Software Requirements Specification

for

Chat Bot Application

M.Charkradhar-16CO226

M.kunal-16CO227

Sri Charan-16CO228

NITK

OCTOBER 1,2018

Table of Contents

1. Introduction

1.1 Purpose

This document will provide all of the requirements for the project college application Chat bot. It will serve as a reference for developers and users during the development of the final version of the system.

1.2 Product Scope

Our Chat bot is an AI chat bot 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, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer. For example, when it receives the question "What time does the gym close today", it will give a response "The gym closes at 10pm today." The main objective is using the Google's dialog flow API, creating sample web messaging interfaces that demonstrate the use of the API. The goal is to provide our college students and faculty a quick and easy way to have their questions answered.

1.3 Intended Audience and Overview

The students of the college are the target audience and the prime users of the application. The application mainly is designed to answer their queries and make their life simpler.

2. Overall Description

2.1 Product Perspective:

Our Chat bot will try to understand the query and provide a definitive answer. There will be four main units to the system working together to understand the question and return an appropriate answer:

- Valid question construction - capable of taking a natural language question and mapping it to an intent.
- Generic answer construction –deriving the answer from the database and forming a text response.
- Intent extraction –Extracting the parameters from the intent obtained.

2.2 Product Features

The major features for our Chat bot will be the following:

- Web API (Dialog flow): An API call will include a question in the form of a query string url parameter and the service will reply in JSON.
- Natural Language Processing: The system will take in questions written in 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 unique responses for different queries.

2.3 User Classes and Characteristics

The two classes of users for this system are described below:
● API users: API users consist of application developers who incorporate the API into their own web applications. They can alter the default responses and the behavior of the bot.
6
● Web app users: These users consist of non-technical users who want to get answers for their questions. These users ask questions and get answers with web messaging interfaces. This class of users include students, teaching faculty, and staff.

2.4 Constraints

- Limited Question Scope: Creating a chat bot able to answer every single question about college is not possible to implement within the duration of the project, so the system will be able to answer questions about limited topics mentioned in the homepage of the web application.
- Language: The system will only understand questions in English.

2.5 Assumptions and Dependencies

Dialog Flow is an API used for NLP. We need to extract text from a webpage, we use python flask for that purpose of integrating our web page and the external API. We will develop the project using Python and MySQL database.

3. External Interface Requirements

3.1 User Interface

The user needs to open the web application which displays information about the features of the application and a link to the console. If the user clicks the console he will be directed to another web

page containing the bot .User can ask questions in the chat bot available. The questions that bot can give reply are based on the topics that are in the Home page of our web application.

3.2 Hardware Interface

Web applications can have any designated hardware, it does not have any direct hardware interfaces. The hardware connection to the database server is managed by the underlying operating system and the web server.

3.3 Software Interface

The software interface plays vital role in the interaction between the user and system. Interfaces are made as simple as possible for smooth interaction between the user and system. Proper security measures will be implemented to avoid the failure of the interfaces. Platform used is browser.

4. Functional Requirements

4.1 API Calls

4.1.1 Client (Web Application) Responsibilities:

The client will send a GET request to the Web API with the question as a parameter. The server will reply with either data or an error. The client will be able to parse the JSON and determine if there was an error.

4.1.2 Server (API) Responsibilities

The server will send all data in JSON response documents
A JSON object will be the root of every API response.

4.2 Dialog Construction

4.2.1 Input & Output Format

The bot will receive a text string from the user. It will identify important words in the sentence and extract the parameters specifying the intents.
The bot will output the sentence as a string with the data obtained from the database and present it to user.

4.3 List of entities

This unit will have a list of generic words related specifically to potential queries mapping them to a certain intent.

4.4 Error Handling

An error during the conversation means that there was a problem in parsing the sentence and extracting the parameters or during the web hook call. In this case, return a message such as “Sorry, I didn’t understand that.”

4.5 Result Extraction & Database

4.5.1 Database

A MySQL database will be used to store all information required to answer user queries. The database is populated by the information extraction unit before the rest of the system is available, so that all information is readily accessible.

The database will use the entities extracted by the API as table names or column headers for easier retrieval of data.

4.6 Supported Question Topics

The Web API will only handle questions from following topics without unexpected error:

4.6.1 Academic calendar

View the academic calendar of the semester and year specified.

4.6.2 Faculty search

Search for faculty and HODs of each department with department name and other parameters specified.

4.6.3 Campus Timings

Get to know the timings of all the amenities in the campus.

4.6.4 Doctor Timings

Get to know the timings and day wise visit of doctors to the HCC.

4.6.5 Food Order

Order food from any night canteens across the college.

4.6.6 Library book check

Check for the availability of the book in the library and tell where the book is at.

4.6.7 Sports Timings

Check the timings of the grounds and ask about the materials availability.

4.7 User Interfaces

The GUI will have a textbox that will accept inputs from a keyboard or microphone.

The GUI chat window displays questions sent to the system and responses from the API.
The chat window will contain all questions and answers from the current session, with a scroll bar if all messages can't fit on the screen.
If there is a network issue, the chat window will display an error message "webhook call failed".

5. Nonfunctional Requirements

5.1 API

Modularity- The system will be designed in such a way that the algorithms for the different will be separate and can be easily modified.
Security- The connection between the Web API and the programs will use HTTPS, for security.

5.2 Web Application

Ease of Use-A new user is likely to make mistakes after 5 minutes of use as the application is easy to understand and use.

6. Proposed system

- User enters his query.
- The bot extracts the action parameters from the given query.
- The parameter are passed onto the system.
- We perform the required action on the database using the passed parameters.
- The appropriate values from the database are returned.
- The bot converts the return values into text format and displays it to the user.
- The conversation is continued until the user stops it or default fall back intent occurs.

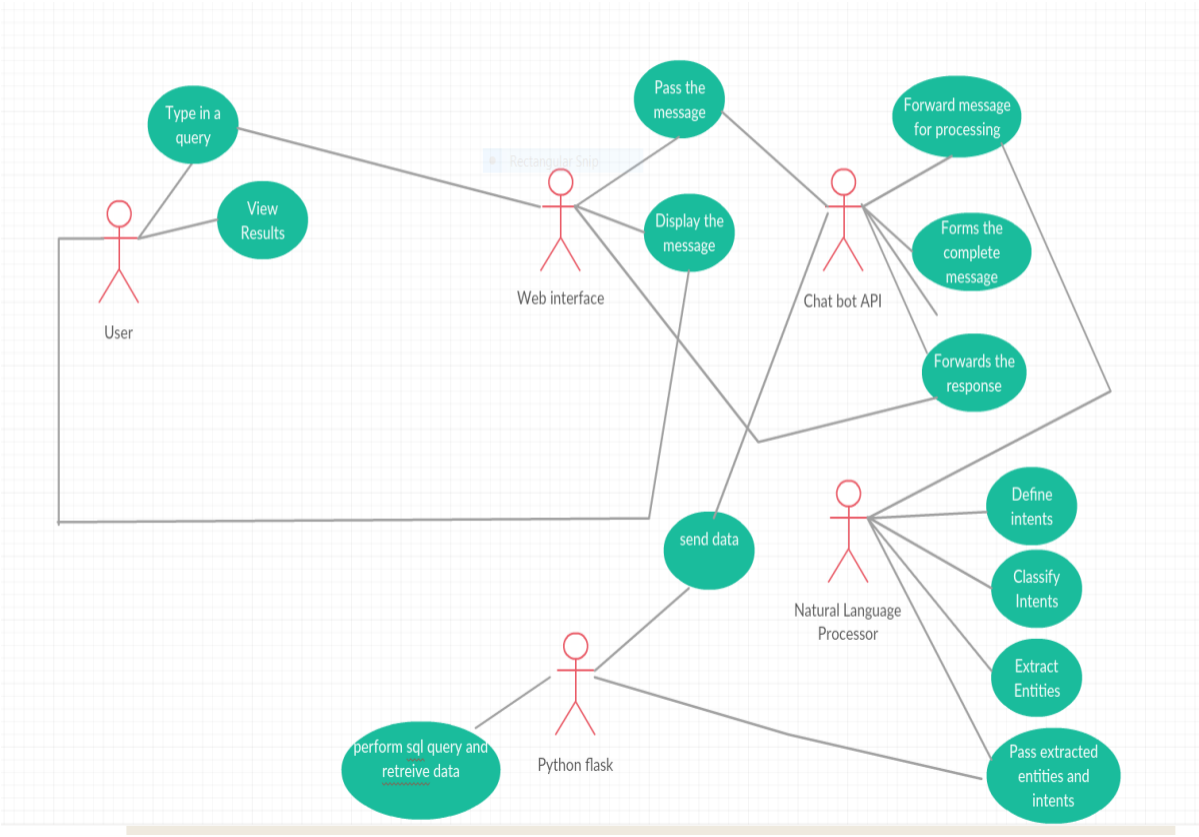
7. Documentation of Intents

The below table contains about the list of intents, what are the list of utterances for every intent, identifying the entities, specifying the actions determined by the intents and the extracted entities

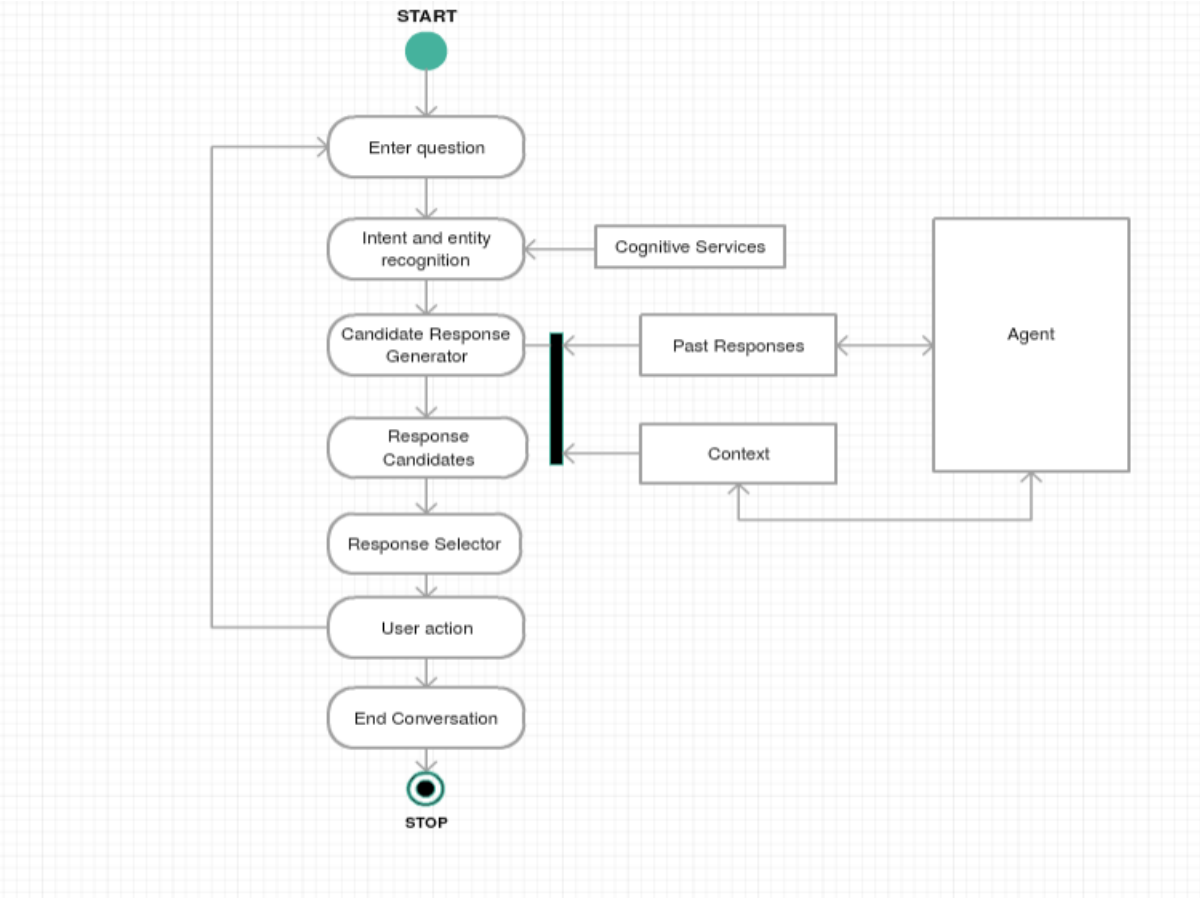
Intent	Sample Utterances	Entities	Actions
Academic_calendar	Show academic calendar for 2018 odd sem 2015 even sem	{Semester} {academic_calendar-year}	Look the Academic_calendar and direct to a web page containing the academic calendar of that year and sem the user wanted.
Faculty.search	I want all the professors with PhD qualification in CS dept.	{Professors} {Department}	Look up the Faculty and display the names of all professors with qualification phd and department CSE.
HostelWardens_Info	can you tell me phone number of warden of Mega tower 1	{Block}	Look up the warden relation and display the phone number of the warden of block mega tower 1
Campus.Timings	What are the timings of Health Care centre	{place}	Look up the Campus_timings relation and display the block timings of the Block name Health care center.
Food.search	Can you tell the items available in 7th NC	{NC_name}	Look up the food relation and direct to web page containing menu of 7th Block
Food.Order	I want one Biryani from 7Th NC	{NC_name} {Item} {Quantity}	Place order in the NC_Order relation with Quantity 1,Item Biryani, NC_name 7Th block
Library.search	Where are the Ansi C books available in the library	{Book}	Look up the Library relation and display the shelf, floor and availability of the book Ansi C
Sports_Complex	When can we play the cricket.	{Game}	Look up the Sports_Complex and display the timings for the game cricket.
HCC.DoctorTimings	When will be the dermatologist in the HCC	{Specialist}	Look up the HCC relation and display the day, time and doctor name of the specialist Dermatologist.
Event.calendar	When is the Engi happening this year?	{event_name}	Look up the event calendar table and respond with the corresponding dates.
Fest.schedule	When is the pro show in incident	{event_name} {Fest_name}	Look up in the schedule table and respond accordingly.

8. UML DIAGRAMS

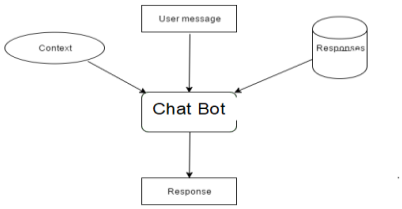
8.1 USE CASE DIAGRAM



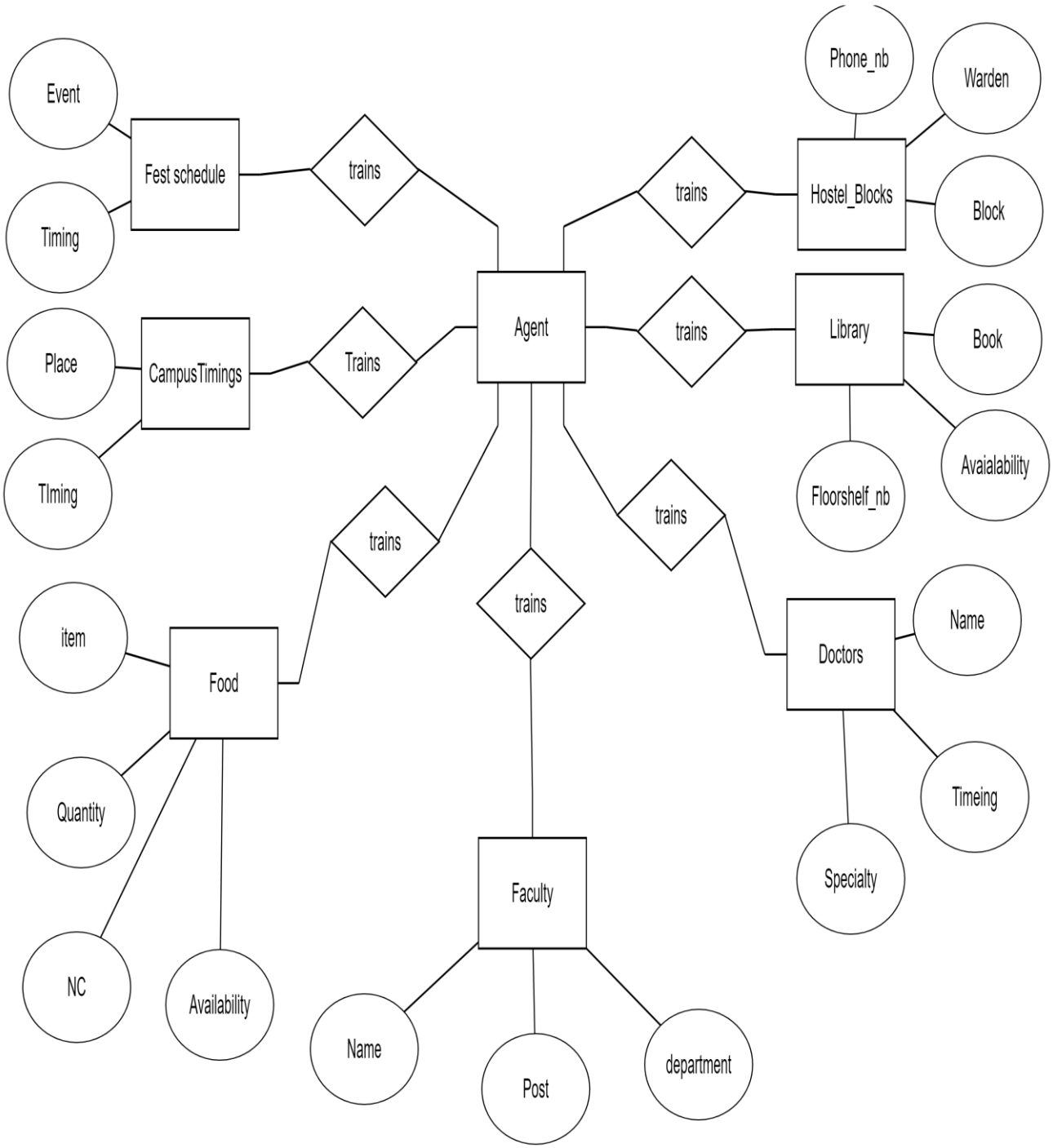
8.2 Activity Diagram



8.3 Data Flow Diagram



8.4 ER Diagram



9. Conclusion

The main objectives of the project were to develop an algorithm that will be used to identify answers related to user submitted questions. To develop a database were all the related data will be stored and to develop a web interface. The web interface is for students in the particular college. A background research took place, which included an overview of the conversation procedure and any relevant chat bots available. A database was developed, which stores information about questions, answers, keywords, logs and feedback messages. A usable system was designed, developed and deployed to the web server.