



HARAMAYA UNIVERSITY
COLLEGE OF COMPUTING AND INFORMATICS
DEPARTMENT OF SOFTWARE ENGINEERING



SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

OUTLINE AND DESCRIPTION

Software Requirements Specification (**SRS**) is a document focused on what the software needs to do and how it must perform. It lays the important groundwork so that every person involved with the project understands the most crucial details.

SEPT. 2023

CHAPTER	CONTENT	PAGE
	i. Declaration Sheet	
	ii. Table of Contents	
	iii. List of Tables	
	iv. List of Figures	
	v. Definition, Abbreviation and Acronyms	
CHAPTER ONE INTRODUCTION	1.1. Document Scope	3
	1.2. Document Purpose	3
	1.3. Document Conventions	3
	1.4. Intended Audience	3
	1.5. Team Organization	3
CHAPTER TWO SYSTEM DESCRIPTION	2.1. System Overview	4
	2.2. System Scope	4
	2.3. User Classes and Characteristics	4
	2.4. Business Rules	4
	2.5. Operating Environment	5
	2.6. Design and Implementation Constraints	5
	2.7. Assumptions and Dependencies	5
CHAPTER THREE SYSTEM REQUIREMENTS	3.1. Functional Requirements	6
	3.2. Use Cases	6
	3.2.1. Use Case Diagram	
	3.2.2. Use Case Description	
	3.3. Data Requirements	6
	3.3.1. Overview	
	3.3.2. Entity, Attribute and Relationships.	
	3.3.3. Entity-Relationship Diagram	
	3.3.4. Data Validation Rules	
	3.4. External Interface Requirements (Optional, commonly used for Embedded Systems)	7
	3.4.1. Hardware Interface	
	3.4.2. Software Interface	
	3.4.3. User Interface	
	3.4.4. Communication Interface	
	3.5. Non-functional Requirements	7
	3.5.1. Performance Requirements	
	3.5.2. Usability Requirements	
	3.5.3. Security Requirements	
	3.5.4. Software Quality Attributes	
APPENDIX A	Group Log	7

CHAPTER ONE

1.1

DOCUMENT SCOPE

Describe what the SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with this section and proceeding through the sections that are most appropriate to each reader type.

CHAPTER ONE

1.2

DOCUMENT PURPOSE

Full description of the main objectives of this SRS document in the context of this project

CHAPTER ONE

1.3

DOCUMENT CONVENTION

Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.

HINT: you can refer senior project document standard.

CHAPTER ONE

1.4

DOCUMENT AUDIENCE

Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. In your case it would probably be you (as developer), the advisor (as project manager) and examiners (as testers).

CHAPTER ONE

1.5

TEAM ORGANIZATION

List group members with their respective role in the project

CHAPTER TWO

2.1

SYSTEM OVERVIEW

This section is a summary of critical information relevant to the project, such as objective, problem statement, scope, significance, and technologies, and tools will be used to build the system. Explain overall workflow of your system. Use **high level context diagram** that can describe overall major activities in the system.

CHAPTER TWO

2.2

SYSTEM SCOPE

Provide a **short** description of the system and its purpose, including major features.

CHAPTER TWO

2.3

USER CLASSES AND CHARACTERISTICS

Description of the characteristics, needs and expectations of the user who will be using and interacting with the system. This section will be subset of project stakeholders.

CHAPTER TWO

2.4

BUSINESS RULE

This section includes list of business rules that impact the system that emerge as constraints in form of policies, guidelines, standards, regulations or calculated formulas. A description of business rule and an indication of the related requirements that is impacted by the rule. **(You can use tabular form)**

CHAPTER TWO**2.5 OPERATING ENVIRONMENT**

Describe the environment the system will operate on after completion in terms of hardware and software such as operating system, software versions, database, server, and other required soft and hard infrastructures.

CHAPTER TWO**2.6 DESIGN AND IMPLEMENTATION CONSTRAINTS**

Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards.

CHAPTER TWO**2.7 ASSUMPTIONS AND DEPENDENCIES**

These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

CHAPTER THREE

3.1

FUNCTIONAL REQUIREMENTS

Functional requirements are product features that developers must implement to enable the users to achieve their goals. It outlines what the system is supposed to do and what task it should perform. Functional requirements need to be clear, simple, and unambiguous. Write the functional requirement based on features.

A. Feature Name

- a. The system shall/will/should

CHAPTER THREE

3.2

USE CASE MODEL

- Use case diagram per actor
- Use case description of major use cases

FROM AUTOMATION TO AI

CHAPTER THREE

3.3

DATA REQUIREMENTS

Describes the data that the system needs to handle, store and manipulate to perform its functions. It outlines the data that the system will consume, process and create.

- ✓ Entity, Attribute and Relationships.
- ✓ Entity-Relationship Diagram

Data Validation Rules: ensure that the data entered into the system meets specified criteria and adheres to defined business rules.

CHAPTER THREE

3.4

EXTERNAL INTERFACE REQUIREMENTS (Optional)

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, and Communications

CHAPTER THREE

3.5

NON-FUNCTIONAL REQUIREMENTS

If you think of functional requirements as those that define **what** a product or system is supposed to do, *non-functional* requirements (NFRs) define **how** the product or system should do it. (*The following things should be written by assuming you are writing for the designers and programmer*)

Performance: defines how fast a software system or a particular piece of it responds to certain users' actions under a certain workload. In most cases, this metric explains how long a user must wait before the target operation happens. *What does your system requires to keep performance (when and why?).*

Security: relates to the system's protection against unauthorized access. *What does your system requires to keep security (when and why?).*

Usability: denotes the ease of use of the system. *What does your system required to be usable (when and why?).*

Software Quality Attributes: Specify any additional quality characteristics for the product that will be important to either the customers or the developers.

GROUP LOG

Include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document. You can add your progress reports if advisor and coordinator signed on. *You can add your individual (with name and role in the project) and one group photo here.*

FOLLOWING GUIDELINE IS MANDATORY

HOWEVER, YOU WELCOME IF YOU A HAVE SUGGESTION TO BE ADDED IN SRS OUTLINE