<Logo of FPTU/ >

<Name of the project>
Software Requirement Specification

Class Code: <Code of the class>

Group Code: <Code of the group>

<Location, issued date of the Document>

RECORD OF CHANGE

*A - Added M - Modified D - Deleted

Effective Date	Changed Items	A* M, D	Change Description	New Version
02/12/2025	Initial	A	Add project overview	

SIGNATURE PAGE

ORIGINATOR: *<Name>*_____ *<Date>*_

<Position>

REVIEWERS: *<Name>*_____ *<Date>*_

<Position>

TABLE OF CONTENTS

1 INTRODUCTION 5

1.1 Purpose 5
1.2 Definitions, Acronyms 5
1.3 References 5

2 OVERALL DESCRIPTION 6

2.1 Product Perspective 6
2.2 Business Process 6
2.3 User classes and characteristics 6

3 FUNCTIONAL REQUIREMENTS 7

3.1 Use Case Diagram 7
3.2 Use Case Specifications 7
3.2.1. < Use Case Name 1> 7
3.2.2. < Use Case Name 2> 8
3.2.3. < Use Case Name 3> 8
3.3 State Diagrams 8
3.4 Data flow Diagrams 8
3.5 Logical Data Model 8

4 NON-FUNCTIONAL REQUIREMENTS 9

4.1 Usability 9
4.2 Reliability 9
4.3 Performance 10
4.4 Reusability 10
4.5 Scalability 10

5 SUPPORTING INFORMATION 11

1 INTRODUCTION

*[The introduction of the **Software Requirements Specification (SRS)** provides an overview of the entire **SRS**. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **SRS**.]*

*[Note: The **SRS** document captures the complete software requirements for the system, or a portion of the system. Following is a typical **SRS** outline for a project using only traditional, natural-language style requirements—with **no use-case modeling.** It captures all requirements in a single document, with applicable sections inserted from the Supplementary Specifications (which would no longer be needed). For a template of an **SRS** using use-case modeling, which consists of a package containing Use Cases of the use-case model and applicable Supplementary Specifications and other supporting information, see rup_srsuc.dot.]*

*[Many different arrangements of an **SRS** are possible. Refer to [IEEE830-1998] for further elaboration of these explanations, as well as other options for **SRS** organization.]*

1.1 Purpose

[Specify the purpose of this SRS. The SRS fully describes the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]

1.2 Definitions, Acronyms

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to the project's Glossary.]

1.3 References

[This subsection provides a complete list of all documents referenced elsewhere in the SRS. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

2 OVERALL DESCRIPTION

[This section of the SRS describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

2.1 Product Perspective

*[Describe the product's context and origin. Is it the next member of a growing product line, the next version of a mature system, a replacement for an existing application, or an entirely new product? Provide a **context diagram** of the system, with explanations as applicable. The context of a system refers to the connections and relationships between the system and its environment.]*

2.2 Business Process

*[Representing processes of an enterprise using **Swimlane Diagram**, so that the current business processes may be analyzed, applied securely and consistently]*

2.3 User classes

[Identify the various user classes that you anticipate will use this product and describe their pertinent characteristics.]

3 FUNCTIONAL REQUIREMENTS

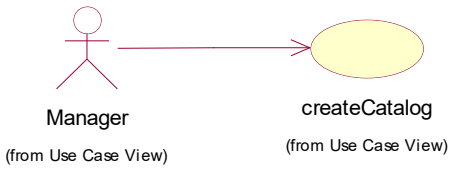
3.1 Use Case Diagram

[The main Use case Diagram of the system]

3.2 Use Case Specifications

[At least 3 Use case Specifications from your Use case Diagram]

3.2.1. < Use Case Name 1>



USE CASE-n SPECIFICATION			
Use-case No.	<UC001>	Use-case Version	<1.0>
Use-case Name	<Name>		
Author	<Members>		
Date	Dd/mm/yyyy	Priority	<High/Normal/Low>

Primary Actor:

Secondary Actor:

Description:

<Briefly describe the used case >

Triggers:

<What does lead in using this case?>

Preconditions:

<List the required pre-conditions for using this case>

Post Conditions:

<List the required post-conditions for using this case>

Main Success Scenario:

<List the main steps for using this case to reach the goal successfully >

Alternative Scenario:

<List other steps for using this case to reach the goal in some alternative conditions >

Exceptions:

<List exceptions of this use case >

Relationships:

<List the relationships that use case relates to>

Business Rules:

<Any concern about the business>

3.2.2. < Use Case Name 2>

……

3.2.3. < Use Case Name 3>

……

3.3 State Diagrams

[At least 3 State Diagrams of the system]

3.4 Data flow Diagrams

[At least 3 Data flow Diagrams – level 1 of the system]

3.5 Logical Data Model

[Provide the Logical Data Model of the System]

4 NON-FUNCTIONAL REQUIREMENTS

[This section describes the non-functional requirements of the system. Some examples are listed as below]

4.1 Usability

[This section includes all those requirements that affect usability. For example,

specify the required training time for a normal users and a power user to become productive at particular operations

specify measurable task times for typical tasks or base the new system's usability requirements on other systems that the users know and like

specify requirement to conform to common usability standards, such as IBM's CUA standards Microsoft's GUI standards]

<Usability Requirement One>

[The requirement description goes here.]

4.2 Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

Availability—specify the percentage of time available (xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.

Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?

Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system's output.

Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point(bugs/function-point).

Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a "critical" bug; for example, complete loss of data or a complete inability to use certain parts of the system's functionality.]

<Reliability Requirement One>

[The requirement description.]

4.3 Performance

<Performance Requirement One>

[The requirement description goes here.]

4.4 Reusability

< Reusability Requirement One>

[The requirement description goes here.]

4.5 Scalability

< Scalability Requirement One>

[The requirement description goes here.]

5 SUPPORTING INFORMATION

[The supporting information makes the SRS easier to use. It includes:

Index

Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the SRS should explicitly state whether or not the appendices are to be considered part of the requirements.]