

Software Requirements Specification

for

Chatter Application

Version 1.0 approved

**Prepared by Devansh Gandhi, Harshil Mavani, Pankaj Badgujar,
Bhargavi Padhya**

Northeastern University

October 4, 2019

Contents

1. Introduction	3
a. Purpose.....	3
b. Scope	3
c. Overview	3
2. Overall Description	3
a. Product perspective	3
i. User Interface:.....	3
ii. Hardware Interfaces:.....	4
iii. Software Interfaces:	4
iv. Operations:.....	5
b. Product functions	5
c. User characteristics	5
d. Constraints	5
3. Specific Requirements.....	6
a. External interfaces	6
b. Functions.....	6
i. Exact sequence of operations.....	6
c. Performance requirements	8
i. Number of terminals supported.....	8
ii. Number of simultaneous users supported	8
iii. Amount and Type of information handled.....	8
d. Logical database requirements	8
i. Frequency of use.....	9
ii. Data entities and their relationships.....	9
iii. Data retention requirements	9
e. Design Constraints.....	9
f. Software System Attributes	9
i. Reliability	9
ii. Availability.....	9
iii. Portability	10

1. Introduction

a. Purpose

This document is intended to explain various aspects of the Chatter web application. Primary audiences include anyone who identify themselves as a stakeholder, user, developer, owner or operations executive. The SRS comprises of 3 main sections and each with certain sub sections to elaborate topics in detail.

b. Scope

Chatter is a platform to exchange messages and media in real time. The platform allows users to send messages to other individual users or groups of users. This application also facilitates easy retrieval of old messages and conversations. Along with retrieving old messages, a user can delete any messages sent by him/her.

c. Overview

The rest of the SRS document contains a comprehensive explanation of the product and its functions, constraints or dependencies if any. It also contains all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. There are detailed descriptions of every input into and output from the system, along with functional requirements that define the actions taking place while accepting and processing inputs in order to generate outputs.

2. Overall Description

a. Product perspective

Chatter is a platform to exchange messages between groups and individuals. Products that are like Chatter include Skype, Slack, etc.

i. User Interface:

This document aims at creating Chatter as a web application. Chatter will be opened inside a browser on any device. It will be responsive and displayed suitably based on which device it is being accessed.

Optimising the interface occurs mainly through making the website responsive. This ensures that maximum information is displayed to the user regardless of the screen on which it is being accessed.

ii. Hardware Interfaces:

Since our responsive web portal does not have any designated hardware it does not have any direct hardware interfaces. The communication between the ports is managed by the tomcat server on which the application is hosted.

iii. Software Interfaces:

The Chatter web app must be supported on the following browsers and their versions. Chatter will not be supported for any version older than these.

				
Chrome 23	IE10 / Edge	Firefox 21	Safari 6	Opera 15
Sep 2012	Sep 2012	Apr 2013	Jul 2012	Jul 2013

The following technologies/software solutions will be used either directly or indirectly to build Chatter -

Name	Pneumonic	Version	Source	Purpose
React.js	react	16.10.2	Opensource	Frontend framework
Java Development kit	JDK	1.8	Oracle	Sending messages over the network
MongoDB	mongo	4.0.1	Mongo	Database to store user information
NodeJs	node	12.11.1	Node	Server side scripting to host suite of APIs required to

				interact with DB
Apache Tomcat	tomcat	9	Apache	Server to broadcast messages.

iv. Operations:

The Chatter app will be monitored by a team of operation associates. These teams can query if the entire system is up and running. Further, they can find out how many messages have been sent overall. An operations associate can also query the system to find out the throughput of the system. A slack channel will be updated incase the application crashes.

b. Product functions

Following are the list of functions that Chatter provides:

Enables users to create secure accounts in order to send messages directly to another user or to a group as well as reply to messages on a personal level or in a group.

Provides an option for the users to reply only to the sender or a subset of the group.

Allows the users to find other users or groups, assuming they are willing to be found. It also allows users to follow other users and groups. Users can view the connections within their circle.

Allows users to forward messages to another user or a group.

Enables users to send media files, voice recordings and graphical texts that include emoticons, to other users or groups.

Provides a search feature that finds and returns the list of communication threads in which the search query was used.

Supports three basic character sets, English, Spanish and French.

c. User characteristics

The user of the Chatter system is anyone who has access to a computer or laptop, and knows how to operate the web browser on the machine. Their machine must also have access to internet or wifi in order to run Chatter.

d. Constraints

Following are the constraints that Chatter needs to abide by:

Hardware Limitations: See section 2.a.2

Audit Functions: Adding to the description in section 2.a.4, the operations team do not have rights to create, edit or remove any user from the system. Their job is solely to monitor the operations of the system.

Safety and security considerations: Chatter will send the government agency, a copy of the communications traffic in the order received and the exact text message without any modifications. It will wrap the messages in the from and to IP addresses, without altering the communications in any way. It will also keep a log of all the communications in the system till date in the message database. It will also provide the login and signup information to the government as and when required. Chatter will not decrypt any messages it receives in an encrypted format.

3. Specific Requirements

a. External interfaces

Since the chatter application is a standalone application, it does not have any inputs coming from or outputs going into any other external system. The only inputs the system receives are messages / media being sent by the sender.

b. Functions

i. Exact sequence of operations

1. Sending a message:

- a. User opens a chat with a user/group he wishes to send a message to.
- b. User types a text message in the text box (or selects media such as images / videos / tunes / recordings).
- c. The user clicks the send button (or press enter on the keyboard).
- d. The message gets delivered to the user/group for which the sender opened the chat.
- e. The message is displayed in the chat (or in group chat in case of group message) for both the sender and the receiver with their respective time stamps (time sent for the sender and time received for the receiver)
- f. The message also gets stored in the message database which we implement using mongoDB

2. Finding other users:

- a. User can type the name of the user he/she wants to search in the search bar provided.
- b. The user can then click the search button (magnifying glass icon) to begin the search.
- c. The name is cross-checked in the database to check if it matches any other user.
- d. The person has to mark his profile public in order to be found in searches.
- e. If it matches, the result can be presented to the user who can then click on the result.
- f. A chat window with the individual person will then be displayed and they can begin communication.
- g. In case no user is found by that name, the result "No users found" will be displayed.

3. Creating a group:

- a. There would be a create group button on the screen which the user can click.
- b. The user can then name the group and select other users to add in the group (as explained in 3.b.ii.2). (maximum limit of the group members is fixed)
- c. Then user can click the OK button to create the group which by default makes the user who created the group as the moderator.

4. Adding a group member/members: (Only group moderators)

- a. The user opens a chat with a group he wishes to add user to.
- b. The user then clicks on the option menu in the group and selects the ADD user option.
- c. The user can then select other users to add in the group(as explained in 3.b.ii.2). (maximum limit of the group members is fixed)
- d. Then user can click the OK button to add the users in the group.

5. Removing a group member/members: (Only group moderators)

- a. The user opens a chat with a group he wishes to delete user from.
- b. The user then clicks on the option menu in the group and selects the Delete user option.
- c. The user can then select other users to delete in the group (as explained in 3.b.ii.2). (maximum limit of the group members is fixed)

- d. Then user can click the OK button to delete the users in the group.
- 6. Sending a private message:
 - a. User opens a chat with a user/group he wishes to send a message to.
 - b. User types a text message in the text box (or selects media such as images / videos / tunes / recordings).
 - c. User will also mark the message as private.
 - d. The user clicks the send button (or press enter on the keyboard).
 - e. The message gets delivered to the user/group for which the sender opened the chat.
 - f. The message is displayed in the chat (or in group chat in case of group message) for both the sender and the receiver with their respective time stamps (time sent for the sender and time received for the receiver)
 - g. The message also gets stored in the message database which we implement using mongoDB

c. Performance requirements

i. Number of terminals supported

Since the chatter application is a standalone application hosted on a web server, it does not involve terminals and can be accessed by merely a web browser from the user's device.

ii. Number of simultaneous users supported

Since the chatter application is hosted on a Tomcat server, it can handle about as many users as supported by its transaction limit of 13000 TPS

iii. Amount and Type of information handled

The type of information handled includes text messages and media ranging from images to audio recordings, videos and emojis.

d. Logical database requirements

MongoDB shall serve as the backend database for the application.

i. Frequency of use

1. The database would be in use every time a chat is opened to retrieve the messages and display them in the chat window.
2. Also, any new message being sent would be added in the database.

ii. Data entities and their relationships

1. Database would have messages, users, and group information as entities with the users sharing many-to-many relationships with the groups.
2. The messages would also share many-to-many relations with both the users and the group entities

iii. Data retention requirements

1. Messages will be stored in the database along with their timestamps for up to certain period of time (say 90 days) after which they can be deleted in scenarios where memory has to be saved.

e. Design Constraints

- i. We need to keep track of every communication so that any change in the message history of a user should not be affected in the database. In case of a subpoena we can provide the data to the government.
- ii. Any messages that are flagged as private by the user, should not be copied by any other users except the sender.
- iii. Messages order should always be preserved i.e messages should be received in the same order they were sent.

f. Software System Attributes

i. Reliability

Apache tomcat is stable even with a high load of about ~100 pings/sec.

ii. Availability

Since Apache tomcat is being hosted on a third-party application the availability of the application would be the same as availability provided by the third-party application.

iii. Portability

This is a responsive web application which means user can use it from any desktop/mobile. It needs a browser (as specified in 2.a.3) and an internet connection to access the application.