
**SOFTWARE REQUIREMENT SPECIFICATION
FOR
Artificial Intelligence Based
CHAT-BOT
[05-07-2019]
CDAC, MUMBAI**

Revision History:

Versio n.	DATE	Author ed By	Review ed By	REASON FOR CHANGE
00	05/07/201 9	Team- 08		1 st release

Table of Contents

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Document Conventions.....	3
1.3 Intended Audience and Reading Suggestions.....	4
1.4 Product Scope.....	4
1.5 References.....	4
2. Overall Description.....	4
2.1 Product Perspective.....	4
2.2 Product Functions.....	5
2.3 User Classes and Characteristics.....	5
2.4 Operating Environment.....	5
2.5 Design and Implementation Constraints.....	6
2.6 User Documentation.....	6
2.7 Assumptions and Dependencies.....	6
3. External Interface Requirements.....	7
3.1 User Interfaces.....	7
3.2 Hardware Interfaces.....	7
3.3 Software Interfaces.....	7
3.4 Communications Interfaces.....	7
4. System Features.....	8
4.1 Description.....	8
4.2 Functional requirements.....	8
5. Other Non-functional Requirements.....	9
5.1 Performance Requirements.....	9
5.2 Safety Requirements.....	9
5.3 Security Requirements.....	9
5.4 Software Quality Attribute.....	9
6. Other Architecture.....	10
7. Other Requirements.....	11

1. Introduction

1.1 Purpose

This software design document describes the architecture and system design of Enquiry Chatbot, A question answering system for the CDAC Courses. So our project focuses on record, resolve and responds to user requests.

1.2 Document Convention

1.3 Headings:-

Text: -Bold

Font-Size: - 14

Highlighting: - Times New Roman

Sub Headings: -

Text: -Bold

Font-Size: - 12

Highlighting: - Times New Roman

Header: -

Text: - Simple

Font-Size: -10

Highlighting: - Times New Roman

Footer: -

Text: - Simple

Font-Size: -10

Highlighting: - Times New Roman

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, users, testers and project managers for the purpose of understanding the design of the system in terms of different perspectives. Further, this document contains functionalities and characteristics of the system along with the working environment. It also includes other information related to the system such as external interface requirements, features, and other non - functional requirements.

1.4 Product Scope

Artificial Intelligence Based CHAT-BOT (a natural language query service) is an AI chatbot that receives questions from users, tries to understand the question, and provides appropriate answers. It does this by converting an English sentence into a machine-friendly query, looking up in the database for the necessary information to answer the question, and finally returning the answer in a natural language sentence. The main objective is to develop a Web API that provides such natural language query service. In addition, sample web, mobile, and text messaging interfaces will be created to demonstrate the use of the API. The goal is to provide the customer with a quick and easy way to have their questions answered, as well as to offer other developers the means to incorporate Artificial Intelligence Based CHAT-BOT into their projects.

1.5 References

<https://getbootstrap.com/>

<https://stackoverflow.com/>

https://www.w3schools.com/html/html_css.asp

<https://rasa.com/>

<https://github.com/RasaHQ>

2. Overall Description

2.1 Product Perspective

Most of the search engines today, like Google, use a system (The Pagerank Algorithm) to rank different web pages. When a user enters a query, the query is interpreted as keywords and the system returns a list of highest ranked web pages which may have the answer to the query. Then the user must go through the list of webpages to find the answer they are looking for. Artificial Intelligence Based CHAT-BOT, however, will try to understand the query and provide a definitive answer.

2.2 Product Functions

- **Web API:** an API call will include a question in the form of a query string URL parameter and the service will reply in JSON. The Administrator will be able to resolve the complaint.
- **Natural Language Processing:** the system will take in questions written in Standard English.
- **Natural Language Responses:** the answer to the question will be written in standard and understandable English.
- **Information Extraction:** there will be a database containing all the information needed, populated using information extraction techniques.

2.3 User Classes and Characteristics

Admin:

The Admin will be able to access the list of queries received and will be able to create the proper generic response for that. The admin will maintain a log of queries that are resolved and the future possible queries that could be generated from it.

User:

They consist of non-technical users who want to get answers to their questions. These users ask questions and get answers with mobile, web, or text messaging interfaces. This class of users includes customers interested in the services provided by the car showroom.

2.4 Operating Environment

❖ Hardware platform:

- Processor - Above Pentium 4, with a clock speed of 2.0 GHz
- RAM - 1 GB or above
- Free disk space of above 1 GB

❖ Software Platform:

- **Front-End:** HTML, CSS, Bootstrap.
- **Back-End:** Python(flask / Django), Tensor Flow.
- **Database :** MySQL.

❖ Supported Tools:

- Visual Studio Code , MySQL Workbench

2.5 Design and Implementation Constraints

Constraints:

- The user interface is in English (In Future Reference other language option are available).
- User can access Anytime 24*7.

2.6 User Documentation

The user documentation provides the user with the complete details about the project, its functionality, users, software used, hardware requirements, environment and so on. Step by step working of the project will be provided in the “Help Menu” in order to help the user to understand how the project operates.

2.7 Assumptions and Dependencies

❖ Assumptions:

- There is an active internet connection with the system.
- The system has an internet browser installed.
- Users know the English language, as the user interface will be provided in English.

❖ Dependencies:

- There is a need to raise a query by the user end.
- The query should contain entity and intent defined in the database. The query must be resolved in order to give a response to the user.

3. External Interface Requirements

3.1 User Interface

The main element is web-pages using HTML, CSS, Bootstrap Material. There are multiple interfaces like log-in pages, home pages, Query Pages, Chat Pages, Feedback page of User and Admin.

The user will select the category of the query and wait for the feedback/status of the query by the system. After the query is been resolved, the user can give feedback.

3.2 In the hardware interface:

The system interacts with hardware given the processor is above P4 with a clock speed of 1.5 GHz with 1 GB RAM and the Hard Disk with 1 GB free space in the memory. In future enhancements, it can be made responsive to be able to work with mobile devices as well.

3.3 Software Interface:

In software interfaces, Python, Tensor-Flow is the back-end technology used along with MySQL Database. The front-end technologies include HTML, CSS and Bootstrap. Data will be communicated between these interfaces accordingly.

3.4 Communication Interface:

The main communication interface for interacting with the system will be the web browser.

4. System Features

4.1 Description

This system will help our Users to ask queries and get their queries solved. This system will help us generate database for generic questions and answers.

4.2 Functional Requirements

4.2.1 Forum

- The system will allow customers to send problems to the CDAC Education and Training.
- The system will show who has sent the query and the response generated.

4.2.2 Database Management

- The system should maintain a log of queries asked.
- The project team pre-populate and periodically update the table with information
- Thus, using this system, we will be able to save a lot of time and energy.

4.2.3 Organization Departments Analysis

The system shall allow the organization Management to analyse different queries and work upon different possible response to those queries. This will help the organization Management understand the different possibilities and areas to improve in future.

5. Other Non-Function Requirement

5.1 Performance Requirements

The system should store all the complaint related database records of each complaint raised by People from different location properly and it should be available for use 24*7 through the server. Also, the application should be user-friendly with a proper user interface which makes it easy for the user to understand. All the options should be present in properly accessible places for user convenience.

5.2 Safety Requirements

Admin only can access the intent and entity file that will help to train the chat-bot.

5.3 Security Requirements

Intent and entity file will not be editable by end user.

5.4 Software Quality Attributes

5.4.1 Availability

The system should run on a variety of operating systems that support the JavaScript Language. The system should run on a variety of hardware.

5.4.2 Accessibility

The software will be accessible to admin, End User (Public) and Departments Heads.

5.4.3 Compatibility

The software will be compatible with multiple platforms.

5.4.4 Durability

The software will be tested for working with multiple users.

5.4.5 Effectiveness

The software will be made to handle operations effectively.

5.4.6 Maintainability

The system should be easy to maintain. There should be a clear separation between the Interface and the business logic code. There should be a clear separation between the data access Objects that map the database and the business logic code.

6. System Architecture

Front End Application: web app and Android app which receives a question from the user and talks to the Query Chat-bot API Service to give the answer.

Query Chat-bot API Service: Receives HTTP GET-POST requests containing user queries and forwards them to Query Chat-bot Main.

Query Chat-bot Main: The main process called by the API, which coordinates the other subsystems.

Generic Question Construction: takes the question from the user, and creates a generic question template by replacing certain nouns with generic representations.

Generic Answer Construction: takes in a generic question template and outputs a generic answer template.

Generic Answer Population: takes a generic answer template and populates it with information from the database to form an answer.

Error Handler: if an error occurs during execution of the system, an exception will be thrown, and the error handler will decide how to respond to the user.

Database: stores information about queries needed by generic question construction and generic answer population.

Information Extraction: finds information through structured or unstructured websites, and stores that information into the database. This is separate from the other subsystems, as it must be running before the others in order to populate the database.

7. Other Requirements

Appendix A: Glossary

SRS: Software Requirement Specification

GUI: Graphical User Interface

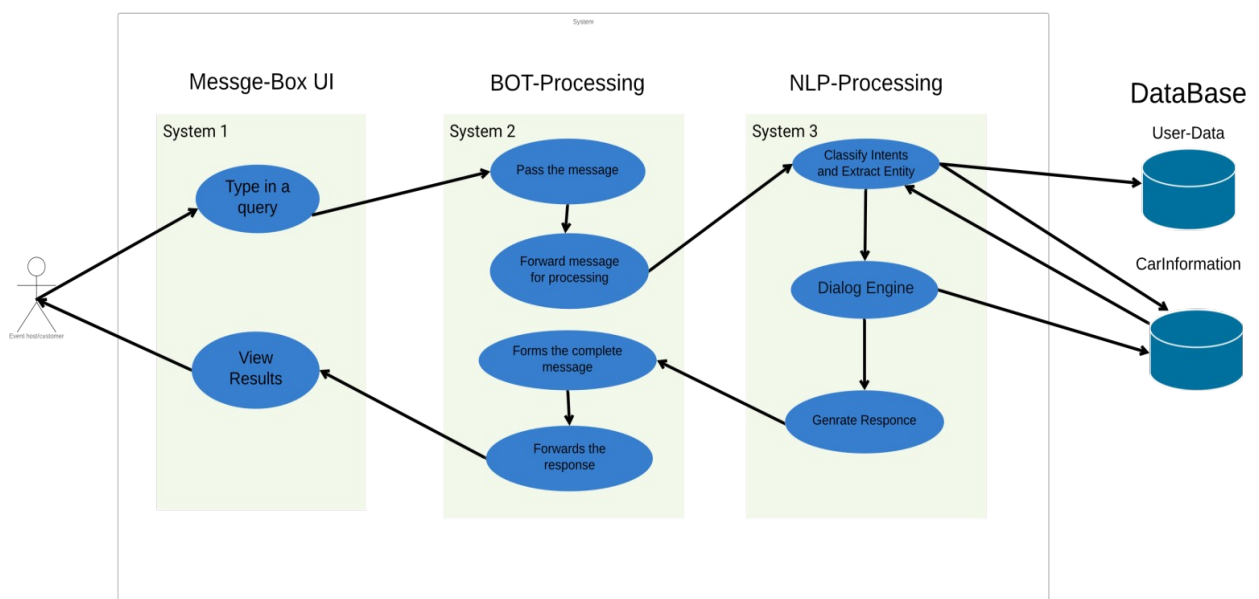
SQL: Structured Query Language

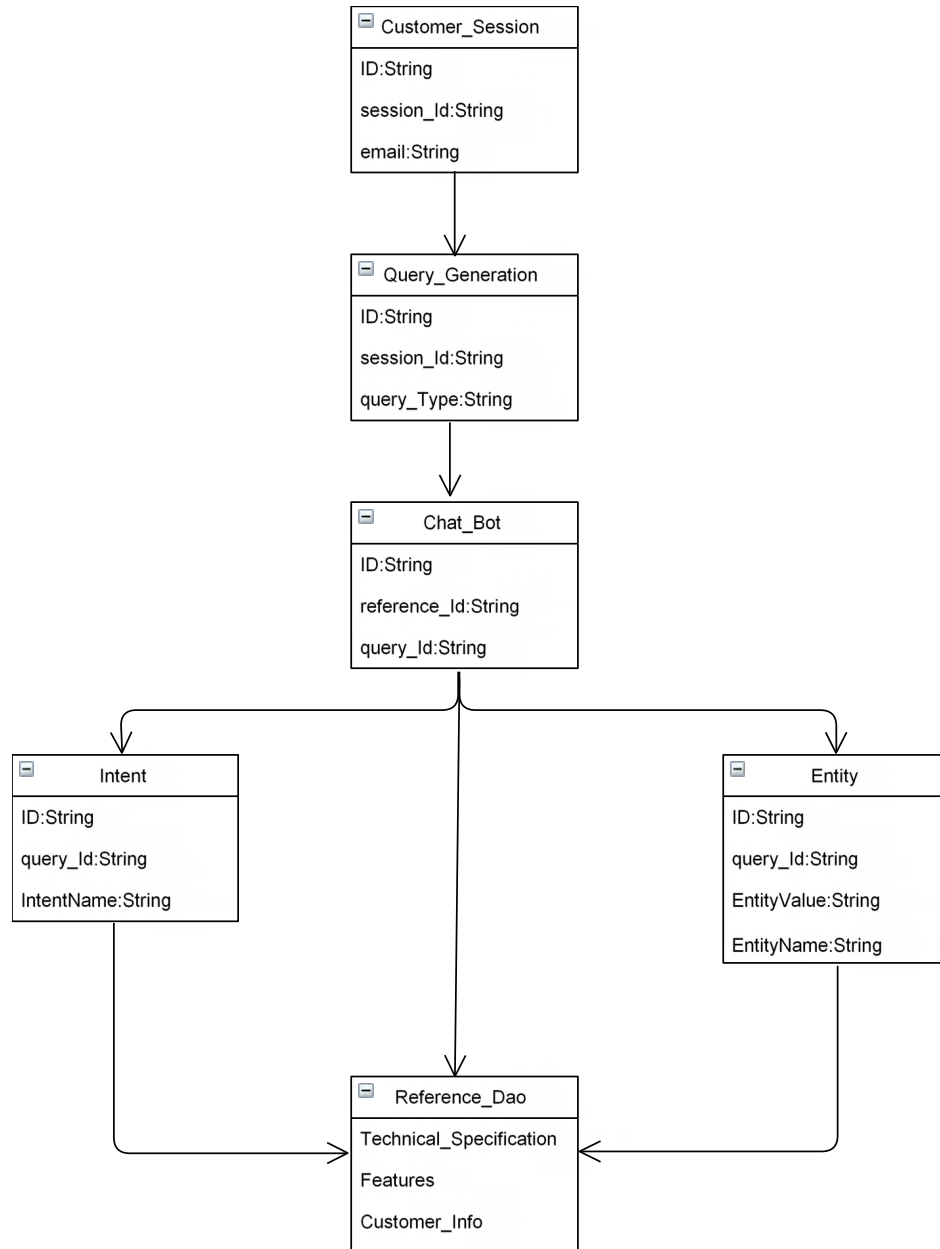
HTML: Hyper Text Markup Language

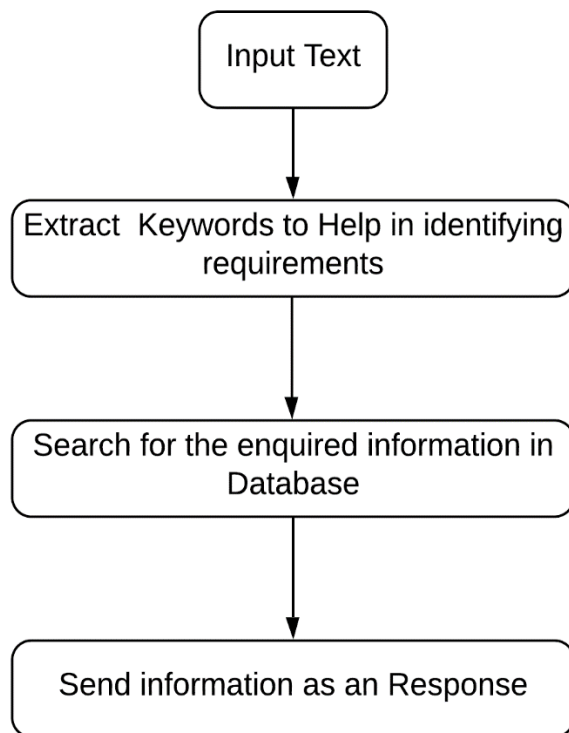
CSS: Cascading Style Sheet

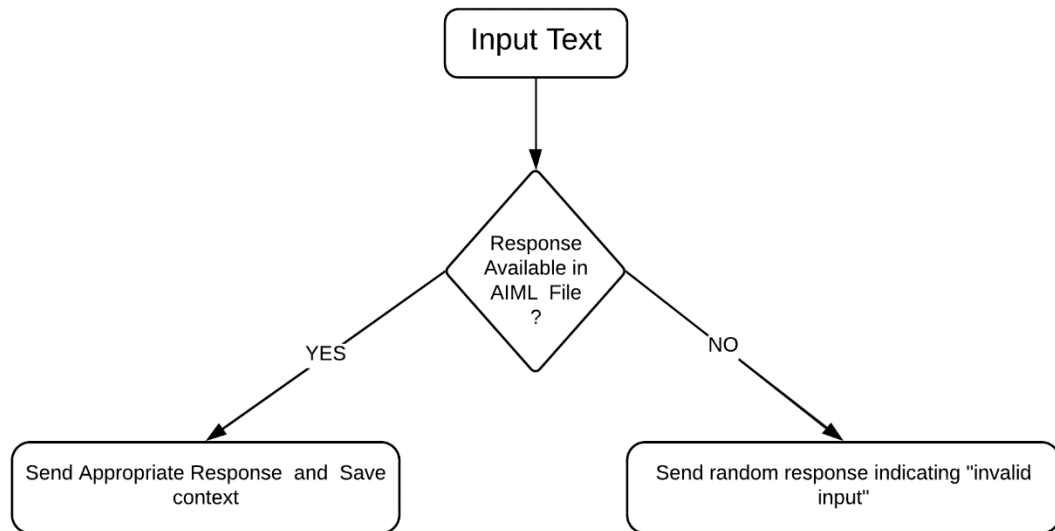
Appendix B: AnalysisModels

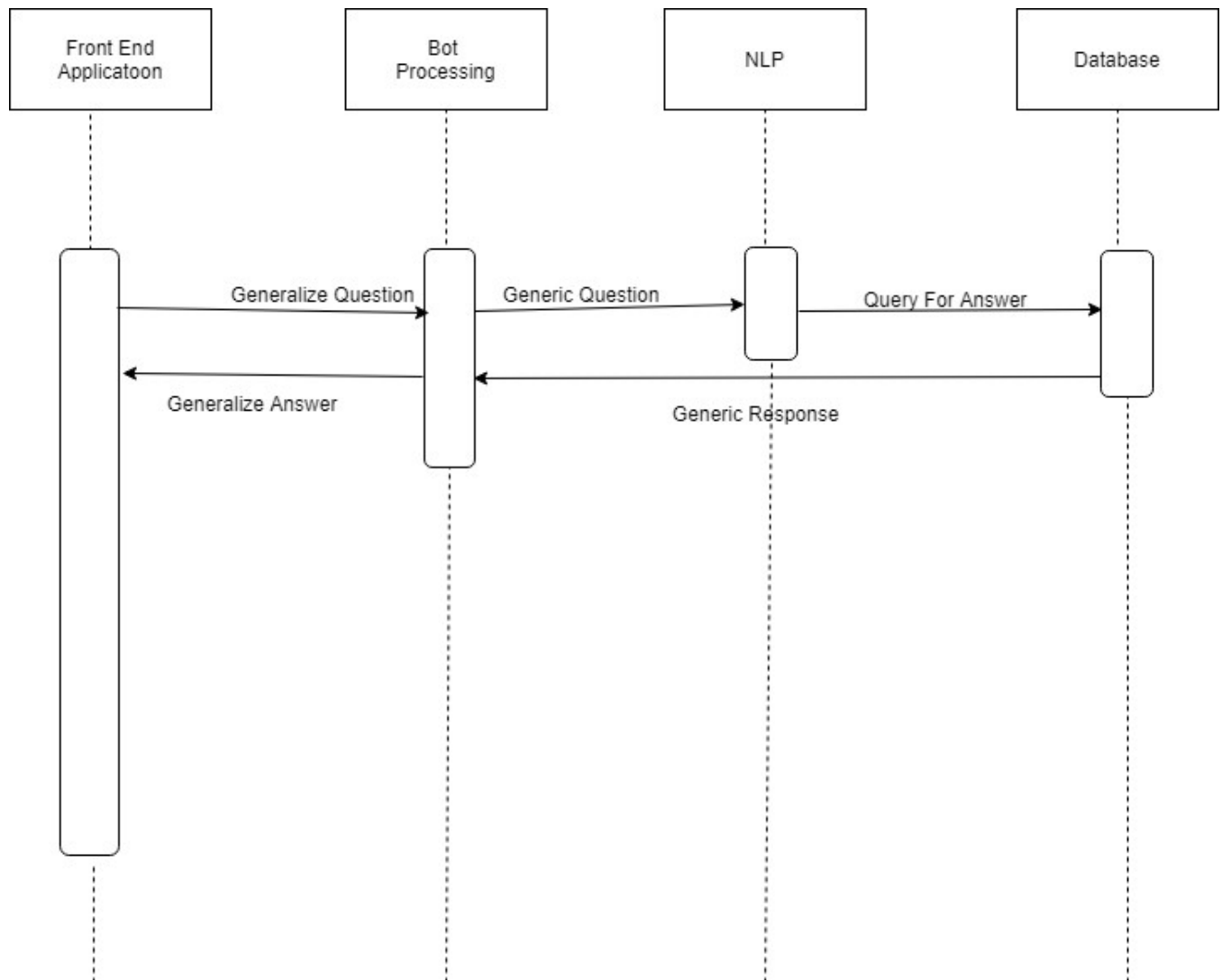
a) Use CaseDiagram



b) Class Diagram

c) Activity Diagram 1

Activity Diagram 2

d) Sequence Diagram1.1:

a) Sequence Diagram1.2: