Software Requirement and Design Specifications

[Project Title]

Version: [xx.xx]

Course Code	
Instructor	
Project Team	
Submission Date	

[Instructions]

- *No section of template should be deleted. You can write 'Not applicable' if a section is not applicable to your project. But all sections must exist in the final document.*
- *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.*
- *This' Instruction' section should also be removed in final document.*

Table of Contents

1. Introduction

1.1. Purpose of Document

[Describe the purpose of this document.]

1.2. Intended Audience

[Describe people who are concerned with or are expected to use this document.]

1.3 Definition of Terms, Acronyms and Abbreviations

[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.]

Term	Description
ASP	Active Server Pages
DD	Design Specification

1.4 Document Convention

[Describe the font and font size that this document will be using]

2. Overall System Description

2.1. Project Background

[This section will establish business context in which system is being built. This will describe background information and will mention the actual problem / opportunity in business that triggered the project.]

2.2. Project Scope

[This section will give an overview of project scope. This of project and will mention project boundaries and main functionalities that will be addressed in the system. Describe what the system will and will not do]

2.3. Not In Scope

[This section will highlight/explicitly mention the functionalities (if any) that are not in the scope of current project.]

2.4. Project Objectives

[This section will describe the objectives of project that how it is going to address the problem\opportunity identified in business environment and what would be the end result of project.]

2.5. Stakeholders

[This section will describe stakeholders of the system. This will include different business user classes that are expected to interact with system and similarly the technical people who are going to be involved in software development/management]

2.6. Operating Environment

[Describe the environment in which the software will operate, including the hardware platform, operating system, network environment and other software components or applications with which it must coexist.]

2.7. System Constraints

[Describe the constraints imposed on the system by the external environment. External environment may be caused by the stakeholders, business conditions, technical issues, academic requirements etc and may include the following:

- *Software constraints*
- *Hardware constraints*
- *Cultural constraints (includes language etc.)*
- *Legal constraints*
- *Environmental constraints (e.g., the environment where the software will be installed, It could be a noisy environment, which may require that there is no sound event in the project).*
- *User constraints (e.g., the project is developed for children, so it may be required that the project has more graphic controls rather than textual controls).*
- *Off the shelf components that might be used in the project may have their constraints that are consequently transferred to the project.]*

2.8. Assumptions & Dependencies

[This section will identify:

- *Any assumptions taken regarding the system or environment*
- *Any dependency of system on any external factor.]*

3. External Interface Requirements

[This section is intended to specify any requirements that ensure that the new system will connect properly to external components. Place a context diagram showing the external interfaces at a high level of abstraction.]

3.1. Hardware Interfaces

[Describe the characteristics of each interface between the software and hardware components of the system. This description might include the supported device types, the nature of the data and control interactions between the software and the hardware.]

3.2. Software Interfaces

[Describe the connections between this system and other external software components (identified by name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify and describe the purpose of the data items or messages exchanged among the software components. Describe the services needed and the nature of the inter-component communications. Identify data that will be shared across software components.]

3.3. Communications Interfaces

[Describe the requirements associated with any communication functions the system will use, including e-mail, web browser, network communications standards or protocols, electronic forms, and so on. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, and synchronization mechanisms.]

4. Functional Requirements

4.1. Functional Hierarchy

[This section will give a big picture of overall system functionality. The main modules/features of system and their sub-functions will be described here in the form of a functional hierarchy so that, before getting into the use case, audience could grab the idea of overall system functions.]

4.2. Use Cases

4.2.1. [Title of use case]

[Use Case Diagram]
[Use Case Description]

Use Case Description	
Use Case name:	
Use Case Description:	
Primary actor:	***Other actors:***
Stakeholders:	
Relationships ▪ ***Includes:*** ▪ ***Extends:***	
Pre-conditions: ▪	
Flow of Events: ***1. Actor does….*** ***3.*** ***4.***	
Alternative and exceptional flows: ***4.1*** ….	
Post-conditions: ▪	

5. Non-functional Requirements

5.1. Performance Requirements

[The performance characteristics of the system that are required by the business should be outlined in this section. Performance characteristics include the speed, precision, concurrency, capacity, safety, and reliability of the software. These characteristics define the performance of the project.]

5.2. Safety Requirements

[Specify the requirements that are concerned with possible loss, damage, or harm that could result from the use of the system. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the system must conform.]

5.3. Security Requirements

[Specify any requirements regarding security, integrity, or privacy issues that affect the use of the system and protection of the data used or created by the system. Define all user authentication or authorization requirements, if any. Identify any security or privacy policies or certifications the system must satisfy.]

5.4. User Documentation

[List the user documentation components that will be delivered along with the software, such as user manuals, online help, context-sensitive help and tutorials.]

SDS

6. System Architecture

[This section should provide a high-level overview of how the functionality and responsibilities of the system are partitioned and then assigned to subsystems or components. The main purpose is to gain a general understanding of how the system is decomposed, and how the individual parts work together to provide the desired functionality].

6.1. System Level Architecture

[The architecture should decompose the system at a top level in a way that provides a foundation for more detailed design work. The architecture discusses relationships and roles of system elements (subsystems, components, modules, etc.), but does not provide internal details. Areas for consideration are:

- *System decomposition into elements*
- *The relationship between the elements*
- *Interfaces to external systems*
- *Major physical design issues such as where elements will execute*
- *Global design strategies such as error handling*
-

NOTE: You will use UML diagrams (Deployment and Component diagrams) to document the overall system architecture.]

7. Design Strategy

[Describe the design strategies or decisions that impact the overall organization of the system and its high-level structures. This information should provide the reader with insights into the key abstractions and mechanisms used in the system architecture.

For the strategy, discuss the reasoning employed (possibly referring to previously stated design goals and principles) and any trade-offs. Areas for consideration include:

- *Future system extension or enhancement*
- *System reuse*
- *User interface paradigms*
- *Data management (storage, distribution, persistence)*
- *Concurrency and synchronization]*

8. Detailed System Design

[A detailed design should include the following:

- *Detailed class diagram along with a detailed description of all attributes, functions or methods specifying interactions between different classes/modules.*
- *Logical data model (E/R model)*
- *Detailed GUIs]*

8.1. Database Design

[A detailed Database design should include the following:

- *Logical data model (E/R model)*
- *Data dictionary]*

ER Diagram

[Entity Relationship Diagram of the system with description]

8.2. Application Design

[A detailed application design should include the following:

- *Detailed Sequence diagram and Collaboration diagram with parameter list*
- *State Transition Diagram*
- *Activity Diagram*

8.2.1 Sequence Diagram

8.2.2 State Diagram

8.2.3 Activity Diagram

9. References

[This section should provide a complete list of all documents referenced at specific point in time. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. (This section is like the bibliography in a published book).]

10. Appendices

[This section should include supporting detail that would be too distracting to include in the main body of the document.]