Software Requirements Specification

for

Slick Chat

Version 1.0

Prepared by

Group Name: Group 2

Devanshi Ganatra	**ganatra.d@husky.neu.edu**
Dipen Patel	**patel.dip@husky.neu.edu**
Jainam Sheth	**sheth.j@husky.neu.edu**
Purvil Bambharolia	**bambharolia.p@husky.neu.edu**

Instructor:	**Michael Weintraub**
Course:	**CS5500 – Foundations of Software Engineering**
Teaching Assistant:	**Dhaval Dedhia**
Date:	**21st Feb 2020**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Devanshi Ganatra Dipen Patel Jainam Sheth Purvil Bambharolia	First draft of the software requirement specifications for the chat application – Slick Communications.	02/21/20

1 Introduction

The following document provides a detailed software requirement specification document for a chat application – Slick Chat.

1.1 Document Purpose

This document layouts the overall description of the software to be developed as well as its intention. Intended audience for this document are - developers, testers, project managers, leadership teams, sales and marketing.

1.2 Product Scope

The team at Slick Communications intends to invent the next big messaging service - Slick Chat, where people can message each other or groups effectively and securely through an easy-to-use interface. The system would allow exchange of information between two or more users of the system, find other users, create groups of users and will be independent of the user's choice of device.

1.3 Document Overview

This SRS document entails functional and non-functional requirements as well as any type of constraints to the project. It also describes intended user behavior and user characteristics. It also enlists any dependencies to the project.

1.4 Definitions, Acronyms and Abbreviations

User - user of this system is a person who intends to use the system to message other users of the system securely and effectively.
Group - A set of users of the system collectively forms a group of users.
Message - An important/unimportant piece of information in the form of text or media.
Thread - A running conversation between a bunch of users related to a specific subject.
Media - A visual entity that can form a message.

1.5 References and Acknowledgments

https://github.ccs.neu.edu/cs5500-fse/team-2-SP20/blob/master/product-backlog.pdf
http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf

2 Overall Description

2.1 Product Perspective

It is a communications system implementing the client-server model. The system provides an effective and secure mechanism for users to share and acquire information among other users or groups of users. The system is expected to evolve over with at least two releases. The first release must support a few hundred/thousand users and must be shipped until the end of April.

2.1.1 System Interfaces

The system uses the command-line interface to interact with users. The system is supported by any type of UNIX OS and Windows based systems.

2.1.2 User Interfaces

The application user-interface provides text-based keywords to allow the user to perform actions.

2.1.3 Software Interfaces

System allows the user to import any type of media file such as images, documents, videos, etc. System allows the user to download any incoming media file. The system also interfaces with a MySQL database to store all the activities being carried out on the system.

2.1.4 Communication Interfaces

The system will communicate between significant components over web sockets - a pub-sub based communication protocol.

2.1.5 System Constraints

The system in its first release must support a few hundreds/thousands of users. Secure messaging is a key aspect of this system.

2.1.6 Operations

User must be able to securely message one or more users. Users must be able to create groups of users.

2.2 Product Functionality

The service will allow people to message each other or message groups of people effectively and securely. Users will be able to retrieve and initiate conversations easily, have information about their communications partners at their fingertips all independent of any device.

2.3 User Characteristics

End-users are expected to have basic proficiency in reading and writing.
They should be familiar and comfortable handling/using/operating smart devices/ like laptops/phones. The users are expected to have basic knowledge and understanding of what the application does and the available functionalities that could be used. They should have basic competency in using chatting applications and how to access them.

2.4 Constraints

2.4.1 Regulatory Policies

Users will be connecting through a US service and are subject to US rules hence, we would provide support for CALEA (Communications Assistance for Law Enforcement Act)

2.4.2 Parallel Operations

Users will be able to access the application through multiple devices at a single time by logging in using their unique username and password.

2.4.3 Reliability Requirements

The message which is sent would preserve its contents and won't be tampered when delivered to the intended user/group.

2.4.4 Safety and security considerations

The messages sent will be encrypted end to end. The level of encryption will be based on the tier that the user has subscribed to. The user can always upgrade to a better subscription tier to improve security.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 User Functions

3.1.1.1 Users should be able to login using different user authentication methods such as native user id and password, LinkedIn, or Husky ID.

3.1.1.2 Users should be able to search for other users and groups enrolled in the system.

3.1.1.3 Users should be able to send and reply to text messages.

3.1.1.4 Users should be able to forward messages to other users and groups.

3.1.1.5 Users should be able to send/reply to a message to a group.

3.1.1.6 Users can choose to show different profile images to different users and groups.

3.1.1.7 Users can mute messages from users and groups for a pre-set time duration.

3.1.2 Message Functions

3.1.2.1 The system allows different service tiers to the users which represents how many messages in history are visible to the user.

3.1.2.2 The system uses UTF-8 character set to support English, French and Spanish languages.

3.1.2.3 Messages should support emoji, emoticons, images, videos, recordings, documents and any other form of media files.

3.1.2.4 Users should be able to delete an already sent message.

3.1.2.5 Incoming messages should be queued for the user if the user is offline. The queued messages should be delivered in order of their timestamps.

3.1.2.6 Messages can be grouped as threads bound to a subject.

3.1.2.7 Messages can be set to expire after a user-specified time span.

3.1.2.8 Messages can use different levels of encryption based on user selected tier.

3.1.3 Group Functions

3.1.3.1 Any user must be able to create a group.

3.1.3.2 A group can be made up of groups.

3.1.3.3 A group must have a moderator.

3.1.3.4 Moderator has the power to remove a user from the group.

3.1.3.5 Moderators can restrict users from joining the group - open or approval based.

3.1.3.6 A user can invite another user to join a group.

3.1.3.7 A user can also invite a group to join their group upon moderator approval.

3.1.3.8 Moderators can set a password on the group for the user to open the group.

3.1.3.9 Moderators can restrict users from sending messages to the group i.e. only moderators can post messages.

3.1.3.10 Users can choose to reply to the group as a whole, subset or to an individual user in the group.

3.1.3.11 Users can create polls in the group for other users to vote on a question.

3.1.4 Government Functions

3.1.4.1 Government will be able to obtain a copy of the conversation.

3.1.4.2 People of interest will be totally oblivious of the existence of any government body overseeing the conversation.

3.1.4.3 Time duration for streaming conversation to the government can be restricted.

3.1.4.4 The conversations will be sent to the government as is without any processing or modifications.

3.1.4.5 The message payload sent to the government will also consist of the IP addresses of the people of interest.

3.1.4.6 Government can choose to be notified whenever people of interest login into the system.

3.1.5 Software Interfaces

The user should be able to interact with the system through a web-based graphical UI for performing chat and user operations.

4 Non-functional Requirements

4.1 Performance Requirements

4.1.1 System will perform messaging operations in sub-second times.

4.1.2 Searching for new users can take up to 5 seconds.

4.1.3 The system will support up to a thousand concurrent users.

4.2 Safety and Security Requirements

The messages sent will be encrypted end to end. The level of encryption will be based on the tier that the user has subscribed to. The user can always upgrade to a better subscription tier to improve security.

4.3 Software Quality Attributes

4.3.1 Availability

The system will always be accessible and available to the user. Also, the system will be accessible from the user's choice of device.

4.3.2 Reliability

The system will be reliable in terms of successfully delivering a message. Messages will not be lost in encryption or communications channels. Users will be able to rely on the system to deliver important pieces of information.

4.3.3 Security

Users will be protected against their identity/user account with utmost security. Messages transported across the system will be encrypted with standard encryption methods.

4.3.4 Portability

Users will be able to access the system using any device, including mobile devices.

4.3.5 Logical Database Requirements

The system should use MongoDB – a NoSQL based database to store all information in the system related to users, groups and messages.