Daniel Rodriguez

Daniel’s Chat Application

Requirements Document

Table of Contents

Introduction

The Daniel's Chat Application Requirements Document outlines the expected functionality of Daniel's Chat Application including expected users and behaviors. The chat application is a program designed to allow two or more users to communicate with each other in real time with the user of a server to relay messages between them. Functionalities include logging in and out of the program, viewing online users available to chat, and sending and receiving a message.

Purpose and Scope

The Requirements Document allows the chat application programmer to write out all the information and expected behaviors of the program to ensure that everything assigned is being accounted for in the design process. The Requirements Documents allows the programmer to create a sort of checklist to compare what is expected versus what is being planned for the design. If all bases are covered, the programmer, and grader, can be assured that the project is on the right path and fewer errors can be expected as the project progresses.

Target Audience

The Requirements Document is intended to be reviewed by both the programmer himself, to review his understanding of the assignment, as well as the professor and teaching assistant, as they will be able to see the programmer's understanding of the assignment and monitor the project's progress.

Terms and Definitions

Chat Application - refers to Daniel's Chat Application, or the main program assigned for this project. It will be hosted on a local computer and communicates with the program server, which contains all the data for the program. Also referred to as "program."

Home Screen - Once logged into the chat application, the default screen is the home screen which includes a list of online users, menu options, as well as a text field--used to send a message out to the entire list of online users.

Program - refers to Daniel's Chat Application, or the main program assigned for this project. It will be hosted on a local computer and communicates with the program server, which contains all the data for the program. Also referred to as "chat application."

Server - a non-local computer that stores the underlying data and data structure for the chat application. A user must successfully login to their account in order to communicate with the server

User - a person at a local computer who is communicating with the chat application, which communicates with a server

Product Overview

Give a high level description of the functionality of the project here. Describe the purpose of this section. It may be useful to give your definition of a user, a stake holder and a use case. If there are scope limitations to the project, i.e. things you will not be doing, or are not required to do, this is a good section to put those.

Users and Stakeholders

The Users and Stakeholders section will define the audience of this program, or who is expected to use it, as well as the parties involved who have an interest in the success of this program.

Daniel Rodriguez, Programmer

Daniel Rodriguez is the sole program designer and programmer for this program. It will be his responsibility to create a project overview, a more detailed outline of requirements, design and analysis, test plan, as well as the final project deliverables, including the actual coding of the project. He will oversee all testing and development of the project, ensuring it meets the time constraints and goals given.

Professor Fei Xie and TA Bin Lin

The CS300 Professor Fei Xie and TA Bin Lin will be reviewing and scoring the project throughout its development--reviewing the Requirements Document, Design Document, Test Plan, Project Report and final deliverables, upon project completion.

Use cases

The following use cases outline the expected behaviors and functionalities of the chat application project. Participants for each functionality will be identified and their roles outlined.

Logging In/Registering User Account

A human user will open the chat application program and enter in login information which will be communicated to a server on the other end, which will verify or add login information in the data structured housed within it. It is the server's responsibility to manage all underlying program data.

Sending/Receiving Messages

A logged in human user will select users to send a message to. Upon submission of message, the local computer will send information to the server on the other end, which will be responsible for managing the behavior of that data--sending message to the correct user or verifying receiving user information.

Logging Off

A human user will indicate that they are finished using the chat application, the local computer will communicate this with the server which will mark the user as offline and store user information appropriately.

Functional Requirements

The following section outlines the functional requirements of the chat application project, or the functionality that is expected from the program in order to be successful and fulfill all requirements.

User Login

Account registration is required in order to connect to the server and utilize the chat application. Upon starting the chat application, the program will show two text fields, prompting the user to enter a username and a password.

User Has A Registered Account

If the user has previously registered an account in the chat application, they should type in their previously registered username and password and submit. The program will send the inputted information to the server which will traverse the underlying program data structure, searching for a matching username and password combination. If the program is unable to find a matching username and password, the server will communicate to the program that no match was found and the user will be notified by the program accordingly. User will have the option to retry their input or register a new account (see User Registration Required below). If a matching username and password combination is found, the server will communicate to the user, via the program, that the account was found and the program will show it's home screen--which consists of a list of online users as well as a list of menu options that the user can choose from. Each user account, within the data structure, will contain a flag field which will indicate whether a user is online or not. Once a matching username and password are found, upon login, the account will be flagged as online. All accounts that are flagged as being online will appear to all users currently signed into the chat application.

User Registration Required

If the user has not yet registered an account, a link will be displayed on the login screen giving the user the option to register a new account. Clicking this link will take them to a page with three text fields: username, password, and confirm password. User will enter a desired username in the username field and a password in the password and confirm password fields. Once submitted, the program will verify that the two password fields match. If they do not, the user will be prompted to re-enter their information. If the two password fields match, the program will send the information to the server; the server will traverse the data structure of user accounts, searching for a matching username as the one the user is trying to register. If a matching username is found, the server will communicate to the user, via the program, that this username already exists. User will be taken back to the login screen to try to login again or have the option to register a new account. If no matching username is found, the server will add the new user account to the data structure and notify the user of the success of account registration. User will then be taken to the login screen to login to the chat application with the newly registered information.

View All Online Users

Once logged into the chat application, the server will be prompted to display to the user, via the program, a list of all online users--found by the flag field indicating if a user is online or not.

Send Message

The core functionality of this program is the ability to send a message to a group or individual.

Send Message to Individual User

The user may double-click on the name of an online user and a new text field will appear enabling the user to type a message. Once the typed message is submitted, the program will send the message to the server, along with the information of who it should be delivered to, find the matching user the message is being sent to, and send that message to them. The receiving user will be notified of the new message and have a text field presented to them to respond to the original message sender. The two may communicate in real time, as long as both users remain logged into the program.

Send Message to All Online Users

In the program’s home screen, there will be a consistent text field at the bottom, where the user can type in a message. Any message typed and submitted here will be sent to all online users.

Receive Message

If a user is logged in and a message is sent to them, a small chat window will open, displaying the message being sent, and it will include a text field at the bottom so that the receiving user may reply to the message. The information of who the other party is will be included and any reply message will be sent to the appropriate user.

Logoff

The chat application home screen will contain an option to log off the program. Selecting this option will change the “online” flag field to off and the user’s username will no longer be highlighted in the list of online users.

Nonfunctional Requirements

Nonfunctional requirements gives specifics for the chat application. These are not functionality specifications but rather software specifications.

Reliability

All user interactions with the chat application should be completed successfully every time. The program should be written in a way that each step is executed appropriately each time a user inputs and submits something. The only problems that could arise are if the local computer or server are having performance issues beyond the scope of the assigned project.

Programming Language

The Daniel's Chat Application will be written in Java using an IDE.

Milestones and Deliverables

This section outlines the timeline for the completion of this chat application project. There are certain benchmarks to be met and submitted to professor and teaching assistant, and those are outlined below with due dates.

Design Document

After the submission of the Requirements Document, the programmer will spend time translating all requirements into a design flow, specifying, in greater detail, the structure of the program. Data structures will be determined and outlined and how the local computer will communicate with the server and protocols will be used. More specifics will be given in regards to the interface that will be used by a user. Attention will be given to system specifications and architecture. The Design Document will be ready for submission by May 9, 2017.

Test Plan

For every use case outlined in the Requirements Document, a test plan should be developed to ensure that all scenarios have a planned path and resolution. These test scenarios will be outlined in the test plan, which will be ready for submission by May 30, 2017. At this point, the programmer will have already started coding the main program as well as these test case scenarios, as testing will need to be done as the main program code is written.

Project Report and Final Deliverables

Finally, after all the design, coding, and testing have been checked and completed successfully, a fully functioning program will be ready for submission by June 8, 2017.