Software Requirements Specification

for

<Project>

**Version <X.X>**

**Prepared by**

**Group Name: Software Architecture Class**

**s**

**CONTENTS II**

**REVISIONS II**

**1** **INTRODUCTION 1**

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

**2** **OVERALL DESCRIPTION 2**

2.1 Product Overview 2

2.2 Product Functionality 3

2.3 Design and Implementation Constraints 3

2.4 Assumptions and Dependencies 3

**3** **SPECIFIC REQUIREMENTS 4**

3.1 External Interface Requirements 4

3.2 Functional Requirements 4

3.3 Use Case Model 5

**4** **OTHER NON-FUNCTIONAL REQUIREMENTS 6**

4.1 Performance Requirements 6

4.2 Safety and Security Requirements 6

4.3 Software Quality Attributes 6

**5** **OTHER REQUIREMENTS 7**

**APPENDIX A – DATA DICTIONARY 8**

**APPENDIX B - GROUP LOG 9**

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# Introduction

*<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>*

## Document 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 SRS, particularly if this SRS describes only part of the system or a single subsystem.*

*TO DO: Write 1-2 paragraphs describing the purpose of this document as explained above.>*

## Product Scope

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.*

*TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>*

## Intended Audience and Document Overview

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the “client” and the professor). Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## Definitions, Acronyms and Abbreviations

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.*

*TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>*

## Document Conventions

*<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.*

*TO DO:* *Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance.* *Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>*

## References and Acknowledgments

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. >*

# Overall Description

## Product Overview

*<Describe the context and origin of the product being specified in this SRS. 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 SRS 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. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.*

*TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level.>*

## Product Functionality

1. *The system shall generate a file with relevant information about the amount of people that will be on campus on a given day and time, that is, generate a weekly schedule with classes and attendants by day*
2. *The system shall automatically arrange attending students to a lab considering its capacity, but it shall be open to overwriting by the Department Director*
3. *The system shall implement an algorithm to divide students in subgroups. (Odd and even ids was the initial proposal but it is not statistically reliable). The student can play different roles depending on their group (maybe in one of them they are required to be "odd" and in others they are required to be "even").*

## Design and Implementation Constraints

1. The system should provide interfaces to read and write CSV files from diverse systems.
2. A file for a given day must be generated at least with a week of time before it's usage. This file cannot be modified in the last week before it's use.
3. A Life activity has a duration of 1 hour and need 1 hour before and 1 hour after. Essentially it takes 3 hours for a Life activity.
4. Any Life area must remain empty 1 hour before and after its usage.
5. A student can only be in at most 2 Life activities, this can change in the future.

## Assumptions and Dependencies

1. There are only few courses selected to be given on campus. Only the teachers in charge of those courses must be able to make a reservation (or to propose sutdents).
2. The students can change their decision about going to campus but they have to be isolated during 10 days before attending to the in person classes.
3. Each laboratory has a different capacity.
4. The system will store only the lowest capacity of a space. A space capacity can be restricted based on its size or on the government indications.
5. It is not possible to connect to the existent systems, but it is possible to extract relevant files from them.
6. The system should consider that students can only have 1 class on campus per day and a maximum of 3 classes per week.
7. The system is pretended to be used during the orange and yellow semaphore. But it cannot be discarded the possibility to use it during the green semaphore.
8. A person is in charge of designing the students' schedules in order to avoid conflicts. It is not task of the system to do it.
9. A student needs to confirm their asistance 10 days in advance

# Specific Requirements

## *External Interface Requirements*

### User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. For your project, you only need to be concerned with the main thermostat (not the mobile app) and can use the graphic from the project description as the basis for your user interface..*

*TO DO: Provide the graphic for the thermostat user interface and provide a basic description as to how users will interact (e.g. tough screen, menus, etc.).>*

### 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. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.*

*TO DO: Please provide a short description of the different hardware interfaces. This can simply be a list of the devices you must interact with at this point. For sensors, you can assume that each has a basic “read” interface to get the current values. Each sensor in this system uses English units.>*

### Software Interfaces

*<Describe the connections between this product and other specific software components (in your case, just the mobile app that can send commands).>*

## Functional Requirements

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions.*

### *The system shall generate a file with relevant information about the amount of people that will be on campus on a given day and time, that is, generate a weekly schedule with classes and attendants by day, automatically arrange attending students to a lab considering its capacity, but it shall be open to overwriting by the Department Director*

### The system shall allow modifications on the final results. There will be circumstances that will require manual adjustments of the teacher or admin.

### Users must be able to create and edit classes and spaces with its corresponding capacity

### The system shall inform for each week to the student their schedule

### The system shall be able to change capacity of a lab due to semaphore and changing laws.

### The system will be able to get from a CSV file the Classrooms and laboratories list with the maximum number of supported people per space

### The system shall fetch from SiDI classes and their students. Import for CSV file

### The system will be able to read from CSV the student preference (per course basis) in CSV files from a Google Forms or a Professor

### Student assignation can be overwritten by directors depending on class mechanics

### The system will be able to get extra student data such as phone and birth date

### The system must have user Authentication for admin and  users

### The system shall warn the teacher and the Department Director when the capacities are being surpassed

### The system must be able to handle different campus and terms (semesters)

### The teacher shall be able to see their classes, their labs and the attending students for each week. Alternatively, this information can be generated by the system, and sent manually to each professor.

### The system shall allow modifications on the final results. There will be circumstances that will require manual adjustments of the teacher or admin.

### Additionally, the system must be able other activities on campus such as LiFE.

### System shall be multi campus, multi term, a user should select only one campus and one term when signing in.

…

## Use Case Model

*TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.*

### Use Case #1 (use case name and unique identifier – e.g. U1)

*TO DO: Provide a specification for each use case diagram*

**Author –** Identify team member who wrote this use case

**Purpose** - What is the basic objective of the use-case. What is it trying to achieve?

**Requirements Traceability –** Identify all requirements traced to this use case

**Priority** - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is depolyed

**Preconditions** - Any condition that must be satisfied before the use case begins

**Post conditions** - The conditions that will be satisfied after the use case successfully completes

**Actors** – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

**Extends –** If this is an extension use case, identify which use case(s) it extends

**Flow of Events**

* 1. Basic Flow - flow of events normally executed in the use-case
  2. Alternative Flow - a secondary flow of events due to infrequent conditions
  3. Exceptions - Exceptions that may happen during the execution of the use case

**Includes** (other use case IDs)

**Notes/Issues** - Any relevant notes or issues that need to be resolved

### Use Case #2

…

# Other Non-functional 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.*

*TODO: Provide performance requirements based on the information you collected from the client/professor. For example, you can say “P1. The secondary heater will be engaged if the desired temperature is not reached within 10 seconds”>*

## Safety and Security 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. 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.*

*TODO:*

* *Provide safety/security requirements based on your interview with the client - again you may need to be somewhat creative here. At the least, you should have some security for the mobile connection.*

## 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.*

*TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Adaptability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change (e.g. adapting for different sensors and heating/AC units, etc.). Please note that you need to include* ***at least*** *2 quality attributes. You can Google for examples that may pertain to your system.>*

# Other Requirements

*<This section is* ***Optional.*** *Define any other requirements not covered elsewhere in the SRS. 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.>*

**Appendix A – Data Dictionary**

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

**Appendix B - Group Log**

*<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>*