Software Requirements Specification

Pineapple Chatbot

Version 1.0 (draft)

Prepared by Kuong Thong, Enrique Castillo, Julio Aguilar, Jose Lopez, Yilin Ruan
CSULA ITS Department
March 22, 2020

Contents

Revision History

Name	Date	Description	Version
Converted certain design document parts to this document	11/21/18	Getting started with skeleton of document	aardvark
Adding external if requirements	11/26/18	Filling out next section on doc	beaver
Adding design constraints	11/29/18	Nothing in section, filling out	cheetah
Adding non-functional requirements and finishing appendixes	12/02/18	Nothing in section, filling out	dingo
Updated diagrams and UI description	02/26/19	Keeping the diagrams and descriptions in line with our plans for the bot	elk
Several updates and tweaks to almost all sections	03/22/19	Done in order to keep all descriptions, module names, and feature requirements in line with the renovated plans for the bot	fox
Updated for new database structure	10/31/19	Updating the document to have our new database	goose
Removed Amazon Lex	11/15/19	Removed Amazon Lex from the application	horse
Updated spaCy	11/22/19	spaCy replaces Amazon Lex	impala
Added simple bayesian model	03/11/20	A simple bayesian model will be used to choose the best ranked URL	jaguar

1 Introduction

1.1 Purpose

The product is an informational/service chatbot for the Cal State LA website. The chatbot will answer queries by the user regarding the university. It will be able to receive feedback from the user. Its main purpose is to assist website visitors in any information they might want throughout their time in school. This document has all of the scope of our project.

1.2 Intended Audience and Reading Suggestions

This document is intended for developers, testers, marketing staff, and project managers as well as the documentation authors. The document contains information on how the application works. On how to use the chatbot application, read the document from top to bottom to get a better understanding on how the chatbot runs.

This SRS contains information on details of the product, the testing of the product and the procedures to run or disable the product. It also contains the limitations and the specifics on how it should be used. The requirement list is also stated and the list of future requirements can also be found on this document. A tester can look at the requirement list and data flow diagrams to see how it works. Customer(s) can look at the list of things the Pineapple chatbot can do. An investor may look into future goals for the product and a high level description of the product.

1.3 Product Scope

Pineapple chatbot uses a web based application that will provide an interaction between the user and an AI chat bot. This software will use MySQL to gather and store information in a database such as the user questions as well as predetermined responses. It will use best matching algorithms like Levenshtein distance to output the closest result. A simple bayesian model is used to determine which website has the best match to the users' intent. Once the software is released, the bot will be able to have a conversation as well as return responses such as answers and links to help the students of CSULA easily navigate through the calstatela website.

1.4 Definitions, Acronyms, and Abbreviations

- SQL - Structured Query Language
- BS4 - Beautifulsoup
- NLTK - Natural Language Toolkit
- DF D - Data Flow Diagram
- SRD - Software Requirement Document
- SDD - Software Design Document

1.5   References

- TBD

2   Overall Description

2.1   Product Perspective

The system uses crawled data and stores it into a database. Then using natural language processing it determines what the user is asking for based on the query, and returns a response. To start we approached the project like we are building a custom search engine for the campus website. The chatbot, in its current state, returns the link to the webpage relevant to the query, if the search is not about a professor. If a professor is searched for, an English answer is given to the user by the chatbot. The chat bot in general is similar to any type of service/chatbot that is offered in the market such as Domino's pizza, Spectrum internet, and many more. Some chatbots that we found, such as Spotify's chatbot, that only allows the user to choose from predetermined choices, motivated us to develop a more sophisticated one that could have a conversation as well as provide best answers to questions.

2.2   Product Functions

The features that are available to users are:

- Communicate with the chatbot
- Ask the chatbot questions
- Chatbot responses with links/answers to users question

2.3   User Classes and Characteristics

- Developers: Those who use the application as well as edit the applications
- Designers: Those who edit, user and change the design of the application
- Users: Those who will use the application but do not have the permission to edit

2.4   Operating Environment

The primary component of the pineapple chatbot will be available to anyone who can access the CSULA webpage.

2.5   Design and Implementation Constraints

- Needs MySQL to store all responses
- Language Requirement: English
- Hardware Limitation: N/A
- Implementation Constraints: The application is written in Python, JavaScript, and MySQL

- Should be designed to accommodate all students and users who can access the CSULA website
- User must have access to the internet

2.6 User Documentation

There are no guidelines or user manuals/tutorials to this application. It will be made public to users on the CSULA website. The CSULA website will feature the chatbot.

2.7 Assumptions and Dependencies

- Python Modules
  - MySQL connector (Database Connection)
  - bs4 (BeautifulSoup, HTML Parser)
  - nltk (Natural Language ToolKit)
  - spaCy (Language Understanding)
  - urllib (URL Opener)
  - re (Regular Expressions, Text Processing)
- MySQL
- NodeJS

2.8 Apportioning of Requirements

In the case that the project is delayed, some requirements may be transferred to the next version of the application. Those releases will be developed in the following Senior Design Year.

3 External Interface Requirements

3.1 User Interfaces

The user will have access to a text box in the chatbot website. This will allow the user to type in queries for the chatbot to answer. The user can type their question in the empty text box, and by pressing ENTER they can submit their query. The website will then display their question back, in the same format as most text messaging applications, the website will then display an answer from the chatbot in the same format. As of now this response only contains a link that has the answer to their question if the question is not about a professor. However, a relevant answer is given in English, if a question about a professor is asked.

3.2 Hardware Interfaces

This program does not need any specific hardware interface requirements other than any device with access to the internet and CSULA's website.

3.3 Software Interfaces

As of right now the chatbot website is only available to those with a VPN. Therefore, the user must use any VPN software in order to access the chatbot website application.

3.4 Communications Interfaces

Access to the internet is required to access the chatbot website. Any VPN, as mentioned above is required. In addition, the chatbot website is not available in Internet Explorer web browser. We recommend the user use any of the other popular ones such as Google Chrome, Firefox, or Microsoft Edge.

4 Requirements Specification

4.1 Functional Requirements

4.1.1 Input Module

1. This module shall allow the user to input queries
2. This module should allow the user to give feedback on the answer given
3. This module may allow the user to use voice as an input

4.1.2 Output Module

1. This module shall display the user's question back to them
2. This module shall display the chatbot's response to the user which will include a direct info response and a link that provides more detailed information

4.1.3 Logic Module

1. This module shall parse the user's input for keywords
2. This module recognize intent from the user's input using language understanding implemented by the spaCy library
3. This module should respond appropriately even when the user's input is unintelligible
4. This module shall send the keywords and intent to the storage module
5. This module shall receive a list of possible URLs from storage module
6. This module should find URL most related to the keywords of the users' input
7. This module shall send the entity, intent, and keywords to data extraction module

4.1.4 Data Extraction Module

1. This module shall use the entity, intent, and keywords to extract the right information from the URL
2. This module should send the information in 4.1 to the output module
3. This module shall parse and organize text extracted from web pages into abstracted information and have spaCy label the information

4.1.5   Storage Module

1. This module shall be able to send queries to the database
2. This module shall be able to receive responses from the database
3. This module shall be able to send information back to the logic module

4.2   External Interface Requirements

Website Text Box Input: This input has the purpose of being the only input available to the user in submitting text strings to be sent to the chatbot server program running on a different port. The source of input is the website text field/box that the user clicks on to select, and then type in the input. Valid range is any text string input possible in JavaScript string. No limitations were enacted on this input in JavaScript. When multiple inputs are entered, the website puts the secondary requests in a queue while waiting for the first input to output a reply. Therefore, several inputs can be typed without causing time sync issues. This input waits for a reply (the Website Chat Bubble Output) output, an entity that is explained in section 2 in 4.2. No screen formats or data formats are needed other than, the input needs to be an English request for information that is generally simplified and direct.

Website Chat Bubble Output: This output has the purpose of being the only output available which functions as a web link reply to the user's input. Destination of output is the website in a UI Chat Bubble that shows the output of the chatbot inside. The valid range is limited to the amount of URLs present in our database, which is limited to the number of web pages on the CSULA website. The accuracy depends on our matching algorithm which matches based on the input given. Timing is based on giving a reply and immediately taking any other input still in the queue until done. Has a strong relationship to input in that the algorithm matches input to appropriate an output in the database. The format is grabbed from web-crawling the CSULA site and gathering links so all link variations are the format. Output also sends an error message when an input that the bot cannot match to output is entered.

4.3   Logical Database Requirements

The database should have a table containing records of individual terms, URLs, and keywords within the URLs.

The database shall accept queries.

The database shall return the queried results.

4.4   Design Constraints

- Web-crawling has hardware limitations on time to crawl, and exporting the .txt file with a dictionary to a MySQL database
- We do not use a language converter, and so, only English queries are accepted for our matching algorithm
- Not compatible with Internet Explorer, but compatible with all other modern browsers

- Users can only access the chatbot when connected to a CSULA network or VPN
- Python restricts our chatbot server to being written in Python as well as, cannot use third-party tools not written in Python for our chat algorithm

5 Other Nonfunctional Requirements

5.1 Performance Requirements

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 REACT server.

Accessed by multiple users at the same time. Currently unaware of the actual limit.

5.2 Safety Requirements

A feedback system will be implemented to keep all reports from remaining anonymous in order to protect the user's information or identity in the case of an unexpected event.

5.3 Security Requirements

There will be two servers, one to host the application and one to store the database and all information. A VPN is necessary to access the web page for further security purposes.

5.4 Software Quality Attributes

The quality attributes that this Chatbot will strive to have the most of is: creating an easier environment for students and users to navigate through the CSULA website by just communicating with the chatbot. By making it simpler, we hope that it will create a more efficient way to save students time.

5.5 Business Rules

All users of this application will have access to this chatbot.

All users with specific permissions in IT will have access to view private contents/data.

6 Other Requirements

Not Applicable.

Appendix A: Glossary

Reference section 1.4

Appendix B: Analysis Models

Pineapple ChatBot will be used in a standalone web application

Appendix C: To Be Determined List

Refer to section 2.8