SOFTWARE REQUIREMENTS SPECIFICATION

for

ConvoCraft — A CLI ChatRoom

Version 1.0

Prepared by :
1. Aditya Rao — IMT2023061
2. Mannat Kaur Bagga — IMT2023071

Submitted to :
Project Guide / Lecturer

November 13, 2025

Contents

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) details the functional and non-functional requirements for **ConvoCraft**, a command-line multi-user real-time chat application. The document will be used by developers, testers, and project managers to guide implementation and verification.

1.2 Scope

Product: ConvoCraft (CLI ChatRoom).
Goal: Provide a text-based, multi-user, real-time chat application accessible via the command line, offering essential communication and status-management features. Possible future extensions include a graphical UI, multimedia sharing, and persistent private messages.

1.3 Definitions, Acronyms and Abbreviations

CLI	Command Line Interface
SRS	Software Requirements Specification
JMS	Java Message Service
ActiveMQ	Message broker (Apache ActiveMQ)
Roll Call	Feature to list currently active users in the chat room

2 Overall Description

2.1 Product Perspective

ConvoCraft is a standalone client-server system intended to operate with a message broker (e.g., Apache ActiveMQ) as the central hub for message exchange. The client is a Java application using JMS to communicate with the broker.

2.2 Product Functions

Core functions include:

- User login (username-based identity)
- Real-time messaging (send/receive messages to the public chat topic)
- Status management (set/view user status)
- Roll call (list active participants)
- Graceful exit (disconnect and notify others)

2.3 User Classes and Characteristics

- **Chat Participant:** Any user with network access and a running JMS client. Responsibilities: log in, send/receive messages, set status.
- **Administrator:** System-level control (manage broker and system health).

2.4 Operating Environment

- Hardware: Standard desktop/laptop capable of running a JVM.
- Software: JDK (recommended version 22 or later), Apache ActiveMQ (running), terminal/console.

3 Specific Requirements

3.1 Functional Requirements

FR 1 — User Login (High) On launch the client prompts the user for a username and establishes identity for the session.

FR 2 — Message Sending (High) The client allows the user to type a message and publish it to the public chat topic.

FR 3 — Message Receiving (High) The client continuously listens for messages from the broker and displays them in real time.

FR 4 — Roll Call (Medium) The client provides a command that lists all currently active users in the chatroom.

FR 5 — Status Setting (Medium) The user can set a status string (e.g., “available”, “away”) that is visible to other users.

FR 6 — User Exit (High) The client provides a command to gracefully disconnect and notify other users of the exit.

3.2 Non-Functional Requirements

3.2.1 Performance

- **Latency:** Typical message delivery latency should be under 500 ms under normal conditions.
- **Throughput:** Support sustained 10 messages/sec with 10 concurrent users.

3.2.2 Security

- **Authentication:** Current version relies on the uniqueness of username; no password-based auth.
- **Data Integrity:** Broker must guarantee no loss/duplication during normal operation (broker responsibility).

3.2.3 Reliability and Availability

- If the message broker is unavailable, the client should show an informative error and attempt reconnects for a defined retry period.

4 Interface Requirements

4.1 User Interface

- Format: Text-based CLI.
- Input: Keyboard input; typed commands and messages.
- Output: Formatted text for messages, statuses and system alerts shown in the terminal.

4.2 Software Interfaces (Communication)

- Protocol: JMS (Java Message Service).
- Message Broker: Interacts with Apache ActiveMQ (or compatible JMS broker) via JMS client libraries.

5 Appendix / Notes

- Implementation language: Java (JDK 22+ recommended).
- Messaging middleware: Apache ActiveMQ (must be installed and running).
- Future extensions: GUI, multimedia, persistent private messaging, authentication enhancements.