Chat Application and Server

Requirements Document

Table of Contents

1 Introduction 3

1.1 Purpose and Scope 3

1.2 Target Audience 3

1.3 Terms and Definitions 3

2 Product Overview 4

2.1 Users and Stakeholders 4

2.1.1 User 4

2.1.2 Programer 4

2.2 Use cases 4

2.2.1 Sending Messages 4

2.2.2 Reviewing Messages 5

2.2.3 Checking Messages 5

3 Functional Requirements 6

3.1 Send Messages 6

3.1.1 Message to Specific User 6

3.1.2 Message to All Users 6

3.2 Online Status> 6

3.3 User Interface 6

4 Nonfunctional Requirements 7

4.1 Sending Messages 7

4.1.1 Message to User 7

4.1.2 Message to All 7

4.2 Online/Offline 7

4.3 Login/Logout 7

5 Milestones and Deliverables 9

5.1 Client/Server communication 9

5.1.1 Development of User Interface 9

5.1.2 Development of Server Backend 9

5.2 System fulfiils Requirements 9

# Introduction

This document is to outline the chat application and its features. The project is to create a program(client) and a server that can connect to each other and allow for messages to be sent from one user, using the client software, to another user over the internet.

## Purpose and Scope

This Document will describe the high level feature set and expected functionality of the system.

## Target Audience

Stakeholders and will be able to approve features and verify expectations.

## Terms and Definitions

Client: program running on user computer that allows connection to the sever.

User: A single account that can send/receive messages through the client software.

Server: Computer that is storing the messages and manages the connection of users.

# Product Overview

The user will be able to send and receive messages (text only) from either another user or broadcast a message to all users through an easy to use interface ascribing to the already established messaging paradigm common in other chat and messaging clients. A user will not be able to send messages to multiple recipients at once

## Users and Stakeholders

Describe the purpose of this section. Only a few sentences are expected here.

### User

For this software the User will only interact with other users.

### Programmer

The programmer will alone build the client application and server to manage the user to user communication. There will not be communication with the programmer and the user

## Use cases

Some examples of how the service is expected to be used.

### Sending Messages

In this example one friend, Sam, is planning an event and using the chat application to communicate with another friend, Sally. Sam will select Sally from the list of people to chat with. If Sally has the chat application open on her computer then her name will be indicated as online on Sam’s computer notifying him that Sally is available to chat with. They exchange some potential ideas and close the application. Each ones names now indicates they are offline.

### Reviewing Messages

Sam being a forgetful human has forgotten what date him and Sally had planned on. Sally is not online but Sam can still select Sally to view the past messages and read the past conversations to see what the date they agreed on was.

### Checking Messages

Sam opens the chat application on his computer. There is an indication that he has new messages from Sally. He can select to view these messages and sees that the date for the event has been moved due to inclement weather.

# Functional Requirements

Describe the purpose of this section and outline its contents. Only a few sentences are expected here. It may help to define a functional requirement.

## Send Messages

A user must be able to send a message to a specified user, or users, by selecting the destination and typing in their message.

### Message to specific user

A user must be able t send a message to a single particular user, group messages or sending a message with multiple users will not be supported

### Message to all

A user can specify a message be sent to all users in which case the message will be delivered to each user individually.

## Online Status

The user must be able to see who is online and who is offline. Some sort of visual symbol with green meaning online and red meaning offline is expected.

## User Interface

The program must be easy to operate and intuitive to use to even the most basic user. Selecting the name of the contact should bring up all messages and allow for the sending of another message.

# Nonfunctional Requirements

Outline of how the features will be implemented and the way that they will work together. This is meant as a guide for the programmer for how they should implement each feature.

## Sending a message

By “sending” a message to another user the content of the message will be added to the data structure that is storing the messaging conversation between the two users. The message will be sent to the server and only when the recipient user requests messages from the server will hat new message be displayed for them.

### Sending to User

From a list of users that will be displayed the user can select to send a message to a destination, this will contact the server and request the chat history between the two. They will have the option of adding a message to the conversation that the other user will be able to view.

### Sending to All

In this case that the user selects all as the destination for their message the contents of their message will be added to whatever preexisting chat history they already have going.

## Online/Offline Status

The online and offline status will be determined by status messages sent to and from the server and the client. A message will be sent from the server at a time interval to the client and if a response is received then that user will be marked as online. If no response is received, then that user will be marked as offline.

## Login/Logout

the user will be prompted to log in with a username and a password. If the username and password match a user in the server data base, then that user will be given access to the messages of that account. This will be know as being logged in. when the user closes the application the user will be “logged out” and the messages and information of that user will be delete from the client, but not the server.

# Milestones and Deliverables

Outlined are specific goals or targets to hit to stay on schedule. Development will start with the most basic features and build up from there until all features of the application are implemented.

## Client /Server Communication

Client application that is capable of sending and receiving communication to the sever. From this platform features such as messaging, user logins, ect. Will be able to be developed.

### Begin Developing User Interface

The user interface will begin the development phase, limited features will exist but backend for communication to server is operational.

### Begin development of backend features

The server will be able to communicate with the client application, but features such as user log in and log out will begin to be flushed out as the User interface is developed.

## Chat client fulfills all requirements

When all the features of the chat application are met the last milestone will be met and application will be in the final stages of development. Small bugs may still exist but overall structure will be completed including: sending messages, receiving messages, user online status, login, and logout.