Chaterbox - Chat for a generic audience

Requirements Document

Table of Contents

Introduction

Welcome to the Chatterbox Design and requirements document! This document is going to include important information, such as the what chatterbox can do and how it’s going to do it. This will include the stakeholders, target audience, functional and nonfunctional requirements, as well as deliverables.

Purpose and Scope

The purpose of this document is to layout the requirements of this assignment and a plan for dealing with them. The scope of this project is 2 months, and the budget is $0.

Target Audience

The target audience for this project are graders and potential hiring managers.

Terms and Definitions

Functional requirements are requirements that can be written as a function.
Non-functional requirements are things that I want the program to do that are not functions.

Product Overview

I aim to create a chat client capable of connecting users to one another. The user of course, being a hiring manager who will be using it to test for bugs in my code, and to see how well it works, or a grader wanting to see if I completed the project requirements.

Users and Stakeholders

Molly Shove

Molly Shove will be entirely in charge designing, implementing and maintaining the Product.

The Grader

The Grader will help evaluate if the functional requirements have been met.

Use cases

Use cases are reasons people might want to use the program

Send a message to a friend

A person might want to communicate with another person who is not in the same room with them. Chatterbox can help with that

Communicate with a group of people

A person might want to talk to a whole group of people, perhaps to plan a hangout, or just for fun. This is also something Chatterbox can help with.

Securely log onto a server

A person will want their information to be safe. Chatterbox will focus on security in order to preserve their user’s sense trust and sense of security

Functional Requirements

The purpose of this section is to inform the reader of the functional requirements of the program. A functional requirement is taken here to mean something which can be written into a program as a function.

Chat Functionality

The goal of this program is to send and receive messages from one or more users connected to the same server.

Send a message to a single user

This facilitates one on one chat between two users

Send a message to multiple users

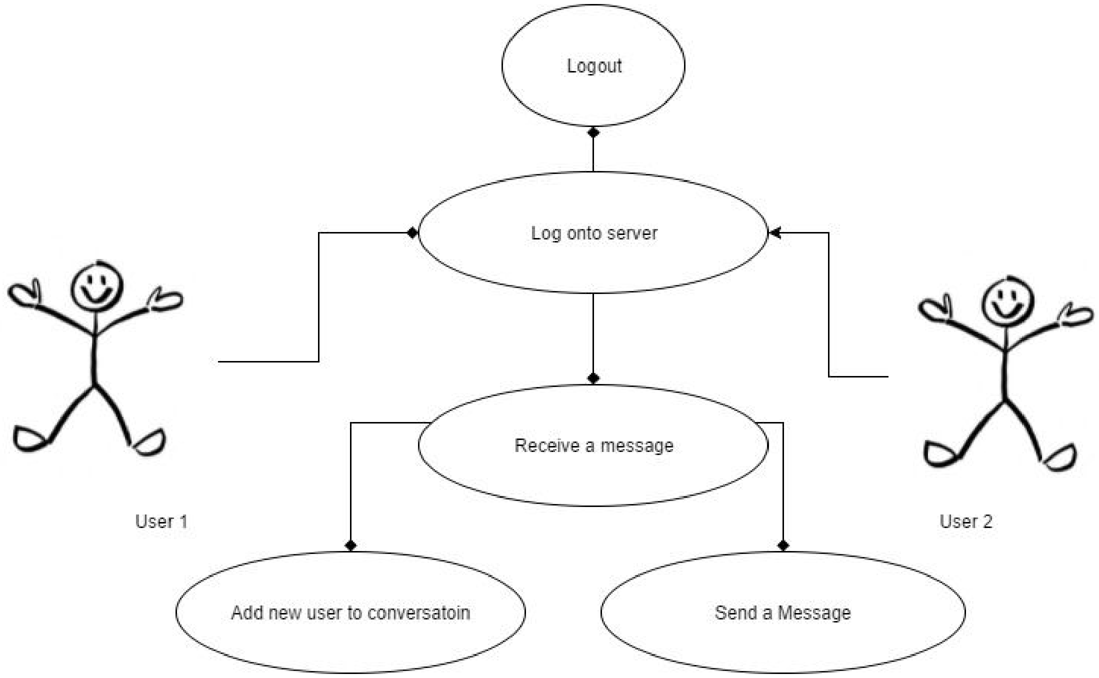
This allows users to engage in conversation with many people, which both implies group send and group receive functional capabilities

Log into the server

This part will have a number of functions, from getting the server working to authenticating a user.

Nonfunctional Requirements

This part of the document will describe the non functional requirements. In this context, nonfunctional requirements. Below is a diagram of the use cases, many of which inform the nonfunctional requirements.



User Interface

One of the most important parts of chatterbox is a functional, elegant and intuitive UI. I want bright colors, a friendly font and a functional, minimalist design.

Onboarding Process

I want a new user's first chat to be with a bot who will send a few messages telling them about the features of the program. This will teach new users the system in a friendly, immersive way. We chose to implement this because it's industry standard with popular chat apps like Slack

List of Online Users

There will be a list of currently online users which the user can choose to chat with. This encourages the user to start new conversations and hang out online to be available to chat with other users.

Secure user data

It's of utmost importance that no one should be able to use buffer overflows or other malicious methods to see a password or access user data of any kind. This means that the design of the login process will be designed to optimize security.

Milestones and Deliverables

The purpose of this section is to put a timeframe on creating the above mentioned features. A milestone is the implementation of the functional and nonfunctional requirements. The following are the deadlines already given to us as part of the class

- April 25: The requirements document is due
- May 9: The design document is due
- May 30: The test plan is due
- June 8: The project report and final deliverables are due

Here are a few deadlines based on the timing of these which can help me deliver the product on time.

- April 30th: Github set up, necessary libraries researched and imported
- May 7th: Graphical user interface tools explored and design finalized
- May 14th: UI coded.
- May 21st: Server coded
- May 28th: Backend coded
- June 2nd: Unit tests written
- June 5th: Quality Assurance

Functional UI

At this point the UI should be visible and look good, even if the user can not meaningfully interact with it yet.

Server Coded

At this point, the user should be able to log in and connect to a server.

Backend coded

This includes many of the main functions of the program, most notably sending and receiving messages from one or more other people in a conversation

Unit Tests Written

This means I have written the code I will need to automatically test the functionality of the backend

Quality Assurance

This is the step at which I debug the code. At the end of this step, the code will be fully functional and bug free