  
   
Software Requirements   
 Specification (SRS)

BabbleBot

Version 1.0

Prepared by:

Group 6

Mad DJs

|  |  |  |
| --- | --- | --- |
| * Morgan McKeithan |  | mckeithanm15@students.ecu.edu |
| * Ashley Fraley |  | fraleya14@students.ecu.edu |
| * Jeremiah Smith |  | smithjere14@students.ecu.edu |
| * Eric Sengmany |  | sengmanye14@students.ecu.edu |
| * Daehan Barnes |  | barnesda14@students.ecu.edu |

|  |  |
| --- | --- |
| Instructor: | Dr. Kamran Sartipi |
| Course: | CSCI 3030 (Software Engineering I) |
| Date: | 10/02/2017 |

Contents

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 8

Appendix A – Data Dictionary 9

Appendix B - Group Log 10

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# *<In this template you will find text bounded by the “<>” symbols. This text appears in italics and is intended to guide you through the template and provide explanations regarding the different sections in this document. There are two types of comments in this document. These comments that are in black are intended specifically for that course. These comments that are in blue are more general and apply to any SRS. Please, make sure to delete all of the comments before submitting the document.*

# *The explanations provided below, do not cover all of the material, but merely, the general nature of the information you would usually find in SRS documents. It is based on the IEEE requirements and was adapted specifically for the needs of Software Engineering courses. Most of the sections in this template are required sections, i.e. you must include them in your version of the document. Failure to do so will result in marks deductions. Optional sections will be explicitly marked as optional. If you have any questions regarding this document please refer to the MiniThermostat SRS example on the course web-site.>*

# Introduction

*BabbleBot is a machine-learning chat bot that answers educational questions. In this section the reader will get a brief overview of BabbleBot along with an overview of this documents and terminology that will be used.*

## Document Purpose

The software requirements of BabbleBot 1.0 are specified in this document. All subsystems of Babble Bot will be introduced and described.

## Product Scope

The software being specified in this document will respond to user inquiries in regard to educational questions. It will be beneficial in helping to answer questions that users happen to have in regard to simple to average Math questions or English sentence structure questions.

## Intended Audience and Document Overview

This document is intended for the client of our system as well as Dr. Sartipi and his reviewers. The remainder of this Software Requirements Specification documents contains specific details regarding our system and how it should be implemented and used. Section 1 provides an overall view of the purpose of this document and a very short look at what our program will entail. Section 2 goes into the details of our project, including the functions and intended users of the program. Section 3 takes an even deeper look into our project by looking at the software and hardware sides. Section 4 discusses performance requirements of our project, safety and security measures, and other non-functional requirements. In order to understand uncommon language or acronyms used in this document, one should refer to section 1.4 Definitions - Acronyms and Abbreviations. To view the functions that our program performs, one should refer to section 2.2 – Product Functionality. To see the intended users of our program, refer to section 2.3.

## Definitions, Acronyms and Abbreviations

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.

TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>

## Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

## References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

TO DO: Use the standard IEEE citation guide for this section. An example citation guide is posted for you on the website.>

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.

TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used.>

## Product Functionality

* Website – Babblebot will be showcased on an aesthetically pleasing platform in order to capture user attention and allow for easier searching.
* Accepting User Input - whether it be the query or adding queries/responses to the database, BabbleBot will read in user input and further analyze it.
* Analyze Queries – BabbleBot will look at each portion of the given query in order to analyze sentence structure to add to its database and to reach an appropriate response.
* Formulate Responses – BabbleBot will use its knowledge to provide an appropriate response to the user.
* Perform arithmetic – BabbleBot can solve math problems inputted by the user.
* Explain Sentence Structure – BabbleBot can analyze sentence structure and explain it to a user of our program.
* Send Responses – BabbleBot will respond to user in designated space.

## Users and Characteristics

<Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.

TO DO:

1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.

3. Distinguish the most important users for this product from those who are less important to satisfy.>

## Operating Environment

The environment will work on any internet enabled browser to which no other components is needed, as it would connect to a website that this would be hosted on, this is the requirements to access it. In reference to where it would be hosted at, the code and software will be on an website and server to hold all of this thus, which will be connected to the user on their internet browser, and the website hosting this would need to have a sufficient bandwidth and able to properly handle large amounts of responses and process at once so that everything can run properly and smoothly.

## Design and Implementation Constraints

5 constraints would be:

1. Internet Access
2. Knowledge and ability to type in the English language.
3. Storage limit on learned words and phrases with a minimum of a gigabyte
4. Requires a database of common words, and their grammatical meanings
5. For programming standards, the code will not be nested in each other and will be broken up into parts so that they may function individually.

## User Documentation

On-line help will be provided to users in the form of on-screen instructions on the home page of our website, this would also include a troubleshooting section incase that any problems should arise as well. So the given manuals will, be the set up tutorial, how it works and it’s algorithm, and trouble shooting help upon encountering an error.

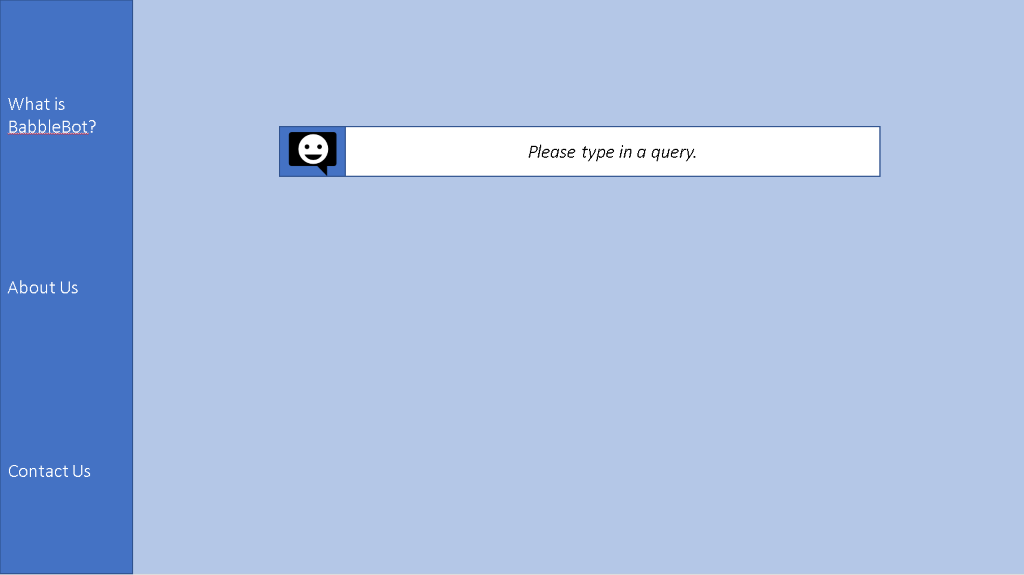
## Assumptions and Dependencies

Some assumed factors are that the database if it becomes full or overloaded will automatically de-load the least uses words/phrases. Dependency on internet connection as well as on the website that would be hosting said database and server, there fore if there were too many users at once it would cause a delay and latency upon the AI that would slow it down massively, there fore there will be a limit on the amount that this can handle at once or a que to which each query will be handled if access from several different points such that it could effectively handle large amounts of users at once. So for the time being until this is finalized the baseline assumption should be no more then 10 due to the limitations of the software used.

# Specific Requirements

## External Interface Requirements

### User Interfaces

**

When a user first enters the page the first thing that should draw their attention is the BabbleBot User input interface or chat box. There are links to other parts of the website on the left-hand side. Clicking on these links will move to a page that provides information for the user based on which link they have chosen.

The figure below is an example of what User interaction with BabbleBot will look like. They will enter a query into the chat box and BabbleBot will respond. The query provided by the user will be displayed along with the answer to the query.

**

### 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. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), 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. 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.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

### 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.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

## Functional Requirements

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

*TO DO: Brake the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

## Behaviour Requirements

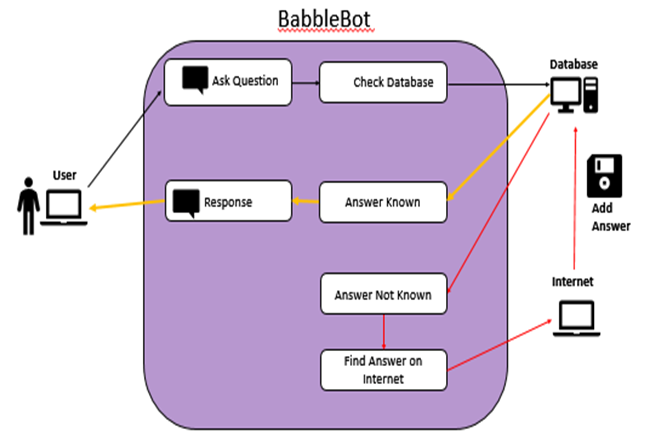
### Use Case View

-*User will ask Babble bot a question. BabbleBot will check its database.*

*-If BabbleBot knows the answer it will send the user a response*

*-If BabbleBot does not know the answer, it will search the Internet for the answer.*

*-Once Answer is found, BabbleBot will store what it learned from the internet and respond to the Users initial query.*



# Non-functional Requirements

## Performance Requirements

-Website must be able to receive user input, up to 140 characters.

-Queries must be in English.

-BabbleBot must be able to semantically and syntactically break down the provided query.

-Using its own database BabbleBot and its understanding of the query, It will form an answer.

-If a word or any other element in the query is not understood, BabbleBot will search the interent for a definition and update its database.

-Once the answer is formulated, BabbleBot will send the response. This should happen in 30 seconds.

## Safety and Security Requirements

-We will ensure that we do not allow BabbleBot to store or utilize any foul language. This can be done by specifying words that we have identified beforehand that it should not use.

-Ensure that we protect our code, so that it does not get copied. We will be able to mitigate these risks if we severely limit access of the users. Only allowing users to interact with BabbleBot.

## Software Quality Attributes

**4.3.1 Adaptability**

It is important to for this program to be adaptable, as it is the main idea behind its functionality. The thing that would make this program different than a pre-designated answering program, is the ability to understand what the User is asking and respond with the appropriate answer. Where it adapts is when a word or query is not understood, by finding the answer to the information it doesn’t know BabbleBot is constantly learning and adapting.

**4.3.2 User Friendly**

It does not matter how ground breaking our product is, if it is extremely difficult for the user to operate or interact with. Our program requires very limited input from the User. All they would need to do is enter a query into the chat bar and wait for the response. The overall theme of the website will be easy to read and ensure that the chat bar is easy to recognize.

**4.3.3 Maintainable**

The over-arching concept of the BabbleBot is that it learns from User interaction. On the administrative side, the only maintenance that needs to be accomplished are: ensuring the website is still functioning and is still compatible as browsers update, fixing any problems that may arise from the different logic checks, and checking the database to verify that there are no anomalies or foul language.

# Other Requirements

<This section is **Optional.** Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>

GroupMe discussion – September 25, 2017: discuss final decisions on additional two functions to be included in project

Group Meeting – September 27, 2017: agree on overall details to be discussed in each section and subsection. Assign subsections to be completed by each group member.

Group Meeting – September 29, 2017: review subsections written by each group member. Provide input from all group members and edit.

GroupMe Discussion – October 2, 2017: discuss final submission and talk about any last-minute edits that need to be completed before submission.