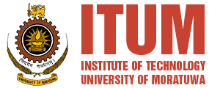
****

**Institute of Technology University of Moratuwa**

**Title of the Project**

**Software Requirements Specification**

**Version 1.0**

*Student Details:*

Your name Index Number

Your Name Index Number

Your name Index Number

Your Name Index Number

*Supervised by:*

<<Supervisor Name>>

Division of Information Technology

Institute of Technology University of Moratuwa

July 2021

# Table of Contents

Do not enter table of contents manually. Use **“Table of Contents”** option in MS Word.

# Introduction

## 1.1 Purpose

Write the purpose of presenting SRS document.

Eg: Reasons of writing the SRS document

## 1.2 Product Scope

### 1.2.1 Aim and Objectives

The primary focus of your research project is usually expressed in terms of aims and objectives.

**Aim** is what you hope to achieve. **Aims** are statements of intent. They are usually written in broad terms. They set out what you hope to achieve at the end of the project.

**Objective** is the action(s) you will take in order to achieve the aim. **Objectives**, on the other hand, should be specific statements that define measurable outcomes, e.g. what steps will be taken to achieve the desired outcome.

When writing your objectives try to use strong positive statements.

Here, you can present the changes according to the comments given by the examiners.

### Project Boundary

Cleary define the boundaries of the project.

# 2 Overall Description

## 2.1 Product Perspective

Draw a simple high level architectural diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful. (*Appendix A)*

Provide a small description of the diagram.

## 2.2 User Classes and Characteristics

Identify the various users and their characteristics. (*Appendix B)*

## 2.3 Operating Environment

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

## 2.4 Design and Implementation Constraints

Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; 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 (for example, if the customer’s organization will be responsible for maintaining the delivered software).

## 2.5 Assumptions and Dependencies

List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. 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, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).

# 

# 3 External Interface Requirements

## 3.1 User Interfaces

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, 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. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.

Eg: mockups of the user interfaces: login page, etc. (Draw at-least 3 basic user interfaces)

## 3.2 Hardware Interfaces

Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.

Eg: RFID reader

## 3.3 Software Interfaces

Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

Eg: Google Map API, Payment gateway API, Shopping cart libraries, Server operating system

## 3.4 Communications Interfaces

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

Eg: HTTP, SMTP, FTP (Name the protocol and reasons of using it)

# 4 System Designs

This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.

## 4.1 Use case Diagram

A diagram of a company structure

Description automatically generated

### 4.1.1 Use case Description

Include use case scenarios or narratives which describe how users will perform functions in the system. Here each use case is represented as a sequence of step identified with pre-conditions and post-conditions. (*Refer the Appendix C template)*

Eg: Refer the Appendix D

# 5 System Features

## 5.1 System Feature 1

Don’t really say “System Feature 1.” State the feature name in just a few words.

Eg: Refer the Appendix E

### 5.1.1 Description and Priority

Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).

### 5.1.2 Stimulus/Response Sequences

List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.

### 5.1.3 Functional Requirements

Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## 5.2 System Feature 2

# 

# 6 Other Nonfunctional Requirements

## 6.1 Performance Requirements

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

## 6.2 Safety Requirements

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.

## 6.3 Security Requirements

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

## 6.4 Software Quality Attributes

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

# 7 References

List down all the sources (including web sites, journals, articles, books) you've used in your project, so readers can easily find what you've cited.

Follow IEEE guidelines when including references in your research paper

<https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf>

# Appendix

Append any other figures, tables or documents related to your document. Properly mention each and every item that you are appending.

Each appendix should deal with a separate topic

* Each appendix must be referred to by name (Appendix A, Appendix B, Appendix C, etc.)
* If there is only one appendix, it is just called Appendix
* Each appendix must also have a title
* Begin each appendix on a separate page with page number

**Appendix A**

**Diagram

Description automatically generated**

Figure 2: Example for the High -Level Diagram

**Appendix B**

**Text

Description automatically generated**

Figure 3: Example for the user and their characteristics

**Appendix C**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | (prefix your Use Case IDs with UC, to distinguish them from other models) | |
| **Use Case** |  | |
| **Priority** | (High, Medium or Low) | |
| **Primary System Actor** |  | |
| **Other participant Actors** |  | |
| **Description** | (brief) | |
| **Pre-conditions** |  | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| 4. If xxx 5. Else 6. etc. |  |
| **Post-conditions** |  | |
| **Alternative Flows** | (briefly describe alternative flows here for base Use Cases; extend this into complete flow of events for Elaborated Use Cases, either here or in the next template) | |
| **Assumptions** |  | |

**Appendix D**

Table

Description automatically generated

Figure 4: Example for the use case narratives

**Appendix E**

**Text, letter

Description automatically generated**

Figure 5: Example for the system feature