

# SE 3XA3: Software Requirements Specification Node Messenger

Team #24, Node Messenger  
Tasin Ahmed - ahmedm31  
Shardool Patel - pates25  
Omar Elemary - elemario

October 5, 2018

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Customers . . . . .	1
1.2.3	Other Stakeholders . . . . .	1
1.3	Mandated Constraints . . . . .	1
1.4	Naming Conventions and Terminology . . . . .	1
1.5	Relevant Facts and Assumptions . . . . .	1
<b>2</b>	<b>Functional Requirements</b>	<b>2</b>
2.1	The Scope of the Work and the Product . . . . .	2
2.1.1	The Context of the Work . . . . .	2
2.1.2	Work Partitioning . . . . .	2
2.1.3	Individual Product Use Cases . . . . .	2
2.2	Functional Requirements . . . . .	2
<b>3</b>	<b>Non-functional Requirements</b>	<b>3</b>
3.1	Look and Feel Requirements . . . . .	3
3.2	Usability and Humanity Requirements . . . . .	3
3.3	Performance Requirements . . . . .	3
3.4	Operational and Environmental Requirements . . . . .	4
3.5	Maintainability and Support Requirements . . . . .	4
3.6	Security Requirements . . . . .	4
3.7	Cultural Requirements . . . . .	4
3.8	Legal Requirements . . . . .	4
3.9	Health and Safety Requirements . . . . .	4
<b>4</b>	<b>Project Issues</b>	<b>5</b>
4.1	Open Issues . . . . .	5
4.2	Off-the-Shelf Solutions . . . . .	5
4.3	New Problems . . . . .	5
4.4	Tasks . . . . .	5
4.5	Migration to the New Product . . . . .	6
4.6	Risks . . . . .	6
4.7	Costs . . . . .	6

4.8	User Documentation and Training . . . . .	6
4.9	Waiting Room . . . . .	6
4.10	Ideas for Solutions . . . . .	6
<b>5</b>	<b>Appendix</b>	<b>7</b>
5.1	Symbolic Parameters . . . . .	7

## List of Tables

1	Revision History . . . . .	ii
---	----------------------------	----

## List of Figures

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

This document describes the requirements for .... The template for the Software Requirements Specification (SRS) is a subset of the Volere template (Robertson and Robertson, 2012). If you make further modifications to the template, you should explicitly state what modifications were made.

# **1 Project Drivers**

## **1.1 The Purpose of the Project**

## **1.2 The Stakeholders**

### **1.2.1 The Client**

### **1.2.2 The Customers**

### **1.2.3 Other Stakeholders**

## **1.3 Mandated Constraints**

## **1.4 Naming Conventions and Terminology**

## **1.5 Relevant Facts and Assumptions**

User characteristics should go under assumptions.

## **2 Functional Requirements**

### **2.1 The Scope of the Work and the Product**

#### **2.1.1 The Context of the Work**

#### **2.1.2 Work Partitioning**

#### **2.1.3 Individual Product Use Cases**

### **2.2 Functional Requirements**

## **3 Non-functional Requirements**

### **3.1 Look and Feel Requirements**

The users will be we greeted by a welcome page with logo of our product, information, and contacts. There will be two buttons named 'Register' and 'Login' which they can use to begin using the Node Messenger. The Node Messenger will be in a different tab, and can be accessed with the click of a button. The welcome page should be visibly appealing and be structured well for the ease of use. The messenger itself should have a simple yet unique design, as to not make it cluttered and difficult to use.

### **3.2 Usability and Humanity Requirements**

Node Messenger can be used by anyone with the need to communicate with others over the Internet. The user should be able to access the website through our URL, be able to use a keyboard and mouse to navigate around our website. Previous knowledge of other messaging apps can be helpful. The user should be able to access the website through our URL, be able to use a keyboard and mouse to navigate around our website. Previous knowledge of other messaging apps can be helpful. Node Messenger can be accessed on any browser, and mobile as we plan to make it responsive.

### **3.3 Performance Requirements**

Node Messenger should be quick to respond to user instructions, while transferring and receiving messages as quickly as possible. It should keep the user's personal data safe, and keep the machine's memory usage to a minimum. Node Messenger should be available to user anywhere on Earth with the access to Internet. It should be able to provide service to many users at the same time. It can be used on any browser without the need to be downloaded on the machine. Our HTML code with account for scalability of the messenger. Node Messenger shall be usable as long as HTML, CSS, and Javascript is supported by the browser.

### **3.4 Operational and Environmental Requirements**

Node Messenger can be used on any browser that supports the latest versions of HTML, CSS, and Javascript. Before Node Messenger can be used, it must be able to run on a local URL, and be responsive. We plan to update Node Messenger as frequently as possible to keep the features up-to-date.

### **3.5 Maintainability and Support Requirements**

We will check for errors before release in order to keep maintenance to a minimum. We will also ensure Node Messenger can run on the majority of the browsers. Node Messenger should be available to anyone with the URL, and access to Internet.

### **3.6 Security Requirements**

Node Messenger can be accessed by anyone, given their machine meets the minimum requirements. It should not let users with invalid credentials login. Node Messenger should preserve the personal data of the users.

### **3.7 Cultural Requirements**

Node Messenger should be respectful of all users of different cultures and backgrounds. It will be available in English, be hosted on North American servers. It will be available worldwide.

### **3.8 Legal Requirements**

Node Messenger shall not disobey any laws.

### **3.9 Health and Safety Requirements**

We will make sure the colors used do not put too much pressure on the eyes.

## 4 Project Issues

### 4.1 Open Issues

Not Applicable.

### 4.2 Off-the-Shelf Solutions

Not Applicable.

### 4.3 New Problems

Node Messenger will occupy just one of the many web addresses being hosted. It should not create any problems, unless the website cannot hold a large amount of people at a time. The product will not have any effect on the installed systems, as nothing is required to be downloaded to be used. Node Messenger might cause problems for the users if used for a prolonged period of time, such as hand pain, or eye soreness.

### 4.4 Tasks

Task	Completer's Role	Timeline
Model Implementation	Software Engineer	Oct 10th
Model Revision	Client	Oct 14th
HTML and CSS implementation	Software Engineers	Oct 25th
Javascript backend	Software Engineers	Nov 1st
Revision	Client	Nov 3rd
Host Node Messenger	Software Engineers	Nov 24th
Maintenance	Software Engineers	Yearly

Task 1: Create a model of the Node Messenger website to visualize how everything will be positioned. This will help to understand how everything in our website will interact with each other. Task 2: Create the front-end of the website using HTML and CSS. Ensure the website is clean and visually appealing. Task 3: Create the back-end of the website using Javascript. Ensure all components of the messenger is fully functioning and creates the desired output. Task 4: Host the messaging app on a free web hosting site.



- 4.5 Migration to the New Product
- 4.6 Risks
- 4.7 Costs
- 4.8 User Documentation and Training
- 4.9 Waiting Room
- 4.10 Ideas for Solutions

## References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

## 5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

### 5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC\_CONSTANTS. Their values are defined in this section for easy maintenance.