

# **Software Requirement and Design Specification (SRDS)**

Documentation Guideline

5th Semester

**Department of Computer Science and Engineering  
July 2025**

# Documentation Rules

- Report must follow this template.
- Minimum page count: **20–30 pages** (excluding title page and TOC).
- Include at least **3 system diagrams** in the diagrams' section.
- Describe at least **7–10 functional requirements**.
- Include **5 non-functional requirement categories**.
- Provide at least **2 algorithm descriptions**.
- The final SRDS report must be printed in **A4 size** on one side of the page only.
- **Submission Deadline: 1st September 2025 (11:59 PM).**
- **Plagiarism or duplicate submissions will result in disqualification. (0 Mark)**

## Design and Formatting Rules

- Cover page must include **Project name, Student id and Name, Department, Semester, Date**. Cover page design should look professional.
- Use spiral binding.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	About the System . . . . .	3
1.2	Purpose of the System . . . . .	3
1.3	Importance of the System . . . . .	4
<b>2</b>	<b>System Overview</b>	<b>5</b>
2.1	System Features . . . . .	5
2.2	Intended Users . . . . .	5
<b>3</b>	<b>Requirements</b>	<b>6</b>
3.1	Functional Requirements . . . . .	6
3.2	Non-Functional Requirements . . . . .	6
3.3	System Environment . . . . .	6
<b>4</b>	<b>System Design</b>	<b>7</b>
4.1	System Components Overview . . . . .	7
4.2	System Diagrams . . . . .	8
4.3	Database Structure . . . . .	8
4.4	Basic Algorithm Descriptions . . . . .	8
<b>5</b>	<b>Appendix</b>	<b>9</b>
5.1	References . . . . .	9

# 1. Introduction

## 1.1 About the System

Briefly describe what the system is about.

- What is the name of the system?
- What is the system designed to do?
- What type of users or organization is it intended for?
- Is it a web, desktop, or mobile application? Or something else?
- What are some real-world problems this system addresses?

## 1.2 Purpose of the System

Explain why this system is being made. Use 5–10 bullet points with descriptions. Mention if it is trying to improve an existing system. If so, describe how?

- What specific problems or limitations does the current system (if any) have?
- What are the main goals of building this system?
- What improvements or innovations does it aim to introduce?
- What are the expected outcomes or benefits of developing this system?

## 1.3 Importance of the System

Mention the benefits or importance of this system.

- How does the system improve efficiency, accuracy, or usability?
- Will it save time, money, or human effort? How?
- Does it bring convenience or automation to a manual process?
- How will it benefit end users or organizations in the long term?
- Are there any social, environmental, or business impacts to consider?

## 2. System Overview

### 2.1 System Features

Describe the key features and functionalities of the system. Organize them into a structured list or table with clear descriptions.

- What are the main modules or functionalities of the system?
- What can a user do with this system (e.g., login, submit forms, generate reports)?
- Which features are essential and which are optional or enhancements?
- Are there any automation or intelligent features included?
- Can any of the features be customized by the user?

### 2.2 Intended Users

Identify the primary users of the system, as well as any secondary stakeholders who may interact with or be affected by the system.

- Who will primarily use this system (e.g., students, admins, customers)?
- Are there different types of users with different roles and permissions?
- Who are the secondary stakeholders (e.g., managers, IT staff, government)?
- How will each user type interact with the system?
- Are there specific needs or limitations of the users that the system considers?

## **3. Requirements**

### **3.1 Functional Requirements**

List all the specific things the system must be able to do.

### **3.2 Non-Functional Requirements**

Describe system performance, security needs, usability, etc.

### **3.3 System Environment**

Mention the hardware, software, and tools needed.

## 4. System Design

### 4.1 System Components Overview

Break down the system into key modules or parts. Take a look at the following example:

- **User Interface (UI):** Responsible for interacting with users. This includes forms, buttons, input fields, and screens where users can perform actions or view information.
- **Authentication Module:** Manages user registration, login, logout, and password management. Ensures that only authorized users can access protected parts of the system.
- **Database Management:** Handles the storage, retrieval, updating, and deletion. This includes maintaining tables for users, transactions, reports, and other related data.
- **Business Logic Layer:** Contains the core logic of the application. Process user requests, perform calculations, apply business rules, and coordinate interactions between the user interface and the database.
- **Report Generation Module:** Allows users to generate and download reports based on stored data, filtered by specific criteria such as date ranges or user types.
- **Notification Service:** Sends notifications such as email, SMS, or in-app to users for important system events such as successful registrations, updates, or warnings.

**Note:** These are some example system components. These could vary depending on your specific software system.



## **4.2 System Diagrams**

Include use case diagrams, flow diagrams, or class diagrams.

## **4.3 Database Structure**

Outline tables and relationships (if your system uses a database).

## **4.4 Basic Algorithm Descriptions**

Explain important algorithms or processes in simple terms.

## 5. Appendix

### 5.1 References

List any documents, books, or websites that are referred to.