

Term Project: Instant Messaging App

Feature Document

Table of Contents

1 Introduction	2
1.1 Purpose	2
1.2 Target Audience	2
1.3 Terms and Definitions	2
2 Product Overview	3
2.1 Product Vision	3
2.2 Product Scope	3
2.3 Operating Environment	3
2.4 Design and Implementation Constraints	3
3 User Stories	5
3.1 Personas	5
3.2 User Scenarios	5
3.2.1 Persona 1: Alice	5
3.2.2 Persona 2: Bob	5
3.2.3 Persona 3: Charlie	6
3.3 Epic User Stories	6
3.3.1 A Day of Remote Work	6
3.3.2 Studying for Finals	7
3.3.3 Talking with Clients	7
4 Feature List	9

1 Introduction

This document provides an overview of the instant messaging application project, detailing its features, purpose, target audience, and user interactions. The application will allow users to connect to a server and send messages to each other in real time over the internet.

1.1 Purpose

The purpose of this document is to outline the features and functionality of the instant messaging application to ensure clear understanding during development.

1.2 Target Audience

The target audience includes software developers, product managers, and stakeholders involved in the development and deployment of the application.

1.3 Terms and Definitions

- **Client:** A user application that connects to the chat server.
- **Server:** A backend system responsible for managing connections and message routing.
- **Real-time Messaging:** The ability to send and receive messages instantaneously.

2 Product Overview

This section provides a high-level overview of the Instant Messaging application, the scope of its features, and why it is being developed.

2.1 Product Vision

This instant messaging application aims to facilitate seamless real-time communication among users through a secure and scalable chat server.

2.2 Product Scope

Describe all the features of this product using a mind map here. You can use existing instant messaging application as an inspiration. Please be as specific as you can. This mind map should be one of your main references when actually designing sprint backlog.

One-on-one messaging

Group chats

Message history

Online/offline status indicators

Secure encryption for messages

2.3 Operating Environment

The application will be cross-platform, supporting Windows, macOS, Linux, iOS, and Android via web browser.

2.4 Design and Implementation Constraints

Programming languages: JavaScript

Frameworks: MERN Stack: MongoDB database, Express.js framework, React.js for user interface, Node.js for runtime environment.

Constraints: Low-latency messaging. System will be initially designed to run on a server from a single machine, so it will likely be difficult to scale it if the user base grows large.

3 User Stories

This section highlights user personas and their interactions with the application. These personas are vital for understanding how the expected users will interact with the product and what features they need and want to make them willing to use the application over competitors.

3.1 Personas

Alice: A remote worker who needs a secure and reliable communication platform. She has experience working with tech and wants a powerful, secure, and bug-free application.

Bob: A student using the app for group discussions. He has some experience with tech, but is more focused on organization and usability than security or response-time.

Charlie: A customer service representative managing client inquiries. He has little experience with tech and his clients have various levels of knowledge. His communications with his clients will include personal information.

3.2 User Scenarios

3.2.1 Persona 1: Alice

Alice is working on a project with her team at work. She needs to send real-time messages to her team so they can communicate effectively and stay on the same page. She must be notified right away when a message is received, since it could affect what she is working on at a given moment.

3.2.2 Persona 2: Bob

Bob wants to create a study group for class discussions. He wants to be able to use multiple group chats to keep conversations for each class separate. He likes to have a customizable interface to color code his groups and keep himself organized.

3.2.3 Persona 3: Charlie

Charlie requires a chat interface to communicate with clients. He talks with clients from a variety of backgrounds and with varying levels of tech competency, so it is important for him that the program is intuitive and easy to use for anyone. He also needs to have accurate and secure records of the messages with clients so that no private information is leaked.

3.3 Epic User Stories

3.3.1 A Day of Remote Work

Epic 1: A Day of Remote Work
<i>As a</i> remote worker <i>I want to</i> send and receive messages with a fast response time, prompt notifications, and encryption <i>So that I can</i> communicate smoothly and safely with my team
Acceptance Criteria <i>Give</i> direct messages with notifications <i>When</i> a message is sent to another user <i>Then</i> real-time communication can occur
User story 1: As a remote worker, I want to be able to communicate with a fast response time.
User story 2: As a remote worker, I want to receive prompt notifications for new messages.
User story 3: As a remote worker, I want secure encryption of passwords.

3.3.2 Studying for Finals

Epic 2: Studying for Finals
<i>As a</i> student <i>I want to</i> have group chats, color coded group chats, and the ability to mute groups <i>So that I can</i> study for finals effectively and ignore unrelated messages when I'm trying to focus
Acceptance Criteria <i>Give</i> a group chat with options to mute and change color <i>When</i> the create group button is pressed <i>Then</i> group communication can occur
User story 1: As a student, I want to have group chats to study for finals.
User story 2: As a student, I want to be able to change the profile color
User story 3: As a student, I want to be able to mute chats so I can focus on the chat for the class I'm studying for.

3.3.3 Talking with Clients

Epic 3: Talking with Clients
<i>As a</i> customer service representative <i>I want to</i> have an easy-to-use interface with direct messaging and reliable message logs <i>So that I can</i> talk to clients from various backgrounds privately and keep records of information I need from them.

<p>Acceptance Criteria</p> <p><i>Give</i> message logs</p> <p><i>When</i> a direct message or group chat is opened</p> <p><i>Then</i> message logs can be viewed and saved</p>
<p>User story 1:</p> <p>As a customer service representative, I want a simple user interface to talk with clients.</p>
<p>User story 2:</p> <p>As a customer service representative, I want direct messaging capability to speak privately with my clients.</p>
<p>User story 3:</p> <p>As a customer service representative, I want reliable message logs to record information.</p>

4 Feature List

Feature 1: Direct Messaging	Priority: High	Effort Required: 3 days
Description: Ability to connect to the server and communicate one-on-one with another user privately in real-time.		
Constraints: Must not be accessible by other users outside of the two users participating in the direct message.		
Comments: This is the most important feature and the first priority.		

Feature 2: Group Chats	Priority: High	Effort Required: 4 days
Description: Ability to connect to the server and communicate with a group of friends in real time. Should allow large groups, and all users in the group will see every message sent to the group chat.		
Constraints: Must not be accessible by other users outside of the users that are part of the group.		
Comments:		

Feature 3: Color-coding	Priority: Low	Effort Required: 1 day
--------------------------------	----------------------	-------------------------------

Description: Ability to change the color of chats to help stay organized and/or make the interface look the desired way.
Constraints:
Comments:

Feature 4: Encrypted Passwords	Priority: Med	Effort Required: 2 days
Description: Encrypt passwords to ensure privacy and safety.		
Constraints:		
Comments:		

Feature 5: Message Log	Priority: High	Effort Required: 2 days
Description: Ability to store records of the messages a user sends and receives.		
Constraints: Important to have high reliability.		
Comments:		

--