## Version [0.0]



Software Requirements Specification

## 

Prepared By: [Your Name]

# Software Requirements Specification Template

*[The SRS template is used provide a recording of the all agreed upon requirements. The Software Requirements Specification is a document created by the system analysts AFTER all the requirements have been collected from all stakeholders. The information recorded in this document will be used by the technical team for actual design of the final solution.]*

## 

## Project Overview

| Customer Name: |  |
| --- | --- |
| Name of Project: |  |
| Project Manager: |  |
| Technical Lead: |  |
|  |  |
|  |  |

# Revision History

| Comments | Name | Date | Description |
| --- | --- | --- | --- |
| Initial document | A N Other | 12/12/2012 | Prepared Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Template Caveats:**

Whilst ever intention has been made to ensure this template is as flexible as possible, each and every software project will differ. This template can only server as a guide to what considerations need to be given due attention. You should always review this document internally before sharing with the customer.

Contents

[Version [0.0] i](file:///C:\Users\Robin\Desktop\!%201%20My%20Courses\29%20Software%20Requirements\Course%20Materials\001_SRS%20Software%20Requirements%20Specification%20Template.docx#_Toc437966844)

[i](#_Toc437966845)

[Software Requirements Specification Template ii](#_Toc437966846)

[Project Overview ii](#_Toc437966847)

[Revision History ii](#_Toc437966848)

[1. Introduction 1](#_Toc437966849)

[1.1 Purpose 1](#_Toc437966850)

[1.2 Scope 1](#_Toc437966851)

[1.3 Definitions, Acronyms, and Abbreviations 1](#_Toc437966852)

[1.4 References 1](#_Toc437966853)

[1.5 Overview 1](#_Toc437966854)

[2. General Description 2](#_Toc437966855)

[2.1 Product Perspective 2](#_Toc437966856)

[2.2 Product Functions 2](#_Toc437966857)

[2.3 User Characteristics 2](#_Toc437966858)

[2.4 General Constraints 2](#_Toc437966859)

[2.5 Assumptions and Dependencies 2](#_Toc437966860)

[3. Specific Requirements 2](#_Toc437966861)

[3.1 External Interface Requirements 3](#_Toc437966862)

[3.1.1 User Interfaces 3](#_Toc437966863)

[3.1.2 Hardware Interfaces 3](#_Toc437966864)

[3.1.3 Software Interfaces 3](#_Toc437966865)

[3.1.4 Communications Interfaces 3](#_Toc437966866)

[3.2 Functional Requirements 3](#_Toc437966867)

[3.2.1 <Functional Requirement or Feature #1> 3](#_Toc437966868)

[3.2.2 <Functional Requirement or Feature #2> 3](#_Toc437966869)

[3.3 Use Cases 4](#_Toc437966870)

[3.3.1 Use Case #1 4](#_Toc437966871)

[3.3.2 Use Case #2 4](#_Toc437966872)

[3.4 Classes / Objects 4](#_Toc437966873)

[3.4.1 <Class / Object #1> 4](#_Toc437966874)

[3.4.2 <Class / Object #2> 4](#_Toc437966875)

[3.5 Non-Functional Requirements 4](#_Toc437966876)

[3.5.1 Performance 4](#_Toc437966877)

[3.5.2 Reliability 4](#_Toc437966878)

[3.5.3 Availability 4](#_Toc437966879)

[3.5.4 Security 4](#_Toc437966880)

[3.5.5 Maintainability 4](#_Toc437966881)

[3.5.6 Portability 4](#_Toc437966882)

[3.6 Inverse Requirements 4](#_Toc437966883)

[3.7 Design Constraints 4](#_Toc437966884)

[3.8 Logical Database Requirements 5](#_Toc437966885)

[3.9 Other Requirements 5](#_Toc437966886)

[4. Analysis Models 5](#_Toc437966887)

[4.1 Sequence Diagrams 5](#_Toc437966888)

[4.3 Data Flow Diagrams (DFD) 5](#_Toc437966889)

[4.2 State-Transition Diagrams (STD) 5](#_Toc437966890)

[5. Change Management Process 5](#_Toc437966891)

[A. Appendices 5](#_Toc437966892)

[A.1 Appendix 1 5](#_Toc437966893)

[A.2 Appendix 2 5](#_Toc437966894)

# 1. Introduction

The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. All Entries to this document must be validated with the customer before an agreement is made to include them in a project revision. All Revisions should be record at the start of this template and sign-off given before work is committed to.

## 1.1 Purpose

*State the purpose of this document, its intended audience and the how it is to be used.*

## 1.2 Scope

*Include in this section all comments related to what is included in the project and what is not. Be sure to identify the major modules that are expected to be delivered. For Example: This project will deliver a simple software solution that will integrate with one known system. The software will include administration, reporting, security and Service levels. This software will not look at addressing the upgrade of existing hardware. Any limitations or constraints imposed by working legacy hardware will be documented* *and flagged to the customer*

# 2. High Level Description

*This section of the SRS should describe the general factors of the proposed system. All references to features should be at a high-level and in user friendly terminology. This section does not state specific requirements.*

## 

## 2.1 Product Functions

This section should provide a list of all high-level features:

1. The software will allow data capture via webpage user interface
2. The software will support ad-hoc reporting and customizable reports can be uploaded.
3. The software will save all data to storage on the internal network.
4. The software…

## 

## 2.2 User Training

All requirements that pertain to user training, training the trainer sessions, hand-over etc. should be listed here.

## 

## 2.3 System Constraints

*If the proposed solution will be limited by the existing hardware, Existing software etc., then these need to be recorded here. All possible constraints should be recorded here.*

## 

## 2.4 Assumptions and Dependencies

This section should list all factors that may have an effect on the requirements defined in this specification.

1. Assumptions may include what existing operating systems you are expecting to use?
2. What version of SQL is available?
3. It is assumed there will be ample storage space for 5 years’ worth of data acquisition?

# 3. Specific Requirements

This section will list all requirements in a formal manner. Specific details will be given, not on the how, but on the what. Software design decisions will be guided by these requirements and failure to capture them can result in project over-run.

Each requirement should be:

* **Specific** – target a specific area for improvement.
* **Measurable** – quantify or at least suggest an indicator of progress.
* **Assignable** – specify who will do it.
* **Realistic** – state what results can realistically be achieved, given available resources.
* **Time-related** – specify when the result(s) can be achieved.

Keep the requirements formal but in a user friendly manner. Be unambiguous and succinct.

## 3.1 Integration Requirements

### 

### 3.1.1 User Interfaces

* What existing systems does this system have to integrate with?
* What system interfaces are available for integration?
* Can existing systems export data in XML or CSV?
* Etc.

### 3.1.2 Hardware Interfaces

* Define all existing hardware interfaces
* What new hardware will be needed?
* Etc.

### 3.1.3 Software Interfaces

* Define all existing software interfaces
* What new software will be needed?
* Will the existing software be replaced or upgraded soon?
* How will you handle the change?
* Etc.

### 3.1.4 Communications Interfaces

* What channels of communication will you need into the system?
* What channels of communication will you need out of the system?
* Are there any messaging like requirements?
* What about popup alerts?
* Etc.

## 3.2 Functional Requirements

This section describes all functional requirements of the proposed software project. Each requirement can be grouped into the following categories. **Functional requirements** define the software-functionality that the developers must build into the product to eempower its users, with the ability to accomplish their tasks, and fulfil their businesses requirements.

### 3.2.1 Business Rules

* Define all business rules the new system will need to follow (one per requirement)

### 3.2.2 Transactions, Adjustments, Postings

* Are there transactional roll backs required?
* How are postings handled?

### 3.2.3 Security Access, Protection

* Define all requirements that relate to security

### 3.2.4 Data Capture, Transformation, Interpretation

* Define all Data capture requirements

### 3.2.5 Performance, monitoring and reporting

* Define all performance requirements

### 3.2.6 External Interactions

* Define all external interactions

### 3.2.7 Legal, Environmental, Regulatory constraints

* What existing Legal requirements are there?
* What environmental requirements are there?
* Are there regulatory constraints, if so, how do you prove you are meeting them?

## 3.5 Non-Functional Requirements

Non-functional requirements are more Constraints or standards that have been imposed upon the system, which it must exhibit or comply with. Non-functional requirements define the system’s qual-ity characteristics.

As a general rule of thumb, non-functional requirements can end with “-ity”, although not always!

### 3.5.1 Performance

* What requirements define the performance requirements?
* How many records per minute does the system need to handle?
* Does the system support concurrency?
* Etc.

### 3.5.2 Reliability

* What requirements are there for reliability?
* How will this be measured?
* Does this need to be monitored?
* Etc.

### 3.5.3 Availability

* What up time requirements are there?
* What process will be used to handle system unavailability?
* What load will cause a downtime?
* Etc.

### 3.5.4 Security

* Who will have access to what areas?
* How will those users be authenticated and then authorized?
* How will new users be managed, added to and removed from the system?
* How will the system be protected from abuse?
* What requirements are there on the data being stored?
* Do passwords and credit card information need to be encrypted? if so, how and to what standards?

### 3.5.5 Maintainability

* How will the system be maintained?
* Does it support automatic updates?
* What requirements are their around maintenance, and any potential consequences?
* Etc.

# 5. Change Management Process

Describe here the process that will be followed to handle changes to requirements and document versioning. Describe the process-flow so that new change requests can be made from both the customer and the vender and the resulting process for validating and integrating them into this specification.

Each change request should record:

* Who requested the change?
* When was it requested?
* What is there reason for the change?
* Are there any consequences of making the change?
* What conflicts may arise from this change?
* Is this a new version change or an upgrade change?
* Etc.

# A. Appendices

Appendices can be used to provide additional information.

## A.1 Appendix 1

## A.2 Appendix 2