Software Requirements Specification

For

Dr. Deb McAvoy

Russ College Chatbot

Version 1.0 approved

Prepared by Team TBA

Jonathan Feige

Joshua Wright

Michael Dinger

Logan Gordon

10/08/19

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 2

3. External Interface Requirements 2

3.1 User Interfaces 2

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 3

4.1 Chatbot Functinality 3

4.2 Web Functinality 3

4.3 Email Implementation……………………………………………………………………..4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 4

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 6

Appendix B: Analysis Models 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Original | 10/01/19 | Original change | 1.0 |
| Grade update | 10/08/19 | Responding to critique of documentation | 1.1 |
| Semester 2 Update | 1/17/20 | Updates for changes over the course of the semester | 2.0 |
| Finals Update | 4/13/20 | End of semester document updates | 3.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

## The Russ College Chatbot, nicknamed Russ Rufus, is a tool to quickly/accurately disseminate information for application and enrollment to the University.

## Document Conventions

Each Higher-level requirement should be treated as having its own priority. These requirements shall be listed in the order of their priority in this document.

## Intended Audience and Reading Suggestions

This document is intended for development team members, the client, and the administrative supervisors. The SRS contains an overview of the chatbot’s requirements and specifications as well as a full description of the features. The document ends with other requirements for safety, security, and quality purposes.

The team members will find the entire document relevant and should read from beginning to end, paying special attention to any information that needs updating or revising, making changes as necessary.

The client may want to focus on the scope (1.4) and the features (4.) portion of the SRS. These will ensure the current version is in line with the client’s vision for the chatbot as well as give the best reference on the status of the project. The remaining information on the additional capabilities of the chatbot will give a more complete understanding of the total product being delivered.

The administrative supervisors are suggested to read all the material, skimming for flaws while understanding that this live document is a work in progress and may simply need updated to reflect the current product.

## Product Scope

The Russ Rufus Chatbot is intended to increase long-term enrollment from high school students transitioning to college by reducing the number of prospective students that do not go to college after the summer. This problem has been identified as being in part caused by a lack of easily available information. This chatbot serves to provide information quickly and concisely to help reduce the drop off caused by anxiety over the long summer break. It also works as a tool to give scholarship and enrollment information increasing overall enrollment.

# Overall Description

## Product Perspective

The Russ College Chatbot is designed to be a self-contained product that can be attached to any webpage. The current destination for the chatbot is a new recruitment focused portion of the Ohio.edu domain1.

## Product Functions

Read questions provided by the user and supply an appropriate answer and information or refer them to a human that can further help them.

## User Classes and Characteristics

* Student: the main user of this product and whom it was designed to be used by as they are the ones who have the questions meant to be answered by the chatbot.
* Developers: the people who will maintain the code, have access to the product’s database of questions and answers, and will be the ones who add to it.

## Operating Environment

The product will run on top of the Russ College Website.

## Design and Implementation Constraints

The product is limited to a set database of questions and answers, so it will have to be maintained by an administrator.

## User Documentation

The product will be documented using PEP 8.

## Assumptions and Dependencies

The product runs using python3 so appropriate libraries will be necessary. It is also dependent on Chaterbot Corpus and Chaterbot.

# External Interface Requirements

## User Interfaces

The chatbot will exist at the bottom right of the webpage. It will exist in a circle icon marked “R2”. On click, the icon will expand to a small window taking up the corner of the webpage. This will have a bar across the top that says “Russ Rufus”, and a box underneath that has a space for the chatbot conversation. At the bottom of the box there will be a space to type and an arrow to submit your questions for Russ Rufus. (Fig. 1).

## Hardware Interfaces

The chatbot will be required to be hosted alongside the existing website. The server may need extra instillation before the chatbot can be fully functional alongside the website.

## Software Interfaces

This software will use Python3, CSS, and HTML5 and the python library Chatterbot. It will be compatible with Windows, MacOS, and Linux. All data will be handled internally as it will be preloaded on the execution of the application.

## Communications Interfaces

The chatbot will request a name and email for follow up questions, if needed. To access the chatbot the user must have a web browser and the ability to access the Russ College of Engineering Website. As the chatbot will be added to an existing webpage, security will fall on the responsibility of both team TBA and the website administrators.

# System Features

## Chatbot Functionality

4.1.1 Description and Priority

A chatbot system that will take a call and return the appropriate response given for a database. This being the first and highest priority.

4.1.2 Stimulus/Response Sequences

A user will be able to see the chatbot and understand its use. The user will make a call in the form of a question and a response will be made from the bot.

4.1.3 Functional Requirements

REQ-1: Install all dependencies.

REQ-2: Collect information for database.

REQ-3 Implement database for a command line chatbot.

## Web Functionality

4.2.1 Description and Priority

The chatbot will seamlessly integrate with a webpage for a better user experience. This is the second highest priority.

4.2.2 Stimulus/Response Sequences

A user will interact with the chatbot through this web interface. By clicking the R2 you will be introduced to the chatbot interface.

4.2.3 Functional Requirements

REQ-1: Designing the R2 chatbot logo

REQ-2: Adding a web-based user interface to the command line bot.

REQ-3: Removing the command dependence

REQ-4: Integration with the already existing website for full functionality

## Email Implementation

4.3.1 Description and Priority

The chatbot will automatically send a notification to the administrator whenever it doesn’t know the answer to a question. This is the third requirement for the chatbot.

4.3.2 Stimulus/Response Sequences

This will require no user response. If the chatbot is not confident in its answer, then it will redirect the question to the administrator with the email attached of the person with the question.

4.3.3 Functional Requirements

REQ-1: Design a system of storing short term emails

REQ-2: Design a system of sending emails to an administrator.

REQ-3: Filtering out bad or unwanted responses.

REQ-4: Minimizing the amount of spam / fake responses coming through.

# Other Nonfunctional Requirements

## Performance Requirements

The performance of the chatbot is the determinant for user retention with the goal of exceeding an 80% user retention rate. Our client also requests an 8.0 second maximum response time. To meet these goals our chatbot’s response time will have a 7.2 second maximum response time, a reactive UI for text interaction with the chatbot. Metrics to capture our attainment of these goals will be the following: total users, active users, engaged users, and new users. More metrics for measurement of our chatbot’s performance are messaging metrics and overall performance metrics which will be computed to ensure our goals stated above are reached. The message metrics are the following: amount of conversation starter messages, bot instruction messages, input messages, missed user messages, total conversations, and new conversations. Using this information, the performance tracking can compute retention rate, goal completion rate, goal completion time and messages, fall back rate, and user satisfaction.

## Safety Requirements

The only safety requirement for the chatbot is to untether all user input from bot training. This will remove the possibility of real-time bot adjustment to better respond to user input, but it will also remove the possibility of intentional user manipulation to access any sensitive data that is potentially introduced as a response from the chatbot.

## Security Requirements

Security requirements for the Russ College Chatbot are limited to using data protection methods to ensure privacy of user email and passwords used to access the chatbot. The client has requested the ability to have users register so missed conversations or contact with a human can be moved to email-based interaction. When users register with their email and/or password, this information should be collected, stored, and transmitted in secure fashion to protect the privacy of user conversations or desired communication with Russ college administrators.

## Software Quality Attributes

The following characteristics are software quality attributes targeted for developers, maintainers, and users: Maintainability, reliability, robustness, testability, and usability. Maintainability is handled by conversation editing tools. Reliability will be based on the performance-metrics specific in section 5.1. Robustness is a metric of user satisfaction with the goal of being greater than 90%. Testability for the chatbot will be measured by the amount of test scripts for each conversation category. All different conversation and information domains1 must have a 100% completion success for every potential response to user input that the bot is trained to handle. This includes the exact phrasing of bot training data and three to five synonymous phrases. Usability of this chatbot will require the chatbot to be accessible to those with and without physical disabilities. This means that the chatbot must be accessible to persons who are deaf, blind, or physically limited in their interaction with mobile devices or traditional mouse-and-keyboard for the web-based solution. Also, the Russ College Chatbot must be branded to Ohio University, and targeted to Ohio University’s current and prospective students. This is a design and user experience attribute.

## Business Rules

The following business rules will be used for maintaining software after the initial development phases of the chatbot:

* Website maintainers oversee updating chatbot integration as new releases of the webpages hosting the chatbot are rolled out.
* Additions to conversation scripts will be added by developers initially. After this initial training, conversation scripts can be added to and edited by the technical maintainer after initial developer work is completed. A format guide and document outlining proper semantics will be included with the program or interface used for script contributions.
* Requests for human or in-person points of contact will be directed to Dr. Deb McAvoy
* Missed conversations will be made as conversation additions

# Other Requirements

**Database** There is the potential need for a database to serve as storage for conversation responses for the bot. It may also serve as storage for missed conversations which can be used for continual improvement of goal success rates of bot conversations.

**Legal** The potential legal requirements of this chatbot include a disclaimer that communicates why an email and password may be collected to user metrics and to facilitate interaction between The Russ College and potential or current students. Since this information will need to be stored somewhere, the user will need to acknowledge that they agree with the use of their email address as a communication channel between the college and users.

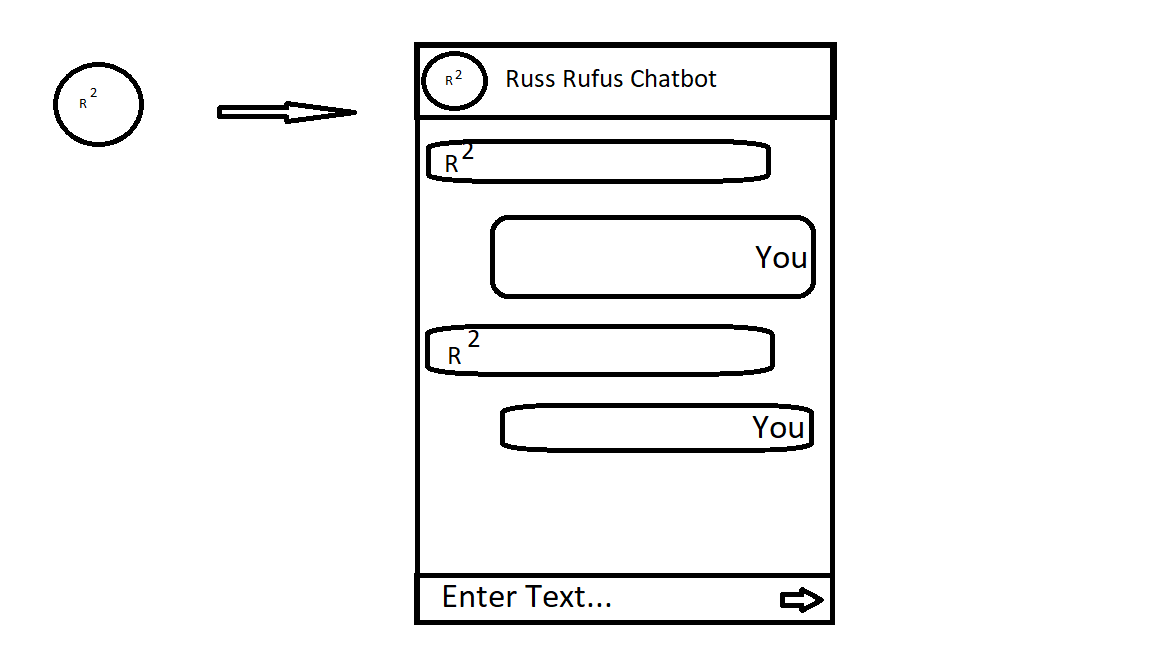
Appendix A: Glossary

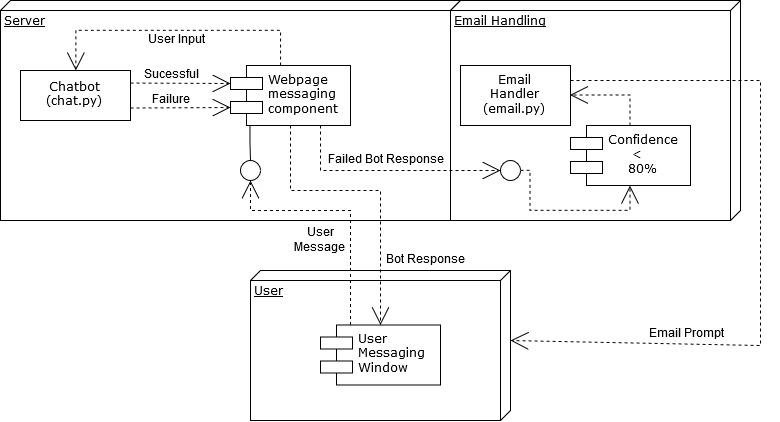
Domain1: A group of computers on a network with common rules.

UI*2: Short for User Interface, how the user and a computer system interact.*

Appendix B: Analysis Models

Chatbot Diagram (Fig. 1):

Data Flow diagram (Fig. 2)



Data Flow diagram (Fig. 3)

