

SOFTWARE REQUIREMENT SPECIFICATION -

A FOUNDATION FOR BUILDING
A SUCCESSFUL APPLICATION



What is a Software Requirement Specification (SRS)?

It is a document that contains a full and clear description of a product to be developed. This document includes functional and non-functional requirements and establishes the tasks that the software is set to accomplish. As one of the core values of the SRS is attaining the customer's feedback, it must be written in a way that is easy to read and understand.

SRS is incredibly important.

And here are some reasons why:

- It allows you to get accurate estimation of costs, risks and schedules
- It helps a client to clarify his or her own vision
- Both the Customer and Vendor have a unified understanding of the product
- It helps to establish the most important features and get rid of all the unnecessary ones
- It serves as a basis for other technical documentation

- A team can optimise the development process, therefore - save time
- There won't be any task duplication
- It allows for any problems to be structured, addressed and solved much easier
- It becomes a basis for the testing and validation verification

Which problems might occur, if you don't have a specification?

- It'll be impossible to get an accurate estimation
- Customer and Vendor might very well have a completely different understanding of the project
- Developers won't be able to tell whether the software they build fully embodies client's requirements
- It'll be hard to create user manuals properly
- There's every possibility of a redesign at a later stage

How does it work in Magora Systems?

Step 1. Initial documentation

Our project manager contacts a client and requests all the existing data concerning the future product, which he passes on to the analyst. This information might include a previous specification, workflow, user scenarios or simply a text description of the software. The analyst might ask the client some additional questions, as it is vital to reach a complete understanding of their requirements and vision of the system.

Step 2. Initial Specification

After processing all the gathered data, the analyst composes the first version of the SRS. It includes a general description of the project, and structure of the application. The document explains how the

interface works and how users interact with the system.

Typically, a specification includes the following:

1. Introduction. Provides an overview of the SRS content.

1.1. Purpose.

1.2. Scope - a section describing what the software should and shouldn't do. It also contains detailed information about the objectives and goals to be achieved.

1.3. Definitions, Acronyms, and Abbreviations. This chapter includes explanations for all the referred specific terms, turning an SRS into a readable and understandable document.

1.4. Organisation of the SRS.

2. Overall Description.

This part outlines the major defining factors for the product and its requirements. You won't find any specific requirements here, as they'll be described in the section 3 below. But the chapter will help to get a better grasp of what they're all about.

2.1. Product Perspective focuses on the product's relation to any other products or projects. It states whether the software is going to be independent or part of a bigger system.

2.2. Product Functions contains a summary of the software functionality presented in such a way, that it would be easy for anyone to read and understand.

2.3. User Characteristics describe future users and how the specific requirements are influenced by them.

2.4. General Constraints depict existing limitations for developers concerning system design.

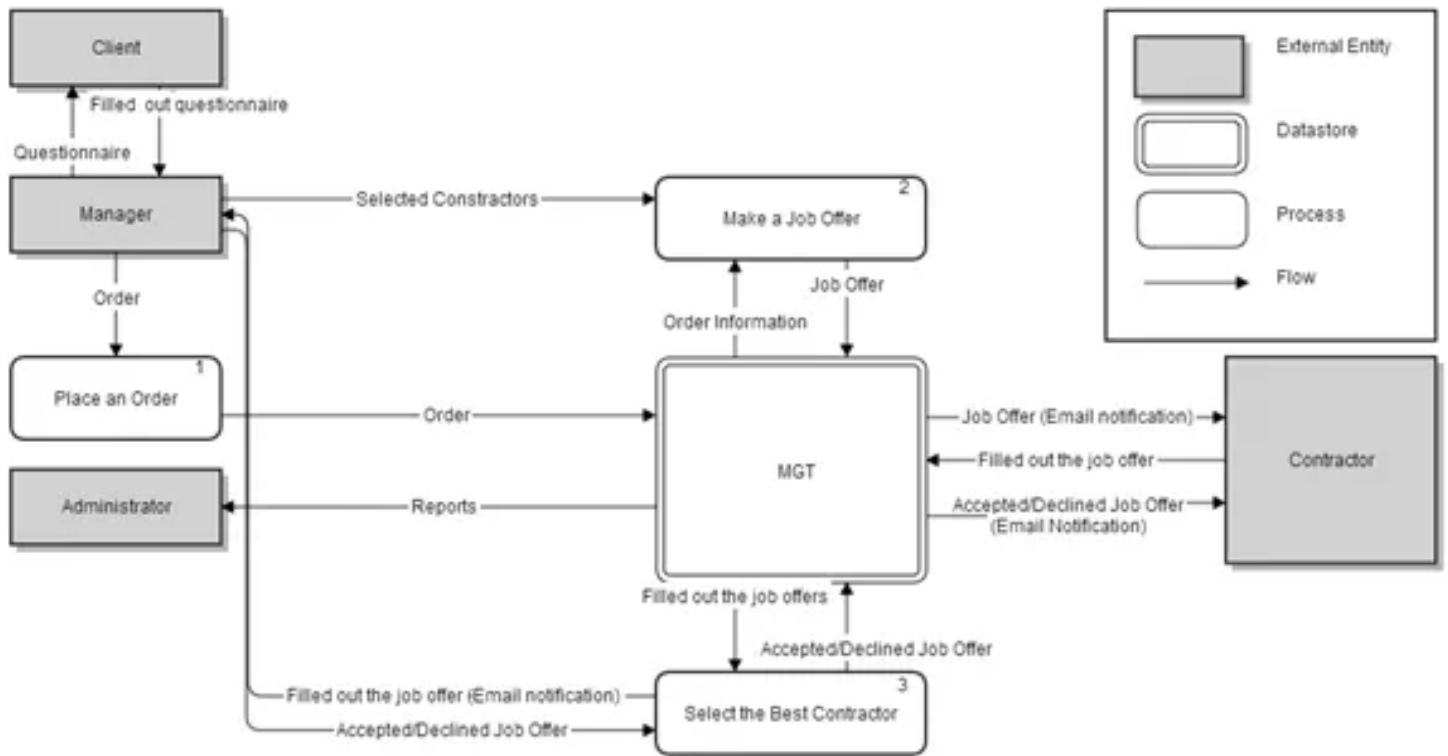
3. Specific Requirements. This is the core part of the SRS. Here the developer will find all the details needed for them to build the software correctly. The Requirements must be represented in a well-structured, logical and readable fashion.

3.1. Functional Requirements are all about the things an app does. This subsection includes a declaration of how each input should be transformed to its output, and a list of the required operations. If necessary, it also includes a rationalisation of some of the requirements.

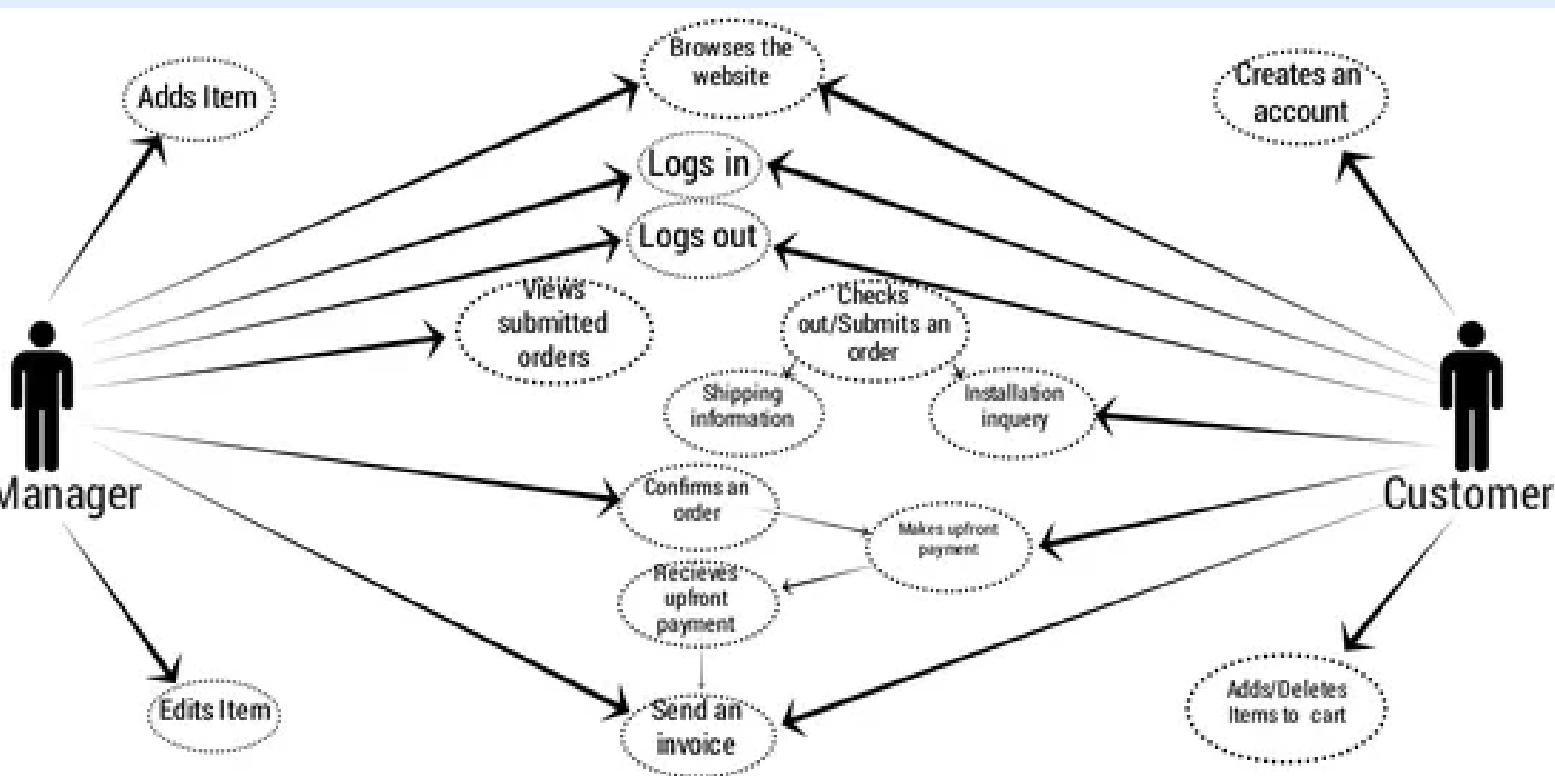
3.2. Non-Functional Requirements define the criteria for the software functioning evaluation, such as productivity, data safety and system security.

4. Special requirements.

4.1. *Data Flow Diagram* determines the input and output data, its sources, destinations and places of storage.



4.2. *Use Case Diagram*. The purpose of this diagram is to demonstrate how objects interact with the software and map out the basic functionality.



5. Mock-ups. Another great way to start a project is to build mock-ups. They allow a client to see the basic structure and usability of the application.

Step 3. Specification is reviewed by a client

The document is passed on to the customer. It will allow them to see the bigger picture and figure out the details. After that they evaluate the specification: approves or sends back for revision.

Step 4. Refinement

The specification is modified until it meets all the customer requirements.

Step 5. Proposal and milestones

The project manager breaks the project down into milestones. The development process is based on the SRS written FOR the client and is based on THEIR INPUT.