ChessEDU

Version <1.0>

[Note: The following template is provided for use with the Unified Process for EDUcation. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
| 25/10/2022 | 1.0 | First Draft | Adair Torres |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Assumptions and Dependencies 4

3. Usability 4

3.1 <Usability Requirement One> 5

4. Reliability 5

4.1 <Reliability Requirement One> 5

5. Performance 5

5.1 <Performance Requirement One> 5

6. Supportability 5

6.1 <Supportability Requirement One> 6

7. Design Constraints 6

7.1 <Design Constraint One> 6

8. Security 6

9. Online User Documentation and Help System Requirements 6

10. Interfaces 6

10.1 User Interfaces 6

10.2 Hardware Interfaces 6

10.3 Software Interfaces 6

10.4 Communications Interfaces 6

11. Applicable Standards 7

# Introduction

[The introduction of the **Supplementary Specifications** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this **Supplementary Specifications**.

The **Supplementary Specifications** captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

* Legal and regulatory requirements, including application standards.
* Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.
* Other requirements such as operating systems and environments, compatibility requirements, and design constraints.]

## Purpose

[Specify the purpose of these Supplementary Specifications**.**]

## Scope

[A brief description of the scope of this **Supplementary Specifications**; what Project(s) it is associated with and anything else that is affected or influenced by this document.]

## Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Supplementary Specifications**. This information may be provided by reference to the project’s Glossary.]

## References

[This subsection provides a complete list of all documents referenced elsewhere in the **Supplementary Specifications**. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Overview

[This subsection describes what the rest of the **Supplementary Specifications** contains and explains how the document is organized.]

# Assumptions and Dependencies

[This section describes any key technical feasibility, subsystem or component availability, or other project related assumptions on which the viability of the software described by this **Supplementary Specifications** may be based.]

# Usability

[This section should include all of those requirements that affect usability. Examples follow:

* specify the required training time for a normal users and power users to become productive at particular operations
* specify measurable task times for typical tasks, or
* specify requirements to conform to common usability standards, for example, IBM’s CUA standards or Microsoft’s GUI standards]

## <Usability Requirement One>

The requirement description.

# Reliability

[Requirements for reliability of the system should be specified here. Suggestions are as follows:

* Availability – specify percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and the like.
* Mean Time Between Failures (MTBF) – this is usually specified in hours but it could also be specified in terms of days, months or years.
* Mean Time To Repair (MTTR) – how long is the system allowed to be out of operation after it has failed?
* Accuracy – specify precision (resolution) and accuracy (by some known standard) that is required in the systems output.
* Maximum bugs or defect rate – usually expressed in terms of bugs/KLOC (thousands of lines of code), or bugs/function-point.
* Bugs or defect rate – categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or complete inability to use certain parts of the functionality of the system.]

## <Reliability Requirement One>

[The requirement description.]

# Performance

[The performance characteristics of the system should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

* Response time for a transaction(average, maximum)
* Throughput (for example, transactions per second)
* Capacity (for example, the number of customers or transactions the system can accommodate)
* Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
* Resource use: memory, disk, communications, and so forth]

## <Performance Requirement One>

[The requirement description.]

# Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

## <Supportability Requirement One>

[The requirement description.]

# Design Constraints

[This section needs to indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

## <Design Constraint One>

[The requirement description.]

# Security

[Identify data to be protected and the type of treats to which each type of data is exposed (treats to physical security; people who may be the sources of these threats; special or unusual security requirements especially with respect to: access the system, encryption of data, auditability). Finally list the objects which require protection by logical or physical security]

# Online User Documentation and Help System Requirements

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, and so on.]

# Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and so forth, so that the software can be developed and verified against the interface requirements.]

## User Interfaces

[Briefly describe the user interfaces that are to be implemented by the software and refer to the user interface prototype document]

## Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

## Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS, but with which this software application must interact.]

## Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so on.]

# Applicable Standards

[This section describes by reference any applicable standards and the specific sections of any such standards that apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]