<PRODUCT NAME>

A

Software Requirements Specifications (SRS)

On

“<Product Name>”

Version X.X approved

Dec-2016

Contents

Revision History

Name	Date	Comments

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Document Conversion

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and reading suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

1.5 Document Copyrights

1.6 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Product Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Features

<Summarize the **Master** features (That can be mentioned in brochure) the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan we are planning to use the Sybase database management studio).>

3. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1 <System Feature 1"Name">

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

3.2 <System Feature 2"Name"> … etc.

4. Interface Requirements

4.1 User Interface

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

4.1.1 System type

<Describe the system type from the operation system type. Ex: Desktop Application compatible with windows 7 and later versions.>

4.1.2 System Colors & Fonts

<Mention all of the colors & fonts that will be used in the GUI design. Ex: Blue (Code: X500), Grey (Code: D6800).>

4.1.3 Keyboard shortcuts

<Write all of the shortcuts that will be implemented in the system.>

Shortcut	Action

4.1.4 Standard buttons

<Define all of the standard buttons that will be displayed in all of the pages as the header, footer, help button, save ...etc.>

4.1.5 Standard system behavior

<Define any standard behavior like the system sound, confirmation messages, pop-up messages...etc.>

4.1.6 Layout constraints

<Define any objects that the customer wants to be standard on any layout like departments logo, company's logo, layout size, full screen, minimize option, resize option, header and footer constraints...etc.>

4.1.7 Language Constraints

<Define any language related constraints for the system, like the names must English...etc.>

4.1.8 Parameters Constraints

<Define if the customer has any requests for naming any part of the system, input and outputs...etc.>

4.1.9 Logical flow Constraints

<Define if the customer has any constraints for the logical flow or the processes order in the line of business.>

4.2 Hardware Interface

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

4.2.1 Devices constraints

<Define if the user wants to run the software on a specific mac addresses for some devices or even control some of the features to function on specific devices.>

4.2.2 Hardware Integration

<Define if there will be any hardware integration between any additional I/O devices other than the screen, mouse, Keyboard.>

4.2.3 Logical Hardware Integration

<Define if the customer wants to let the company design or he will provide some logical boards designed for this product specially.>

4.3 Software Interface

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.3.1 Software Integration

<Define if the customer wants to integrate any other systems in this product like a current CRM system that he/she has implemented already.>

4.3.2 Hardware usage constraints

<Define if the customer wants to have any constraints on the hardware resources usage like the RAM management, CPU usage…etc.>

4.3.3 Integrated services

<Define if there will be any other services added or used by the product itself like Email clients, web browsers, synchronizing mechanisms, encryption requests or even data transfer rates control.>

4.4 Communication Interface

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4.4.1 Software communication constraints

<Define if the customer wants to have any constraints on the way of communication between this product and any other software like encrypted messages, requests…etc.>

4.4.2 Hardware communication constraints

<Define if the customer wants to add any constraints on the communication protocols between the software and the hardware just in case he/she has a pre-installed firewall device, cloud servers…etc.>

4.4.3 Network communication constraints

<Define if the customer has any constraints for the network protocols like the HTTP, FTP…etc.>

5. Non-Functional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

7. Legal confirmation

I am<Delegate name> representing the company <Company's name> and I have read all of the document with the name (Software Requirements Specifications) or SRS, I agree to all of the mentioned in all of the parts, I agree that those requirements and all of the specifications will be working fine with my business processes and my signature here is a prove from my side and the <Company's name>'s side.

I acknowledge that I know that there will be no changes what so ever in the requirements from my side and <Delegate name> side as well and that there will be no support provided from <Delegate name> in any requirements or functionalities that is not mentioned in this document. I also acknowledge that this document is effective starting the date of my signature below.

<Customer's> Signature	Owner Signature
.....	.....
Date of Signature	Date of Signature
.....	.....