**Software Requirements Specification (SRS) Document**

**<Client Name>**

**<Project Name>**

**<Version Number>**

# Document Versioning Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Action** | **Author** | **Approver** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Contents

[Document Versioning Details 2](#_Toc9335576)

[Contents 3](#_Toc9335577)

[1. Introduction 5](#_Toc9335578)

[1.1. Purpose 5](#_Toc9335579)

[1.2. Document Overview and Intended Audience 5](#_Toc9335580)

[1.3. Definitions and abbreviations 5](#_Toc9335581)

[1.4. References 5](#_Toc9335582)

[2. High-Level Description 7](#_Toc9335583)

[2.1 Background 7](#_Toc9335584)

[2.2 Project Scope 7](#_Toc9335585)

[2.3 User/Actor Roles and Characteristics 7](#_Toc9335586)

[2.4 Constraints 8](#_Toc9335587)

[2.5 Assumptions 8](#_Toc9335588)

[2.6 Dependencies 9](#_Toc9335589)

[3. Functional Requirements 10](#_Toc9335590)

[3.1 System Features 10](#_Toc9335591)

[3.2 Business Process Flow 10](#_Toc9335592)

[3.3 Wireframes/Prototype 11](#_Toc9335593)

[3.4 User Interface Requirements 11](#_Toc9335594)

[3.5 Use Case Listing 11](#_Toc9335595)

[4. Non-Functional Requirements 12](#_Toc9335596)

[4.1 Performance Requirements 12](#_Toc9335597)

[4.2 Usability Requirements 12](#_Toc9335598)

[4.3 Security Requirements 13](#_Toc9335599)

[4.4 Training Requirements 13](#_Toc9335600)

[4.5 Recovery Requirements 13](#_Toc9335601)

[5. Other Requirements 15](#_Toc9335602)

[5.1 Quality Control and Testing Requirements 15](#_Toc9335603)

[5.2 System Maintenance and Additional Requirements 15](#_Toc9335604)

[6. Risks 16](#_Toc9335605)

[7. Completion Criteria/Exit Criteria 17](#_Toc9335606)

[8. Appendix 18](#_Toc9335607)

# Introduction

*[A Software Requirement Specifications (SRS) document is a detailed and structured requirements document that contains the functional requirements (depicts behavior), non-functional requirements (illustrates characteristics), and any other requirements that the software must fulfill. An SRS document is also called a 'Product Requirements Document (PRD)' and 'System Requirements Specification' document.*

*This section provides an overview of the SRS document and briefly describes the business problem that is being solved, the scope of project activities, including the stakeholders involved in the project.]*

## Purpose

*[Define the specific intent behind creating the SRS document and what the reader can expect to learn from this document.]*

## Document Overview and Intended Audience

*[This section should contain a general overview of the contents of the SRS document and the details of the desired readers and audience of this document.]*

## Definitions and abbreviations

*[The meaning and description of any project-specific terminology, technical abbreviations, and business acronyms used in the document should be mentioned here. If the project used terms in a different context than the general understanding, even those details should be listed here.]*

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## References

*[List down any of the other reference documents that are mentioned in the document. It may include:*

* *Documents on the shared project repository*
* *URLs or network locations*
* *Websites*
* *White papers*

*Also, the information against the reference artifacts shall be captured in the below table:]*

|  |  |
| --- | --- |
| **Reference Document Details** | **Location** |
| *<Name/details of the artifact>* | *<Respective location in the network or shared project repository>* |
|  |  |
|  |  |
|  |  |
|  |  |

# High-Level Description

## Background

*[This information could be taken from the project charter or the project vision documents and contains the following particulars:*

* *Details of the client's existing business environment*
* *Issues and problems being faced by the client today*
* *What kind of solution is being looked at to solve those problems? (brief explanation)*
* *The objective behind creating the specific product or software]*

## Project Scope

*This section contains an executive summary of what is included and what isn't in the scope of activities under this project. The requirements under this section should be specific, clear, and crisp while also being feasible.*

* **In Scope:** *Provide a high-level summary of what is to be included (in scope) for project completion and should be considered as a part of project efforts. All the features, functionalities, and modules which are 'in scope' define the overall project boundaries.*

*Care should be taken with this section, and only the 'What' of the solution should be included here (and not 'How').*

*Also, items under this section should be subjected to a thorough review (from both internal and external stakeholders) as they will be referenced in case of change requests discussions and disputes.*

* **Out of Scope:** *Provide a high-level summary of what should not be considered (out of scope) for project completion and shouldn't be a part of the current project scope (these requirements can be a part of future phases of the project).]*

## User/Actor Roles and Characteristics

*[Every application or software product being created will have some users interacting with the system. Based on their level of interaction with the system, these actors will have different roles, and the same should be specified under this section; in the below format:*

* **Role Name:** *The name of the role as per the application nomenclature*
* **Role Designation/Title:** *The designation or title of the users to whom this role can be assigned*
* **Role Description*:*** *A brief description of the role itself, its underlined rights & privileges on the application and the characteristics of the user to whom this role is assigned*
* **Frequency of use:** *Specify how often the user with this role uses the system]*

## Constraints

*[Constraints are defined as any 'factors' that marks the boundaries around the project functions or 'dependencies' that describes how the project activities are to be carried out.*

*There could be several factors that could be a project constraint, for example:*

* *User Interface Design: A UI design that is difficult to achieve technically*
* *Code Review/Due diligence: Code reviews couldn't be avoided, but they obviously slow down the application development pace*
* *Human Resources: Getting skilled/specialized resources to work on the project or a fixed size of the development team*
* *Legal: Certain laws or regulation that restricts project activities*
* *Organization Process: Processes that are defined at the organizational level and needs to be adhered to*
* *Quality: Certain quality or conformance related requirements that should be fulfilled*
* *Technical: Any requirement that is not possible to achieve owing to a technical limitation*
* *Duration: Completion of a particular activity within a stipulated time duration*
* *Budget: Project budget is another common constraint that controls the amount of money that could be spent and thus places a boundary on a lot of project functions.]*

## Assumptions

*[Any belief or events or premise that are considered as to be true (even without proof) in the near future is called an assumption. The reason why assumptions are made is since one doesn't know everything about the project aspects right from the very start, certain things are assumed to be true, and future plans are made based on this.*

*In the absence of assumptions, a lot of decisions and plans couldn't be made at all.*

*However, since all the assumptions don't tend to be true, it's imperative to document them and communicate with all the stakeholders about the same.*

*There are some common areas where assumptions are taken:*

* *Resource assumptions: Assumptions that the skilled resources will be available as per the schedule and will work with their full capacity*
* *Technology assumptions: Assumptions that there will be no technical limitations or hindrances*
* *Availability assumptions: Assumptions that all the functionalities, resources, and vendor-related deliverables will be available at their scheduled time*
* *Infrastructure assumptions*
* *Requirements assumptions: Assumptions that all the requirements are thoroughly thought about, reviewed, and approved*
* *Accuracy of schedule: Assumptions that all the in the project tasks and functionalities will be delivered as per the schedule]*

## Dependencies

*[Typically, many project activities are dependent on each other, and the important ones should be listed here.*

*Some examples include:*

* *Resource-based dependencies*
* *Task-based dependencies*
* *Schedule dependencies*
* *Dependencies on vendors]*

# Functional Requirements

*[Functional Requirements describes the operations, capabilities, and activities a system must be able to perform. This section contains a granular level of requirements to enable the reader with a better understanding and includes the specific behaviors, characteristics, and attributes that are expected from the application or the software being developed. Additionally, Flow diagrams, Wireframes, Use cases list, or other similar types of information should be supplemented here.]*

## System Features

*[A functionality is defined as a desired output or result expected from the system after a (series of) input. For any system with a moderate level of complexity, such functionalities are grouped to form a 'feature', and similar features constitute a 'module'.*

*Each of the requirements should be backward traceable by explicitly referencing its source (business requirement ID) in earlier documents and should be organized in the below format:*

1. *Module A: Description of the module*
   1. *Feature 1: Description of the feature*
      1. *Functionality a*
      2. *Functionality b*

*..*

*..*

* + 1. *Functionality n*
  1. *Feature 2:*

*..*

*..*

* 1. *Feature n: ]*

|  |  |  |
| --- | --- | --- |
| **Business Req. ID** | **Module Name** | **Feature name** |
|  |  |  |
|  |  |  |
|  |  |  |

## Business Process Flow

*[Most of the time, software applications follow a hierarchical/sequential flow of events that are initiated after a specific sequence of inputs is carried out.*

*Any such details and/or diagrams related to the solution process flow, data flow, and information flow should come here. Also, if there are multiple processes within the system, the details of where those processes fit on, how they are interconnected (if they are), and to what effect they are used should be included.]*

## Wireframes/Prototype

*[Prototype or mockup is an initial version of a product and visually depicts the end product. A prototype or wireframe of the software being created should be included under this section to show any data or process-based navigation, represent different scenarios, and project the general look and feel of the application being created.*

*The aim of the prototypes should be to validate the understanding of the requirements and get initial feedback against the user interface and design elements.]*

## User Interface Requirements

*[This section contains the requirements about the look and feel of the interface using which the user will interact with the software. Following details shall be included under this section:*

* *The general layout of the application*
* *The way content, as well as data, is presented to a user*
* *Different kinds of navigation available in the software*
* *Representation of dynamic design elements (widgets & menus)*
* *Does the user interface differ based on the user's role/user group?]*

## Use Case Listing

*[Use cases are 'requirement specification document' that accurately describes how a user (actor) will interact with the system being developed, through a flow of events.*

*This section should contain a listing of all the use cases that are supposed to be prepared during the analysis phase of the project, along with a brief description of the functionalities, scenarios, and flows contained within.]*

# Non-Functional Requirements

*[The 'operational characteristics' or non-functional requirements of the system like system response time, performance, scalability, and usability are included here. While trying to figure out what kind of requirements should come under here, try to see them as 'qualities' that the system should have.*

*This section should only contain the details around the requirement without specifying how these requirements are supposed to be satisfied]*

## Performance Requirements

*[List all the application/software performance-related attributes and characteristics over here. All such details will help the technical team carry out the capacity planning for the servers, hardware, and software components of the system being developed.*

*It should have information for:*

* *Details about average load on the system (number of users)*
* *System response time (minimum acceptable and maximum)*
* *System turnaround time*
* *Throughput for optimal performance*
* *Maximum (peak) workload the system should handle*
* *Scalability (ability to control the additional workload if additional hardware/computation processors are added)*
* *Any assumptions regarding the performance*
* *Specify limits for how long it's tolerable for different types of fault to remain undetected (fault detection and prevention)*
* *How long will the system be available (all day or at specific times?)*
* *How will the users learn of unavailability?*
* *Any fallback activities needed in case of non-availability]*

## Usability Requirements

*[Applications and products should be easy to understand and efficient to use. Any requirements regarding the usage of the system being developed should come here; examples include:*

* *Ease of use*
* *Effectiveness of use*
* *Speed of operations*
* *Intuitiveness/understandability*
* *Satisfaction]*

## Security Requirements

*[These requirements define how secure (from unethical and non-permitted users) the application and its network, servers, operating systems and infrastructure are, and depends on the kind of software or application being built and the data it contains.*

*The analyst should spend a great deal of time with the technical & IT department of the customer and try to get all such details. However, in the absence of the same, the standard security requirements applicable in the industry should be proposed, like:*

* *User authentication*
* *User authorization*
* *Data access*
* *Access control*
* *Data integrity*
* *Vulnerability (to hacking) testing]*

## Training Requirements

*[Any kind of requirement where the users have to be trained regarding the usage of the system should come here.*

*It includes:*

* *End-user training*
* *Training of support personnel*
* *Preparation of training manuals/guides/videos]*

## Recovery Requirements

*[These are the requirements that ensure that the business and services are up and running even in the event of any natural disasters and unforeseen circumstances (also called Business Continuity).*

*It should contain:*

* *What will qualify as a disaster (its definition)?*
* *Creation of Disaster Recovery (DR) plan: Plan which defines how the services will be restored in the event of a disaster*
* *Recovery Point Objective (RPO): Acceptable amount of data that can be lost in the event of a disaster*
* *Recovery Time Objective (RTO): Acceptable amount of time the application and/or services will be down in the event of a disaster*
* *What are the performance degradation tolerance limits in the event of a disaster?*
* *How often a DR mock drill needs to run?]*

# Other Requirements

## Quality Control and Testing Requirements

*[This section contains all the requirements and activities that need to be exercised to ensure that the quality of the application is of high standards and stays within the defined limits through the project development life cycle.*

*These activities include, but are not limited to:*

* *Defining the extent of testing required (testing scope and not in scope)*
* *Identifying the nature of the testing (manual or automated)*
* *Determining the type of testing to be performed (functional, performance, security, etc...)*
* *Testing techniques, methodologies, and strategies to be used*
* *Frequency of the testing activities*
* *What all types of documentation need to be prepared against the testing*
* *Metrics that needs to be captured to know the progress of testing on the project*
* *Any other testing related requirement]*

## System Maintenance and Additional Requirements

*[Requirements that don't fit in any of the sections defined above should come in here.*

*Some of the sections that could be included here are:*

* *Project management requirements: Any requirements related to how the complete project (or some specific sections thereof) should be managed, monitored, or controlled should come in here.*
* *System maintenance: Any requirements related to how the system needs to be maintained post its completion and go-live*
* *Storage requirements: Any requirements as to how the data should be stored, segregated, or clustered (particularly in data or process heavy applications)*
* *Resource/workforce requirements: Special notes regarding the type, experience levels, competency of resources that needs to be engaged for developing the application]*

# Risks

*[Any uncertain events or factors that pose a threat to the successful implementation of the project activities are classified as a risk. This section should contain any of the initial risks foreseen based on what is known about the project to date.*

*Some of the areas from where risk could emanate are:*

* *Availability of resources/skilled resources*
* *Incorrect/overly optimistic estimations*
* *Assumptions tend to be false*
* *Unclear/poorly defined requirements*
* *Scope creep*
* *Learning curve*
* *Lack of change management*
* *Requirements are not reviewed/signed off*
* *Vague or inadequate project communications*
* *Technical/implementation risks]*

# Completion Criteria/Exit Criteria

*[It's essential to have the definition of 'complete' defined from the perspective of the project, and the criteria that need to be fulfilled to mark the completion of project tasks are detailed out under this section. For example:*

* *Compliance of the developed software with the respective documented requirements*
* *Unit testing of functionality*
* *Code review activities that should be carried out*
* *System testing*
* *Integration testing*
* *No. of permitted defects]*

# Appendix

*[This section includes any supplementary material and supporting artifacts that will help the reader strengthen their understanding of the SRS document. However, essential information should always be provided in the main document body*

* *More information on the current state of processes*
* *Any type of graphs, charts, tables, figures, diagrams, etc…*
* *Process flows, detailed specifications, technical details, etc…*
* *Any other information set that is too detailed or out of place for the main body of the SRS template*

*It should be noted that if possible, the original documents should not be attached here; instead, the names and links to the location of those documents should be provided]*