Software Requirements Specification

JamBox

Software Requirements Specification

0.0.1

1/24/2020

Frank Pasqualini

Software Engineer

Prepared for

JamBox

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 1/24/2020 | 0.0.1 | Frank Pasqualini | Title Information |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Grace Brueggman | Software Eng. | 1/24/2020 |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

2. General Description 1

2.1 Product Perspective 1

2.2 Product Functions 1

2.3 Constraints, Assumptions and Dependencies 1

3. Specific Requirements 1

3.1 External Interface Requirements 2

3.1.1 Software Interfaces 2

3.2 Functional Requirements 2

3.2.1 <Functional Requirement or Feature #1> 2

3.2.2 <Functional Requirement or Feature #2> 2

3.3 Use Cases 2

3.3.1 Use Case #1 2

3.3.2 Use Case #2 2

3.4 Classes / Objects 2

3.4.1 <Class / Object #1> 2

3.4.2 <Class / Object #2> 2

3.5 Non-Functional Requirements 2

3.5.1 Performance 3

3.5.2 Reliability 3

3.5.3 Availability 3

3.5.4 Security 3

3.5.5 Maintainability 3

3.5.6 Portability 3

3.7 Logical Database Requirements 3

3.8 Other Requirements 3

4. Analysis Models 3

4.1 Use Case Diagrams 3

4.2 Class Diagrams 3

4.3 Sequence Diagrams 3

4.4 Activity Diagrams (DFD) 3

4.5 State-Transition Diagrams (STD) 3

5. Change Management Process 4

A. Appendices 4

# 1. Introduction

Brief overview

## 1.1 Purpose

*SRS Purpose and Audience*

## 1.2 Scope

*This subsection should:*

*(1) Software product’s name*

*(2) What the software product will do and won’t do (just high level, no details)*

*(3) Describe the application of the software being specified. Describe all relevant benefits, objectives, and goals as precisely as possible.*

## 1.3 Definitions, Acronyms, and Abbreviations

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.*

# 2. General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## 2.1 Product Perspective

*This subsection of the SRS puts the product into perspective with other related products or*

*projects. (See the IEEE Guide to SRS for more details).*

## 2.2 Product Functions

This subsection of the SRS should provide a summary of the functions (just high level, not the functional/ non-functional requirements) that the software will perform.

## 2.3 Constraints, Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS.

# 3. Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 3.2. Overall, this section 3 will capture the requirements that are used to guide the project’s software design, implementation, and testing.

## 3.1 External Interface Requirements

### 3.1.1 Software Interfaces

## 3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.2.1 <Functional Requirement or Feature #1>

Description

### 

### 3.2.2 <Functional Requirement or Feature #2>

…

## 3.3 Use Cases

### 3.3.1 Use Case #1

### 3.3.2 Use Case #2

…

## 3.4 Classes / Objects

### 3.4.1 <Class / Object #1>

3.4.1.1 Attributes

3.4.1.2 Methods

### 3.4.2 <Class / Object #2>

…

## 3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### 3.5.1 Performance

### 3.5.2 Reliability

### 3.5.3 Availability

### 3.5.4 Security

### 3.5.5 Maintainability

### 3.5.6 Portability

## 3.6 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

## 3.7 Other Requirements

Catchall section for any additional requirements.

# 4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

## 4.1 Use Case Diagrams

## 4.2 Class Diagrams

## 4.3 Sequence Diagrams

## 4.4 Activity Diagrams

## 4.5 State-Transition Diagrams (STD)

# 5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*