of a prototype does not end until it has met the criteria for acceptance, thus it continues to be tweaked and improved upon for the next iteration. The system was implemented as follows:

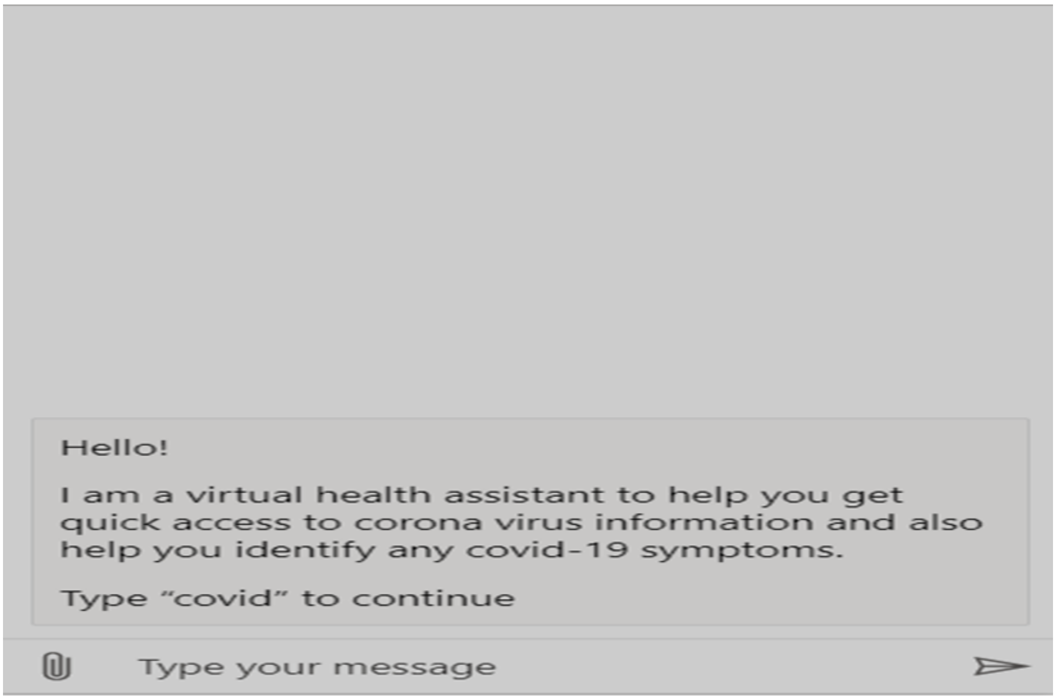
Step 1 – Create bot Instance

Since the Microsoft Bot Composer Framework abstracts the coding interface used for the execution of the project, the first step was to create the bot instance within the bot composer framework. Azure Web functions were loaded into the back-end of the system. This enabled the structure for a triggers and dialog to be created.

Step 2 –Create Triggers and Dialog

In order for a chat session to be created, a dialog instance would be created. In a dialog trigger and all the chat interactions, greetings and introductions would be created therein.

In the covid bot, a trigger phrase is used to start a chat session; in this case the word "covid" would be used as the trigger phrase to start the session. Thus, whenever the word "covid" is used in a chat session, a dialog will begin. Fig 7 shows the bot greetings and trigger word introduction.



Hello!

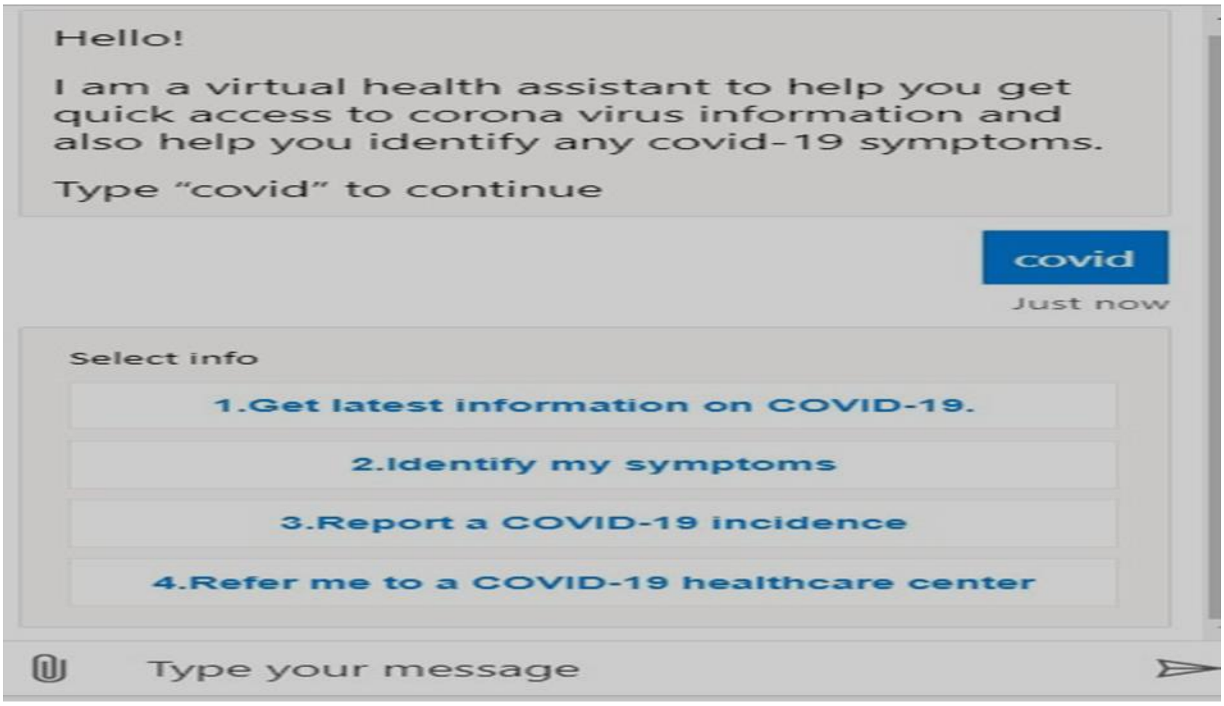
I am a virtual health assistant to help you get quick access to corona virus information and also help you identify any covid-19 symptoms.

Type "covid" to continue

Fig 7. Bot greetings and introduction

Step 3 –Web App Implementation

At this stage, the dialog and trigger phrases have been created, but it has not been put in a form where it can be used in the real world. Consequently, steps were taken to integrate the model in a web application that can also be accessible on a mobile phone. For this process, a web chat emulator running locally within the Microsoft Bot Composer was used for the initial implementation before it was later deployed fully on Azure application services. Fig 8 shows a chat session on the chat bot.



Hello!

I am a virtual health assistant to help you get quick access to corona virus information and also help you identify any covid-19 symptoms.

Type "covid" to continue

covid

Just now

Select info

1.Get latest information on COVID-19.

2.Identify my symptoms

3.Report a COVID-19 incidence

4.Refer me to a COVID-19 healthcare center

Type your message

Fig 8 Bot Chat session

6. Conclusion

This study focused on overcoming the problems of disseminating critical healthcare information due to the outburst of the novel Corona virus. This system adopted the use of a conversational artificial intelligence chatbot to deliver health information on covid-19 giving answers to students concerning its latest spread, help centers and information on its symptoms. This was done to combat the propagation of fake news on social media and the risk of spreading the virus through crowded health care centers.

This research study was limited to promptly answering questions on COVID-19 and its symptoms as well as providing additional information on its spread in real-time from the students in the University of Benin, Computer science department. In addition, since there was little coding required in building this bot, the bot functions by using predefined questions and answers stored on the cloud with manual input and gives the output of the associated question to the user/students.

References

[1] C. B. Mann (2020) 'Can conversing with a computer increase Turnout? Mobilization using chatbot communication". Journal of Experimental Political Science 1–12, Cambridge University Press. doi:10.1017/XPS.2020.5.

[2] R. Dale (2016). "The return of the chatbots," Nat. Lang Eng, 22, 811–817. Cambridge University Press, doi:10.1017/S1351324916000243

[3] S. Janarthanam (2017). ‘Hands-On Chatbots and Conversational UI Development: Build Chatbots and Voice User Interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills”, Packt Publishing Ltd.

[4] A. Shevat (2017). ‘Designing Bots: Creating Conversational Experiences”, Sebastopol, CA: O'Reilly Media. Inc. Retrieved July 8, 2021 from https://www.oreilly.com/library/view/designing-bots/9781491974810

[5] J. Wang; L. Gu and M. Aiken (2019). A comparison of three conversational agents.
Asian Journal of Science and Technology 10(9), 10051-10055. Retrieved February 25, 2022 from https://journalajst.com/sites/default/files/issues-pdf/7316.pdf

[6] S. Kundu (2020). The Evolution of conversational Al. The Research Nest. Retrieved February 10, 2022 from https://medium.com/the-research-nest/the-evolution-of-conversational-ai-d4e564ff851f

[7] Y. Zhang; S. Sun; M. Galley ; Y. Chen ; C. Brockett ; X. Gao ; J. Gao ; J. Liu and B. Dolan (2020): “DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation”, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstration. Association for Computational Linguistics (ACL), 270–278, doi:10.18653/v1/2020.acl-demos.30 Retrieved July 5, 2021 from https://aclanthology.org/2020.acl-demos.30.pdf

[8] L. Zhou ; J. Gao; D. Li and H-Y. Shum (2018). "The Design and Implementation of XiaoIce, an Empathetic Social Chatbot", Retrieved July 9, 2021 form https://arxiv.org/abs/1812.08989

[9] N. Lewis (2020). "Robot friends: Why people talk to chatbots in times of trouble". CNN, Retrieved July 10, 2021 from https://edition.cnn.com/2020/08/19/world/chatbot-social-anxiety-spc-intl/index.html