Software Requirements Specification

# TeamName

Student – 1

Student – 2

Student – 3

Student – 4

Date of Submission

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Section/Sub-section Title** | **Author**  **[person with primary responsibility for this section]** | **Page #** |
| **1. Introduction**  **1.1 Document Purpose**  **1.2 Intended Audience**  **1.3 Project Scope** |  |  |
| **2. Overall Description**  **2.1 Product Context**  **2.2 Product Features**  **2.3 Use Classes and Characteristics**  **2.4 Operating Environment**  **2.5 Design and Implementation Constraints**  **2.6 User Documentation**  **2.7 Assumptions and Dependencies** |  |  |
| **3. Use Case Descriptions**  **3.1 First Use Case**  **3.2 Next Use Case (and so on)** |  |  |
| **4. External Interface Requirements**  **4.1 User Interfaces**  **4.2 Hardware Interfaces**  **4.3 Software Interfaces**  **4.4 Communications Interfaces** |  |  |
| **5. Nonfunctional Requirements**  **5.1 Performance Requirements**  **5.2 Safety Requirements**  **5.3 Security Requirements**  **5.4 Software Quality Attributes** |  |  |
| **6. Other Requirements** |  |  |
| **Appendix A: Requirements Traceability Matrix** |  |  |
| **Appendix B: Analysis Models** |  |  |
| **Appendix C: Issues List** |  |  |

### General Instructions

* This document must contain all the following sections and subsections.
* This is a generic template and some sections will not apply to this course project. When this is the case, simply comment why a section is not applicable to this project.
* Use bullets, figures, & tables as much as possible to make the presentation concise and readable.
* All diagrams, tables, & figures need to be numbered by section [i.e. Table 4.1] and have either a title or caption. Be consistent in the use of a title or caption.
* Ensure your terms are used consistently throughout the document. For example, don’t use task leader in one section and have the same role called team leader in another section.
* Do not use ambiguous terms such as user. Specify a particular role.

### Remove ALL

### highlighted instructions and

### instructions delimited by < > from your final deliverable.

# Introduction

## Document Purpose

*<Identify the purpose of this SRS document [****not*** *the project] and explain what it contains that will allow this document to serve as a contractual instrument between the software engineers and the client.>*

## Intended Audience

*<Organized* ***in bullets format*** *– List & describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers.>*

## Project Scope

*<Provide a short description of the software being specified including relevant benefits, objectives, and goals.* [*Goals-vs-objectives*](https://dl.dropboxusercontent.com/u/60888300/CurrentCourses-Public/Common-Deliverables/ResearchPaper/Goals-vs-objectives.pdf) *>*

* + 1. **Goals**
    2. **Objectives**
    3. **Benefits**

# Overall Description

## Product Context

*<Describe the context and origin of the product being specified in this SRS. For example,*

1. *state whether this product is a follow-on member of a product family,*
2. *a replacement for certain existing systems, or*
3. *a new, self-contained product.*

*In this section provide a* ***simple context diagram*** *with an explanation of the diagram that shows*

1. *the major components of the overall system,*
2. *subsystem interconnections, and*
3. *external interfaces.>*

## Product Features

*<Using* ***bullets*** *- List & summarize the* ***major*** *features the product contains or the* ***significant*** *functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS.>*

## Use Classes and Characteristics

*<Identify the various* ***use classes*** *that you anticipate will use this product. Use classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each use class. Distinguish the favored use classes from those who are less important to satisfy.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Roles | ... | Roles |
| Frequency of Use |  |  |  |
| Subset of Product Functions |  |  |  |
| Technical Expertise |  |  |  |
| Security or Privilege Levels |  |  |  |
| Experience |  |  |  |

*In this section include a* [***conceptual***](http://www.agilemodeling.com/artifacts/classDiagram.htm)class diagram(s). This diagram only shows that a class exists,

* 1. lists its’ **attributes**, and
  2. lists the **principal responsibilities** of the class. [At the conceptual level you do not identify specific methods to specify the interface of a class, only list responsibilities.]  
       
     More detail [such as operators] will be added in the SDS version of the class diagram*>*

## Operating Environment

*<Describe the environment in which the software will operate, including:*

1. *the hardware platform,*
2. *communications,*
3. *operating system and versions, and*
4. *any other software components or applications with which it must peacefully coexist.>*

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers.   
  
These will include, but not limited to:*

1. *corporate or regulatory policies;*
2. *hardware limitations (timing requirements, memory requirements);*
3. *interfaces to other applications;*
4. *specific technologies, tools, and databases to be used;*
5. *parallel operations;*
6. *language requirements;*
7. *communications protocols;*
8. *security considerations;*
9. *design conventions or programming standards).*

*If a n item is not applicable to the project, simply explain why not.  
If there is an issue(s) not listed, add items as necessary.>*

## User Documentation

*< List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## Assumptions and Dependencies

*< Using* ***bullets*** *- list any* ***assumed*** *factors (as opposed to* ***known*** *facts) that could affect the requirements stated in the SRS. These could include:*

1. *third-party or commercial components that you plan to use,*
2. *issues around the development or operating environment, or constraints.*

*The project could be affected if these assumptions are incorrect, are not shared, or change.  
Add as many items as necessary.>*

# Use Case Descriptions

*<Introduce this section don’t just start with the table.*

1. *In a* ***Table*** *[Roles in columns and actions/use cases in rows] detail the specific Use Cases that will be described in this section [This table will help to illuminate overlap and help identify a good inheritance structure] and also*
2. *explain the component ratings methodology [priority, benefit, cost, risk] that will be used  
   What are you measuring [i.e. Benefit is How valuable is the use case to the client*.*>*

**Be very careful to describe what needs to be in the software, not how you are going to implement the needs. This is requirements analysis, not design.**

**No specific type of interface is prescribed for the prototype. Any decisions about such tools as buttons, drop-down boxes, etc. are purely a matter of design and are not to be considered or listed during the requirements analysis phase.**

## Title of Use Case

*<This template illustrates organizing the* ***functional requirements*** *for the product [the major services provided by the product] by use cases.>*

|  |  |
| --- | --- |
| **Title of Use Case** | **Description and Priority**  <Provide a   * short **description** of the feature, * identify which **use class(es)** will participate in this use case, and * **indicate** whether it is of High, Medium, or Low priority and include specific priority **ratings** [benefit, penalty, cost, and risk] with **explanation** [why does this case have these ratings]. Rate each category on a relative scale from a low of 1 to a high of 9.> |
| Precondition(s) | * <Describe the state the system must be in before this use case can begin.> |
| Main Flow of Event(s) | * <Describe step-by-step the events that comprise this use case.> * Next event (and so on) |
| Postcondition(s) | * <Describe the state the system will be in after this use case terminates.> |
| Exceptional Flow of Event(s) | * <Describe what might go wrong.> * Next event (and so on) |

## [Next Use Case and so on]

|  |  |
| --- | --- |
| **Title of Use Case** |  |
| Precondition(s) |  |
| Main Flow of Event(s) |  |
| Postcondition(s) |  |
| Exceptional Flow of Event(s) |  |

# External Interface Requirements

## User Interfaces

1. *<Describe the logical characteristics of each interface between the software product and the users.*
2. *Include User Interface* ***mockup screen images****, which will reflect any GUI standards or product family style guides [if they* ***exist****] 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.>*

## 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:*

1. *configuration characteristics,*
2. *the supported device types,*
3. *the nature of the data and control interactions between the software and the hardware, and*
4. *communication protocols to be used6.>*

## Software Interfaces

1. *<Describe the connections between this product and other specific software components (name and version), including*

* *databases,*
* *operating systems,*
* *tools,*
* *libraries, and*
* *integrated commercial components.*

1. *Identify the data items or messages coming into the system and going out and describe the purpose of each.*
2. *Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols.*
3. *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.>*

## Communications Interfaces

1. *<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.*

1. *Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP.*
2. *Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# Nonfunctional Requirements

## 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.>*

## Safety Requirements

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

## Security Requirements

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

## 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.>*

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include:*

1. *database requirements,*
2. *internationalization requirements,*
3. *legal requirements,*
4. *reuse objectives for the project, and so on.*

*Add any new sections that are pertinent to the project.>*

**Appendix A: Requirements Traceability Matrix**

*<Develop a table that lists and identifies by* ***unique number*** *all the system requirements documented in the SRS. Summarize the requirement description. In this table indicate the* ***section*** *of the SRS where the requirement is stated and explained. This matrix must be detailed and complete, as it will also be used in later phases to ensure all requirements appear as design elements documented in the System Design Specification (SDS) and are covered by explicit test cases in the Testing Document.>*

Store this matrix table in a separate.gdoc file. ***Do not embed*** the matrix table in this document, but rather ***link*** to the separate document containing the matrix table.

* **Create a Trello card for each Priority I requirement in the SRS Requirements Traceability Matrix.  
    
  If some requirements can be easily grouped [and many will be], they can alternatively be made into Checklists & items on a single card rather than every requirement being an individual card.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Priority**  **Required=1**  **Desired=2**  **Optional=3** | **Requirement # by Category**  **[Categories defined by project team]** | **Description** | **SRS Section** |
| [ i.e. 1.2.3.4] | [ i.e. 1.2.3.4] | Taken from SRS |  |
| [Category-No Priority] | 4.0.0 | **Secure Data Transmission** |  |
| [SubCategory-No Priority] | 4.1.0 | ***Secure File Transmission*** |  |
| 2 | 4.1.1 | DNS requests must be forwarded through the proxy. |  |

**Appendix B: Analysis Models**

*<Place your use case diagram(s), activity diagram(s), and state diagram(s) in this section. >*

*See Course UML Text.*

**Appendix C: Issues List**

*< This is a dynamic list of the open requirements issues that remain to be resolved, including TBD’s, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>*

**Appendix D: Definitions, Acronyms, and Abbreviations**

*<Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret this document.>*

Grading Criteria

* *Is the Deliverable organized & structured as required –* ***YES/NO***
* *Was the Deliverable Template used and used correctly --* ***YES/NO***
* *Was the Table of Contents compete* ***– YES/NO***
* *Were all sections present and adequate –* ***YES/NO –***



*-- Check in the text for specific instructor comments on content --*



***Please DO NOT delete this grading criteria.***