

Instant Messaging Software Requirements Specification -- 1.0

1. Introduction

1.1 Purpose: Mission Statement

Develop a web-based instant messaging software modeled after WhatsApp, Facebook Messenger, and WeChat. Implement their main features and optimize the user experience as much as possible.

1.2 Scope

The purposed web-based Instant Messaging system will provide mechanisms for signing up/in users, searching users, adding/deleting friends, sending/receiving messages to/from friends, building group chats, recording chat history, and allowing users to post messages as their status, allowing users' friend to see users' status.

More functions could be added after the functions above are implemented. Customized functions are always considered after the functions above.

1.3 Definition, Acronyms, and Abbreviations

Message --- Information in text, image, audio, video, link, or hybrid.

Friend --- A relationship between users that gives both parties a quick link to each other.

Status --- A message posted by a user that can only be seen by the user's friends.

Chat --- A connection tube between two users.

Group Chat --- A connection tube among three or more users.

The system --- represents the Instant Messaging Software.

1.4 Overview

Requirements have been divided into key functional areas. Functional and nonfunctional requirements exist within the laid-out document. The order of the leading sections and the corresponding requirements should be interpreted as a priority.

2. Overall Description

2.1 Product Perspective

This system is software that can be accessed remotely on the website by users. The product is an instant messaging application used to transport messages among users. The objective is to get familiar with the operation of instant messaging software, find the shortcomings, and improve them. If the development progress is far ahead of the expected progress, the development of Android and IOS versions can be considered.

2.2 Product Functions

The functions of this product will be divided into 3 categories:

1. Identity and Relationship Functions: describes all possible operations of setting and changing users' identities and relationships.
2. Communication Functions: describes all possible operations of sending/receiving messages.
3. System Scheduling: describes how the system should be designed.

2.3 User Characteristics

The users of this system will be all signed-up users and all designers and programmers.

Stakeholder	Interests	Constraints
Clients	The software is convenient, easy, and comfortable to use	Complex operations may mean inconvenient
Instructor	The software is cost-effectiveness and completed on schedule	None
UI Designer	The software is in good looking and easy to understand	An easy-to-understand UI also requires a clear, uncluttered system structure.
Structure Designer	The software is easy to maintain, secure and scalable	A secure system also requires a secure database designing.
Database Designer	The software's data is stored safely, easy to change and get.	An easy-to-use database cannot be built without a nice structural design.
Programmers	All requirements are implemented	only requirements designed can be implemented

2.4 Constraints

The system is not able to carry a huge user volume.

2.5 Assumptions and Dependencies

All users have at least a phone number/email.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Identity and Relationship Functions

- 3.1.1.1 Users provide phone number/email address, nickname, and password to sign up to the system.
- 3.1.1.2 Users provide their phone number/email address and password to sign in.
- 3.1.1.3 Users can modify their own profiles by changing personal information.
- 3.1.1.4 Users search for other users by phone number/email address/username.
- 3.1.1.5 A user can apply to be a friend of another user.
- 3.1.1.6 A user can approve/reject a friend request from another user.
- 3.1.1.7 A user can delete a friend.
- 3.1.1.8 A user can create a group chat by inviting friends as group members.
- 3.1.1.9 A user can leave a group chat.
- 3.1.1.10 A user as a group member of a group chat can invite their friends to join the group chat.

3.1.2 Communication Functions

- 3.1.2.1 Users can only communicate with their friends.

- 3.1.2.2 Only the user him/herself and the friend can see messages in a chat.
- 3.1.2.3 All users in a group chat can see messages of all users in the group chat.
- 3.1.2.4 A user can view the chatting histories of chats and group chats.
- 3.1.2.5 Users can send text, image, audio, video, link, or hybrid type messages in chats.
- 3.1.2.6 Users can only see their own and their friends' status.
- 3.1.2.7 Users can write text as comments to status.
- 3.1.2.8 Users can post text, image, audio, video, link, or hybrid type messages as their status.

3.1.4 System Scheduling

- 3.1.4.1 The system shall use the MVC model.

3.2 Page Requirements

- 3.2.1 The list shows all friends' chats and group chats should exist the whole time after signing in.
- 3.2.2 Signing in/up page should be a separate page from the chatting pages.
- 3.2.3 The avatar list of all users in a group chat should be shown when users are in a group chat window.
- 3.2.4 All messages sent by a user should appear with his/her avatar simultaneously.

3.3 Performance Requirements

- 3.4.1 Any operation on the website should be performed in 5 seconds.

3.4 Design Constraints

- 3.5.1 The system shall be usable at least on Chrome.
- 3.5.2 The system shall support PC and mobile.
- 3.5.3 The system must comply with the relevant privacy legislation.

3.5 Software Quality Attributes

- 3.6.1 Users can only see messages in chats between themselves and their friends and group chats they joined.
- 3.6.2 Users can only view the history of messages that they can see.
- 3.6.3 The system can be remotely accessible to all users.
- 3.6.4 The system shall protect all users' information.

3.6 Other Requirements

- 3.7.1 The hard deadline for the system is the end of April of next year for project completion.