Software Requirements Specification

**Messenger**

**Client – Server Application**

**Simple Message Protocol (SMP)**

**Version 1**



# Table of Contents

[Table of Contents 2](#_Toc190790573)

[Revision History 3](#_Toc190790574)

[1 Introduction 4](#_Toc190790575)

[2 Requirements 4](#_Toc190790576)

[2.1 SMP Server Requirements 4](#_Toc190790577)

[2.1.1 Requirement 1: *Application Start* 4](#_Toc190790578)

[2.1.2 Requirement 2: *Show Messages* 5](#_Toc190790579)

[2.1.3 Requirement 3: *Write SMP PUT Message to File* 5](#_Toc190790580)

[2.1.4 Requirement 4: *Read SMP GET Message from File* 5](#_Toc190790581)

[2.1.5 Requirement 5: *Requirement Name* 5](#_Toc190790582)

[2.2 SMP Client Producer Requirements 6](#_Toc190790583)

[2.2.1 Requirement 1: *Send Message* 6](#_Toc190790584)

[2.2.2 Requirement 3: *Requirement Name* 6](#_Toc190790585)

[2.3 SMP Client Consumer Requirements 6](#_Toc190790586)

[2.3.1 Requirement 1: *Get Message* 6](#_Toc190790587)

[2.3.2 Requirement 3: *Requirement Name* 6](#_Toc190790588)

[3 Considerations 7](#_Toc190790589)

[3.1 Dependencies 7](#_Toc190790590)

[3.2 Assumptions 7](#_Toc190790591)

[3.3 General Constraints 7](#_Toc190790592)

[3.4 Guidelines 7](#_Toc190790593)

[3.5 Development Methods 7](#_Toc190790594)

[4 Appendix 7](#_Toc190790595)

[4.1 Acronyms 7](#_Toc190790596)

[4.2 Engineering Terms 7](#_Toc190790597)

[4.2.1 Software Development 7](#_Toc190790598)

[5 Notes 8](#_Toc190790599)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Date | Authors | Comments |
| 1 |  |  | 1. Initial release. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

Perhaps the most important piece of the software system initiation, definition and planning stage is the definition of the software requirements. The role of the software requirements specification document is to define in detail the functionality of the software system. The Software Requirements Specification document defines the software requirements for each software system component. The software requirements specification document typically includes software configurations, software installation procedures, database installation procedures, and training standards.

Because the software requirements specification document defines the system functionality in great detail, the software requirements definition process is usually a very formal and time-consuming process. Using an agile software development methodology, requirements gathering is an iterative process. During each iteration, the team develops a better understanding of the system and the system’s operating environment.

This Software Requirements document contains SMP client and server application development requirements and other requirements specific to customers contracting with Never Crash Software Services and vendors developing a software or hardware product or software subsystem for Never Crash Software Services.

NOTE: This document contains software application development requirements and other requirements specific to customers contracting with Never Crash Software Services and vendors developing a software or hardware product or software subsystem for Never Crash Software Services. This document is NOT intended to serve as a non-disclosure agreement, letter of intent, request for proposal or quotation, joint marketing agreement or purchase agreement. These or other documents that may be required to define any business relationship or agreements between any vendor and Never Crash Software Services shall be executed separately as required.

All licenses for this software application are owned by the principal contracting agency, Reality Software.

# Requirements

## SMP Server Requirements

### Requirement 1: *Application Start*

* Type: User Interface
* Description: This feature allows the SMP server to start processing SMP client requests. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button, in the case of a graphical user interface (GUI).
* Security Issue: Local (inside the network’s firewall) unauthorized accessed to the SMP server that’s hosting the SMP server application allows the unauthorized actor to read, update, and delete SMP messages.
* Security Issue: Remote (outside the network’s firewall) unauthorized accessed to the SMP server application allows the unauthorized actor to read, update, and delete SMP messages.

### Requirement 2: *Show Messages*

* Type: User Interface
* Description: Reads SMP message records from the SMP message file and displays the SMP version, message priority, message date and time, and message content for each of the messages in the file and outputs the messages. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button in the case of a graphical user interface (GUI).
* Security Issue: Local (inside the network’s firewall) unauthorized accessed to the SMP server that’s hosting the SMP server application allows the unauthorized actor to read SMP messages.

### Requirement 3: *Write SMP PUT Message to File*

* Type: Security Feature
* Description: Server receives an SMP PUT message packet from a SMP Client Producer. The message packet consists of the SMP version, message priority, message date and time, and an encrypted message. The server adds the message packet to a file associated with the message priority.
* Security Issue: Local (inside the network’s firewall) unauthorized accessed to the SMP server that’s hosting the SMP server application allows the unauthorized actor to read SMP unencrypted messages.
* Security Issue: The encryption method and the encryption key management is to be determined.

### Requirement 4: *Read SMP GET Message from File*

* Type: Security Feature
* Description: Server receives an SMP GET message packet from a SMP Client Consumer. The SMP GET message packet consists of the SMP version and a message priority. Based on the message priority, the server reads and consumes a SMP message from the appropriate file and sends a SMP message packet to the SMP Client Consumer. The server deletes the record from the file after the client acknowledges that it has received message packet.
* Security Issue: The encryption method and the encryption key management is to be determined.

### Requirement 5: *Requirement Name*

* Type:
* Description:
* Security Issue:

## SMP Client Producer Requirements

### Requirement 1: *Send Message*

* Type: User Interface
* Description: The SMP Message Producer client program is designed to send an SMP PUT request to the server. This feature allows a user to send an SMP message to the server. The message sent to the server consists of the SMP version, message priority, message date and time, and an encrypted message. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button in the case of a graphical user interface (GUI).
* Security Issue: If unencrypted, the message content can be read by a network packet analyzer (packet sniffer). It’s recommended that the message content is encrypted.
* Security Issue: The encryption method and the encryption key management is to be determined.

### Requirement 3: *Requirement Name*

* Type:
* Description:
* Security Issue:

## SMP Client Consumer Requirements

### Requirement 1: *Get Message*

* Type: User Interface
* Description: The SMP Message Consumer client program is designed to send an SMP GET request to the server to retrieve the next message. This feature allows a user to retrieve an SMP message from the server. The message sent from the server back to the client consists of the message date and the encrypted message. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button in the case of a graphical user interface (GUI).
* Security Issue: If unencrypted, the message content can be read by a network packet analyzer (packet sniffer). It’s recommended that the message content is encrypted.
* Security Issue: The encryption method and the encryption key management is to be determined.

### Requirement 3: *Requirement Name*

* Type:
* Description:
* Security Issue:

# Considerations

Describes the issues that need to be addressed before implementing a software solution.

## Dependencies

Describe any dependencies.

## Assumptions

Describe any assumptions that may be wrong.

## General Constraints

Describe any constraints that could have an impact on the implementation of the software.

## Guidelines

Describe any guidelines for the implementation of the software.

## Development Methods

Describe the software development method that will be used.

# Appendix

## 

## Acronyms

SMP – Simple Message Protocol

## Engineering Terms

### Software Development

Client – A software application that sends and receives messages to and from a server application.

Server – A software application that processes messages from client applications.

Producer – A software application that produces messages.

Consumer – A software application that consumes messages.

# Notes