# Software Requirements Specification (SRS)

**Submission Date: 23/08/24 – Friday**

# What is SRS Document?

A software requirements specification (SRS) is a document that describes **what the software will do and how it will be expected to perform**. It also describes **the functionality the product needs to fulfill all stakeholders (business, users) needs**.

An SRS can be simply summarized into four Ds:

* Define your product's purpose.
* Describe what you are building.
* Detail the requirements.
* Deliver it for approval.

In SRS we have to DEFINE the purpose of our product, DESCRIBE what we are building, DETAIL the individual requirements, and DELIVER it for approval.

An SRS gives you a complete picture of your entire project. It provides a single source of truth that every team involved in development will follow.

# How to Write an SRS Document ?

Contents of SRS Document:

## Introduction

* 1. Purpose
  2. Definitions and Acronyms

## Overall Description

* 1. User Needs
  2. Assumptions and Dependencies

## System Features and Requirements

* 1. Functional Requirements
  2. Nonfunctional Requirements

## ​Introduction

### Define the Purpose

Define the Purpose of the product. **Purpose** of the digital product is a clear and concise statement that defines the intent of the solution. This statement should address your needs, outlining what the app will achieve once completed .

### 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.

## ​Overall Description

Here, give a description of what you’re going to build. Is it a new product?

Why is this product needed? Who is it for?

### User Needs

Describe who will use the product and how. Understanding the user of the product and their needs is a critical part of the process.

Who all will be using the product? How they will be using the product?

### Assumptions and Dependencies

What all 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 assumptions to better understand when our product would fail or not operate perfectly.

## ​System Features and Requirements

In order for your development team to meet the requirements properly, we must include as much detail about them as possible.

### Functional Requirements

Functional requirements of a product means the functionality provided by the product/system. **Functionality** of the product describes the features, capabilities, and limitations or constraints that might affect the software’s performance.

Asking yourself the question “does this add to my tool’s functionality?” Or “What function does this provide?” can help with this process.

*Note: Format of wrting Functional Requirements is given separately*

### Nonfunctional Requirements

Non-functional requirements state constraints on the design and construction of the product.

Nonfunctional requirements ensure that a product will work the way users and other stakeholders expect it to. They are as important as functional ones.

A system can still work if non functional requirements are not met, but it may not meet user or stakeholder expectations.

Examples of nonfunctional requirements are:

* + - Performance requirements
    - Safety requirements
    - Security requirements
    - Usability requirements
    - Scalability requirements

Write only the non functional requirements that are relevant/ applicable to your product.