<Project Name>

Requirements Specification and Analysis

<Version>

<Date>

<Your Name>

Prepared for

SOFT3102 Software Engineering Practice



Table of Contents

[1. Introduction 1](#_Toc65941015)

[1.1. Purpose of the System 1](#_Toc65941016)

[1.2. Scope of the System 1](#_Toc65941017)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc65941018)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc65941019)

[1.5. Overview 1](#_Toc65941020)

[2. Current System 1](#_Toc65941021)

[3. Proposed System 1](#_Toc65941022)

[3.1. Overview 1](#_Toc65941023)

[3.2. Functional Requirements 2](#_Toc65941024)

[3.3. Nonfunctional Requirements 2](#_Toc65941025)

[Usability 2](#_Toc65941026)

[Reliability 2](#_Toc65941027)

[Performance 2](#_Toc65941028)

[Supportability 2](#_Toc65941029)

[Implementation 2](#_Toc65941030)

[Interface 2](#_Toc65941031)

[Packaging 2](#_Toc65941032)

[Legal 2](#_Toc65941033)

[3.4. System Models 2](#_Toc65941034)

[Scenarios 2](#_Toc65941035)

[Use case model 2](#_Toc65941036)

[Dynamic model 2](#_Toc65941037)

[User interface—navigational paths and screen mock-ups 3](#_Toc65941038)

[3.5. Project Schedule 3](#_Toc65941039)

[4. Glossary 3](#_Toc65941040)

[5. References 3](#_Toc65941041)

REQUIREMENTS ANALYSIS DOCUMENT[1]

In agile project management approaches, the Requirement Analysis Document (RAD) should be developed incrementally like the product itself, after each sprint planning, that is, when the requirements are clearer and can be described in detail. The requirements are going to be updated throughout the development process as getting new feedbacks in review and refinement activities.

Please set your word processor’s language to English, enable spell checker to detect the misspellings, and follow the formatting in this document.

# Introduction

The purpose of this section is to provide a brief overview of the function of the system and the reasons for its development, its scope, and references to the development context (e.g., reference to the problem statement written by the client, references to existing systems, feasibility studies). The introduction also includes the objectives and success criteria of the project[1].

## Purpose of the System

## Scope of the System

## Objectives and Success Criteria of the Project

## Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the RAD.

## Overview

This subsection should:

* Describe what the rest of the RAD contains.
* Explain how the RAD is organized.

# Current System

If the new system will replace an existing system, this section describes the functionality and the problems of the current system. Otherwise, this section describes how the tasks supported by the new system are accomplished now.

# Proposed System

Documents the requirements elicitation and the analysis model of the new system.

## Overview

Presents a functional overview of the system.

## Functional Requirements

Describes the high-level functionality of the system. Functional requirements are the descriptions of the features which provide value and can be used by the end users.

## Nonfunctional Requirements

Describes user-level requirements that are not directly related to functionality. This includes usability, reliability, performance, supportability, implementation, interface, operational, packaging, and legal requirements.

### Usability

### Reliability

### Performance

### Supportability

### Implementation

### Interface

### Packaging

### Legal

## System Models

Describes the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.

### Scenarios

A scenario is an instance of a use case.

### Use case model

A use case is a generalization of several scenarios. Therefore, the number of scenarios must be equal to or greater than the number of use cases.

### Dynamic model

The dynamic model is depicted with sequence diagrams and with state machines. Sequence diagrams represent the interactions among a set of objects during a single use case. State machines represent the behavior of a single object (or a group of very tightly coupled objects). The dynamic model serves to assign responsibilities to individual classes and, in the process, to identify new classes, associations, and attributes to be added to the analysis object model.

When working with either the analysis object model or the dynamic model, it is essential to remember that these models **represent user-level concepts, not actual software classes or components.**

### User interface—navigational paths and screen mock-ups

## Project Schedule

Add your sprint start dates, end dates, sprint goal and the scope (list of PBIs committed) for each sprint.

# Glossary

To establish a clear terminology, developers **identify the participating objects** for each use case. Developers should **identify, name, and describe them** unambiguously and collate them into a glossary.

# References

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.