Software Design Specification

**Messenger**

**Client – Server Application**

**Simple Message Protocol (SMP)**

**Version 1**



# Table of Contents

[Table of Contents 2](#_Toc132796890)

[Revision History 4](#_Toc132796891)

[1 Introduction 5](#_Toc132796892)

[2 Design Overview 5](#_Toc132796893)

[3 Design Goals 6](#_Toc132796894)

[3.1 Phase 1 6](#_Toc132796895)

[3.1.1 Design Goal 1: Develop the SMP Protocol Packet Message Format 6](#_Toc132796896)

[3.2 Phase 2 6](#_Toc132796897)

[3.2.1 Design Goal 3: Develop the SMP Server Data Store 6](#_Toc132796898)

[3.2.2 Design Goal 4: Develop the SMP Server Prototype 6](#_Toc132796899)

[3.3 Phase 3 6](#_Toc132796900)

[3.3.1 Design Goal 3: Develop the SMP Client Producer Prototype 6](#_Toc132796901)

[3.3.2 Design Goal 4: Develop the SMP Client Consumer Prototype 6](#_Toc132796902)

[4 System Architecture 6](#_Toc132796903)

[4.1 System Network Topography – One LAN 6](#_Toc132796904)

[4.1.1 Diagram -- SMP Client Producer Sends SMP PUT Request 6](#_Toc132796905)

[4.1.2 Diagram -- SMP Client Consumer Sends SMP GET Request 7](#_Toc132796906)

[4.2 System Network Topography – Two Separate LANs 7](#_Toc132796907)

[4.2.1 Diagram -- SMP Client Producer Sends SMP PUT Request 8](#_Toc132796908)

[4.2.2 Diagram -- SMP Client Consumer Sends SMP GET Request 8](#_Toc132796909)

[4.3 SMP Server Component 9](#_Toc132796910)

[4.4 SMP Client Producer Component 9](#_Toc132796911)

[4.5 SMP Client Consumer Component 9](#_Toc132796912)

[5 System Design Details 10](#_Toc132796913)

[5.1 Design Detail 1: *Name of Design Detail* 10](#_Toc132796914)

[5.2 Design Detail 2: *Name of Design Detail* 10](#_Toc132796915)

[5.3 Design Detail 3: *Name of Design Detail* 10](#_Toc132796916)

[6 Design Considerations 10](#_Toc132796917)

[6.1 Dependencies 10](#_Toc132796918)

[6.2 Assumptions 10](#_Toc132796919)

[6.3 General Constraints 10](#_Toc132796920)

[6.4 Design Guidelines 10](#_Toc132796921)

[6.5 Development Methods 10](#_Toc132796922)

[7 Definitions 11](#_Toc132796923)

[8 References 11](#_Toc132796924)

[9 Appendix 11](#_Toc132796925)

[9.1 Acronyms 11](#_Toc132796926)

[9.2 Engineering Terms 11](#_Toc132796927)

[9.2.1 Software Development 11](#_Toc132796928)

[10 Notes 11](#_Toc132796929)

# Revision History

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

Introduction

The Software Design Specification details the software application’s design. It describes the software application’s overall architecture, including the architecture of each software component and each of the software component’s interfaces. The Software Design Specification is written by senior software systems designers. When completed, the Software Design Specification is passed to the software development team to give the team an overview of the software system’s architecture. A Software Design Specification helps to ensure that the software system’s software requirements and the software technical specification are clear. The Software Design Specification definition process is an iterative process. During each iteration, the team develops a better understanding of the system and the system’s operating environment. The Software Design Specification may include any of the following:

• Software Component APIs.

• Software Component Diagrams

• Software Component Interface Diagrams.

• Software/Hardware Interfaces.

• Flowcharts illustrating important use cases.

• Protocols, protocol data formats, and protocol messages.

NOTE: This document contains software application design specifications 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.

# Design Overview

The SMP protocol allows client applications to send messages to a SMP server and retrieve messages from a SMP server. The server application stores the message in a datastore, such as an ASCII text file. The datastore is designed to implement the behavior of a message queue. A queue is a first-in, first-out data structure. Messages are sent to the SMP server and inserted into the queue in the order they’re received. The messages are retrieved from the queue in a first-in, first-out fashion. In other words, the first message inserted in the queue is the first message retrieved from the queue.

# Design Goals

## Phase 1

### Design Goal 1: Develop the SMP Protocol Packet Message Format

## Phase 2

### Design Goal 3: Develop the SMP Server Data Store

### Design Goal 4: Develop the SMP Server Prototype

## Phase 3

### Design Goal 3: Develop the SMP Client Producer Prototype

### Design Goal 4: Develop the SMP Client Consumer Prototype

# System Architecture

The section provides an overview of how the functionality and responsibilities of the system were divided and assigned to subsystems.

## System Network Topography – One LAN

This configuration illustrates that the SMP Server, SMP Client Producer, and the SMP Client Consumer are all connected to the same LAN.

### Diagram -- SMP Client Producer Sends SMP PUT Request



SMP Client Producer sends an SMP PUT Request packet to the SMP Server



SMP Server sends back an SMP PUT Response Packet

### Diagram -- SMP Client Consumer Sends SMP GET Request



SMP Client Consumer sends an SMP GET Request packet to the SMP Server

Graphical user interface, application

Description automatically generated

SMP Server sends back an SMP GET Response Packet

## System Network Topography – Two Separate LANs

This configuration illustrates that the SMP Server is connected to a separate LAN, and the SMP Client Producer, and the SMP Client Consumer are connected to a different LAN.

### Diagram -- SMP Client Producer Sends SMP PUT Request

Graphical user interface, application

Description automatically generated

SMP Client Producer sends an SMP PUT Request packet to the SMP Server

Graphical user interface, application

Description automatically generated

SMP Server sends back an SMP PUT Response Packet

### Diagram -- SMP Client Consumer Sends SMP GET Request

Graphical user interface, application

Description automatically generated

SMP Client Consumer sends an SMP GET Request packet to the SMP Server



SMP Server sends back an SMP GET Response Packet

## SMP Server Component

Optional: Describe the architecture of the SMP Server.

## SMP Client Producer Component

Optional: Describe the architecture of the SMP Client Producer.

## SMP Client Consumer Component

Optional: Describe the architecture of the SMP Client Consumer.

# System Design Details

## Design Detail 1: *Name of Design Detail*

## Design Detail 2: *Name of Design Detail*

## Design Detail 3: *Name of Design Detail*

# Design Considerations

Describes the issues that need to be addressed before creating a design 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 design of the software.

## Design Guidelines

Describe any guidelines for the design of the software.

## Development Methods

Describe the software design method that will be used.

# Definitions

# References

# 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