CSE AUTO-RESPONSE BOT

Group 11

Anh Nguyen - Emily Wang - Trieu Hung Tran - Yuxuan Luan

Software Requirements Specification

Document

Version: (3) **Date: 11/12/2017**

Revision History

Version	Description of Change(s)	Author(s)	Date
1	Initial draft of this requirement specifications	Anh Nguyen, Emily Wang, Trieu Hung Tran, Yuxuan Luan	09/10/2017
2	Revise requirement specifications and edit the functional requirements for the implementation process	Anh Nguyen, Emily Wang, Trieu Hung Tran, Yuxuan Luan	10/05/2017
3	Add the revision history table and edit the functional requirements for the testing process	Anh Nguyen, Emily Wang, Trieu Hung Tran, Yuxuan Luan	11/12/2017

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the CSE Auto-Response Bot. It will explain the interfaces, constraints, and functions of the application. It is intended to be a reference to develop the first several versions of the system.

1.2 Scope

The CSE Auto-Response Bot gives CSE staffs and professors more time to focus on complicated questions asked by the students that need further advanced solutions instead of wasting too much time on repeated questions. By using the built-in database provided by CSE department, the system can automatically recognize and retrieve the most appropriate answers for different questions from the students or notice the system managers that there might be new questions or the ones that need further support. Beside its advantages in student support system, the CSE Auto-Response Bot can also use its ability to automatically extend the database. By this way, it can respond to different type of requests in multiple styles; hence, students will feel of it as a human being when having the conversation.

The CSE Auto-Response Bot is first implemented to work well on all Windows OS devices. It needs the Internet to interact with the database as well as to let the system managers know about complicated questions.

1.3 Definitions, Acronyms, and Abbreviations.

Term	Definition
Cross-platform	Able to be used on different types of computers or with different software packages
CSE	Computer Science and Engineering
GUI	Graphical User Interface
Java	An Object-Oriented programming language
JRE	Java Runtime Environment
MB	Megabyte - a unit of information
Object-Oriented	Using a methodology that enables a system to be modeled as

	a set of objects that can be controlled and manipulated in a modular manner
OS	Operating system
System Managers	People who are given specific permission for managing and controlling the system

1.4 Overview

The remainder of this document includes two main sections, a maintenance note and an appendix. The first section provides an overview of the system functionality and interfaces. The second one describes the requirement specifications in detail including all function designs and system attributes. The appendix in the end of the document contains the application's expected user interface.

2. The Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

The CSE Auto-Response Bot does not require any additional system to be able to run on all Windows OS computers.

2.1.2 Interfaces

The first release of CSE Auto-Response Bot provides a simple GUI for users to interact with the system. With the support of computer's input devices such as keyboard and mouse, it allows users to enter their questions and navigate through the application.

The application may be downloaded and installed on mobile devices in future releases when users can use the touchscreen to communicate with the system.

2.1.3 Hardware Interfaces

The CSE Auto-Response Bot is designed to be compatible with all Windows OS computers at first and with mobile OS such as iOS or Android in subsequent releases.

2.1.4 Software Interfaces

The CSE Auto-Response Bot is going to use a simple GUI that requires a working computer for installation and utilization to interact with users.

2.1.5 Communications Interfaces

All users must use their CSE account and password to log in. This application needs the Internet to request, modify, and retrieve data stored on the CSE server.

2.1.6 Memory

The CSE Auto-Response Bot only needs a simple GUI design which does not use much memory.

2.1.7 Site Adaptation Requirements

All data tables created for this system must be stored on CSE server to make it more convenient to modify and retrieve data in execution time. The CSE Auto-Response Bot also requires a working computer with JRE installed for installation and utilization.

2.2 Product Functions

The CSE Auto-Response Bot is designed to be one of students' best friends who will help or show them how to get help for frequently problems in college life. In the first release, it

should be able to help all CSE students, staffs, and professors to quickly search for an answer of frequently questions or classes' descriptions from any related words.

2.3 User Characteristics

Because of its dependency on CSE department's database, this program is first designed mostly for CSE students, staffs, and professors. If permitted, future releases may include the database from other departments as well. There should be a basic tutorial for first-time user. Due to the popularity of automatic texting, everyone should know how to chat with a bot.

2.4 Constraints

Using a GUI created by Java - a cross-platform programming language, there will not be any concerns or limited to any working computer. However, Internet connection will be a constraint because of the interaction between this application and CSE server. In addition, the CSE Auto-Response Bot is limited by the capacity of the database. The database cannot cover all questions provided by users but it takes time for the system to learn and update. Besides, since the database is shared between different users, the application's processes may delay when there are many requests at the same time.

2.5 Assumptions and Dependencies

Chatbot program only depends on Java. Therefore, as long as Java does not have any big update, the application will be supported for all current and future computer operating systems.

2.6 Apportioning of Requirements.

The core function of the Chatbot program is searching for the questions provided by users in the database to retrieve the most related answers to the interface. They should be completed in the first version of the application. Login, report and rating functions could be delayed if time is in rush.

3. Specific Requirements

3.1 External Interfaces

In subsequently releases, users should be able to see the login page (Appendix 1) when the application is opened. They are required to enter their CSE username and password to be able to interact with the system.

The main page (Appendix 2) should appear after users enter their CSE account correctly. If users are students, they should find a textbox to enter their questions and a space to display the whole conversation. If users are CSE professors or staffs, there will be another option for them to choose which is updating the database. In the updating page (Appendix 3), they can enter a new question along with its answer.

In the first release, the main page will appear no matter what users type in the CSE account spaces and all users will be treated as CSE professors and staffs which allows them to access the updating feature.

3.2 Functions

3.2.1 First Release

3.2.1.1 Chat

3.2.1.1.1 This is the main function of the whole application which retrieves the most appropriate answers to user's input. Note: Users can only retrieve answers typed in English.

3.2.1.1.2 There will be a white blank text-box for users to input their questions and a space to display system's results which are retrieved by a searching algorithm using the input's keywords.

3.2.1.1.3 Associated functional requirements:

3.2.1.1.3.1 The system will be able to check for empty input.

3.2.1.1.3.2 The system will be able to identify if the input question is too broad/not in the database/matches one or several questions in the database and inform users with an appropriate message.

3.2.1.1.3.3 If the searching algorithm cannot find a specific question in the database, the system will send a report about this problem to the database. The system manager can then use the reports to update the database appropriately.

3.2.1.2 Update

3.2.1.2.1 This application uses the auto-update function to extend its database so that it can handle different types of question from future users.

3.2.1.2.2 Users can navigate to the update screen from the chatting screen. There will be two blank text-box for users to input their question and the corresponding answer.

3.2.1.2.3 Associated functional requirements:

3.2.1.2.3.1 The system will be able to check for empty question and/or answer.

3.2.1.3 Log in

3.2.1.3.1 This function gives the system the ability to manage its users and limit the range of incoming questions to CSE related problems.

3.2.1.3.2 After starting the application, a login page will display which requires users to input their CSE username and password to be able to interact with the Bot.

3.2.1.3.3 Associated functional requirements:

3.2.1.3.3.1 The system will be able to check for invalid username and/or password.

3.2.1.4 Report

3.2.1.4.1 Users can report any errors that appear when using the application so that the developing team can handle them as soon as possible.

3.2.1.4.2 Users can click on the report button on every frame to type in and send their problems to the system.

3.2.1.4.3 Associated functional requirements:

3.2.1.4.3.1 The system needs a database to store all the reports from users.

3.2.1.4.3.2 The system will be able to check for empty reports.

3.2.2 Subsequent Releases

3.2.2.1 Advanced update

3.2.2.1.1 This function will be used to upgrade the original update system described in the first release.

3.2.2.1.2 Users (Professor/Staff) will not be limited to only adding new questions to the database but also be able to remove or edit existing questions.

3.2.2.1.3 Associated functional requirements:

3.2.2.1.3.1 The system will be able to retrieve all questions or make use of the searching algorithm mentioned in the chatting system to search for a specific question from the database.

3.2.2.2 Report management system

3.2.2.2.1 This function will allow staffs and professors to manage all reports from users.

3.2.2.2.2 Users can look at the list of all reports and remove resolved problems.

3.2.2.2.3 Associated functional requirements:

3.2.2.2.3.1 The system will be able to retrieve a list of all reports and match each of them with the corresponding users who sent the report.

3.2.2.3 Rating system

3.2.2.3.1 Each response from the system can be rated from 1 to 5 stars based on its correctness and helpfulness for users. The system manager can then look at the ratings update bad answers to the more suitable ones.

3.2.2.3.2 After an answer is retrieved by the system and sent to the screen, users can click on it and choose a rate between 1 and 5 for its correctness and helpfulness.

3.2.2.3.3 Associated functional requirements:

3.2.2.3.3.1 The relationship table which established the relationship between questions and answers needs an additional column which will store the rating for each question-answer combination.

3.3 Performance Requirements

Because the program only perform text processing on input questions and communicate with the database, there will not be any lag or pause. However, if there are too many questions requested through the server at the same time, it may take a little bit longer to retrieve the answer.

3.4 Logical Database Requirements

In the first release, this application's database will be designed to fit into one table in general where each row consists of one question and its appropriate answer. In future releases, different types of questions/requests may have different tables if needed for clarification. Data may be not only text but also images, videos, or locations, etc. Accounts and Reports tables should be created to store all information for login and reporting system.

The system will search through the database to retrieve an answer whenever a question is sent by users. Only CSE professors and staffs can request to extend the database.

3.5 Design Constraints

The CSE Auto-Response Bot is planned to use no more than 10 MB of the hard drive and no more than 10 MB of memory usage. The database which is first stored in the CSE server can have up to 500 MB in space; however, it can be optimized to increasing system performance.

Another constraint of the system is that it will be made for Windows OS for the first release, so it will only be available on Windows OS computers. Other OS may be supported in future releases.

3.6 Software System Attributes

3.6.1 Reliability

The CSE Auto-Response Bot must be able to retrieve valuable answers for 80-90% of valid questions from users and it is expected to reach 95-100% after several improvements in the future.

3.6.2 Availability

The working computer should be connected to the Internet and should have access to CSE server in order for the application to communicate with the database. Besides that,

the whole system should run perfectly at all time. Even if there happens to crash or bug, users can reopen the application easily.

3.6.3 Security

Users must have their CSE accounts to be able to interact with the system. There will be instruction from CSE department on how to create an account for new user. The question text that send from the user should be sanitised. Any text statement that will modify or damage the database should be prevented from the system.

3.6.4 Maintainability

There will be a monthly maintenance of database for this system. Moreover, future upgrades will be adding more database from other departments and more classes throughout the whole university by using pre-built functions. Test environment should be built in each function for unit testing.

3.6.5 Portability

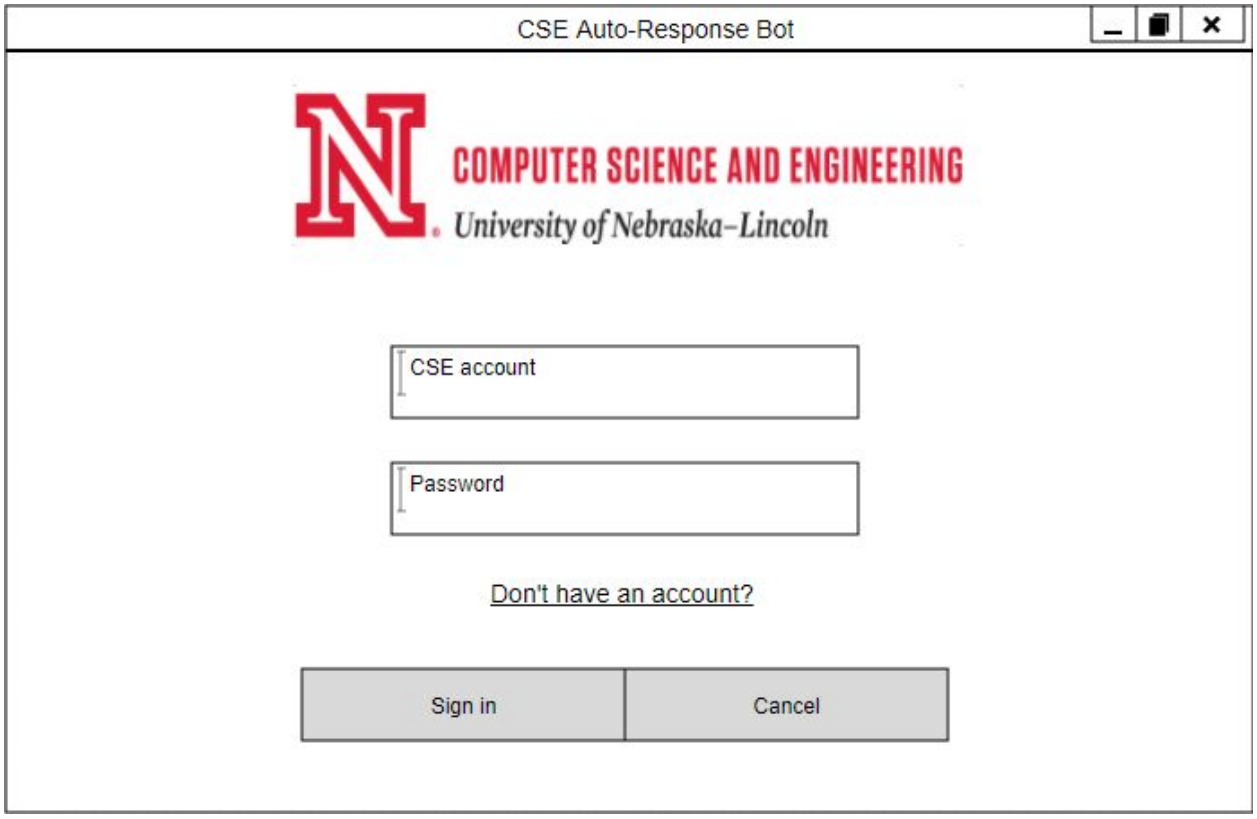
The CSE Auto-Response Bot is designed for Windows OS so that it should run perfectly on all Windows OS computers.

4. Change Management Process

When using this system, users can easily evaluate how reliability of the application. To change or upgrade the system, users should contact the maintenance team with all information that they want to include. For example, if university wants to scope out the system to all departments, database team need to ask for permission to access all departments' database, after that there will be a team meeting on how to spread out the work.

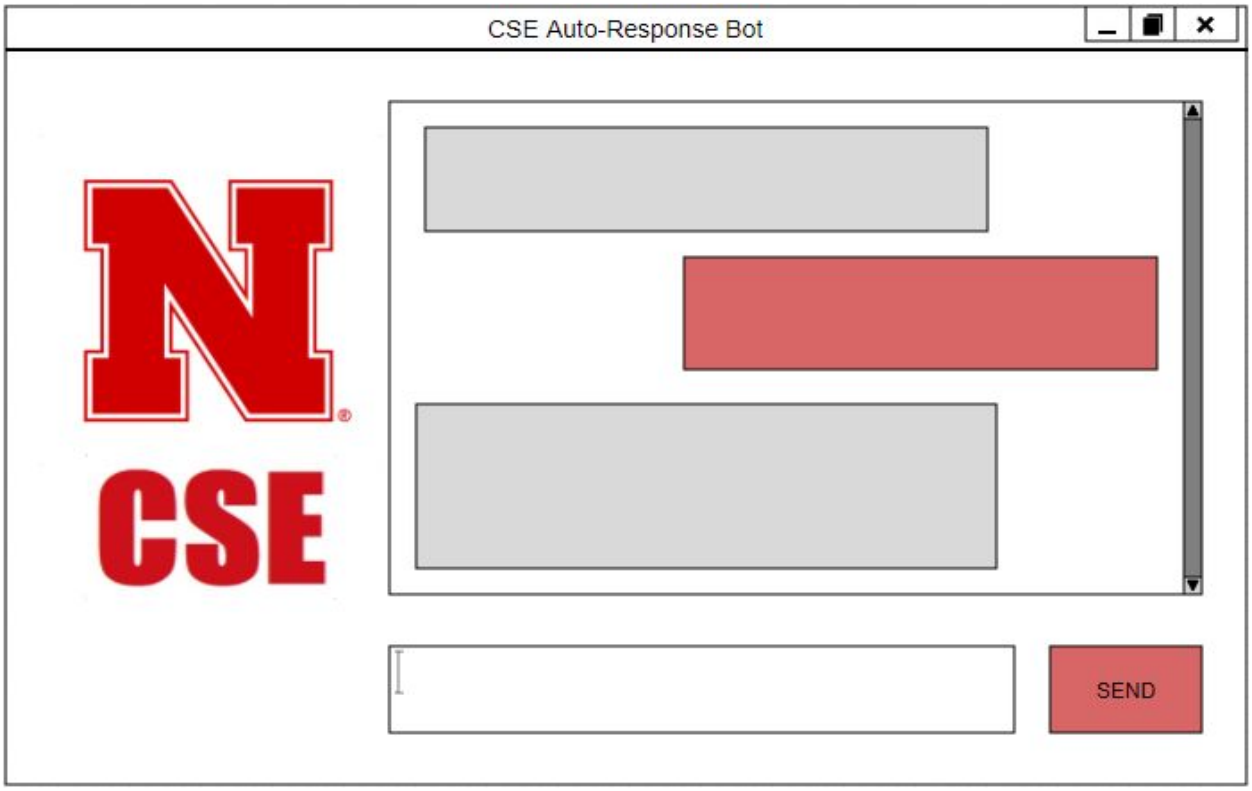
5. Appendix

5.1. Login Page

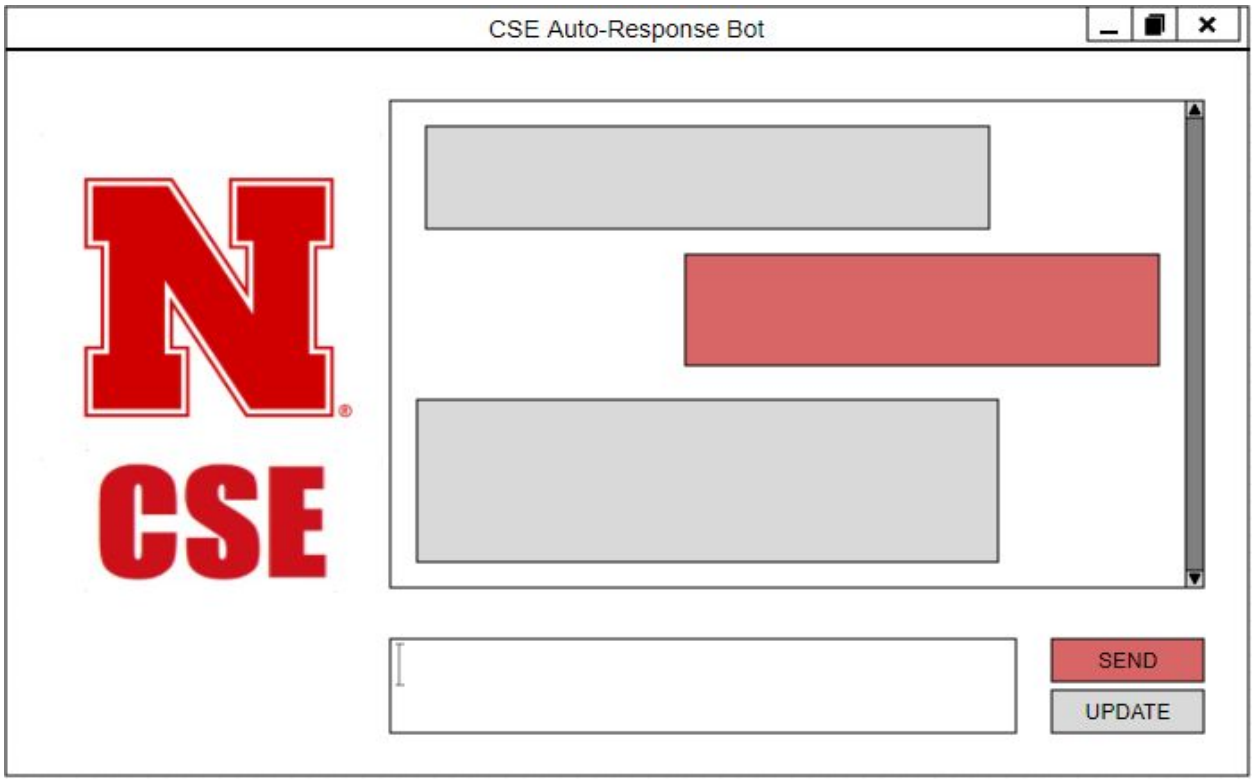


5.2. Chatting page

5.2.1 For CSE Student



5.2.2 For CSE professors and staffs



5.3 Updating page