Toaz - software requirements specification on chatbot

B.tech CSE (Lovely Professional University)

Software Requirements Specification

for

ChatBot

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

1. Introduction

1.1 Purpose of Document

This document will provide all of the requirements for the project Chatbot. It will serve as a reference for developers and customers during the development of the final version of the system.

1.2 Project Scope

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, 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 "I want to register a complaint?", it will give a response “Register it now!”
The main objective is creating a Web API, and sample web, mobile, and text messaging interfaces that demonstrate the use of the API.
The goal is to provide student, women and all other people a quick and easy way to have their questions answered, as well as to offer other developers the means to incorporate Chatbot into their projects.

1.3 Overview of Document

1.Revision History: Provide the date of, reason for, and people who were involved with the modification of this document.
2.Introduction: Provide an overview of the application, explain the objectives and goal of the project and describe the document structure.
3.Overall Description: Provide the specification of the system model, the classes model and the main constraints.
4.Functional Requirements: Provide the analysis of the requirements by feature.
5. Non -functional requirements: Provide some other constraints that affect performance, safety and security.
6.Use Cases: Provide possible scenarios where the user interacts with the Web API and sample applications.
7.Glossary: Definitions of terms used.

2. Overall Description

2.1 Product Perspective

● Generic question construction - capable of taking a natural language question and making it more generic.

- Generic answer construction - capable of taking a generic question template and providing a generic answer template.
- Generic answer population - capable of taking a generic answer template and populating it with information from the database to form an answer.
- Information extraction - capable of finding information through structured or unstructured websites, and storing that information in a database.

2.2 Product Features

The major features for Bot Chatbot will be the following:

- **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.
- **Natural Language Processing:** T he system will take in questions written in standard English.
- **Natural Language Responses**: T he answer to the question will be written in standard and understandable English.
- Information Extraction: T here will be a database containing all the information needed, populated using information extraction techniques.

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 want to incorporate Bot Chatbot API into other software applications.

- **Mobile app/Web app/SMS users**

These users consist of non-technical users who want to get answers for their questions. These users ask questions and get answers with mobile, web, or text messaging interfaces. This class of users include Bot's current and prospective students, teaching faculty, and staff.

2.4 Constraints

2.4.1 Limited Question Scope

Creating a chatbot able to answer every single question about Bot is not possible to implement with current technology and within the duration of the project, so the system will be able to answer questions about limited topics.

2.4.2 Language

The system will only support questions in standard English.

3. System Features

- ***<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>***
- *The bot will respond to any input it receives.*

If the bot understands the input, it will respond with correct information. All use cases If the bot needs more information to find an answer, it will ask for more information. The bot will use a text recognition API to understand the input.

3.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

3.2 System Feature 2 (and so on)

4. External Interface Requirements

4.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

4.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

4.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

5. Other Nonfunctional Requirements

5.1 API

5.1.1 Modularity

R5.1.1.1 The system will be designed in such a way that the algorithms for the four main units will be able to be easily swapped out. **Priority 1**

5.1.2 Accuracy

R5.1.2.1 The overall accuracy of the Web API's response will be measured using a developer-made testing set. **Priority 1**

R5.1.2.2 The overall accuracy is calculated by dividing total number of correct answers by the number of questions asked. **Priority 1**

R5.1.2.3 The accuracy of the Generic Question Construction part will be close to 80%. **Priority 2**

R5.1.2.4 The accuracy of the Generic Answer Construction unit will be close to 70%. **Priority 2**

R5.1.2.5 The accuracy of the Generic Answer Population unit will be close to 70%. **Priority 2**

The accuracy for each supported topic will be as follows:
R5.1.2.6 Bot facilities' locations and schedules will have accuracy greater than 70% **Priority 2**

R5.1.2.7 Bots staff's office locations, contact information, and positions will have accuracy greater than 70% **Priority 2**

R5.1.2.8 Bots policies including academics, admissions, information technology etc. will have accuracy greater than 70%. **Priority 2**

R5.1.2.9 On-campus dining locations, hours, food types, etc. will have accuracy greater than 50%. **Priority 2**

R5.1.2.10 Food trucks' general locations, hours, and food types on Bots Campus will have accuracy greater than 40%. **Priority 3**

R5.1.2.11 Official events listed on Bot website, location and hours will have accuracy greater than 40%. **Priority 3**

5.1.3 Fast Response

R5.1.3.1
The average time for the server to respond, over the question testing set, will be less than or equal to 2 seconds. **Priority 2**

5.1.4 Security

R5.1.4.1 The connection between the Web API and the programs will use HTTPS, for security. Priority 3

5.2 Web interface/Mobile application

5.2.1 Ease of Use

R5.2.1.1 A new user will make less than 3 mistakes in 5 minutes after 5 minutes of use. **Priority 1**

Appendix A: Glossary

- Chatbot: An interface, usually text based, specializing in the mimicry of natural language conversation. AKA “artificial conversational entity.”
- GUI: Graphic User Interface, a type of user interface that allows users to interact with the software through graphical icons (e.g. buttons, etc.).
- HTML: Hypertext Markup Language, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on webpages.
- JSON: JavaScript Object Notation, a data-interchange format that is commonly used in exchanging data over the Internet.
- Pagerank: PageRank is an algorithm used by Google to rank websites. It works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.
- SMS: Short Message Service, the text messaging protocol of cellular telephones.
- Standard English: the language that can be understood by English-speaking high school graduates.
- URL: Uniform Resource Locator, an address to a resource on the Internet.
- URL parameter: parameters whose values are set in a webpage’s URL.
- Web API: an application programming interface (API) for either a web server.
- Web scraping: web scraping is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database.