**Content in Blue color to be deleted and write the document for CRS**

**Software Requirement Specification Template**

**Course Registration System**

**1. Define the Purpose with an Outline (Or Use an SRS Template)**

**1. Introduction**

           1.1 Purpose

           1.2 Intended Audience

           1.3 Intended Use

           1.4 Product Scope

           1.5 Definitions and Acronyms

**2. Overall Description**

           2.1 User Needs

           2.2 Assumptions and Dependencies

**3. System Features and Requirements**

            3.1 Functional Requirements

            3.2 External Interface Requirements

            3.3 System Features

            3.4 Nonfunctional Requirements

**2. Define your Product’s Purpose**

This introduction is very important as it sets expectations that we will come back to throughout the SRS.

Some items to keep in mind when defining this purpose include:

*Intended Audience and Intended Use*

Define who in your organization will have access to the SRS and how they should use it. This may include developers, testers, and project managers. It could also include stakeholders in other departments, including leadership teams, sales, and marketing. Defining this now will lead to less work in the future.

*Product Scope*

What are the benefits, objectives, and goals we intend to have for this product?

This should relate to overall business goals, especially if teams outside of development will have access to the SRS.

*Definitions and Acronyms*

Clearly define all key terms, acronyms, and abbreviations used in the SRS. This will help eliminate any ambiguity and ensure that all parties can easily understand the document.

If your project contains a large quantity of industry-specific or ambiguous terminology or acronyms, you may want to consider including a reference to a project glossary, to be appended to the SRS, in this section.

*Need to create a PRD?*

**3. Describe What You Will Build**

Description of what you’re going to build.

Why is this product needed?

Who is it for? Is it a new product?

Is it an add-on to a product you’ve already created?

Is this going to integrate with another product?

Understanding and getting your team aligned on the answers to these questions on the front end makes creating the product much easier and more efficient for everyone involved.

*User Needs*

Describe who will use the product and how. Understanding the various users of the product and their needs is a critical part of the SRS writing process.

Who will be using the product?

Are they a primary or secondary user?

What is their role within their organization?

What need does the product need to fulfill for them?

Do you need to know about the purchaser of the product as well as the end user?

For the development of medical devices and med device software, you may also need to know the needs of the patient.

*Assumptions and Dependencies*

What are we assuming will be true?

Understating and laying out these assumptions ahead of time will help with headaches later. Are we assuming current technology?

Are we basing this on a Windows framework?

We need to take stock of these technical assumptions to better understand where our product might fail or not operate perfectly.

Finally, you should note if your project is dependent on any external factors. Are we reusing a bit of software from a previous project?

This new project would then depend on that operating correctly and should be included.

**4. Detail Your Specific Requirements**

In order for your development team to meet the requirements properly, we must include as much detail as possible. This can feel overwhelming but becomes easier as you break down your requirements into categories. Some common categories are functional requirements, interface requirements, system features, and various types of nonfunctional requirements:

*Functional Requirements*

[Functional requirements](https://www.perforce.com/blog/alm/what-are-functional-requirements-examples) are essential to your product because, as the name implies, they provide some sort of functionality.

Asking yourself questions such as “does this add to my tool’s functionality?” or “what function does this provide?” can help with this process. Within medical devices especially, these functional requirements may have a subset of domain-specific requirements.

You may also have requirements that outline how your software will interact with other tools, which brings us to external interface requirements.

*External Interface Requirements*

External interface requirements are specific types of functional requirements. These are especially important when working with embedded systems. They outline how your product will interface with other components.

There are several types of interfaces you may have requirements for, including:

* User
* Hardware
* Software
* Communications

*System Features*

System features are a type of functional requirements. These are features that are required in order for a system to function.

*Nonfunctional Requirements*

[Nonfunctional requirements](https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples), which help ensure that a product will work the way users and other stakeholders expect it to, can be just as important as functional ones.

These may include:

* Performance requirements
* Safety requirements
* Security requirements
* Usability requirements
* Scalability requirements

The importance of each of these types of nonfunctional requirements may vary depending on your industry. In industries such as medical device, life sciences, and automotive, there are often regulations that require the tracking and accounting of safety.

**5. Deliver for Approval**

We made it! After completing the SRS, you’ll need to get it approved by key stakeholders. This will require everyone to review the latest version of the document.

\*\*\*\*\*\*\*