Internet Relay Reborn Chat (IRRC)

Requirements Document

Table of Contents

1 Introduction

This document outlines the requirements for the IRRC internet chat application. It introduces the primary users, most common use-cases, and all functional/non-functional requirements for the application to fulfill.

1.1 Purpose and Scope

This document is to be used to create a working design which fulfills the requirements. This document is not for development purposes, and does not describe specific details for implementation.

1.2 Target Audience

This document is to be used strictly by the design team of the application. All details described in this document are for outlining the requirements to create a working design.

1.3 Terms and Definitions

IRC is an acronym which stands for Internet Relay Chat. This is a loose definition which describes existing infrastructure for sending and receiving text over an internet connection through a relay (server). This acronym is different from IRRC (internet relay reborn chat), which is the working name of the application for which this document is written.

2 Product Overview

The IRRC is a two-application infrastructure which emulates the functionality of IRC. These two applications are server and client. One server may host multiple clients. However, any client may only be connected to one server at a time. Users log in with a username and password. If the user is new, then they may create a username and password. Once logged in, a user can either send all-chat messages to all other online users, or they may directly send private messages to any online user.

2.1 Users

The primary users of this software include the chat client users and server operators.

2.1.1 Server Operators

The server operators are the users who run and maintain the server application. Setup and use for the server application is to be as simple as possible.

2.1.2 Chat Users

The majority of IRRC's users are expected to be client application users. This application is to be created for a general audience, therefore the user-interface should be clean and intuitive.

2.2 Use cases

This section will briefly describe the most common situations that will happen during use of IRRC.

2.2.1 Login

Upon starting the client application and specifying a server to connect to, the user will be greeted with a login screen. If it is the first time the user is connecting to this server, they may create a new username and password by choosing the "New User" option below the login prompt. Shown is a non-functional prototype of login page to show placement of elements on the screen.

2.2.2 All-Chat Messages

Once the user is logged in, they can send and receive messages on the server. The default channel is all-chat, which is a server-wide relay where all messages can be seen by all users. This relay is also the channel where server notifications can be read, such as when users come online or server announcements.

2.2.3 Direct Messages

In addition to sending and receiving all-chat messages, users may also send private messages to each other. The user can see all other online users in a list on the left-hand side of the application window. If the user wishes to send a private message to a specific user, all they need to do is double-click the desired user's name to open a private channel.

3 Functional Requirements

Which section will describe the main functional requirements which will heavily influence the shape of the design for the IRRC application.

3.1 User Database

The server application must store message history for users. Therefore, it must be able to recognize returning users based on their name and password.

3.1.1 Registration

When a new user connects to a server, they must register with a new username and password. The username must be unique in order to avoid shared message history or other similar problems. To create a username and password, a new user must click on the “New User” option beneath the login prompt. From there, a prompt will appear asking the user to input a desired name and password (to be entered twice in different boxes). An indicator next to the username box will update constantly to let the user know if the entered username is available. Once registration is complete, the server will store this information and log the user into the chat.

3.1.2 Storage of User Information

User information is to be stored server-side. The username and password are to be stored in a data file.

3.2 Online Users

When a user logs in, a message is to be displayed in the all-chat channel announcing that the user has come online. The server is to keep track of everyone that logs in and out, so that all online users can be seen on the left-hand side of the client application.

3.3 Sending Messages

The client application must simultaneously be reading from the server and sending messages at the same time. When the client sends a direct message to a user, the server will look up the user and add that message to their personal message history.

4 Nonfunctional Requirements

This section will describe the model that has been adopted for this software, and briefly talk about the UI for both applications.

4.1 Server-Client Model

As previously described, the software is to be a two-application framework of client and server. The server will do most of the computational work, while the client is to be the conduit through which the user interacts with the server.

4.1.1 The Server

The server application will keep track of all user data, including usernames, passwords, online status, and message history. This data will be stored in plain text files and manipulated with server-side functions invoked by requests from connected clients.

4.1.2 The Client

The client contains non-computational functions which primarily interact with the connected server. When a user executes a command, such as sending a message, the client will pass the metadata to the server along with a command – such as “write this message to all-chat and display”.

4.2 Client UI

The user-interface for the client is to be simple and intuitive. On the left-hand side of the window there will be a list of online users. On the right-hand side will be a list of open channels, such as direct-message conversations and all-chat. The middle of the screen will be displaying the message history of the current channel, and at the bottom there will be a text box for writing a new message to the open channel.

4.3 Server UI

The server will open in a console window. This window will display all incoming commands from connected clients. Additionally, server commands may be executed by typing into the console window.

5 Milestones and Deliverables

This section will describe the different milestones for the development of this software.

5.1 Prototypes

The first milestone will be reached when a basic prototype of the software is developed. This prototype will showcase the appearance of full functionality for every requirement. While the software will not actually be usable for its final purposes, it will demonstrate what each functionality would look like. The user-interface will be as basic as possible, and only some test-cases will be completed. Data will be hard-coded and input/output may not be functional yet.

5.2 Alpha

The second and final milestone will consist of a fully functional client and server. The user-interface will be in its final stage, and functionality is expected to be 100% complete for the original requirements. All or most test-cases should be complete, and input/output should be fully functional.