# Functional Specification

# Stefanini – Room Scheduler

Phase 1

**Release:** Draft

**Date:** 14 November 2017

**Authors:** Raluca Nechita

# Report History

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision date** | **Author** | **Version** | **Summary of Changes** |
| 22 Sept 2017 | Raluca Nechita | V1.0 | Created First Draft |
| 2Oct 2017 | Raluca Nechita | V1.1 | Updated the draft |
| 2Oct 2017 | Raluca Nechita | V1.2 | Updated the draft with the feedback from the Kick-Off Meeting |
| 9 Oct 2017 | Irina Pandrea | V1.3 | Updated the draft with the requirements ID |
| 11 Oct 2017 | Irina Pandrea | V1.4 | Updated the draft with details from PM |
| 11 Oct 2017 | Irina Pandrea | V1.5 | Updated the draft with details from PM |
| 07.11.2017 | Alina Pila | V1.6 | Updated the draft with details from PM |
|  |  |  |  |
|  |  |  |  |

Contents

[Functional Specification 1](#_Toc495478122)

[Stefanini – Room Scheduler 1](#_Toc495478123)

[Report History 2](#_Toc495478124)

[Revision History 2](#_Toc495478125)

[Purpose 4](#_Toc495478126)

[Naming conventions 4](#_Toc495478127)

[Scope 5](#_Toc495478128)

[1. Global requirements 5](#_Toc495478129)

[Modules 6](#_Toc495478130)

[- 2. Login section 6](#_Toc495478131)

[- 3. Attendee section 6](#_Toc495478132)

[- 4. Host Section 7](#_Toc495478133)

[- 5. Admin section 7](#_Toc495478134)

# **Purpose**

This document provides outline functional specifications and requirements for the Room Scheduler web application, phase 1. Later phases will be determined at a later date, for which, a new document will be created.

It is designed to guide system development and design, including:

* Database structure
* Application architecture
* Site functionality
* Administrative toolset
* Use cases

# Naming conventions

* An **event** is something happening in an interval for a specific room
* **Attendee** is the default user role (the role that an AD user will automatically have it. This role is used by default for a newly created user). Attendee might be referred as “user” in this document.

The requirements set out here are ranked in **M**o**SC**o**W** order:

* **( M )** – Must Have
* **( S )** – Should Have
* **( C )** – Could Have
* **( W )** – Would like to have

# 

# **Scope**

## **Global requirements**

* *1.01* The application will be localized. The supported languages will be English and

Romanian. The default language is Romanian. - **( C )**

*1.02* Responsive Design**: - ( S )**

Supported browsers / devices: Chrome, Edge, Firefox for Windows, Samsung Galaxy Tab 7, Samsung S6 and Safari for iPhone 6s.

The application will have: - **( M )**

* *1.03* The application will have 3 user roles:
  + **Attendee** (the user that will book a room)
  + **Host** (the room host -> default will be “unknown”)
  + **Admin**
* *1.04* The application will have 2 types of events:
  + **Booked** (any user role has access to this event)
  + **Blocked** (only the Admin has access to this event)
* *1.05* An event can have one of the following statuses:
  + **Present** (available only by the host or the admin)
  + **Absent** (available only by the host or the admin)
  + **Cancelled** (available for any role)
  + **Waiting** (available for the Host to go to Present or Absent.

Available for the user for cancel)

* *1.06* The Booked event status flows are:

From waiting, an event can be marked as:

* + Cancelled
  + Present
  + Absent

From Present, Absent, Cancelled, no other statuses will be available.

* *1.07* The Blocked event has no statuses.

## 

## **Modules**

The application will have the following modules and features:

## **2. Login section**

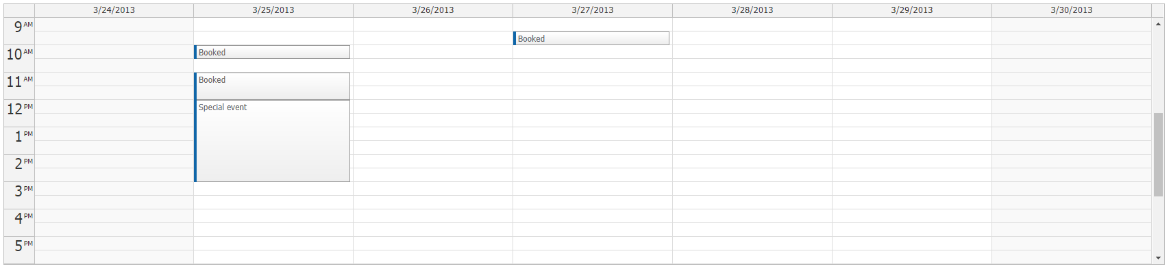
* *2.01* All active domain users will have access to this application - **( M )**
* *2.02* There will be also the possibility to access the application using a username and a password - **( M )**
* *2.03* (If only the login functionality will be available, the Admin -> User Management and Roles sections will become - **( M )**

### **3. Attendee section**

The user section will have:

* *3.01* A calendar, displaying the available time intervals and highlighting the busy ones. - **( M )**
* *3.02* The list of rooms will appear on the right corner as a filter.

By default, the first room defined in the database is loaded.

* *3.03* This page will also have a legend, in which the hosts and their time interval will be displayed.

Example using: <https://javascript.daypilot.org/demo/lite/calendar/nextprevious.html>

* *3.04* The already booked events will be displayed as “Booked” and highlighted in grey. No information about the user that booked will be displayed. Only the user that booked the room will see an “X” button on each **future** event that is assigned to itself.
* *3.05* The events created by the user that is seeing the calendar will be highlighted differently
  + - * Past events will be highlighted in green if they happened = Present
      * Past events will be highlighted in red if they didn’t happened = Absent
      * Future events that will happen “today” in less than 30 minutes from “now” will be grey = Waiting
      * Future events will be blue = Waiting
      * Only for these events the user will see the “X” button / possibility to cancel the event.
      * Once cancelled, the event history will be stored and the event will have the cancelled status for the user that cancelled it (please also store the cancel date and time).
      * Blocked events will be orange = Blocked.
* *3.06* Cancelled events will not be displayed
* *3.07* The calendar can have an additional filter, for hosts.

By default all hosts are checked.

If one of the hosts is deselected, the time interval in which the host is the owner will be disabled from the calendar - **( W )**

Actions available on the calendar - **( M )**:

* *3.08* The possibility to book a free interval by click -> drag & drop - **( M )**

Once stored, the event will be of type “booked” and will have the status “Waiting”. Please make sure to store the date and time when the user submitted the event.

* *3.09* Restrictions:
  + - * + Events should be of:
    - 30 minutes
    - 60 minutes
      * + Events should start at:

X:00

X:30

* + - * + Events should not span outside availability of the room.
        + Events should not overlap events within the same room.
        + Events cannot be created in the past.
        + An event can be cancelled until 15 minutes before the actual start (value that should be configurable, for now, stored in the DB) - **( C )**
        + User is able to create events at any time before the actual start.
        + User can add only one event of 60 minutes or two events of 30 minutes/day regardless of the room.
        + User will not be allowed to make bookings if user has more than 3 missed bookings/last 30 days. Booking rights will be enabled 15 days after the 3rd missed Booking. The events already booked in the interval of 15 days will be cancelled and the room can be booked by other users. The penalized user will be able to book other rooms.
        + Booking is allowed for the current month and the next full month.
        + Cancellation is only available to own bookings (if you are a normal user).
        + The host can cancel any future event
* *3.10* Cancel/Delete an event - **( M )**

## **4. Host Section**

* *4.01* A calendar page, as defined for Attendee. Accessible only by the host. - **( M )**
* *4.02* The host will be able to do the following actions:
  + For today’s past events, the host can mark an event as Present or Absent (while the event is still in “Waiting” status and only after the event started). There will also be a pop-up message for confirming the new status of the event when the host tries to change it.
  + If the host marks an event as “Absent”, the event will become free if there is at least an interval of 30 minutes still available.
  + If there will be two events in the same time (for example, one Present and one Absent), both events will appear in the host calendar.
  + If the host failed to mark the today’s event and is already tomorrow, all yesterday’s event that are still in “Waiting” state will automatically be marked as Present.
  + The host is able to see who created an event.
  + When Host cancels an event, the user that created the event is notified by email. - **( W )**

## **5. Admin section**

* *5.01* User management - **( S )**
* *5.02* Events management
* Block an interval
* See who created an event
* Change the status of an event
* *5.03* Roles Management - **( W )**
* *5.04* Change restrictions- **( W )**
* *5.05* Rooms Managementand Hosts management - **( C )**
* *5.06* Reports- **( W )**