**Requirements Specification Document**

**<Project Name>**

**<Designer Organization>**

**<date created>**

**Contents**

Revision History 1

1 Introduction 2

1.1 Purpose 2

1.2 Project Scope 2

1.3 Glossary of Terms 2

1.4 References 2

1.5 Overview 2

2 Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 Assumptions and dependencies 3

3 System Features 3

3.1 System Feature 1 3

3.2 System Feature 2 (and so on) 4

4 External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 4

5 Other Non Functional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

6 Other Requirements 5

7 Analysis Models 5

Appendix: Issues List 5

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Changes | Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 1 Introduction

### 1.1 Purpose

<*Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this RS, particularly if this RS describes only part of the system or a single subsystem.*>

### 1.2 Project Scope

<*Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies.*>

### 1.3 Glossary of Terms

<*Define all the terms necessary to properly interpret the RS, including acronyms and abbreviations.*>

### 1.4 References

<*List any other documents or Web addresses to which this RS refers. These may include user interface style guides, contracts, standards, system requirements specifications or use case documents. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.*>

### 1.5 Overview

<*Provide an overview of the sections and contents of the document*>

## 2 Overall Description

### 2.1 Product Perspective

*<Describe the context and origin of the product being specified in this RS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the RS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful but not required.>*

### 2.2 Product Features

<*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 RS.*>

### 2.3 User Classes and Characteristics

<*Identify the various user classes that you anticipate will use this product. User 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 characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.*>

### 2.4 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.*>

### 2.5 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).*>

### 2.6 Assumptions and dependencies

<*List any assumed factors (as opposed to known facts) that could affect the requirements stated in the RS. 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.*>

## 3 System Features

*<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.>*

### 3.1 System Feature 1

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

#### 3.1.1 Description and Priority

<*Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority.*>

#### 3.1.2 Functional Requirements

<*Where applicable* - *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 related use case(s). 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. These could be requirements that the clients provided directly or were defined by the designer group as a result of rendering the feature.>*

*<Each requirement should also include information a(1) bout Backward Traceability (the rationale for the requirements and the source – RFP and which section in it, client meeting and which notes from that meeting, etc.. and (2) Forward Traceability (how the requirement can be verified by the users)>*

REQ-1:

REQ-2:

#### 3.1.3 Use Case(s) associated with the Feature or Functional Requirements

<*This is the use case specification. For each Use Case*, *list the dialog elements in the use case that elaborates or is related to this feature or one of its functional requirements, i.e. sequences of user actions and system responses that stimulate the behavior defined for this feature/functional requirement*>

### 3.2 System Feature 2 (and so on)

## 4 External Interface Requirements

### 4.1 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.>*

### 4.2 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.>*

### 4.3 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. Identify data that will be shared across software components.>*

### 4.4 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.>*

## 5 Other Non Functional Requirements

### 5.1 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.>*

### 5.2 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.>*

### 5.3 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.>*

### 5.4 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.>*

## 6 Other Requirements

*<Define any other requirements not covered elsewhere in the RS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

## 7 Analysis Models (Use Case Model, Entity relationship diagrams, Data Flow Diagrams, Glossary)

*<For each analysis model include a short description explaining its elements and relationships, and how they might relate to other artifacts (requirements, use cases) or models described in other parts of the document>*

## Appendix: Issues List

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