11/21/2023

Chatbot Ticketing

Deliverable 1: Proof of Concept

Ron Coveney

X00204989 4th Year Project Cloud computing with devops

Table of Contents

[Executive Summary 2](#_Toc151661126)

[History of Chatbots 2](#_Toc151661127)

[Project Purpose and Scope 3](#_Toc151661128)

[Purpose 3](#_Toc151661129)

[Scope 4](#_Toc151661130)

[Technologies Planned for Use 4](#_Toc151661131)

[Zendesk 4](#_Toc151661132)

[Python Programming Language 5](#_Toc151661133)

[RASA 5](#_Toc151661134)

[Zendesk API 5](#_Toc151661135)

[Webhooks 5](#_Toc151661136)

[Chat Platform Integration 5](#_Toc151661137)

[Authentication 5](#_Toc151661138)

[Development Environment 5](#_Toc151661139)

[Cloud Hosting 5](#_Toc151661140)

[Ngrok 5](#_Toc151661141)

[Proof of Concept 5](#_Toc151661142)

[Prerequisites 5](#_Toc151661143)

[Run Proof of Concept 7](#_Toc151661144)

[Conclusion 8](#_Toc151661145)

[Bibliography 9](#_Toc151661146)

## Executive Summary

The history of chatbots traces back to the 1960s, with significant milestones in natural language processing and artificial intelligence. From early programmes like ELIZA and PARRY to modern advancements like GPT-3 and GPT-4, chatbots have evolved, finding applications in customer service, IT support, and more. In the context of- IT support, the integration of chatbots with ticketing systems has become pivotal for streamlining processes and improving user experience.

The purpose of the proposed chatbot integrated ticketing system is to enhance IT support within an organisation. It aims to provide user support and query resolution, automate ticket creation, facilitate communication and updates, integrate with knowledge bases, and offer efficient ticket management, including incident resolution and request fulfilment.

The technologies planned for use include Zendesk for ticketing, Python as the programming language, RASA for natural language processing, and Microsoft teams for deployment. Authentication will be ensured through OAuth or API Keys. The development environment will use PyCharm and Azure DevOps for version control, and Azure is considered for cloud hosting to ensure scalability. Ngrok will be employed for secure tunnelling to expose the local machine to the internet.

The proof of concept involves creating a basic chatbot that connects with a fictional IT ticketing system. The steps include installing required packages, creating a simple Flask web server, exposing local machine ports using Ngrok, and running the bot to elicit a response from the server thereby proving the fundamental connection needed to transfer ticketing information.

## History of Chatbots

The first chat bot was created in 1966 by Joseph Weizenbaum at the Massachusetts Institute of Technology. The bot known as ELIZA *(Eliza 1999),* used an early form of natural language processing to simulate a psychotherapist and engaged in text-based conversations with users. Eliza used simple pattern matching to respond to user inputs. There are still online versions of Eliza however it is used more as a curiosity then a research tool.

The next key advancement came in 1972 with the introduction of PARRY. Parry was developed by Kenneth Colby and was used to simulate a person with paranoid schizophrenia. Parry got the affectionate name ‘Eliza with attitude’ and in 1973 during an international computer conference ELIZA and PARRY got to speak to each other when Vint Cerf set up both machines to speak to each other in what resulted was a very vague and strange conversation *(Garber, 2014).*

In 1984 Mindscape, a computer games manufacturer founded by Roger Bouy released RACTER *(Retterberg, 1983)*. Racter (short for raconteur) was developed by William Chamberlin and Thomas Etter was a program that generated prose and poetry and was one of the first attempts to create a chatbot that could produce creative content.

The first attempt to incorporate artificial intelligence into a chat bot came in 1995 when Dr Richard Wallace developed A.L.I.C.E (Artificial Linguistic Internet Computer Entity) as part of the artificial Intelligence Markup Language (AMIL) project. ALICE used early natural language processing to engage users in open-ended conversations *(Rouse, 2023).*

In 2001, with the internet beginning to explode but with instant messaging still in its infancy, ActiveBuddy in conjunction with America Online (AOL) developed SmarterChild. SmarterChild became a popular chatbot that provided users with weather updates and the ability to paly simple games through chat. SmartedChild was eventually shutdown in 2008 but for many non tech people it was their first interaction with an AI based chatbot *(Rice, 2023)*.

Big advancements were made in natural language processing and with the release of the Apple iPhone 4S in 2011, the world first met SIRI. SIRI was the first widely available voice-controlled AI technology and while not a traditional chatbot, it did mark a significant advancement in natural language processing and voice recognition technology. Originally developed by The Scientific Research Institute in Stanford University, SIRI used a speech recognition engine provided by Nuance Communications, although Apple wont publicly confirm this *(Kay, 2014)*.

When Facebook introduced messenger bots in 2016 it opened up chatbots to enterprise. Businesses were able to create chatbots to interact with users to manage customer service enquiries and facilitate transactions. This was a big turning point in the evolution of chatbots because I proved that they had advanced sufficiently to have a trusted commercial use and were being adapted for mainstream use.

In 2020 OpenAI released GPT-3 (Generative pre-trained Transformer 3), while not a traditional chatbot it is used by many developers as a backend for chatbots. The front facing chatbot sends the query from the user to GPT and the output is used by the chatbot. Its newest version GPT-4 released in 2023 advanced the previous version by using larger datasets (gpt-3 uses 17 gigabytes gpt-4 uses 45 gigabytes) and has an increased capacity to interpret idiomatic expressions and sarcasm *(Daivi, 2023).*

## Project Purpose and Scope

A chatbot integrated ticketing system serves to streamline and enhance the IT support process within an organisation. The scope of the system will encompass various aspects of IT service management, including incident resolution, communication with and end user and request fulfilment.

### Purpose

#### User Support and Query Resolution

* Level 1 support

The chatbot can provide initial support by addressing and dealing with common IT issues, answering frequently asked questions and guiding users through troubleshooting steps.

* Immediate Assistance

Users can get instant responses and guidance, leading to quicker issue resolution and improved user experience.

#### Automated Ticket Creation

* The chatbot will be able to understand and interpret user queries, identify IT issues, and automatically create tickets for problem resolution.
* Efficiency

Automation will reduce manual effort, accelerate ticket creation, and ensure that tickets are logged correctly.

#### Communication and Updates

* Status Updates

The chatbot will be able to provide status updates on existing tickets to keep users informed of their ticket process.

* Notifications

Automated notifications can be sent through the chat bot to inform users of system outages, updates, or any other relevant information.

#### Workflow Automation

* Task automation

Integrating with a ticketing system will allow the chatbot to automate certain tasks or workflows, improving the overall efficiency of the IT process.

#### Knowledge Base Integration

* The chat bot will integrate with a knowledgebase to provide relevant information or solutions to common problems. This will empower users to resolve issues without intervention from the IT team.

### Scope

#### Ticket Management

* Creation

Automatically generate tickets based on user interactions and issue identification.

* Updates

Provide real-time updates to users on the status and progress of their tickets.

#### User Interaction

* Queries

Handel user queries, providing instant responses and solutions.

* Guidance

Assist users in troubleshooting issues to resolve common IT issues.

#### Knowledge Base Integration

* Access

Integrate with the organizations IT Knowledge base to provide relevant information and solutions.

* Update

Keep the Knowledge Base up to date with added information and issue resolutions.

#### Automation

* Task Automation

Automate routine tasks to improve efficiency.

* Escalation

Automatically escalate tickets to human agents when necessary.

#### Reporting

* Identify trends and recurring problems to improve overall IT support.

## Technologies Planned for Use

### Zendesk

Zendesk is a widely used customer service platform that provides a variety of tools and features to help businesses manage customer support. It offers cloud-based service to allow companies to manage their customer service needs. The key feature that will be used is the Zendesk Ticketing System.

### Python Programming Language

Python has a clean and easy to understand syntax making it an ideal language to program the chatbot in. It also supports many libraries and frameworks that can be leveraged in the development process.

### RASA

RASA is an open source, Natural Language Processing (NLP) framework that integrates with Python. It can be used to create AI chatbots and virtual assistants. It provides tools and libraries for developing chatbots that understand user inputs.

### Zendesk API

This will be used to integrate the chatbot with Zendesk ticketing system. This API will enable the creation, update and retrieval of tickets and allow access to other Zendesk features.

### Webhooks

Webhooks enable real-time communication between the chatbot and Zendesk. They allow the bot to receive notifications and updates from Zendesk and to trigger actions in response.

### Chat Platform Integration

I would like to deploy my chatbot in Microsoft Teams due to its wide use in enterprise. Microsoft Teams is the industry standard in instant virtual communication and to integrate with that would make the user experience more familiar.

### Authentication

There will need to be a secure authentication method between the chatbot and the Zendesk API. This will include using API Keys or OAuth.

### Development Environment

The code editor used will be PyCharm and I will be using Azure DevOps for version control.

### Cloud Hosting

I would like to explore hosting the chatbot on Azure. This would allow for scalability in the case of high traffic.

### Ngrok

Ngrok is a tool that can create a secure tunnel to a localhost. This allows a user to expose a local machine to the internet.

## Proof of Concept

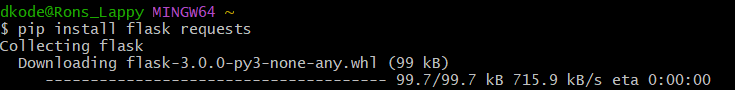
This proof of concept will create a basic chatbot that connects with a fictional IT Ticketing system.

### Prerequisites

* Python installed on local machine.
* Flask for creating a simple webserver.
* Requests library for HTTP requests.

#### Step 1

Install required packages.



#### Step 2

Create a simple flask webserver with a single endpoint “/webhook.” This can manage incoming POST requests. The endpoint expects JSON data containing a user message, and it responds with a placeholder message.

A screenshot of a computer program

Description automatically generated

#### Step 3

Expose local pc to the internet using Ngrok to expose port 5000

Using command: Ngrok http 5000

A screenshot of a computer

Description automatically generated

#### Step 4

Create a simple Chatbot.

A screenshot of a computer program

Description automatically generated

### Run Proof of Concept

1. Expose port 5000 to internet.

Navigate to my local folder.

C:\Users\dkode\ngrok

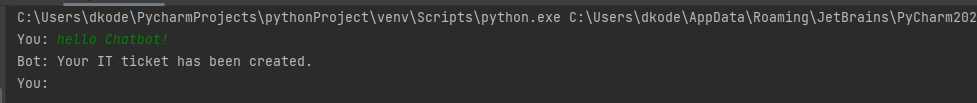
Open cmd line and enter command: Ngrok http 5000.

1. Run Web server (app.py)

A screen shot of a computer

Description automatically generated

1. Run Chatbot (chatbot.py) and give it an input.



This proof of concept shows that a simple chatbot can connect to a webhook endpoint (‘/webhook’) This endpoint simulates interaction with the IT ticketing system. This is the basic connection needed to transfer ticketing information.

Finally Commit and push working POC to Azure DevOps

A computer screen shot of a program

Description automatically generated

A screenshot of a computer

Description automatically generated

## Conclusion

In conclusion, the proposed chatbot integrated ticketing system represents a significant step forward in optimising IT support processes. By leveraging advancements in natural language processing and combining them with established technologies such as Zendesk and Python, the system aims to enhance user support, automate ticket management, and improve overall efficiency. The proof of concept successfully demonstrates the feasibility of connecting the chatbot to a webhook endpoint, setting the stage for further development and integration with real world ticketing systems. As technology continues to advance, this project aligns with the ongoing trend of incorporating AI-driven solutions to streamline organisational processes and improve user experience in the realm of IT support.

## Bibliography

1. *Eliza, the Rogerian therapist* (no date) *Eliza, Computer Therapist*. Available at: <https://psych.fullerton.edu/mbirnbaum/psych101/eliza.htm>

(Accessed: 13 November 2023).

1. Garber, M. (2014) *When Parry met Eliza: A ridiculous chatbot conversation from 1972*, *The Atlantic*. Available at: <https://www.theatlantic.com/technology/archive/2014/06/when-parry-met-eliza-a-ridiculous-chatbot-conversation-from-1972/372428/>

(Accessed: 13 November 2023).

1. Retterberg, S. (1983) *Racter*, *ELMCIP*. Available at: <https://elmcip.net/creative-work/racter>

(Accessed: 13 November 2023).

1. Rouse, M. (2023) *Artificial linguistic computer entity - techopedia*, *ALICE (Artificial Linguistic Computer Entity)*. Available at:

<https://www.techopedia.com/definition/380/artificial-linguistic-computer-entity-alice>

(Accessed: 13 November 2023).

1. Rice, T. (2023) *What happened to Smarterchild? (the aim chatbot)*, *Animas Marketing*. Available at: <https://animasmarketing.com/what-happened-to-smarterchild-the-aim-chatbot/>

(Accessed: 13 November 2023).

1. Kay, R. (2014) *Behind Apple’s siri lies nuance’s speech recognition*, *Forbes*. Available at: <https://www.forbes.com/sites/rogerkay/2014/03/24/behind-apples-siri-lies-nuances-speech-recognition/?sh=77d9ec2f18e5>

(Accessed: 13 November 2023).

1. Daivi (2023) *GPT3 vs gpt4-battle of the holy grail of ai language models*, *ProjectPro*. Available at: <https://www.projectpro.io/article/gpt3-vs-gpt4/816#:~:text=Compared%20to%20GPT%2D3’s%2017,accurate%20results%20than%20GPT%2D3>.

(Accessed: 13 November 2023).