|  |
| --- |
| **Disclaimer**  This is a **template** for the Software Requirements Specification (SRS) that students may use. It povides a **starting point** for the preparation of SRS.  **Note to authors**  If you add any new sections to the document please make sure that you maintain the header and text styles.  Before submission of the first draft of this document please make sure to update the Table of Contents and to delete this page.  **Author**: Dr. C. Constantinides <cc@cse.concordia.ca> |

**Software Requirements Specification**

Version 1.0

for

LOTUS Calendar

Prepared by

|  |  |  |
| --- | --- | --- |
| Alexander Rosser | 27543069 | arosser95@gmail.com |
| Costa Papadakos | 26665691 | cotsop@gmail.com |

|  |  |
| --- | --- |
| Instructor: | Dr. C. Constantinides |
| Course: | SOEN 343 |
| Date: | 25/11/2016 |

**Document history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/11/2016 | 1.0 | Rough Draft | 1. Rosser |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of contents**

[1. Introduction](#_gjdgxs)

[Purpose](#_30j0zll)

[Scope](#_1fob9te)

[Definitions, acronyms, and abbreviations](#_3znysh7)

[References](#_2et92p0)

[2. Overall description](#_tyjcwt)

[Product perspective](#_3dy6vkm)

[Product functions](#_1t3h5sf)

[User characteristics](#_4d34og8)

[Constraints](#_2s8eyo1)

[Assumptions and dependencies](#_17dp8vu)

[3. Specific requirements](#_3rdcrjn)

[External interfaces](#_26in1rg)

[Functionality](#_lnxbz9)

[Actor goal list](#_35nkun2)

[Use case view](#_1ksv4uv)

[Reliability](#_z337ya)

[Usability](#_3j2qqm3)

[Efficiency](#_1y810tw)

[Maintainability](#_4i7ojhp)

[Portability](#_2xcytpi)

[Design constraints](#_1ci93xb)

[(On-line) user documentation and help](#_3whwml4)

[Purchased components](#_2bn6wsx)

[Licensing requirements](#_qsh70q)

[Legal, copyright and other notices](#_3as4poj)

[3. Analysis Models](#_1pxezwc)

**List of figures**

# Introduction

The introduction of the Software Requirements Specifications Document provides an overview of the entire document.

## Purpose

This section defines the role or purpose of the Software Requirements Specifications Document and briefly describes the structure of the document. Identify the intended audience for the document is identified, with an indication of how they are expected to use the document.

## Scope

A brief description of what the Software Requirements Specifications Document applies to; what is affected or influenced by this document.

## Definitions, acronyms, and abbreviations

Provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Software Requirements Specifications Document. This information may be provided by reference to the project’s Glossary.

## References

Provide a list of all documents referenced in the SRS.

# Overall description

## Product perspective

Is the product self-contained? If not, then put the product into perspective with other related products. Use a block diagram to show the big picture.

## Product functions

The product allows users to view the current conference room reservations. It also allows them to create new reservations for available time slots, as well as modify and cancel existing reservations. The system permits users to add themselves to a waitlist if the desired room’s time slot is already reserved. If a time slot is freed, the system will allocate the time period to the user at the top of the wait list. It shall then remove the user from all other wait lists with the same time slot.

## User characteristics

The product is intended to be used by college faculty members. It is expected that the educational level of the average user will be above-average. Their technical expertise is expected to vary greatly and it cannot be assumed that each will have average to above average technical experience. Each user on the system will have the same security, privilege, and accessibility levels.

## Constraints

The software is required to allow a multitude of students to view the room reservations simultaneously. The system also has the constraint of providing safety for all write functions. Only one student at a time may access a room to create, modify or cancel a reservation, to allow for mutual exclusion. Furthermore, the product must provide liveness and fairness. The last constraint placed on the software is that it must place a maximum on the number of active reservations per user.

## Assumptions and dependencies

The product is will authorize access to faculty members registered to the college. To comply, the software depends on access to the school’s existing database to authorized personnel access. It is assumed that the product has no requirement of creating or modifying user profiles to the database as these functions will be handled by the school’s existing systems.

Are there any hardware or OS assumptions??

What assumptions are there? For example, a specific operating system should be present on a given hardware platform. If not, this document would have to be changed.

# Specific requirements

This section contains all requirements in detail: Functional as well as non-functional requirements (quality attributes and constraints). The quality attributes are listed according to the *ISO/IEC 25010* standard that classifies software quality in a structured set of characteristics and sub-characteristics.

## External interfaces

A detailed description of all inputs into the system and all outputs from it (in terms of content and form).

## Functionality

Functional requirements capture the intended behaviour of the system. This section contains the *Actor Goal List* and the *Use Case view*.

## Actor goal list

|  |  |
| --- | --- |
| Actor | Goal |
|  |  |

## Use case view

The use case model is shown in Figure 1.

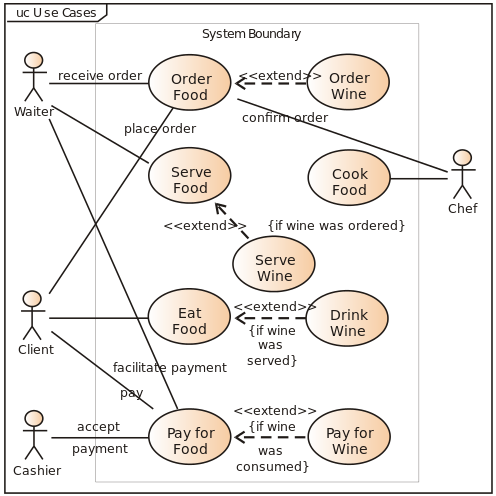


Figure 1. Use case model.

## Reliability

Description goes here.

## Usability

Description goes here.

## Efficiency

Description goes here.

## Maintainability

Description goes here.

## Portability

Description goes here.

## Design constraints

Decisions that must be followed, such as languages, processes, prescribed use of tools, architectural and design constraints, purchased components, class libraries, etc.

## (On-line) user documentation and help

Description.

## Purchased components

Description.

## Licensing requirements

Description.

## Legal, copyright and other notices

Description.

# Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

Illustrate (system) ***UML sequence diagrams*** (one for each critical scenario), identify system operations and describe operation contracts, one per critical system operation. You may also use ***UML state diagrams*** to describe critical use cases. Additionally, create a **domain model** for the system. Make sure that each model is traceable to the requirements.