****

**Software Design   
 Specification (SDS)**

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: | 10/22/2017 |

Contents

[Revisions ii](#_Toc117484243)

[1 Introduction 1](#_Toc117484244)

[1.1 Purpose 1](#_Toc117484245)

[1.2 System Overview 1](#_Toc117484246)

[1.3 Definitions, Acronyms and Abbreviations 1](#_Toc117484247)

[1.4 Supporting Materials 1](#_Toc117484248)

[1.5 Document Overview 1](#_Toc117484249)

[2 Architecture 2](#_Toc117484250)

[2.1 Overview 2](#_Toc117484251)

[2.2 Component 1..n 2](#_Toc117484252)

[3 High-Level Design 3](#_Toc117484253)

[4 Low-Level Design 4](#_Toc117484254)

[4.1 Modules Overview 4](#_Toc117484257)

[4.2 Module Specifications 4](#_Toc117484258)

[4.2.1 Module X1 4](#_Toc117484259)

[4.2.2 Module X2 5](#_Toc117484260)

[Appendix A – Group Log 6](#_Toc117484261)

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Morgan McKeithan, Eric Sengmany, Jeremiah Smith, Ashley Fraley, Daehan Barnes | Complete first version of the BabbleBot Software Design Specification document | 10/22/17 |

# Introduction

## Purpose

This document provides a complete example of the first version of a Software Design Specification document for the machine learning chat bot, BabbleBot. This document is primarily based on the SRS document that was earlier prepared. In the remainder of section 1, we specify the purpose of this document, the overview of BabbleBot, and the sources used in the production of this document.

## System Overview

BabbleBot Software System runs is accessed through a website. The minimal requirements related to the user accessing the website are listed in section 2.4 (Operating Environment) of the BabbleBot SRS v. 1.0 document. The system will make use of the BabbleBot database by sending user input to the program and accessing the database to formulate a response to then send back to the website to show the user. The database will interact on the Internet with Scrub dictionary in order to gain the part of speech of new words to be entered into the database. The main purpose of BabbleBot is to educate users on sentence structure.

## Definitions, Acronyms and Abbreviations

AI – artificial intelligence

ML – machine learning

## Supporting Materials

**The following standards apply**:

J-STD-016-1995 IEEE/EIA Standard for Information Technology, Software Lifecycle Processes, Software Development, Acquirer-Supplier Agreement

IEEE-STD-P1063 IEEE Standard for Software User Documentation

**The following texts and documents have been used in the process of developing this document**:

[1] SDS for MiniThermostat template

## Document Overview

The next section of the BabbleBot SDS v. 1.0 provides the architectural view of the system. It shows the BabbleBot system broken down into subsystems and the reasons for each subsystem. Subsections of section 2 describe the subsystems and their corresponding interfaces. Section 3 provides a control view of BabbleBot and describes the details of each state of the system. Section 4 provides a low-level design of the system.

# Architecture

This section provides the architecture design of the BabbleBot software system. It includes the final version of the system component diagram which shows the different subsystems, their interfaces, and their dependencies on related subsystems.

## Overview

<This section provides a high level overview of the structural and functional decomposition of the system. Focus on how and why the system was decomposed in a particular way rather than on details of the particular components. Include information on the major responsibilities and roles that the system (or portions of it) must play.

TO DO: This section is a much more detailed version of section 1.1.

1. Provide a high level component diagram with all of the required interfaces.

2. Provide a more detailed explanation to the reasons that led you to break the system down in that particular way.

3. Make sure to talk about the non-functional qualities achieved by this Architecture. These non-functional qualities should be: Maintainability and Understainability.>

## Website

Babblebot will be showcased on an aesthetically pleasing platform to capture user attention and allow for easier conversing.

* Input box
* Conversation History box (Output)
* Sentence structure diagram area (Output)

## Input to BabbleBot

Whether it is a question or statement, BabbleBot will read in user input and analyze it.

* Send input from website to BabbleBot program

## BabbleBot Database

The BabbleBot database consists of stored words and their parts of speech. The stored words are words that have been previously inputted to BabbleBot. After each new input, BabbleBot will add to the database in order to improve itself.

## Sentence Structure

## Conversation History

# High-Level Design

<This section describes in further detail elements discussed in the Architecture. Normally this section would be split into separate documents for different areas of the design.

High-level designs are most effective if they attempt to model groups of system elements from a number of different views.

TO DO:

1. Explain the purpose of this section and provide an overview of the following sections.
2. You will be using Statechart diagrams to represent the high-level control view of the system. Provide an overall system Statechart here that illustrates how the interfaces in the Component Diagram will be used (as labels of the transitions) to provide the whole services of the system. Explain in words the different states and transitions from one state to another.>

# Low-Level Design

## *<This section provides low-level design descriptions that directly support construction of modules. Normally this section would be split into separate documents for different areas of the design.*

## *TO DO: Provide a brief introduction, explaining the purpose of this section, strategies, methodologies and techniques used to obtain the suggested modular structure. >*

## Modules Overview

*<This section provides the reader with a brief overview of the modules that comprise the entire system.*

TO DO: Use the following template to describe every module in your system.>

|  |  |
| --- | --- |
| **Name:** | *The name of the module* |
| **File Name:** | *The file name for the module* |
| **Naming Convention:** | *The specific naming convention (prefix, or postfix) used to identify functions related to this module.* |
| **Short Description:** | *A short description of the module – include the main tasks performed by the module, etc.* |
| **Container Component:** | *The name of Component in which the module is located. Also explain the rationale for this design decision.* |

## Module Specifications

*<This section refers to two major types of module specifications. The first concentrates on module interface and the second on its design. In the following sections you will provide a detailed description of the module interface and its design. You will illustrate its design using Statecharts.*

*TO DO: Start with providing a short introduction of what the reader should expect to find in this section. >*

### Module X1

*< There are many different techniques used to specify both – the interface and the design. For interface the techniques can be TDN, GDN, etc. For the design a pseudo-code might prove to be useful. In this template we use a hybrid of several different techniques to specify the interface, and Statechart to specify the design. Remember, module interface is like a tip of an iceberg, it should only show what the others must see.>*

*TO DO: Use the following template to specify the module interface for Module X1.*

|  |  |
| --- | --- |
| **Used External Modules:** | List the modules this module has a USES relation with. |
| **Used External Data Type:** | List the data types, provided by other modules, that this module uses, that will prove to be important in understanding its interface or design. |
| **Internal State Variables:** | List the module’s internal state variables. |
| **Internal Constants:** | List (if any) the internal constants |

|  |  |
| --- | --- |
| **Exported Function(s)** | **Description** |
| The name of the functions | Provide a description of the function, specifying its inputs, outputs and tasks it performs. |

|  |  |
| --- | --- |
| **Internal Function(s)** | **Description** |
| The name of the functions | Provide a description of the function, specifying its inputs, outputs and tasks it performs. |

*<TO DO cont-d: Provide a* ***Statechart*** *that will illustrate the detailed internal design of the module, i.e., what events cause the transitions of the stated above internal state variables.>*

### Module X2

<The same as above>

Appendix A – Group Log

GroupMe discussion – October 18, 2017:

Group meeting – October 19, 2017:

GroupMe discussion – October 21, 2017: