|  |
| --- |
| **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  Philippe Kuret  Adriel Fabella | 26665691  27392680  27466005 | cotsop@gmail.com  philippekuret@gmail.com  adriel.fab@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,   P. Kuret,  C. Papadakos |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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)

4[. Analysis Models](#_1pxezwc)

|  |  |  |
| --- | --- | --- |
| LOTUS Calendar | Version: 1.0 |  |
| Software Requirements Specification | Date: 12/11/2016 |  |

**List of figures**

[Figure 2. Use case model.](#_2jxsxqh)

# Introduction

This introduction provides a description and an overview of the Software Requirements Specifications of our project.

## Purpose

The purpose of this document is to provide a report on the requirements for the “Lotus” room reservation system. It will provide an overall description on the development of system by providing and explaining the system’s functions, constraints, assumptions, dependencies and user characteristics. It will also provide specific requirements for development. This document is intended for approval by the stakeholder and for the development team.

## Scope

The “Lotus” room reservation system is an online reservation system that allow users to reserve timeslots on a college facility room. Multiple options like create, remove and modify reservation are offered to the user. These options can be performed for each room that are saved in the directory of rooms. The user can also view all the reservations made on all rooms to look at the availabilities. All system information like the users and the rooms are maintained in a database, which is located on a web-server

## Definitions, acronyms, and abbreviations

Table 1. Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Someone who interacts with the system |

## References

**[1]**<http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf>

# Overall description

## Product perspective

The product is a new software tool and not a continuation or adaptation of an existing tool. The product is self-contained and does not rely on pre-existing products to function with the sole exception that it uses a client's pre-existing database to authenticate its user access.

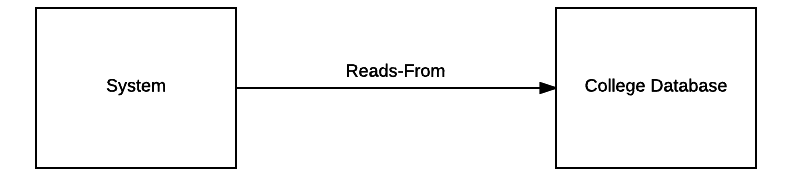


Figure 1: Product Perspective Block Diagram

## 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

(not certain, ask professor)

## 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 |
| User | To log in |
| To log out |
| To view all bookings of a specific date |
| To change and view the bookings of a different date |
| To make a reservation |
| To cancel a reservation made by the user |
| To modify a reservation made by the user. |

## Use case view

The use case model is shown in Figure 1.

Use Case View V2.png

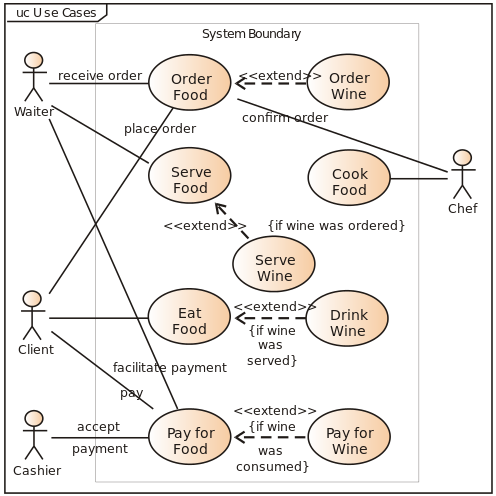


Figure 2. Use case model.

## Reliability (Costa)

The System shall always be available for use(24/7).

The expected behaviour of the System shall not be offline for no longer than 1 hour, upon system failure.

The System shall accommodate 100 simultaneous Users.

## Usability

The System shall be understandable without any explanations required.

The User shall need no longer than 5 minutes to become familiar with the System.

The User shall not need any training in order to operate the system effectively.

## Efficiency

The System shall take no longer than 3 seconds to perform any operation.

The System shall use no more than 70% resources when operating below the maximum user base.

## Maintainability

The components of the System shall not be highly coupled.

The System shall be reusable for the implementation of any reservation or waitlist systems..

Impacts or defects initially present or possibly caused due to recent modification of the system shall be easily assessed and analyzable.

The System shall be relatively modifiable and be able to be modified for a variety of different reservation systems and needs.

The System shall be easily testable for all critical scenarios.

## Portability

The system shall be highly adaptable.

The system shall be easily installable across multiple environments without much modification or support needed.

The system shall be able to replace most basic reservation systems based on a time and location.

## Design constraints (Adriel and Chen)

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

How to setup the system.

## Purchased components

Description.

## Licensing requirements (Saif and Alex)

Description.

## Legal, copyright and other notices(Saif and Alex)

Description.

# 4. 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.