

**University Institute of Information Technology,**

**PMAS-Arid Agriculture University,**

**Rawalpindi Pakistan**

**Project Name**

***By***

**Sheharyar Kalim 17-ARID-1574**

**Natalia Nisar 17-ARID-1557**

**Faizan Khan 17-ARID-1466**

***Supervisor*Mr Zeeshan Javaid**

***Bachelor of Science in Computer Science (20xx-20xx)***

***OR Information Technology OR Software Engineering***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Student Name 1 Student Name 2 Student Name 3

Sheharyar Kalim Natalia Nisar Faizan Khan

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS/IT/SE) “Deep Learning on Chatbots” was developed by “**Natalia Nisar 17-ARID-1557”**, “Sheharyar Kalim 17-ARID-1574**”** and “**Faizan Khan 17-ARID-1466”** under the supervision of “Supervisor Name” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Science/Information Technology/Software Engineering.

Sir Zeeshan Javid

---------------------------------------

**Supervisor**

Maria Iqbal

----------------------------------------

**External Examiner (If any)**

---------------------------------------

**Administrator UIIT**

**Executive Summary**

In public places, there is often a need for monitoring people and different activities going on, which can be referred later for many reasons including security. Appointing humans for this task involves many problems such as increased employee hiring, accuracy problem, trust, no proof for later use, and also the fact that a human can remember things till a certain time limit. Talking about the current security system, they use dumb still cameras with a continuous recording facility ir-respective of the fact that any event may happen or not. Moreover they are usually pointing at a specific user defined locations so more than one cameras are required to cover the entire region.

To prevent all these problems from prevailing, the CSCS is developed. It is a surveillance system, which provides solution to many of these problems. It is a stand-alone application which doesn’t require any computer to operate. It monitors different situations using a camera which is able to rotate intelligently based on sensor messages and captures the scene in the form of video or photos later reference as well.

**C**ustomizable **S**urveillance **C**ontrol **S**ystem **(CSCS)** is a surveillance system that can be assigned a sensor type as in our case a heat sensor is used, it works accordingly, rotates the camera upon event detection and perform user defined actions like capturing video and stores them, for the future use.

It is an embedded system consisting of Linux fox kit with embedded a running server application also a camera, USB storage device and a sensor node base station is attached with fox kit. LAN communication is used by user to download the videos and to operate the system manually.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Sir Zeeshan Javaid” and our Co-Supervisor “Dr. Tariq Ali” for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him/her/them for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Student Name 1 Student Name 2 Student Name 3

Sheharyar Kalim Natalia Nisar Faizan Khan

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Requirement Specification |
| **PC** | Personal Computer |
|  |  |
|  |  |
|  |  |

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_Toc268523777) 2

[1.2 Relevance to Course Modules](#_Toc268523779) 2

[1.3 Project Background](#_Toc268523780) 3

[1.4 Literature Review](#_Toc268523780) 3

[1.5 Methodology and Software Life Cycle](#_Toc268523782) 5

**Problem Definition** 6

[2.1 Purpose](#_Toc268523795) 7

[2.2 Product Functions](#_Toc268523796) 7

[2.3 Proposed Architecture](#_Toc268523804) 7

[2.4 Project Deliverables](#_Toc268523806) 8

[2.5 Operating Environment](#_Toc268523807) 8

[2.6 Assumptions and Dependencies](#_Toc268523808) 8

**Requirement Analysis** 9

[3.1 Functional Requirments](#_Toc268523823) 10

[3.2 Non – Functional Requirments](#_Toc268523825) 11

[3.2.1 Usability](#_Toc268523787) 11

[3.2.2 Reliability](#_Toc268523787) 11

[3.2.3 Performance](#_Toc268523787) 11

[3.2.4 Supportability](#_Toc268523787) 11

[3.2.5 Design Constraints](#_Toc268523787) 11

[3.2.6 Licensing Requirements](#_Toc268523787) 11

[3.3 Use case Model](#_Toc268523823) 12

[3.3.1 Use Case Diagarm](#_Toc268523787) 12

[3.3.2 Actors Discription](#_Toc268523787) 16

[3.3.3 Use Case Discription](#_Toc268523787) 17

**The Design** 21

[4.1 UML Structural Diagrams](#_Toc268523830) 22

[4.1.1 Component Diagram](#_Toc268523787) 22

[4.1.2 System Component Diagram](#_Toc268523787) 24

[4.1.3 Package Diagram](#_Toc268523787) 25

[4.1.4 Deployment Diagram](#_Toc268523787) 26

[4.2 UML Behavioral Diagrams](#_Toc268523830) 27

[4.2.1 Activity Diagrams](#_Toc268523787) 27

[4.2.2 State Machine Diagrams](#_Toc268523787) 29

[4.3 UML Interaction Diagrams](#_Toc268523830) 30

[4.3.1 Sequence Diagrams](#_Toc268523787) 30

[4.4 Node Structure](#_Toc268523830) 31

[4.5 Communication Design Protocol](#_Toc268523830) 32

**Implementation** 33

[5.1 Communication Protocol Implementation](#_Toc268523830) 34

[5.2 PC Application Implementation](#_Toc268523830) 36

[5.3 Embedded Application Implementation](#_Toc268523830) 38

[5.4 Wireless Sensor Application Implementation](#_Toc268523830) 51

**Testing and Evaluation** 52

[6.1 Verification](#_Toc268523830) 53

[6.1.1 Functional Testing](#_Toc268523787) 53

[6.1.2 Static Testing](#_Toc268523787) 57

[6.2 Validation](#_Toc268523830) 57

[6.3 Usability Testing](#_Toc268523830) 57

[6.4 Unit Testing](#_Toc268523830) 57

* 1. [Integration Testing](#_Toc268523830) 57

6.6 [System Testing](#_Toc268523830) 57

**GUI** 58

**Future Work** 70

**References** 72

**List of Figures**

Fig 1.1 Block Diagram 8

Fig 2.1 Use Case Diagram 9

# Chapter 1: Introduction

# Brief

Chatbots are intelligent agents which have ability to understand the spoken language and use speech communication as user interface. In first generation of chatbots has limited capabilities it just allowing them to respond to simple rule-based queries; however, due to recent AI advancements and the abundance of available data, chatbots can now perform more complex tasks and even complete proactive transactions. Government agencies also start adopted AI techniques in different complex tasks in diverse domains, e.g. health, social welfare, public safety, taxation, and education. A recent report given by Harvard he identifies 6 types of governmental problems, for which AI applications are considered as highly suitable: resource allocation, large datasets, expert shortage, predictable scenarios, procedural and repetitive tasks, diverse data aggregation and summarization. There are lots of benefits. The reported applications of AI in government are mainly focused on customer (citizens) service support through bots giving responses to simple queries of citizens and providing relevant information. As a result from the above, issues related to the guidance of stakeholders through the space of available data and the indication of relevant information to facilitate and augment interaction between government and citizens, either for information seeking purposes for handling routine daily transactions, are of major importance.

# Relevance to Course Modules

We have not studied any course related to Deep Learning. As it is our new experience so no relation can be made right now.

# Project Background

The purpose of chatbots is to support and scale business teams in their relations with customers. Our project aims to provide a deep and detailed learning of chatbots that exist in this environment. Simultaneously, we will be providing the specifications of the existing chatbots and how they are being evolved as per the requirement of time and different companies. Our smart chat bot will communicate with people by typing or by speech. As there are many chatbots working but they can’t response to the commands which contain errors. Our smart chat bot will understand the command even if it is not totally correct. Our smart chatbot will take command from user in the form of speech or text, process the data using trained data set and communicate with the user. It is used to automate customer service processes and can settle their affairs easier and faster, companies can save time and costs redirecting people to more creative activities.

The dialog agent should understand what the user is trying to inquire and provide a function of comprehension. Chatbots are provided with input through texts which is then analyzed with NLP tools and thus provides the desired response. Chatbots are meant to help and deliver immediate actions where humans can’t reach due to timing or budget. Generally, chatbots are used for answering customer queries. Chatbots could be trained for answering simple questions of customers on their own like "location of the company", "career email id", etc. If the query of a customer is not generic then it can also be transferred to a human agent. Chatbots can answer the queries of customers but this couldn’t be achieved just by the employees, a system has to be built in order to cater these specific needs. Chatbots streamline interaction between people and services and enhances customer experience. It provides an opportunity to improve customer engagement. Companies can get real insight into what they’re costumers prefer, and therefore provide improved offers and answers to their users. There’s lots to be learned from most searched and used words in regards to what interests the users most. In a world where mobile and digital are first and musts it makes little sense for companies looking to succeed to no harness the benefits of deploying chatbots as a key tool to strengthen their relationships with their clients both internally and externally.

# Literature Review

There are many chatbots which available in market for different purposes, we will discuss few of them in brief here:

**Ensemble-based deep reinforcement learning for chatbots**

The approach for training Deep Reinforcement Learning (DRL) chatbots. It uses an ensemble of 100 DRL agents based on clustered dialogues, clustered actions, and rewards derived from human-human dialogues without any manual annotations. The task of the agents is to learn to choose human-like actions (sentences) out of candidate responses including human generated and randomly chosen sentences. Our ensemble trains specialised agents with particular dialogue strategies according to their dialogue clusters. At test time, the agent with the highest predicted reward is used during a dialogue. Experimental results using chitchat dialogue data report that DRL agents learn humanlike dialogue policies when tested on training data, but their generalisation ability in a test set of unseen dialogues (with mostly unseen sentences, only 4.62% seen sentences to be precise) remains a key challenge for future research in this field.

(Reference: Ensemble-based deep reinforcement learning for chatbots Heriberto Cuayáhuitl a,1,∗ , Donghyeon Lee b , Seonghan Ryub , Yongjin Cho b , Sungja Choi b , Satish Indurthi b , Seunghak Yub , Hyungtak Choi b , Inchul Hwang b , Jihie Kimb a University of Lincoln, School of Computer Science, Lincoln Centre for Autonomous Systems (L-CAS), Brayford Pool, Lincoln LN6 7TS, United Kingdom b Samsung Research, Artificial Intelligence Group, 56 Seongchon-gil, Yangjae, Seocho-gu, Seoul, South Korea)

**Deep Learning for Chatbots**

Deep Learning helps eliminate the complexity of feature engineering from the traditional machine learning process and truly learns the underlying data patterns. Architectures like RNN, LSTM and sequence to sequence model overcome the limitation of context recognition, an essential part of NLP. Generation-based Networks have it made possible to create a “true” chatbot whose responses are based only on the training data. These advances can be leveraged and explored to build life-like chatbots which can make real conversations with a human. By creating robust chatbots which can be customized according to the training data, large scale automation is possible. Finally, the responses that a chatbot makes are mostly a “good guess”. NLP is a vast field, and for machines to truly understand the complexity of human interaction would require coordination of many branches of science such as psychology, literature and linguistics.

(Deep Learning for Chatbots Vyas Ajay Bhagwat San Jose State University, A Thesis Presented to The Faculty of the Department of Computer Science San Jose State University, Dr. Robert Chun Department of Computer Science Dr. Katerina Potika Department of Computer Science Mr. Vinit Gaikwad Cisco, Inc.)

**An Overview of Machine Learning in Chatbots**

We used a mind-mapping approach to present an overview of chatbots, after reviewing papers published from 1998 to 2018. This can help researchers develop a better understanding of the current implementation techniques and usages of chatbots. This is important because chatbots are becoming increasingly popular, especially for customer service in the industry and as an intelligent virtual assistant for personal use. This paper outlines many machine learning techniques which could improve the performance of chatbots because they allow chatbots to learn and adapt through experience. Having the ability to improve itself with every interaction will likely improve the chatbot’s capability of understanding the content and context of the user’s input, which would help the chatbot generate a more accurate, relevant response. However, existing chatbots have a few limitations. The main challenge for a chatbot right now is understanding the context in a conversation and generating a relevant response. Hence, future intelligent chatbots should:

1) Implement improved natural language processing techniques to accurately recognize the content of the user input;

2) Learn to understand the context of conversations and respond accordingly with emotions or personalized content. The ultimate goal of chatbots is to replicate human-human interaction, which requires improved machine learning and natural language processing techniques.( An Overview of Machine Learning in Chatbots Prissadang Suta, Xi Lan, Biting Wu, Pornchai Mongkolnam and Jonathan H. Chan, International Journal of Mechanical Engineering and Robotics Research Vol. 9, No. 4, April 2020)

### **World Health Organization (WHO)**

First in our list of chatbots is The [World Health Organization (WHO)](https://www.who.int/). WHO is one of the leading sources of trusted information for the coronavirus (COVID-19) spread. WHO built [a bot on WhatsApp](https://www.who.int/news-room/feature-stories/detail/who-health-alert-brings-covid-19-facts-to-billions-via-whatsapp) called the WHO Health Alert to share information related to the pandemic. The WHO Health Alert bot helps people protect themselves from infection, offers travel advice, and busting myths about the Coronavirus. The service is initially launched in English, but reportedly will be available in all six United Nations languages within the coming weeks (English, Arabic, Chinese, French, Russian and Spanish.)

Reference: https://manychat.com/blog/chatbot-examples/

## **Helping Insomniacs Get through the Night**

If you suffer from insomnia, as I do, you’ll know that the feeling of almost suffocating loneliness – the idea that everyone else in the world is resting peacefully while your own mind betrays you with worries and doubts – is among the worst parts of not being able to sleep. Enter Casper’s amazingly named [Insomnobot 3000](http://insomnobot3000.com/) (which truly is one of the most tongue-in-cheek, retro-futuristic names for a chatbot I’ve ever come across), a conversational agent that aims to give insomniacs someone to talk to while the rest of the world rests easy. At this point, Insomnobot 3000 is a little rudimentary. As you can see in the screenshot above, the responses offered by the agent aren’t quite right – next stop, Uncanny Valley – but the bot does highlight how conversational agents can be used imaginatively. I’m not sure whether chatting with a bot would help me sleep, but at least it’d stop me from scrolling through the never-ending horrors of my Twitter timeline at 4 a.m.

Reference: https://www.wordstream.com/blog/ws/2017/10/04/chatbots

## **MedWhat: Making Medical Diagnoses Faster**

One of my favorite pastimes is radically misdiagnosing myself with life-threatening illnesses on medical websites (often in the wee hours of the night when I can’t sleep). If you’re the kind of person who has WebMD bookmarked for similar reasons, it might be worth checking out MedWhat. This chatbot aims to make medical diagnoses faster, easier, and more transparent for both patients and physicians – think of it like an intelligent version of WebMD that you can talk to. MedWhat is powered by a sophisticated machine learning system that offers increasingly accurate responses to user questions based on behaviors that it “learns” by interacting with human beings. In addition to the ever-growing range of medical questions fielded by MedWhat, the bot also draws upon vast volumes of medical research and peer-reviewed scientific papers to expand upon its already considerable wealth of medical expertise. In many ways, MedWhat is much closer to a virtual assistant (like Google Now) rather than a conversational agent. It also represents an exciting field of chatbot development that pairs intelligent NLP systems with machine learning technology to offer users an accurate and responsive experience.

Reference: https://www.wordstream.com/blog/ws/2017/10/04/chatbots

# Analysis from Literature Review (in the context of your project)

We have discussed the multiple bots in literature review and we have come to the conclusion that there is only one way of communication for the user. User can only give command in text or speech. We are proposing a system which will take the command by user in both ways text or speech.

# Methodology and Software Lifecycle for This Project

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. We will adopt agile methodology as it is being used worldwide. Changes can be made easily in the system by using this methodology. It is a process by which a team can manage a project by breaking it up into several stages and involving constant collaboration with stakeholders and continuous improvement and iteration at every stage.

* + 1. Rationale behind Selected Methodology

Agile Methodology includes iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. As changes can be made at any development stage, the clients can tell us their queries and what their expectations are from our chatbot. This will help us in providing the final deliverable according to the customers’ requirements. For instance, if the customer wants to change their query or ask various queries it could be done quite easily by adopting this methodology.

.

**Chapter 2: Problem Definition**

# Problem Statement

In this era of technology it is becoming challenge for companies or business industries to maintain their customer relations with them. As consumers are increasing day by day so it is becoming difficult for companies to handle a huge number of users for their customer support. They have to hire more staff in order to resolve customer complaints within time. In a result it costs them very high which increases their expenses.

In order to resolve these issues the developers has already proposed a solution named chatbot. **There are many chatbots to resolve these issues but there are some communication issues between user and chatbot. There accuracy level is not fulfilling the requirements. Client can only give commands in text or only speech which causes communication barriers. Some chatbots only understand formal language which causes misunderstanding.**

# Proposed Architecture

We are proposing a solution which will resolve the traditional problems facing by the companies or business enterprises. We are going to develop a system named Smart Chatbot. It will use NLP for language processing. We will train our system with data set to use it with NLP. Smart Chatbot will take input from the user in form of text or speech then process the input using NLP with the trained data set. Our system will process both text and speech given by the user. To make the system user friendly we will use tkinter for User Interface. We will further use at designer to make our UI attractive. Our system will also overcome slow user response. Immediate responses will be provided by the chatbot to redeem the need for the user*.*

# Deliverables and Development Requirements

* Desktop Application will be delivered
* Source Code
* System Design
* Large amount of data will be required.
* Hardware
* Software
* Training of dataset

# Operating Environment

**Hardware:** The refer hardware for our system is minimum Core i3 4th generation:

* Processor: 1 gigahertz (GHz) or faster
* RAM: 2 gigabyte GB (64-bit) or more
* Hard disk space: 20 GB (64-bit) or more

**Software:** In order to run our system on desktop Windows 8 will be the minimum software requirement.

# Assumptions and Dependencies

We are assuming that our client want to run the system on cross platform.

# Chapter 3: Requirement Analysis

Software Requirements Specification (SRS) report should be included in this chapter.

# Use Cases

Use cases are a widely used and highly regarded format for capturing requirements. Before writing functional requirement use cases can help you to understand the requirements in the way user expect. Following table presents you not only the template to write use case(s) as well as guides you to write each section with example.

|  |  |
| --- | --- |
| **Use Case ID:** | ARID-1466 |
| **Use Case Name:** | Type Message |
| **Actors:** | User. |
| **Description:** | The Use case will get the message and verify language. |
| **Trigger:** | Submission of message |
| **Preconditions:** | 1. **Language should be English** 2. **Roman English is not acceptable.** |
| **Postconditions:** | 1. After verification the answer will be generated. |
| **Normal Flow:** | 1. User will type the query. 2. System validates if the language is English. 3. System validates if the language is not Roman English. 4. System will look for answer. |
| **Alternative Flows:** | 2a: If the language of the message is not in English.   1. User will be asked to type in English Language. 2. User will type query in English   3a: If the language of the message is Roman English.   1. The User will be asked to use English vocabulary to develop better understanding. |
| **Exceptions:** | 2a: If the language of the message is not in English.   1. User will be asked to type in English Language. 2. User will type query in English   3a: If the language of the message is Roman English.   1. The User will be asked to use English vocabulary to develop better understanding. |
| **Includes:** | Included->Verify Language |
| **Special Requirements:** | Keypad. |
| **Assumptions:** | The Chatbot understands only English. |
| **Notes and Issues:** | 1. What is the maximum size of the message that a user can type 2. What if the user is not familiar with English language? |

|  |  |
| --- | --- |
| **Use Case ID:** | ARID-1574 |
| **Use Case Name:** | Retrieve answer |
| **Actors:** | Chatbot |
| **Description:** | The use case will search query in dataset |
| **Trigger:** | Submission of message |
| **Preconditions:** | 1. Language should be English 2. Roman English is not acceptable. |
| **Postconditions:** | 1. After processing an answer will be generated. |
| **Normal Flow:** | 1. Chatbot will search in dataset 2. System will retrieve answer. |
| **Exceptions:** | 2a: If the message do not match the data in dataset   1. System will ask the user to search another query related to context. |
| **Includes:** | Included-> Match dataset |
| **Special Requirements:** | Dataset and messege. |
| **Assumptions:** | The chatbot will retrieve answer from the context given. |
| **Notes and Issues:** | 1. What if the user come up with message that differs in context? |

|  |  |
| --- | --- |
| **Use Case ID:** | ARID-1557 |
| **Use Case Name:** | Display Answer |
| **Actors:** | Chatbot.  User. |
| **Description:** | The use case will display answer. |
| **Trigger:** |  |
| **Preconditions:** | Chat screen should be open.  User must understand English. |
| **Postconditions:** | Message will be displayed on screen. |
| **Normal Flow:** | Chatbot will display answer in the chat panel.  User can read the answer to the query |
| **Exceptions:** | If the message is out of context than the user will type message again |
| **Includes:** | No use case included |
| **Special Requirements:** | Mobile Phone. |
| **Assumptions:** | The chatbot will display the answer. |
| **Notes and Issues:** | 1. What if the user can’t understand English? |

# Functional Requirements

|  |  |
| --- | --- |
| **Functional Requirement No.** | **Functional Requirement Description** |
| FR1 | User will give command to system as input. |
| FR2 | System will respond to the user query efficiently without consuming time. |
| FR3 | System will take input in both formats (text, speech). |
| FR4 | System will only take input in English language. |
| FR5 | System will not understand the roman English language. |
| FR6 | System will provide a friendly user interface. |

# Non-Functional Requirements

|  |  |
| --- | --- |
| **Non-Functional Requirement No.** | **Non-Functional Requirement Description** |
| NFR1 | System will take input from user and process it by using NLP. |
| NFR2 | System will be trained by using data sets, which will be further used for Language processing purpose. |
| NFR3 | System will provide interface by using tkinter. |
| NFR4 | AI libraries will be used in python for language processing purpose. |

1. **Usability:** System should be easy to extend. The code should be written in a way that it favors implementation of new functions. It will provide the up to date information with good performance to satisfy user needs.
2. **Reliability:** This app should provide appropriate answers to the user. This app should be able to interact efficiently with the user.
3. **Integrity:** This desktop application will requires specific android version to run. It also requires an active internet connection to work and to exchange queries to provide information to the user.
4. **Performance:** This section specifies any numerical / statistical requirements imposed on the software such as:

* Two terminals need to be supported, one for the python portion of the bot, the other terminal to run the SQL server.
* Accessed by multiple users at the same time. Currently unaware of the actual limit.

1. **Licensing Requirements:** The Windows 8 should be licensed.

**3.4. Use Case:**

**Diagram

Description automatically generated**

# Use Cases and Description:

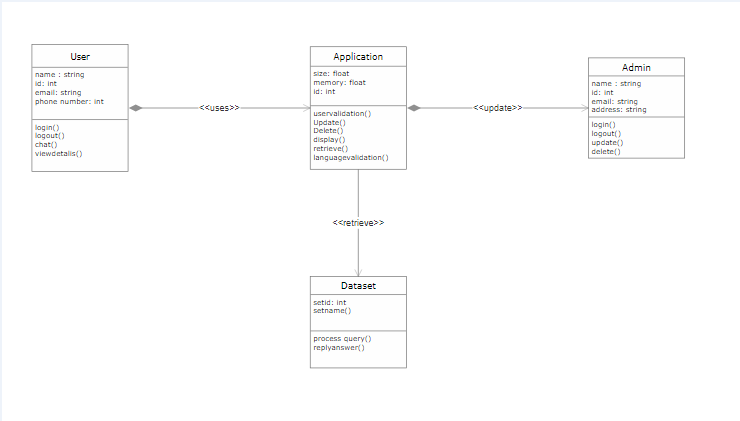
* **Query:** Query is the question that the client will ask through the chatbot.
* **Resolved Query:** System will analyze the query and get back to the client.
* **Retrieve:** System will retrieve command given by the user either in the form of text or voice.
* **Validate:** System will validate the query through NLP and get back to the user.
* **Update Dataset:** Admin will update the dataset to provide the basic knowledge about chatbot.
* **Resolve Query:** The system will resolve the query and return or get back the answer to the client.

# Chapter 4: Design and Architecture

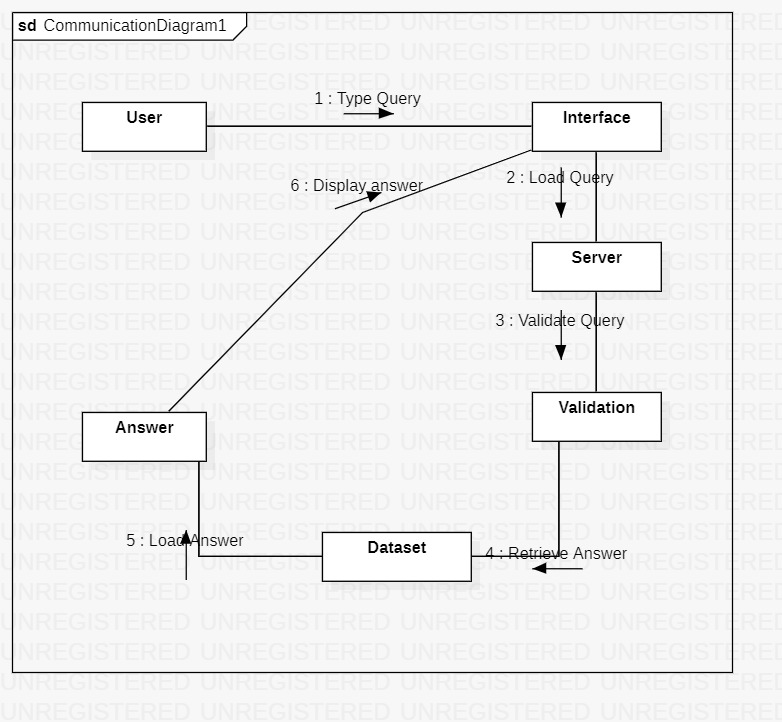
This chapter will discuss the design and architecture of your system.

# System Architecture

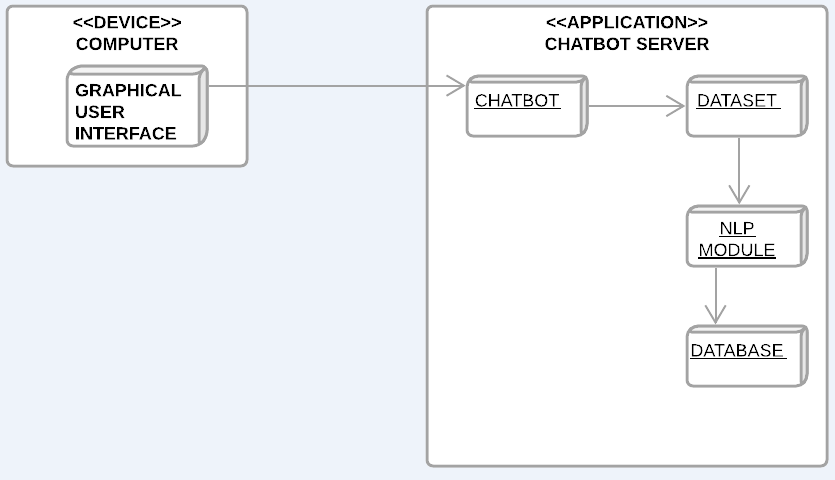
**Class Diagram:**



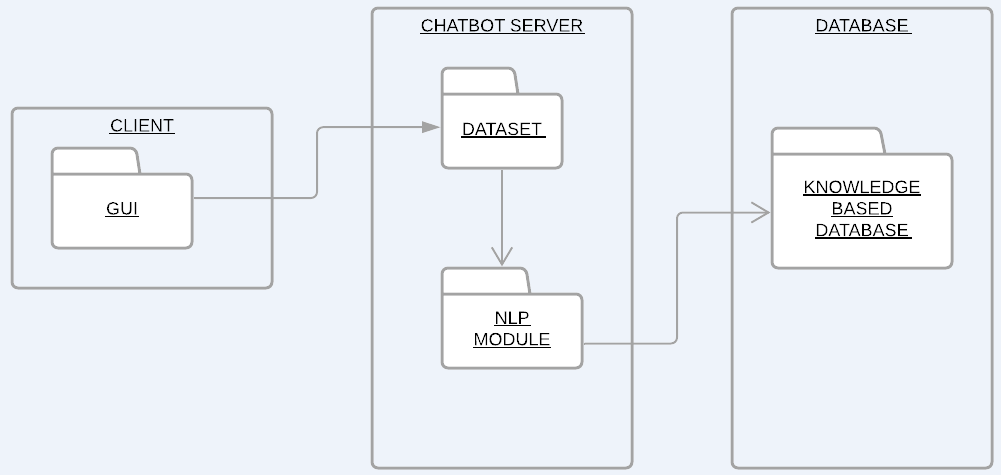
**Communication Protocol Diagram:**



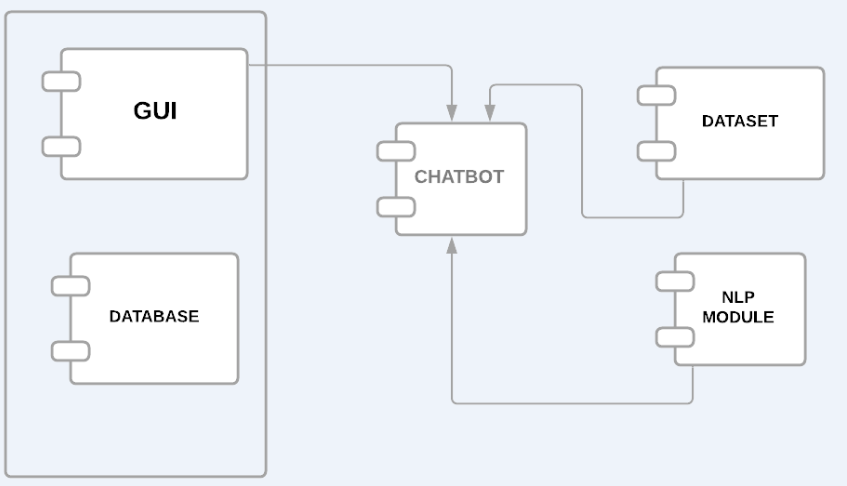
**Deployement Diagram:**



**Package Diagram:**



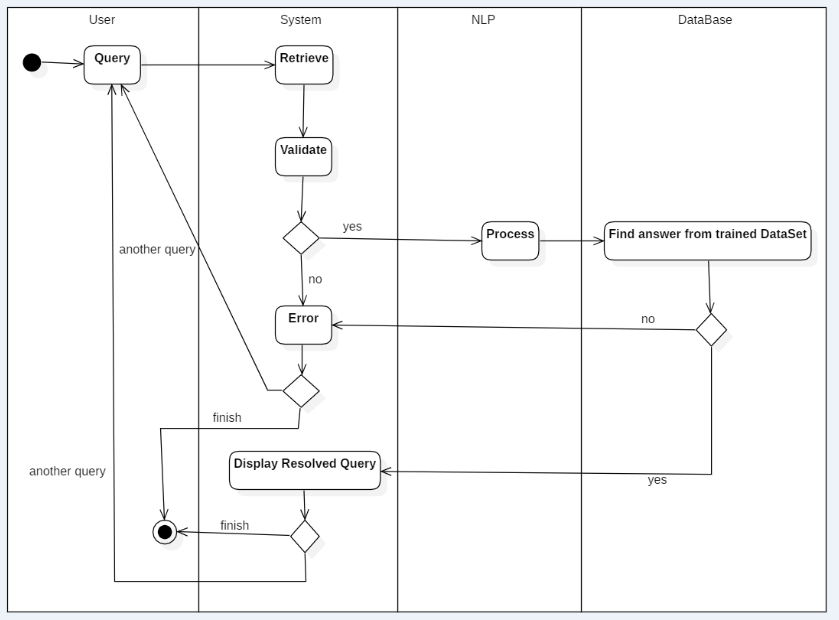
**Component Diagram:**



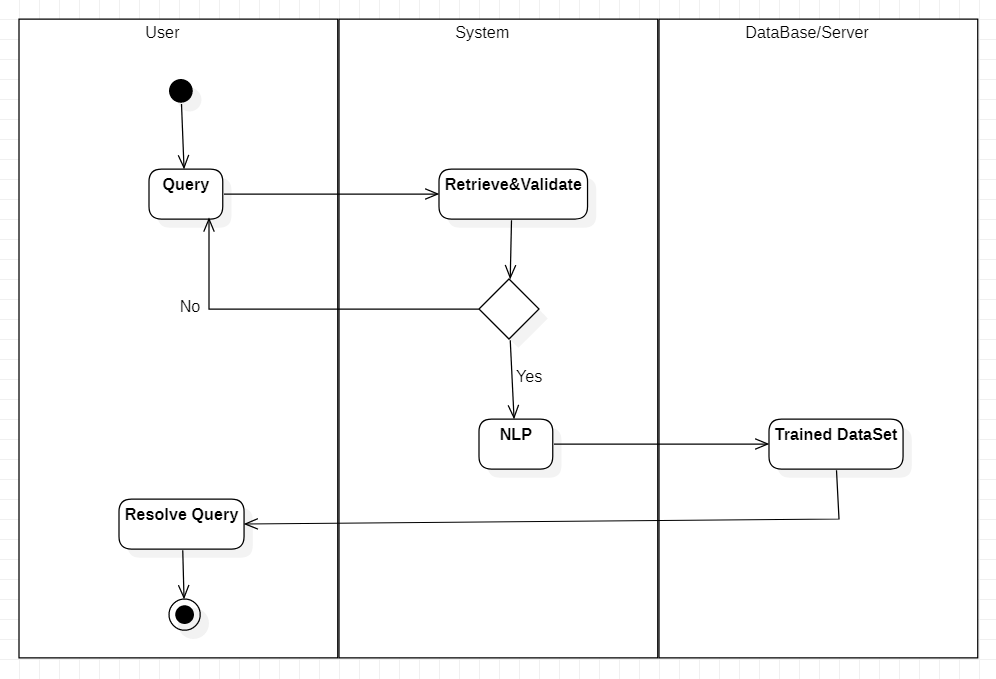
# System Design

**Activity Diagram:**

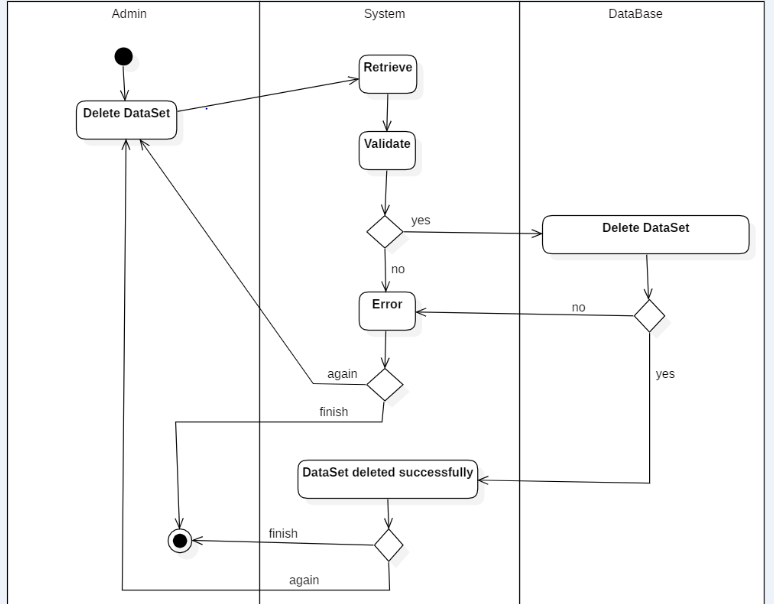
**QUERY:**



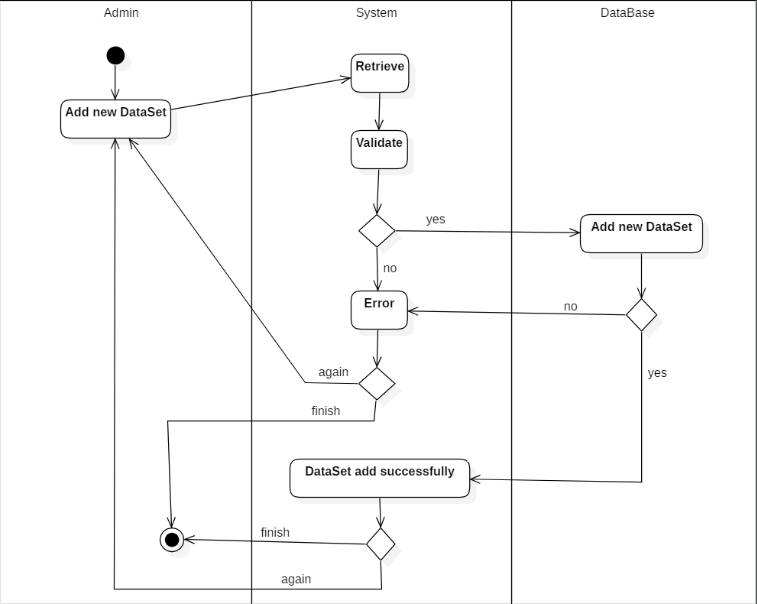
**RESOLVE QUERY:**



**DELETE DATASET:**

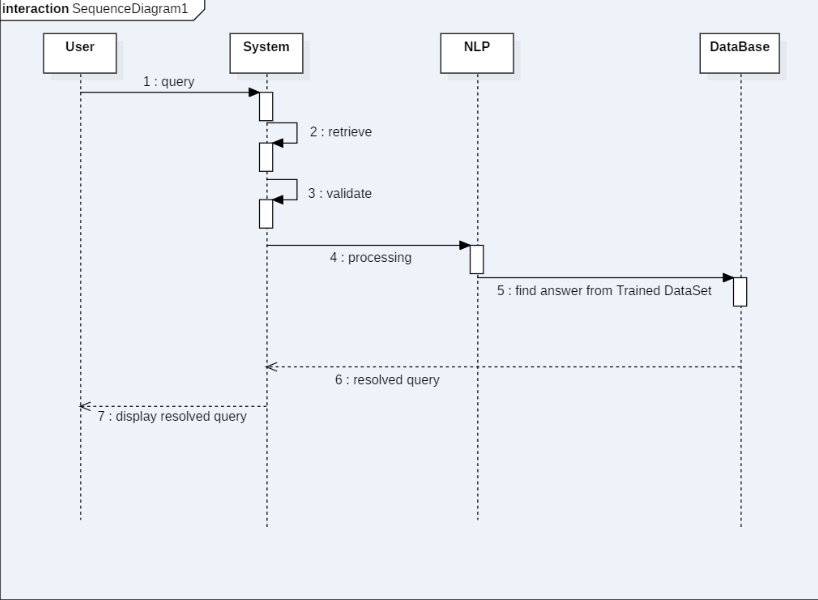


**ADD DATASET:**

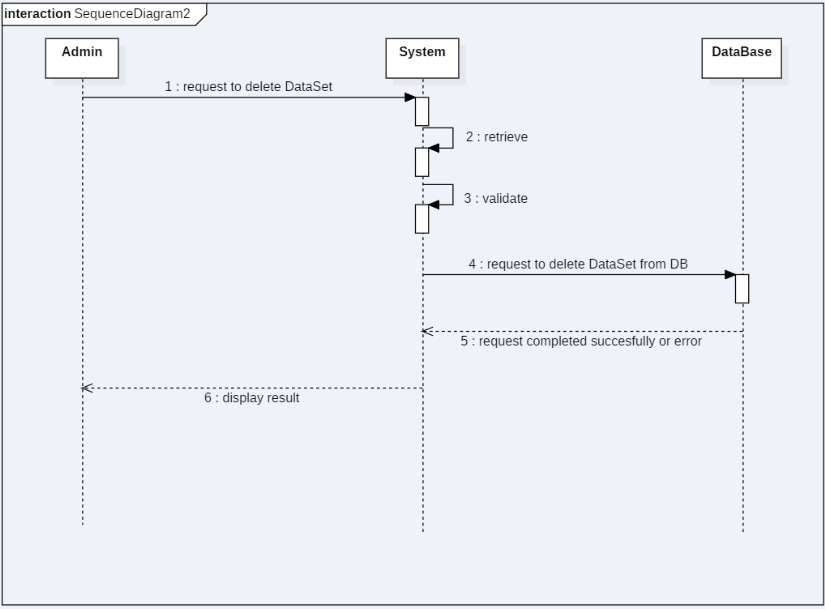


**Sequence Diagram:**

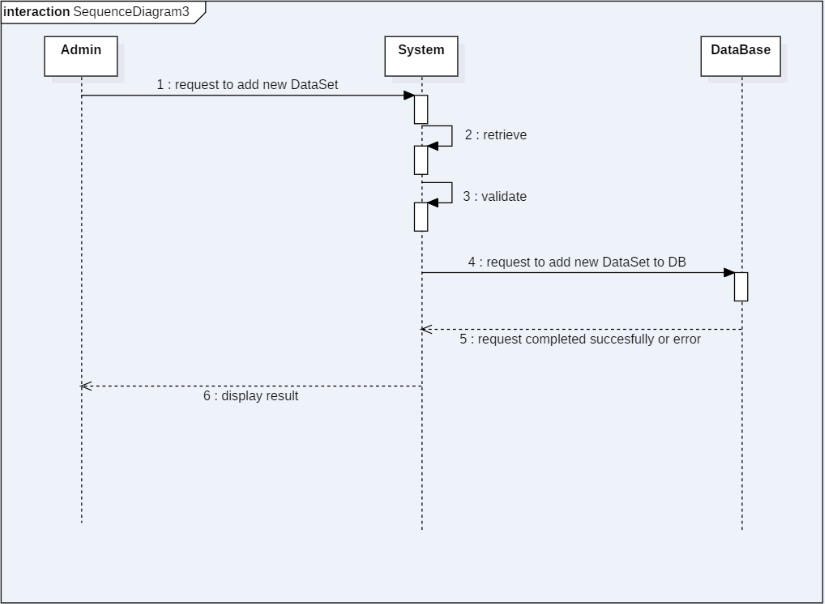
**QUERY:**



**RESOLVE QUERY:**



**ADD DATASET:**



# Chapter 5: Implementation

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

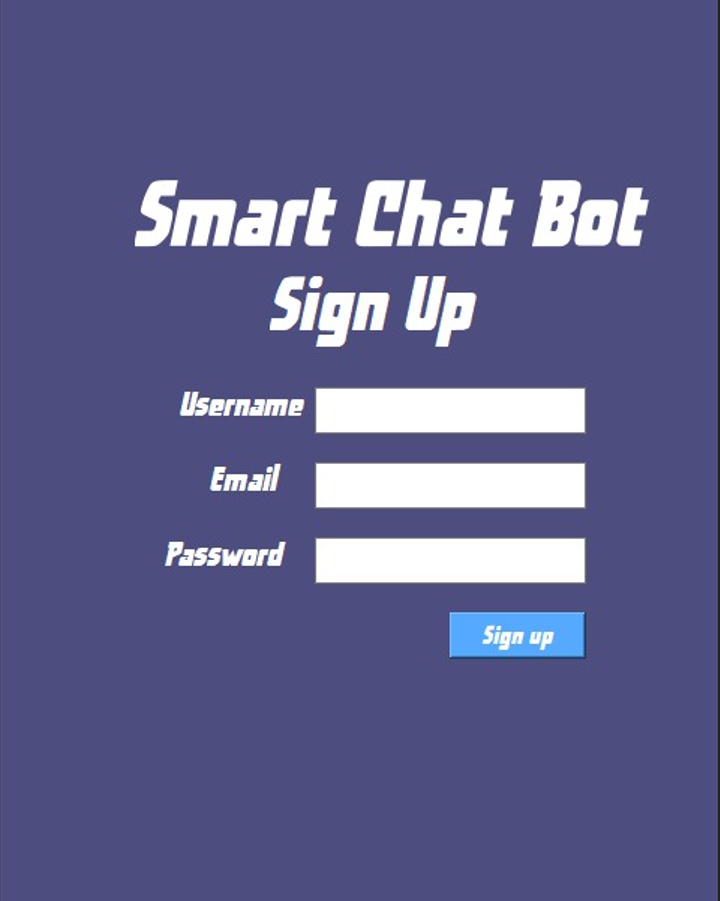
# Libraries and Design:

* Speech Recognition: Speech recognition is one of the most important tasks in the domain of human computer interaction. It allows the software to understand the human voice.
* nlp
* MySQL: It is a database system that runs through web.
* **NLTK** (Natural Language Tool Kit): NLTL is an open-source series of programs and libraries for developing programs in python languages.

NLTK is also referred to as the Natural Language Tool Kit. NLTK offers an easy-to-use interface with diverse corpora and lexical resources, like WordNet, with a series of text processing libraries for tokenization, tagging, semantic reasoning, parsing, stemming, and classification.

* **pyQt5:** PyQt5 is cross-platform GUI toolkit, a set of python bindings for Qt v5. One can develop an interactive desktop application with so much ease because of the tools and simplicity provided by this library.
* **Pyttsx3:** Pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.
* **ChatterBot:** ChatterBot is a machine-learning based conversational dialog engine build in Python which makes it possible to generate responses based on collections of known conversations. The language independent design of ChatterBot allows it to be trained to speak any language.
* **Threading:** Thread, in this module, nicely encapsulates threads, providing a clean interface to work with them. When you create a Thread, you pass it a function and a list containing the arguments to that function.
* TKinter: TKinter is the standard GUI library for Python. Python when combined with TKinter provides a fast and easy way to create GUI applications.

# User Interface



A picture containing text, toiletry

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated with low confidence

Shape, rectangle

Description automatically generated

**Chapter 6: Testing and Evaluation**

Testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided its design and development.

# Verification

We are going to develop a system named Smart Chatbot. It will use NLP for language processing. We will train our system with data set to use it with NLP. Smart Chatbot will take input from the user in form of text or speech then process the input using NLP with the trained data set. Our system will process both text and speech given by the user. To make the system user friendly we will use TKinter and Designer for User Interface. We will further use at designer to make our UI attractive.

# Validation

From the customer requirement perspective, the study of design code of different software modules and also from Graphic user interfaces through which the interacts with the system. Test-Case specifications in performed for system testing by keeping in mind several issues, which are discuss in following subtopics.

# Usability Testing

# Module / Unit Testing

Unit testing.

### 

### **Test Case Name: Send Query**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Deep Leaning Based Chatbot** | | | | | | |
| **Test Case 1** | | | | | | |
| **Test Case ID:** ARID-1466 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | | **Test Designed Date :** | | |
| **Module Name:** Send Query | | | |  | | |
| **Test Title:** Test Send Query | | | | **Test Execution Date :** | | |
| **Description** Chatbot will get the query & verify language. | | | |  | | |
| **Pre-Condition:**   * Language should be English * Roman English is not acceptable | | | |  | | |
| **Dependencies:** User must message | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1- |  |  | After Verification the answer will be generated. | Verification of language has been completed. | Pass |  |

### **Test Case Name: Retrieve Answer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Deep Leaning Based Chatbot** | | | | | | |
| **Test Case 2** | | | | | | |
| **Test Case ID:** ARID-1574 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | | **Test Designed Date :** | | |
| **Module Name:** Retrieve Answer | | | |  | | |
| **Test Title:** Test Retrieve Answer | | | | **Test Execution Date :** | | |
| **Description** Chatbot will search for query in dataset. | | | |  | | |
| **Pre-Condition:**   * Language should be English * Roman English is not acceptable | | | |  | | |
| **Dependencies:** User must message | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1- | 2 | Large Data | After searching the query, answer will be generated. | Answer has been generated. | Pass |  |

### **Test Case Name: Display Answer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name: Deep Leaning Based Chatbot** | | | | | | |
| **Test Case 3** | | | | | | |
| **Test Case ID:** ARID-1557 | | | |  | | |
| **Test Priority (Low/Medium/High):** High | | | | **Test Designed Date :** | | |
| **Module Name:** Display Answer | | | |  | | |
| **Test Title:** Test Display Answer | | | | **Test Execution Date :** | | |
| **Description** Chatbot will display answer. | | | |  | | |
| **Pre-Condition:**   * Chatting panel must be open * User must understand English | | | |  | | |
| **Dependencies:** User must message | | | |  | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1- |  |  | Message will be displayed on the screen. | Message has been Displayed. | Pass |  |

# Integration Testing

Integration testing.

# System Testing

All the modules will be tested for their properly functionality, to see the efficient working of the whole software.

# Acceptance Testing

Acceptance testing.

# Stress Testing

The software would be tested with large datasets to see if it would be working effectively under large sums of data.

# Hardware Configuration for Testing

The preferred hardware for our system is minimum Core i3 4th generation:

* Processor: 1 gigahertz (GHz) or faster
* RAM: 2 gigabyte GB (64-bit) or more
* Hard disk space: 20 GB (64-bit) or more

# Evaluation

The chatbot would be evaluated to confirm that it’s different modules are working properly and the chatbot is functional.

.

Chapter 7: Conclusion and Future Work

This chapter concludes the project and highlights future work.

# Conclusion

Our FYP project is about developing a SMART Chatbot, which would allow the user to interact and ask questions.

Our main focus was to create a chatbot based on deep learning, which would automatically store the data of the datasets and the queries asked by user. Till now we are able to create a self-trained chatbot, which can function well in audio and text input. The user can give audio input as well as text input, to which the chatbot would response effectively. Till now the chatbot is working efficiently. Datasets are being uploaded through which the chatbot would be further trained for full functionality.

# Future Work

Future working requires the chatbot to be trained through various datasets. We will improve the GUI (frontend) and backend for efficient working.

# References

References to any book, journal paper or website should properly be acknowledged. Please consistently follow the style. The following are few examples of different resources i.e. journal article, book, and website.

1 Lyda M.S. Lau, Jayne Curson, Richard Drew, Peter Dew and Christine Leigh, (1999), Use Of VSP Resource Rooms to Support Group Work in a Learning Environment, ACM 99, pp-2. (Journal paper example)

2 Hideyuki Nakanishi, Chikara Yoshida, Toshikazu Nishmora and TuruIshada, (1996), FreeWalk: Supporting Casual Meetings in a Network, pp 308-314 (paper on web) http://www.acm.org/pubs/articles/proceedings/cscw/240080/p308-nakanishi.pdf

3 Ali Behforooz& Frederick J.Hudson, (1996), Software Engineering Fundamentals, Oxford University Press. Chapter 8, pp255-235. (book reference example)

4 Page Author, Page Title, http://www.bt.com/bttj/archive.htm, Last date accessed. (web site)