

College of Science and Technology

School of Science and Technology

# SOFT30121: Advanced Analysis and Design

# Systems Analysis Design and Implementation

*By*

N0683858: Hamid Mujtaba

T0084963: Haowei Goh

N0688119: Elliot Harding

N0756079: Steffan Walker

N0704050: Michael Abraham

N000000: Hamed

NTU Stores Management System

￼

SOFT30121: Advanced Analysis and Design i

Systems Analysis Design and Implementation i

1. Introduction 4

1.1. Purpose 4

1.2. User Characteristics 4

1.3. Assumptions 4

1.4. Scope and Constraints 4

1.5. Glossary of terms 4

1.6. Overview 4

2. Functional Requirements 5

2.1. Feature 1 5

3. Non-Functional Requirements 6

3.1. Usability requirements 6

3.2. Reliability requirements 6

3.3. Performance requirements 6

4. Interfaces 7

4.1. User interfaces 7

4.2. Hardware interfaces 7

4.3. Software interfaces 7

5. Use Case Modelling 8

5.1. Use Cases 8

5.2. Misuse cases 8

6. Project Plan: 9

References 10

7. 10

Bibliography 10

8. 10

# Introduction

Welcome to the sample template for your systems development report. You can alter this document as you like – even discard it and start one of your own. There are many examples of requirements specification templates online for you to use as well.

## Purpose

Specify the purpose of this Software Requirements Specification. The SRS fully describes the external behaviour of the application or subsystem identified. It also describes non-functional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

## User Characteristics

Create general customer profiles for each type of user who will be using the product. Profiles should include:

· Student/faculty/staff/other

· Experience

· Technical expertise

· Other general characteristics that may influence the product

## Assumptions

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

## Scope and Constraints

A brief description of the software application that the Software Requirements Specification applies to.

Also describe any items that will constrain the design options, including

· Parallel operation with an old system

· Audit functions (audit trail, log files, etc.)

· Access, management and security

· Criticality of the application

· System resource constraints (e.g., limits on disk space or other hardware limitations)

Other design constraints (e.g., design or other standards, such as programming language or framework)

## Glossary of terms

Outline any acronyms, abbreviations and definitions that will be used throughout the document

## Overview

What does the rest of this document contain/how is it organised?

# Functional Requirements

These should be described in enough detail for the designers to design a system satisfying the requirements. These should also be testable.

They should also be organised appropriately (could be by feature) and prioritized (could be by MoSCoW).

A good requirement is:

* Correct
* Unambiguous
* Complete
* Consistent
* Ranked for importance
* Verifiable
* Modifiable
* Doesn’t specify any particular design
* Traceable

## Feature 1

# Non-Functional Requirements

Describe the characteristics of the system. Explain the rationale of each of these to aid design choices later. Be as specific as possible. Not all these requirement types may apply, so only use ones appropriate the system.

## Usability requirements

## Reliability requirements

## Performance requirements

# Interfaces

Describe the logical characteristics of each interface between the software product and the users, and any external systems. This may include sample screen images (consider using balsamiq!), any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on.

## User interfaces

## Hardware interfaces

## Software interfaces

# Use Case Modelling

## Use Cases

* Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first-time reader. Include use cases and business scenarios or provide a link to a separate document (or documents). A business scenario:
* Describes a significant business need
* Identifies, documents, and ranks the problem that is driving the scenario
* Describes the business and technical environment that will resolve the problem
* States the desired objectives
* Shows the “Actors” and where they fit in the business model
* Is specific, and measurable, and uses clear metrics for success

## Misuse cases

* Provide some examples of functions that should not be possible to perform in the system. These may help you meet any security requirements you have.

# Project Plan:

This may be considered as separate from the rest of the document, as it is not concerned with the system, but how you are going to organise your team to build it.

Explain which agile methodology you will use, what tools you will use to track progress, source control etc.

Take a look at your interfaces and assign them “T-shirt sizes” (XS, S, M, L, XL).

References

## 

References here in Harvard referencing style. Use MS Word citing and referencing feature.

Bibliography

## 

Bibliography here in Harvard referencing style.