# PROFESSIONAL TRAINING REPORT

**at**

# Sathyabama Institute of Science and Technology

## (Deemed to be University)

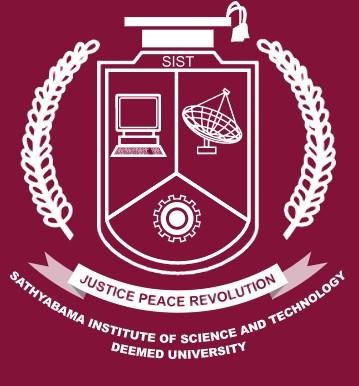
Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

# By CHINTHALACHERUVU GOVARDHAN

**REDDY**

# (Reg.No.39110235)



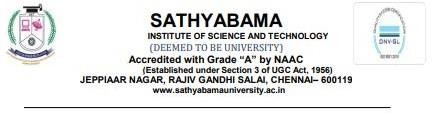
## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

# CHENNAI - 600119,TAMILNADU

# NOVEMBER 2021



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of

## CHINTHALACHERUVU GOVARDHAN REDDY (Reg. No. 39110235) who carried

out the project entitled “**CHAT BOTE** ” under my supervision from September 2021 to November 2021.

# Internal Guide

Ms. Vinodhini ,

# Head of the Department

**Dr. L. Lakshmanan, M.E., Ph.D.,**

## Submitted for Viva voce Examination held on

**Internal Examiner External Examiner**

# DECLARATION

**I, CHINTHALACHERUVU GOVARDHAN REDDY (Reg. No**

**39110235)hereby declare that the Professional Training Report on “CHAT BOT” done by me under the guidance Ms.Vinodhini, at Sathyabama institute of science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.**

## DATE:

**PLACE: SIGNATURE OF THE CANDIDATE:**

# ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA., ME., Ph.D**., Dean, School of Computing **Dr. L. LAKSHMANAN,ME., Ph.D**., Dr.S.Vigneshwari M.E., Ph.D., Head of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. Sankari M, M.E.,(Ph.D),**for her valuable guidance ,suggestions and

constant encouragement paved way for the successful completion of my project

work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project.

# CERTIFICATE

**ABSTRACT**

ChatBot can be described as software that can chat with people using artificial intelligence. These software are used to perform tasks such as quickly responding to users, informing them, helping to purchase products and providing better service to customers. In this paper, we present the general working principle and the basic concepts of artificial intelligence based chatbots and related concepts as well as their applications in various sectors such as telecommunication, banking, health, customer call centers and e-commerce. Additionally, the results of an example chabbot for donation service developed for telecommunication service provider are presented using the proposed architecture.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE**  **ABSTRACT** | **PAGE NO.**  **6** |
|  | **LIST OF FIGURES** | **9** |
| **1** | **INTRODUCTION** | **10** |
|  | 1.1 PROBLEM STATEMENT | 11 |
|  | 1.2 MOTIVATION | 11 |
|  | 1.3 OBJECTIVE | 11 |
|  | 1.3.1 PROPOSED SYSTEM | 11 |
| **2** | **TECHNOLOGIES LEARNT** | **12** |
|  | 2.1 PYTHON | 12 |
|  | 2.1.1 ADVANTAGES OF PYTHON | 12 |
|  | 2.1.2 DISADVANTAGES OF PYTHON | 13 |
|  | 2.2 MACHINE LEARNING | 13 |
|  | 2.2.1CHALLENGES IN MACHINE  LEARNING | 14 |
|  | 2.2.2 TYPES OF MACHINE LEARNING | 15 |
|  | 2.2.3 ADVANTAGES OF MACHINE  LEARNING | 16 |
|  | 2.2.4 DISADVANTAGES OF MACHINE  LEARNING | 17 |
|  | 2.2.5 APPLICATIONS OF MACHINE  LEARNING | 18 |
|  | 2.3 MODULES USED IN PROJECT | 18 |  |

|  |  |  |
| --- | --- | --- |
|  | 2.4 INSTALL PYTHON STEP-BY-STEP  IN WINDOWS | 20 |
| **3** | **EXPERIMENTAL OR MATERIALS AND METHODS** | **22** |
|  | 3.1 CHATBOT | 22 |
|  | 3.2 TYPES OF CHATBOT | 23 |
|  | 3.2.1 GENERATIVE BASED | 24 |
|  | 3.2.2 RETRIEVAL BASED | [2](#_bookmark0)4 |
|  | 3.3 NATURAL LANGUAGE PROCESSING | 25 |
|  | 3.4 SYSTEM SPECIFICATION | 26 |
|  | 3.4.1SOFTWARE REQURIMENTS | 26 |
|  | 3.4.2 HARDWARE REQURIMENTS | 26 |
| 4 | **DESIGN** | **27** |
|  | 4.1 UNIFIED MODEL DIAGRAM | 27 |
|  | 4.1.1 USE CASE DIAGRAM | 28 |
|  | 4.1.2 SEQUENCE DIAGRAM | 29 |
|  | 4.1.3 CLASS DIAGRAM | 30 |
|  | 4.1.4 COMPONENT DIAGRAM | 31 |
|  | 4.1.5 ACTIVITY DIAGRAM | 33 |
|  | 4.1.6 [FLOW CHART DIAGRAM](#_bookmark1) | 34 |
|  | **SUMMARY AND CONCLUSION** | **35** |
|  | REFERENCES | 35 |
|  | APPENDIX | 37 |
|  | A.[SCREENSHOTS](#_bookmark4) | 37 |
|  | B.SOURCE CODE | 40 |

**LIST OF FIGURES**

**FIGURE NO. FIGURE NAME PAGE NO.**

3.1 Chat bot 22

3.2 Generative Based 23

3.3 Retrieval Based 24

3.4 Natural Language Processing 25

* 1. Use case Diagram 28
  2. Sequence Diagram 29
  3. Class Diagram 30
  4. Component Diagram 31
  5. Activity Diagram 33
  6. Flowchart Diagram 34
  7. Importing the packages 35
  8. Tokenization 35
  9. Stop word 35
  10. Tokenization and stop words 37
  11. Completed chat bot 38

# CHAPTER 1 INTRODUCTION

## PROBLEM STATEMENT

Artificial Intelligence (ΑΙ) increasingly integrates our daily lives with the creation and analysis

of intelligent software and hardware, called intelligent agents. Intelligent agents can do a

variety of tasks ranging from labor work to sophisticated operations. A chatbot is a typical

example of an AI system and one of the most elementary and widespread examples of

intelligent Human-Computer Interaction (HCI) [[1](https://europepmc.org/article/pmc/pmc7256567" \l "CR1)]. It is a computer program, which responds

like a smart entity when conversed with through text or voice and understands one or more

human languages by Natural Language Processing (NLP) [[2](https://europepmc.org/article/pmc/pmc7256567" \l "CR2)]. In the lexicon, a chatbot is

defined as “A computer program designed to simulate conversation with human users,

especially over the Internet” [[3](https://europepmc.org/article/pmc/pmc7256567" \l "CR3)]. Chatbots are also known as smart bots, interactive agents,

digital assistants, or artificial conversation entities.

Chatbots can mimic human conversation and entertain users but they are not built only

for this. They are useful in applications such as education, information retrieval, business,

and e-commerce [[4](https://europepmc.org/article/pmc/pmc7256567" \l "CR4)]. They became so popular because there are many advantages of

chatbots for users and developers too. Most implementations are platform-independent and instantly available to users without needed installations. Contact to the chatbot is spread

through a user’s social graph without leaving the messaging app the chatbot lives in, which provides and guarantees the user’s identity. Moreover, payment services are integrated into

the messaging system and can be used safely and reliably and a notification system re-engages inactive users. Chatbots are integrated with group conversations or shared just like any other contact, while multiple conversations can be carried forward in parallel. Knowledge in the use

of one chatbot is easily transferred to the usage of other chatbots, and there are limited data requirements. Communication reliability, fast and uncomplicated development iterations, lack

of version fragmentation, and limited design efforts for the interface are some of the advantages

for developers too [[5](https://europepmc.org/article/pmc/pmc7256567" \l "CR5)]..

## **MOTIVATION**

We have previously outlined the potential implications and opportunities that chatbots hold for

the field of HCI [[1](https://interactions.acm.org/archive/view/september-october-2018/chatbots" \l "R1)]. In this article, we dig into what we see as a key challenge with chatbots

from a user-centered perspective. Developers and designers have an urgent need to know

more about how people experience chatbots and to understand the user needs that motivate

the future use of chatbots. We therefore need to ask how chatbots resonate with user needs

and desires and, in turn, how these same needs and desires evolve as users get more

experience with chatbots.

.

**1.3 OBJECTIVE:**

Chatbots boost operational efficiency and bring cost savings to businesses while

offering convenience and added services to internal employees and external

customers. They allow companies to easily resolve many types of customer queries

and issues while reducing the need for human interaction.

Chatbots are very intelligent. You train them once and they will communicate with

your target audience in their language.Multilingual chatbots have saved you from

investing much on hiring different languages resources.

**1.3.1 PROPOSED SYSTEM:**

**Namrata Mahakalkar [10], proposed an artificial chatbot using NLP (Natural Language Processing)** which can be done in two ways the first via written text and the second is via verbal or voice communication. Written communication is much easier than the verbal communication.

# CHAPTER 2 TECHNOLOGIES LEARNT

## PYTHON

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following –

1. Machine Learning
2. GUI Applications (like Kivy, Tkinter, PyQt etc. )
3. Web frameworks like Django (used by YouTube, Instagram, Dropbox)
4. Image processing (like OpenCV, Pillow)
5. Web scraping (like Scrapy, BeautifulSoup, Selenium)
6. Test frameworks
7. Multimedia

## ADVANTAGES OF PYTHON

1. Extensive Libraries
2. Extensible
3. Embeddable
4. Improved Productivity
5. IOT Opportunities
6. Readable
7. Object-Oriented
8. Free and Open Source
9. Portable
10. Interpreted

## DISADVANTAGES OF PYTHON

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

1. Speed Limitations
2. Weak in Mobile Computing and Browsers
3. Design Restrictions
4. Underdeveloped Database Access layers
5. Simple

## MACHINE LEARNING

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means

of building models of data.

## 2.2.1 CHALLENGES IN MACHINE LEARNING

While Machine Learning is rapidly evolving, making significant strides with cyber

security and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are

## Quality of data

Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

## Time-Consuming task

Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

## Lack of specialist persons

As ML technology is still in its infancy stage,availability of expert resources is a tough job.

## No clear objective for formulating business problems

Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

## Issue of over fitting & under fitting

If the model is over fitting or under fitting, it cannot be represented well for the problem.

## Curse of dimensionality

Another challenge ML model faces is too many features of data points. This can be a real hindrance.

## Difficulty in deployment

Complexity of the ML model makes it quite difficult to be deployed in

real life.

## 2.2.2 TYPES OF MACHINE LEARNING

**Supervised Learning**

This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

## Unsupervised Learning

This involves using un labelled data and then finding the

underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

## Semi-supervised Learning

This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

## Reinforcement Learning

This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

## 2.2.3 ADVANTAGES OF MACHINE LEARNING

**Easily identifies trends and patterns**

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e- commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

## No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwaree, they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

## Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data, you have keeps growing, your algorithms learn to make more accurate predictions faster.

## Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

## Wide Applications

You could be an e tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

## 2.2.4 DISADVANTAGES OF MACHINE LEARNING Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

## Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

## Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

## High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

## 2.2.5 APPLICATIONS OF MACHINE LEARNING

These are the applications :

1. Emotion analysis
2. Sentiment analysis
3. Error detection and prevention
4. Weather forecasting and prediction
5. Stock market analysis and forecasting
6. Speech synthesis
7. Speech recognition
8. Customer segmentation
9. Object recognition
10. Fraud detection
11. Fraud prevention
12. Recommendation of products to customer in online shopping

## MODULES USED IN PROJECT NUMPY

Numpy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

1. A powerful N-dimensional array object
2. Sophisticated (broadcasting) functions
3. Tools for integrating C/C++ and Fortran code
4. Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

## PANDAS

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging And preparation.It had very little contribution towards data analysis. Pandas solved this problem.Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc

## MATPLOTLIB

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

## SCIKIT – LEARN

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

## INSTALL PYTHON STEP-BY-STEP IN WINDOWS

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3. Before you start with the installation process of Python. First, you need to know about your system Requirements. Based on your system type i.e. operating system and based

processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.Download the Correct version into the system.

**Step 1**: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link**: [https://www.python.org](https://www.python.org/)**

Now, check for the latest and the correct version for your operating system.

**Step 2**: Click on the Download Tab.

**Step 3**: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

**Step 4**: Scroll down the page until you find the Files option.

**Step 5**: Here you see a different version of python along with the operating system. Here we will install Windows x86-64 web-based installer. Here your first part regarding.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation.

**Note**: To know the changes or updates that are made in the version you can click on the Release Note Option.

## Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.

**Step 2:** Before you click on Install Now, make sure to put a tick on Add Python 3.7 to path.

**Step 3:** Click on Install now, after the installation is successful. Click on Close. With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

# CHAPTER 3

**EXPERIMENTAL OR MATERIALS AND METHODS**

1. **1CHATBOT**

A chatbot is artificial intelligence (AI) software that can imitate a natural language discussion (or chat) with a user via messaging apps, websites or mobile

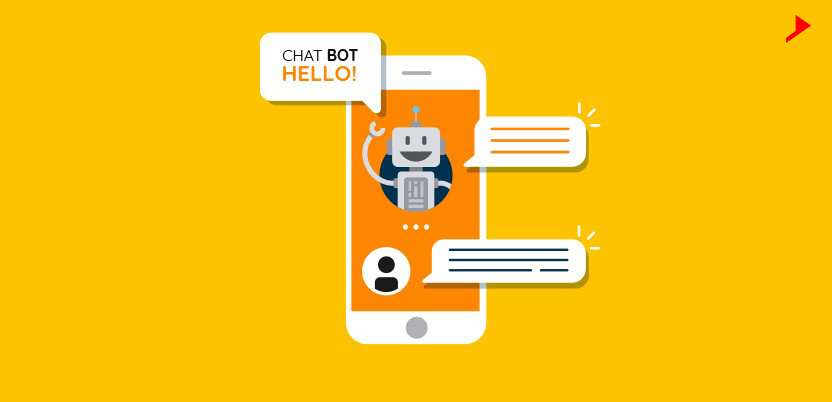
apps.

What is the significance of chatbots? A chatbot is frequently described as one

of the most advanced and promising forms of human-machine interaction.

Chatbots can automatically simulate interactions with customers based on a

set of predefined conditions or events.



**Fig 3.1 chatbot**

**3.2 TYPES OF CHATBOT**

# **3.2.1 Generative Based**

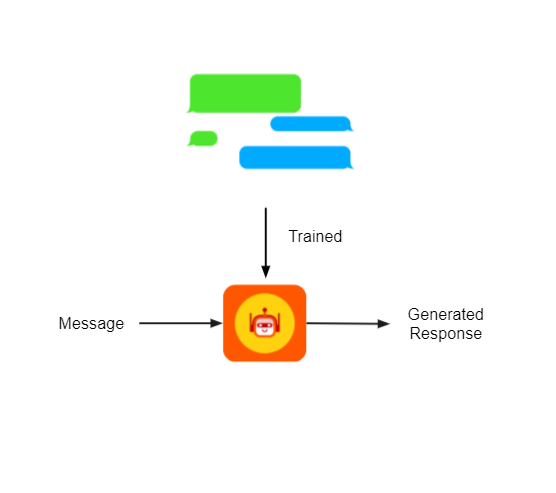
Generative chatbots use a combination of supervised learning, unsupervised

learning & reinforcement learning. A generative chatbot is an open-domain

chatbot that creates unique language combinations rather than selecting from

a list of pre- defined responses. Retrieval-based systems are limited to predefined responses. Chatbots that use generative methods can generate new dialogue

based on large amounts of conversational training data.



# Fig 3.2 Generative based

**3.2.2** Retrieval Based:

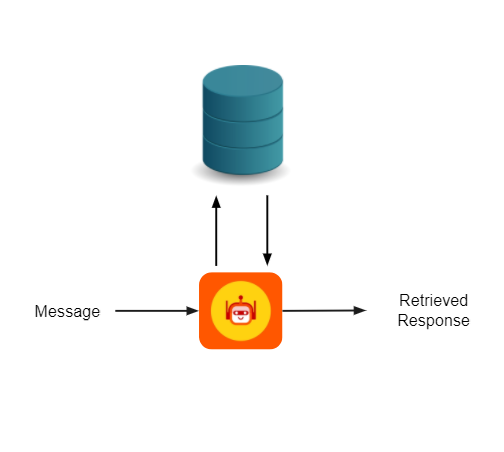
Retrieval based chatbots, employ techniques such as keyword matching, machine learning, and deep learning to find the most appropriate response.

These chatbots,regardless of technology, solely deliver predefined responses

and do not generate fresh output. From a database of predefined responses,

the chatbot is trained to offer the best possible response. The responses are

based on previously collected data.



**Fig 3.3 Retrieval Based**

Since the Generative Based Chatbots require huge amount of data to train

on and construct the appropriate response, we will be using Retrieval Based approach for our chatbot

**3.3** Natural Language Processing:

Natural language processing is a subfield of linguistics, computer science, and

artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large

amounts of natural language data.

We will be using a three-step process to transform our "content" into a form that could

be understood by a computer algorithm and with which it can extract meaningful

insights.



#### Fig 3.4 NLP

**3.4SYSTEM SPECIFICATION**

## 

## 3.4.1SOFTWARE REQUIREMENTS

Functional requirements for a secure cloud storage service are straight forward:

1. The service should be able to store the user’s data;
2. The data should be accessible through any devices connected to the Internet;
3. The service should be capable to synchronize the user’s data between multiple
4. devices (notebooks, smart phones, etc.);
5. The service should preserve all historical changes (versioning);
6. Data should be shareable with other users;
7. The service should support SSO; and
8. The service should be interoperable with other cloud storage services, enabling
9. data migration from one CSP to another.
10. **Operating System:** Windows
11. **Coding Language:** Python 3.7

## 3.4.2 HARDWARE REQUIREMENTS

1. **Processor** – Intel core i5
2. **Speed –** 1.19 GHz
3. **RAM -** 512 MB (min)
4. **Hard Disk** - 20 GB
5. **Key Board -** Standard Keyboard
6. **Monitor –** 15 VGA Colour

# CHAPTER 4 DESIGN

## UNIFIED MODELING LANGUAGE

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it’s one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: “a picture is worth a thousand words”. By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

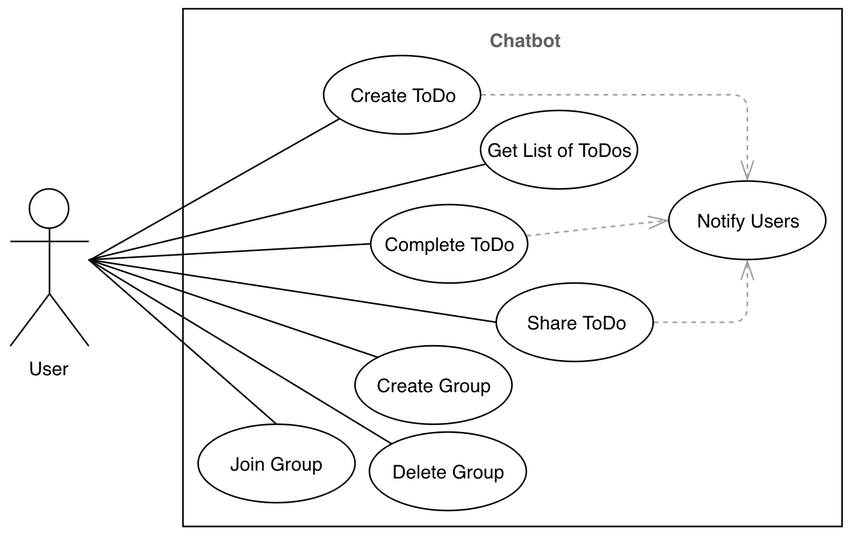
UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that
2. they can develop and exchange meaningful models.
3. Provide extend ability and specialization mechanisms to extend the core concepts.
4. Be independent of particular programming languages and development process.
5. Provide a formal basis for understanding the modeling language.
6. Encourage the growth of OO tools market.

## USE CASE DIAGRAM

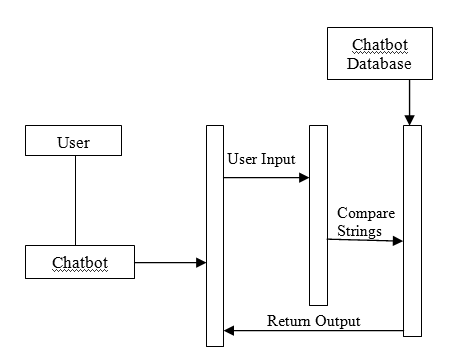
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



## Fig 4.1 Use case Diagram

* + 1. **SEQUENCE DIAGRAM**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



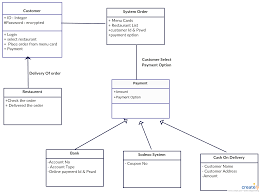
## Fig 4.2 Sequence diagram

* + 1. **CLASS DIAGRAM**

In software engineering, a class diagram in the Unified Modeling Language (UML) I

s a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationhips among

the classes. It explans which class contains information



## Fig 4.3 Class diagram

## Fig 4.3 Class diagram

* + 1. **COMPONENT DIAGRAM**

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus,from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

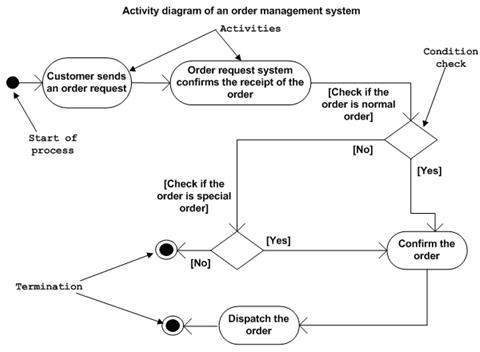
UML Component diagrams are used in modeling the physical aspects of object oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

### [IMG_256](https://in.pinterest.com/pin/158118636905483587/)

## Fig 4.4 Component diagram

* + 1. **ACTIVITY DIAGRAM**

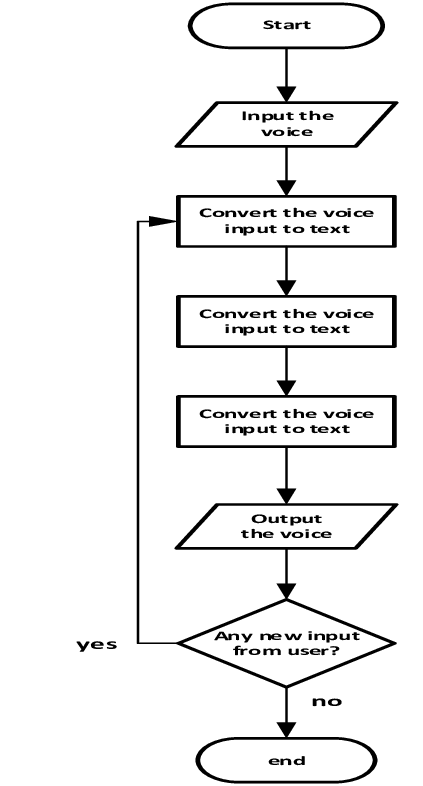
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



## Fig 4.5 Activity diagram

* + 1. **FLOW CHART DIAGRAM**

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.



**Fig 4.6 Flow chart diagram**

# CHAPTER 5 SUMMARY AND CONCLUSION

At the beginning of the project, User Stories are best suited for the understanding and

getting clarity for the complicated requirements. Incase of Rule based Bots, Flow charts

are helpful in getting the entire workflow on the chart. When it comes to NLP based

Bots, it required unconventional method to prepare requirements considering Intents and Entities.However based on size of the projects and time frame, best approach should be selected.Happy Chatting !!!

## REFERENCES

1. Alepis, E., & Virvou, M. (2011). Automatic generation of emotions in tutoring agents

for affective e-learning in medical education. Expert Systems with Applications, 38(8):

9840–9847.

1. Ashok, G., Brian, C., Mithun, K., Shanu, S., Abhinaya, S., & Bryan, W. (2015). Using

Watson for Enhancing Human-Computer Co-Creativity. AAAI Symposium: 22–29.

1. Avalverde, D. (2019). A Brief History of Chatbots. Perception, Control, Cognition.

Retrieved March 9, 2019 from: [https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/](https://www.google.com/url?q=https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/&sa=D&sntz=1&usg=AOvVaw35Eko-ZvjVSDGKzCqCYeCX" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank)

1. Ayedoun, E., Hayashi, Y., & Seta, K. (2015). A Conversational Agent to Encour

age Willingness to Communicate in the Context of English as a Foreign Language.

Procedia Computer Science, 60(1): 1433–1442.

1. Ben Mimoun, Mohammed Slim, & Poncin, I. (2015). A valued agent: How ECAs

affect website customers' satisfaction and behaviors. Journal of Retailing and Consumer Services, 26: 70– 82.

1. Chatbot Magazine (2019). A Visual History of Chatbots. Retrieved March 9, 2019

from: [https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2](https://www.google.com/url?q=https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2&sa=D&sntz=1&usg=AOvVaw348XQu7kw_PZMselCnkUmo" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank)

1. Colace, F., De Santo, M., Lombardi, M., Pascale, L., Pietrosanto, A. (2018). Chatbot for
2. Learning: A Cases Study. International Journal of Mechanical Engineering and Robotics Research Vol. 7, No. 5, September.
3. Egencia (2018). What is a Chatbot and How does it work? Retrieved March 9, 2019

from: [https://www.youtube.com/watch?v=38sL6pADCog](https://www.youtube.com/watch?v=38sL6pADCog" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank)

1. Hattie, J. (2012). Visible learning for teachers: Maximizing impact on learning:

Routledge. [https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2](https://www.google.com/url?q=https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2&sa=D&sntz=1&usg=AOvVaw348XQu7kw_PZMselCnkUmo" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank)

1. Lip ko, H. (2018). Meet Jill Watson: Georgia Tech's first AI teaching assistant.

Retrieved on March 9, 2019 from: [https://pe.gatech.edu/blog/meet-jill-watson-georgia-techs-first-ai-teaching-assistant](https://www.google.com/url?q=https://pe.gatech.edu/blog/meet-jill-watson-georgia-techs-first-ai-teaching-assistant&sa=D&sntz=1&usg=AOvVaw0PtCZ9_cPkBtkHRT1nLGbh" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank).

1. Maruti Techlabs. (2018). Why can chatbots replace Mobile Apps immediately?

Retrieved March 9, 2019 from: [https://www.marutitech.com/why-can-chatbots-replace-mobile-apps-immediately/](https://www.google.com/url?q=https://www.marutitech.com/why-can-chatbots-replace-mobile-apps-immediately/&sa=D&sntz=1&usg=AOvVaw1HL8N9pYWXUcIrzgnBJdRn" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank)

1. [Nguyen](https://www.google.com/url?q=https://www.businessinsider.com/author/mai-hanh-nguyen&sa=D&sntz=1&usg=AOvVaw0u1T5Ilc4EosCDTg4b4z6i" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank), M. (2017). How artificial intelligence & machine learning produced robots

we can talk to. Business Insider. Retrieved March 9, 2019 from: [https://www.businessinsider.com/what-is-chatbot-talking-ai-robot-chat-simulators-2017-10](https://www.google.com/url?q=https://www.businessinsider.com/what-is-chatbot-talking-ai-robot-chat-simulators-2017-10&sa=D&sntz=1&usg=AOvVaw38ei-h_0y94DMrnf-iIiHB" \t "https://sites.google.com/view/chatbots-in-mobile-learning/_blank)

## APPENDIX

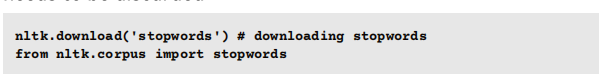
## SCREENSHOTS

# 

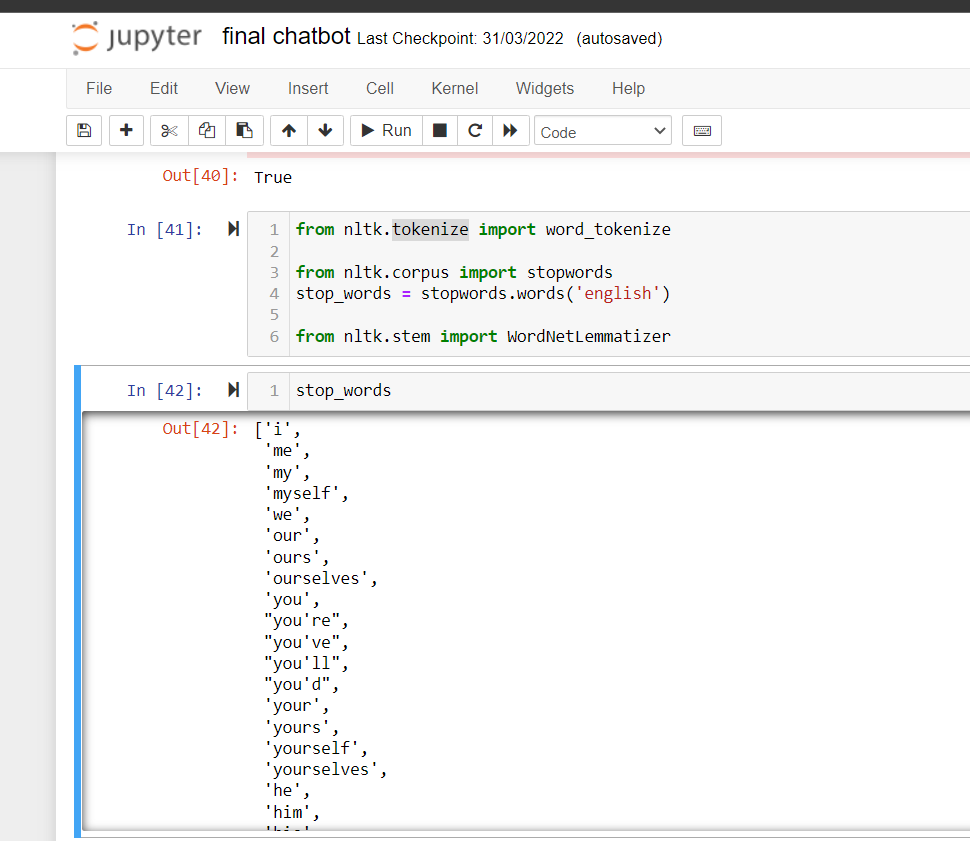
# **Fig 5.1 Importing libraries and downloading packages**



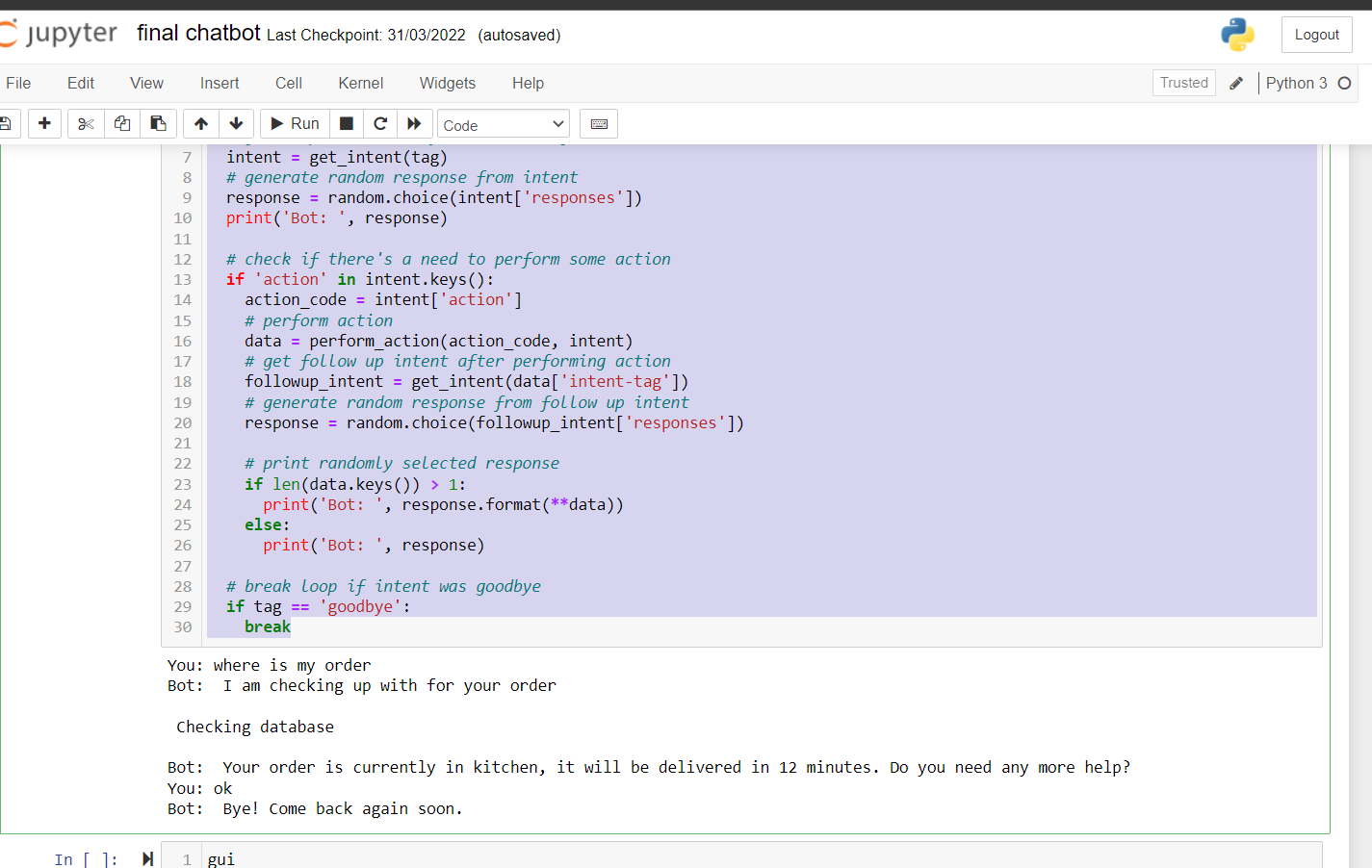
#### Fig 5.2 Tokenization



# Fig 5.3 Stop Words

****

#### Fig 5.4 Tokenization and stop words



**Fig 5.5 complete chatbot**

## SOURCE CODE:

import nltk

import numpy as np

# downloading model to tokenize message

nltk.download('punkt')

# downloading stopwords

nltk.download('stopwords')

# downloading wordnet, which contains all lemmas of english language

nltk.download('wordnet')

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

stop\_words = stopwords.words('english')

from nltk.stem import WordNetLemmatizer

#function to clean Text

def clean\_corpus(corpus):

# lowering every word in text

corpus = [ doc.lower() for doc in corpus]

cleaned\_corpus = []

stop\_words = stopwords.words('english')

wordnet\_lemmatizer = WordNetLemmatizer()

# iterating over every text[a,b,c]='a b c'

for doc in corpus:

# tokenizing text

tokens = word\_tokenize(doc)

cleaned\_sentence = []

for token in tokens:

# removing stopwords, and punctuation

if token not in stop\_words and token.isalpha():

# applying lemmatization

cleaned\_sentence.append(wordnet\_lemmatizer.lemmatize(token))

cleaned\_corpus.append(' '.join(cleaned\_sentence))

return cleaned\_corpus

#Loading and cleaning intent

!wget -O intents.jsonn <https://techlearn-cdn.s3.amazonaws.com/bs_swiggy_chatbot/intent.json>

import json

with open('intents.jsonn', 'r') as file:

intents = json.load(file)

corpus = []

tags = []

for intent in intents['intents']:

# taking all patterns in intents to train a neural network

for pattern in intent['patterns']:

corpus.append(pattern)

tags.append(intent['tag'])

cleaned\_corpus = clean\_corpus(corpus)

cleaned\_corpus

# #Vectorizing intents

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(cleaned\_corpus)

from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder()

y = encoder.fit\_transform(np.array(tags).reshape(-1,1))

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Dense, Dropout

model = Sequential([

Dense(128, input\_shape=(X.shape[1],), activation='relu'),

Dropout(0.2),

Dense(64, activation='relu'),

Dropout(0.2),

Dense(y.shape[1], activation='softmax')

])

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()

history = model.fit(X.toarray(), y.toarray(), epochs=20, batch\_size=1)

# if prediction for every tag is low, then we want to classify that message as no answer

INTENT\_NOT\_FOUND\_THRESHOLD = 0.40

def predict\_intent\_tag(message):

message = clean\_corpus([message])

X\_test = vectorizer.transform(message)

#print(message)

#print(X\_test.toarray())

y = model.predict(X\_test.toarray())

#print (y)

# if probability of all intent is low, classify it as noanswer

if y.max() < INTENT\_NOT\_FOUND\_THRESHOLD:

return 'noanswer'

prediction = np.zeros\_like(y[0])

prediction[y.argmax()] = 1

tag = encoder.inverse\_transform([prediction])[0][0]

return tag

print(predict\_intent\_tag('How you could help me?'))

print(predict\_intent\_tag('swiggy chat bot'))

print(predict\_intent\_tag('Where\'s my order'))

import random

import time

def get\_intent(tag):

# to return complete intent from intent tag

for intent in intents['intents']:

if intent['tag'] == tag:

return intent

def perform\_action(action\_code, intent):

# funition to perform an action which is required by intent

if action\_code == 'CHECK\_ORDER\_STATUS':

print('\n Checking database \n')

time.sleep(2)

order\_status = ['in kitchen', 'with delivery executive']

delivery\_time = []

return {'intent-tag':intent['next-intent-tag'][0],

'order\_status': random.choice(order\_status),

'delivery\_time': random.randint(10, 30)}

elif action\_code == 'ORDER\_CANCEL\_CONFIRMATION':

ch = input('BOT: Do you want to continue (Y/n) ?')

if ch == 'y' or ch == 'Y':

choice = 0

else:

choice = 1

return {'intent-tag':intent['next-intent-tag'][choice]}

elif action\_code == 'ADD\_DELIVERY\_INSTRUCTIONS':

instructions = input('Your Instructions: ')

return {'intent-tag':intent['next-intent-tag'][0]}

while True:

# get message from user

message = input('You: ')

# predict intent tag using trained neural network

tag = predict\_intent\_tag(message)

# get complete intent from intent tag

intent = get\_intent(tag)

# generate random response from intent

response = random.choice(intent['responses'])

print('Bot: ', response)

# check if there's a need to perform some action

if 'action' in intent.keys():

action\_code = intent['action']

# perform action

data = perform\_action(action\_code, intent)

# get follow up intent after performing action

followup\_intent = get\_intent(data['intent-tag'])

# generate random response from follow up intent

response = random.choice(followup\_intent['responses'])

# print randomly selected response

if len(data.keys()) > 1:

print('Bot: ', response.format(\*\*data))

else:

print('Bot: ', response)

# break loop if intent was goodbye

if tag == 'goodbye':

break