**ASSIGNMENT-2**

(Important things to learn)

**TASK-1: Software Requirement Specification (SRS)**

Prepare a Software Requirement Specification (SRS) document for either a Shopping or E-commerce Project. Outline the functional and non-Functional requirements, user stories, and system

specifications.

**Software Requirements Specification (SRS):**

* 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 fulfil the needs of all stakeholders (business, uses).The elements that comprise an SRS can be simply summarized into four Ds:
  + [**Define** your product's purpose.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#purpose)
  + [**Describe** what you're building.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#describe)
  + [**Detail** the requirements.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#detail)
  + [**Deliver** it for approval.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#approve)
* We want to DEFINE the purpose of our product, DESCRIBE what we are building, DETAIL the individual requirements, and DELIVER it for approval. A good SRS document will define everything from how software will interact when embedded in hardware to the expectations when connected to other software. An even better SRS document also accounts for the needs of real-life users and human interaction.

### Why Use an SRS Document?

* An SRS gives you a complete picture of our entire project. It provides a single source of truth that every team involved in development will follow. It is our plan of action and keeps all our teams — from development and testing to maintenance — on the same page.
* An SRS not only keeps your teams aligned and working toward a common vision of the product, it also helps ensure that each requirement is met. It can ultimately help you make vital decisions on our product’s lifecycle, such as when to retire an obsolete feature.
* It takes time and careful consideration to create a proper SRS. But the effort it takes to write an SRS is gained back in the development phase. It helps our team better understand your product, the business needs it serves, its users, and the time it will take to complete.

## How to Write an SRS Document

* Creating a clear and effective SRS document can be difficult and time-consuming. But it is critical to the efficient development of a high quality product that meets the needs of business users.
* Here are five steps you can follow to write an effective SRS document.

### 1. Define the Purpose With an Outline

* Our first step is to create an outline for your software requirements specification. This may be something you create yourself, or you can use an existing SRS template.
* If you’re creating the outline yourself, here’s what it might look like:
  + **Introduction**
    - Purpose
    - Intended Audience
    - Intended Use
    - Product Scope
    - Definitions and Acronyms
  + **Overall Description**
    - User Needs
    - Assumptions and Dependencies
  + **System Features and Requirements**
    - Functional Requirements
    - External Interface Requirements
    - System Features
    - Non-functional Requirements

### 2. Define your Product’s Purpose

* 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.
* **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.

### 3. Describe What You Will Build

### Your next step is to give a 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.
* **Assumptions and Dependencies:**We need to take stock of these technical assumptions to better understand where our product might fail or not operate perfectly.

### 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 non-functional 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. 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.

#### Non-functional Requirements: [Non-functional 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 non-functional 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**

* 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.

**Software Requirement Specification (SRS) Document for E-commerce Project**

**1. Introduction**

* **Purpose:** The purpose of this Software Requirement Specification (SRS) document is to provide a detailed description of the requirements for the E-commerce Project. This document will outline the functional and non-functional requirements, user stories, and system specifications necessary to develop a robust and user-friendly e-commerce platform.
* **Scope:** The e-commerce platform will allow users to browse, search, and purchase products online. It will include functionalities for user registration, product catalog management, shopping cart, order management, payment processing, and user account management. The platform will be accessible via web browsers and will provide an intuitive and responsive user interface.
* **Definitions, Acronyms, and Abbreviations:**
* **SRS:** Software Requirement Specification
* **UI:** User Interface
* **UX:** User Experience
* **API:** Application Programming Interface
* **SKU:** Stock Keeping Unit
* **References:** IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications. ISO/IEC/IEEE 29148:2018 Systems and software engineering — Life cycle processes — Requirements engineering

**2. Overall Description**

* **Product Perspective:** The e-commerce platform is a standalone web application that will be built using modern web technologies. It will interact with third-party payment gateways and shipping providers. The platform will consist of client-side and server-side components.
* **Product Functions:**
* User registration and authentication
* Product catalog browsing and searching
* Product detail viewing
* Shopping cart management
* Order placement and tracking
* Payment processing
* User profile management
* Admin panel for managing products, orders, and users
* **User Classes and Characteristics:**
* **Shoppers:** Users who browse and purchase products.
* **Admin Users:** Users who manage the platform, including product listings, orders, and user accounts.
* **Operating Environment:**
* Web browsers (Chrome, Firefox, Safari, Edge)
* Server environment (Linux-based servers with Apache or Nginx)
* Database (MySQL or PostgreSQL)
* Payment gateways (e.g., Stripe, PayPal)
* **Design and Implementation Constraints:**
* Compliance with web accessibility standards (WCAG 2.1)
* Secure handling of user data and payment information (PCI-DSS compliance)
* Scalability to handle high traffic during peak times
* **Assumptions and Dependencies:**
* Users have internet access and a modern web browser.
* Payment processing depends on third-party payment gateways.
* Shipping and delivery services are provided by third-party providers.

**3. Functional Requirements**

* User Registration and Authentication
* Users shall be able to register using an email address and password.
* Users shall be able to log in using their registered email and password.
* Users shall be able to reset their password if forgotten.
* Product Catalog
* Users shall be able to browse products by category.
* Users shall be able to search for products by name, category, or SKU.
* Users shall be able to view detailed information about a product.
* Shopping Cart
* Users shall be able to add products to the shopping cart.
* Users shall be able to view the contents of their shopping cart.
* Users shall be able to update the quantity of items in the cart.
* Users shall be able to remove items from the cart.
* Order Management
* Users shall be able to place an order for the items in their cart.
* Users shall receive an order confirmation via email.
* Users shall be able to track the status of their orders.
* Payment Processing
* Users shall be able to pay for their orders using a credit card.
* Users shall receive payment confirmation via email.
* Payment information shall be securely processed and stored.
* User Profile Management
* Users shall be able to view and update their personal information.
* Users shall be able to view their order history.
* Users shall be able to manage their shipping addresses.
* Admin Panel
* Admin users shall be able to manage product listings.
* Admin users shall be able to manage orders.
* Admin users shall be able to manage user accounts.

**4. Non-Functional Requirements**

* Performance Requirements
* The platform shall handle up to 10,000 concurrent users.
* Page load times shall not exceed 2 seconds under normal conditions.
* Security Requirements
* User data shall be encrypted both in transit and at rest.
* The platform shall implement multi-factor authentication for admin users.
* The platform shall comply with GDPR for user data privacy.
* Usability Requirements
* The platform shall be accessible to users with disabilities (WCAG 2.1 compliance).
* The platform shall provide a responsive design for use on various devices.
* Reliability Requirements
* The platform shall have an uptime of 99.9%.
* The platform shall implement data backup and recovery procedures.
* Maintainability Requirements
* The platform codebase shall follow industry best practices for code quality.
* The platform shall include automated tests for critical functionalities.

**5. User Stories**

* Shopper User Stories
* As a shopper, I want to register an account so that I can make purchases.
* As a shopper, I want to search for products so that I can find items I am interested in.
* As a shopper, I want to add items to my cart so that I can purchase them later.
* As a shopper, I want to check out my cart so that I can complete my purchase.
* As a shopper, I want to track my order so that I know when it will arrive.
* Admin User Stories
* As an admin, I want to add new products so that they are available for purchase.
* As an admin, I want to update product information so that it is accurate.
* As an admin, I want to view and manage orders so that I can ensure they are fulfilled.
* As an admin, I want to manage user accounts so that I can provide support and resolve issues.

**6. System Specifications**

* Hardware Requirements
* Server: Minimum of 8 CPU cores, 16GB RAM, 500GB SSD.
* Database Server: Minimum of 4 CPU cores, 8GB RAM, 250GB SSD.
* Software Requirements
* Operating System: Linux (Ubuntu 20.04 or later)
* Web Server: Apache 2.4 or Nginx 1.18
* Database: MySQL 8.0 or PostgreSQL 12
* Programming Language: Python 3.8 or later, JavaScript (Node.js 14 or later)
* Frameworks: Django 3.2 (for backend), React 17 (for frontend)
* Network Requirements
* Internet: High-speed internet connection with a minimum bandwidth of 100 Mbps.
* Security: SSL/TLS for secure data transmission.