Logan Miller

Logan's Chat Application

Requirements Document

Table of Contents

Introduction

This section provides an overarching view of what will be included in the following document. This section will provide the purpose and scope of the product. The document will describe requirements and functionality of a new chat application called “Logan’s Chat Application” (LCAP).

Purpose and Scope

The purpose of this document is to describe LCAP; a chat application that will allow users to communicate with one another by sending and receiving text based messages entered into the standard input stream. It is intended that users will be able to use LCAP to conduct conversations in real time over the internet.

Target Audience

The target audience is for persons wishing to send, receive and save text based messages with one another in real time over the internet.

Terms and Definitions

- Client (User) - Any person who creates an account on the application in order to communicate with other users
- Global Chat - The stream of inputs that all online users are able to view
- LCAP - Logan’s Chat Application
- Private Chat - The stream of inputs between two distinct users
- Receiver - A client that receives a message through either the public or private chat
- Sender - A client that sends a message through either the public or private chat
- Server - The actor that will manage message traffic as well as user privileges
- UI - The user interface, the interface the user will use in order to engage with LCAP

Product Overview

The purpose of this document is to provide an overview of a new chat application and the functionality that will be required for this product to allow it's user base to interact with one another. A user will be defined as any actor who uses the application to create an account and chat with other users.

Users and Stakeholders

The purpose of this section is to give a detailed description of the actors that will be using LCAP in order to identify and describe pertinent parties and their role in the software.

Client (User)

The client will be the prime user of the software. They will be required to create an account, which will consist of a username and password, prior to being given access to LCAP.

Server

The server will act as the administrator, saving all user data as well as directing the flow of messages between users.

Software Developer (Logan Miller)

The software developer must ensure that all functionalities of LCAP are implemented in the allotted time and to the specified requirements annotated within this document.

Use Cases

The purpose of this section is to annotate the models of how the software product and the users will interact with each other. Each case should highlight specific functionalities the end product will possess.

Send Message

The send message use case enables a user to send a typed message, input through the standard input stream, to be sent to the server which will then in turn save the message, and finally pass the message onto the intended recipient(s).

Receive Message

The receive message use case enables a user to receive a typed message from another

user. It requires the server to pass on a message from the sender to the receiver.

Save Message

The save message use case enables the server to record a message sent by a user. It requires user to send a message, either through the global or private chat.

Create account

The create account use case enables a user to create an account on LCAP. It requires the user to open the application, request to create an account, submit required information to the server, and the server to save the information for the user.

View History

The view history use case allows a user to view the entirety of their message history. The user will be able to see transcripts of their message history with any user or with the global chat. It will require a request be sent to the server and the server to respond with the user's message history

Inform User

The inform user use case allows the server to update the user on what other users are currently connected to the server.

Open Chat

The open chat use case allows the user the functionality to begin sending messages in either the global or private chat. It will require the user to send a request to the server and the server to respond with contacting the desired user(s).

Functional Requirements

The purpose of this section is to describe in detail the specific actions that LCAP will be able to perform. Given a specific input LCAP will supply a determined output. Each functional requirement covers how the intended use cases will be handled.

Send Message

Users should be able to type messages into the standard input stream and send them to other users through LCAP. This will require three actions from the server.

Send Message Aspect 1

The server receives a message from the client.

Send Message Aspect 2

The server saves the received message in the user's chat history.

Send Message Aspect 3

The server sends the message to either the desired client or to the global chat.

Receive Message

Users should be able to receive messages from other users. This will require the server to supply the receiver with the input given from the sender.

Receive Message Aspect 2

Upon the user receiving a message the server will determine the required recipient and display the message in the user's terminal.

Save Message

When a user sends a message this message should be saved by the server. This will require the server to maintain distinct lists of chat history between users, both in private and global chats.

Save Message Aspect 1

The server receives a message from a user.

Save Message Aspect 2

The server determines whether the message received must be saved to a user's private chat history or to the global chat history.

Save Message Aspect 3

The server records the message in the necessary chat history.

Create Account

A user must be able to create an account in order to gain access to LCAP. When a client creates an account, the server must maintain all aspects unique to the user such as username, password and chat history.

Create Account Aspect 1

User requests the server to create an account. The server prompts the user to input a username and password.

Create Account Aspect 2

The user supplies the server with a unique username and password.

Create Account Aspect 3

The server saves the user's account information and gives the user access to the features supplied in LCAP.

View History

A user will be able to request to see their chat history. Upon request the server will fetch the requested history and output the data in the user's terminal.

Inform Users

When a user logs onto LCAP the server will provide all users with a notification that the user has come online.

Open Chat

A user can double click on another user. This will serve as a request to the server to open a chat session with either another user in private chat or all users in public chat.

Open Chat Aspect 1

User requests the server to open a chat instance.

Open Chat Aspect 2

The server determines whether the user requested to open a chat instance in public or private chat.

Open Chat Aspect 3

The server gives the user access to input text into the desired chat instance.

Nonfunctional Requirements

The purpose of this section is to describe qualities of the LCAP product itself. These properties do not pertain to what the product will do, but how it's tasks will be accomplished.

Language Requirement

The entirety of the code supplied for the software will be written using Oracle's Java 8. Communication with the user base will have to be established in order to ensure that they have access to the necessary software in order to run the software.

User Interface

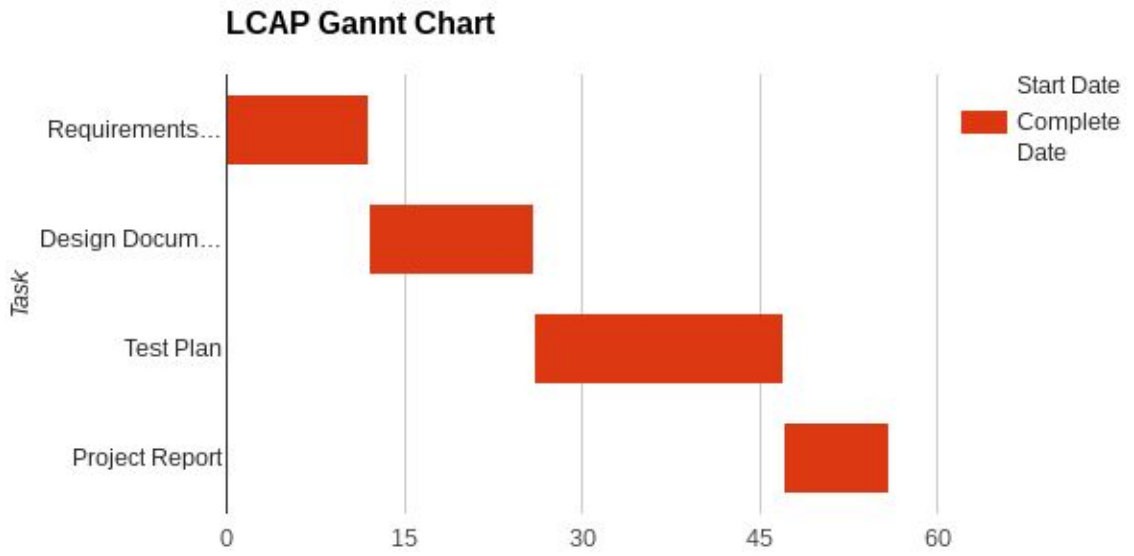
The UI must be user friendly, giving users the ability to easily navigate features of LCAP as well as quickly identify what abilities they will have while using the software.

Response Time

A client must be able to send a message to another user in real time. There should be little to no noticeable lag time in between messages. The application should allow users to conveniently mimic a spoken conversation using typed messages.

Milestones and Deliverables

The purpose of this section is to annotate and describe the major components and phases of the projects creation and when the client will receive the previously described features. This is to ensure a working timeline is set in order to meet deadlines in the required time allotted for the software's creation.



Requirements Document

The first deliverable will be this requirements document entailing the full desired capabilities of LCAP as well as identifying who will be using the software.

Design Document

The second deliverable will describe in detail how the software will accomplish the tasks prescribed in the requirements document.

Test Plan

The third deliverable will describe the resources used for testing the software as well as a schedule for the order that the different functional elements of the program will be tested in order to ensure the software works to the required specifications.

Test Plan Stage 1

Prior to the commencement of testing, modules for the software will have to be created. As each single bit of functionality is created it will have to be tested vacuously.

Test Plan Stage 2

Upon completion of the full range of base functionality, the software will have to be tested using a start to finish methodology where a all use cases will have to tested in a single run, with all functionalities working together in unison.

Project Report

The final deliverable will be a project report in which the software developer for LCAP will ensure the software meets all specifications through a sit down and a presentation of the application's full functionality.