WASHINGTON STATE UNIVERSITY
VANCOUVER

Software Requirements Specification

for

Event Center Scheduling System

Version <1.0>

Prepared by

Group Name: Scofield-Eavenson Ltd.

Matthew Scofield	**11666037**	**matthew.scofield2@wsu.com**
Benjamin Eavenson	**11561401**	**benjamin.eavenson@wsu.com**

Date: 10/24/2019

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Final 1.0	Matthew Scofield Benjamin Eavenson	First publishing.	10/24/19

1 Introduction

The product will be a software system that can be used to track scheduled events for an event center. The product will benefit those who work with clients to schedule their events by making it easier to keep track of preexisting events that must be scheduled around. The users of this product will be able to add/modify/delete events and the product will make viewing these events efficient.

1.1 Document Purpose

This document's purpose is to specify the requirements for the Event Center Scheduling System v1.0. It will cover the requirements relating to viewing, adding events to, and modifying a schedule detailing the availability of various rooms at an event center.

As such the document will clearly lay out the design constraints given by the client to produce the desired product. This SRS will describe the product as a whole system and detailed enough to include all constraints and requirements of the product.

1.2 Product Scope

The product will be a software system that can be used to track scheduled events for an event center. The product should allow for the easy viewing and management of an event schedule. The product will benefit those who work with clients to schedule their events by making it easier to keep track of preexisting events that must be scheduled around.

1.3 Intended Audience and Document Overview

The intended audience of this document is the client who is an office worker at an event center. They are aware of our project in detail and will be the users of this product. The other audience of this document is our professor who will grade this project. Our professor has knowledge of application development.

The suggested order to read this document would be to read through 1.x to get a general sense of what our product is about and how this document is laid out. Sections 2.x will be an overview of how our product will function and should be read next. If more information is wanted, then continue on to section 3.x where this document goes into more detail on how the product functionality will be laid out and the systems the client can expect to be in place. The final section 4.x will describe the constraints and non-functional requirements that our team will provide to the product.

1.4 Definitions, Acronyms and Abbreviations

Not applicable

1.5 Document Conventions

Font Conventions:
Font size in this document is 11 and using arial font.
Italics are used for comments.
Spacing Conventions:
The document is single spaced and maintains a 1” margin.
The document is double spaced between main sections.
List Conventions:
Sections of bulleted lists are bolded.
Lists of requirements are bulleted.

1.6 References and Acknowledgments

Not applicable

2 Overall Description

2.1 Product Perspective

This product is a new, self-contained product. The aim of this product is to provide an event coordinator staff an easy to use interface for scheduling event rooms using this product as a scheduling application. It stores the data for each event in a file on the web server. Below will be a context diagram to show this.

2.2 Product Functionality

- **Add Event**:
  - The user shall be able to schedule new events in the calendar.
- **Modify Event**:
  - The user shall be able to modify existing events in the calendar.
- **Conflict Detection**:
  - The program shall detect scheduling conflicts and prevent the user from creating/modifying events that would cause these conflicts.
- **Event Saving**:
  - The program shall store relevant info *about each event in the planner (Event Owner, Name, Room, Start Time, End Time, Admission Price, Description).*
- **Delete Event:**
  - The user shall be able to delete events in the calendar.
- **Display Events:**
  - The program shall display all the events in the calendar to the user.
  - The display page will refresh at an interval in case events change while a user has a display page open.
- **Login:**
  - The user should be able to log in with their username and password.
  - The user should be able to sign up for an account.

2.3 Users and Characteristics

All users in the system will have the same level of access to the product. The users that this product is intended for are office workers at an event venue. It is expected that they will communicate with each other about the events they did not create if they are to modify or delete them. In this sense, the program will allow all users in the system to modify/delete other users' events. This product should be easy to use for anyone who knows enough to navigate to the web page and login with their credentials.

2.4 Operating Environment

The product should run in the Chrome version 77.x.x.x and Firefox version 69.x.x browsers on windows 10 and OSX Catlina. The program should not have any special software needed for installation besides the above-mentioned browsers and operating systems. This program is intended to be run on a private web server owned by the client company.

2.5 Design and Implementation Constraints

- **Response Time:**
  - The program should have a response time for display of at most 5 seconds.
  - Longest period of response time should be during event save and load type events such as display refresh or product initialization.
- Time to complete the project shall be a constraint on product completion.
- Product shall be designed as a static web app.
- The product should be able to store at most 1000 event entries.

2.6 User Documentation

This software will be delivered to an office staff in charge of scheduling events. The staff will be given a user manual with all of the products functionality explained in detail. The user manual will also contain web links to our company website for more detailed information on how to operate our software, such as tutorials and helpful guides to get them started using the software.

2.7 Assumptions and Dependencies

Assumptions:

The only people with access to the program will be employees of the client company who have access to a company-controlled shared drive on which the application is stored, removing the need for a secure login system to keep unwanted people from accessing it.

The program may be accessed by multiple employees of the client company at the same time, which may lead to conflicts when two or more users attempt to schedule an event at the same time.

The client company will not schedule events that last multiple days.

The client company will never have more than 1,000 events that they wish to track scheduled at the same time.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The product's UI should be designed with ease of access to information in mind. The main screen should display a list of days, the size of which would be specified by the user. Each day will include a list of events scheduled for that day. The displayed event info should include the event name, the start and end times, and the room/location.

Users shall be able to click on a button to open up an interface which allows them to schedule a new event. Similarly, users shall be able to click on buttons that appear on each individual event; one which will bring the user to a screen allowing them to modify the event, and one which will allow the user to delete the event.

The UI for event creation/modification shall include fields for inputting the event name, start time, end time, location, client name, and client contact info. This UI shall also have buttons for saving the event and deleting the event.

3.1.2 Hardware Interfaces

The product (consisting of the webapp and should be hosted on a secure web-server belonging to the client company. All intended users should have access to this web-server. The web-server should be capable of handling traffic from multiple users.

3.1.3 Software Interfaces

The user will use a supported browser Firefox or Chrome to use our software. These should be the only programs necessary to access our software and run it.

The user will be able to specify which events they wish to display on the display screen. During the add or modify event, the user will input the event details (Event Name, Event Date/Time, and Room). This data will be saved to the event database.

3.1.4 Communications Interfaces

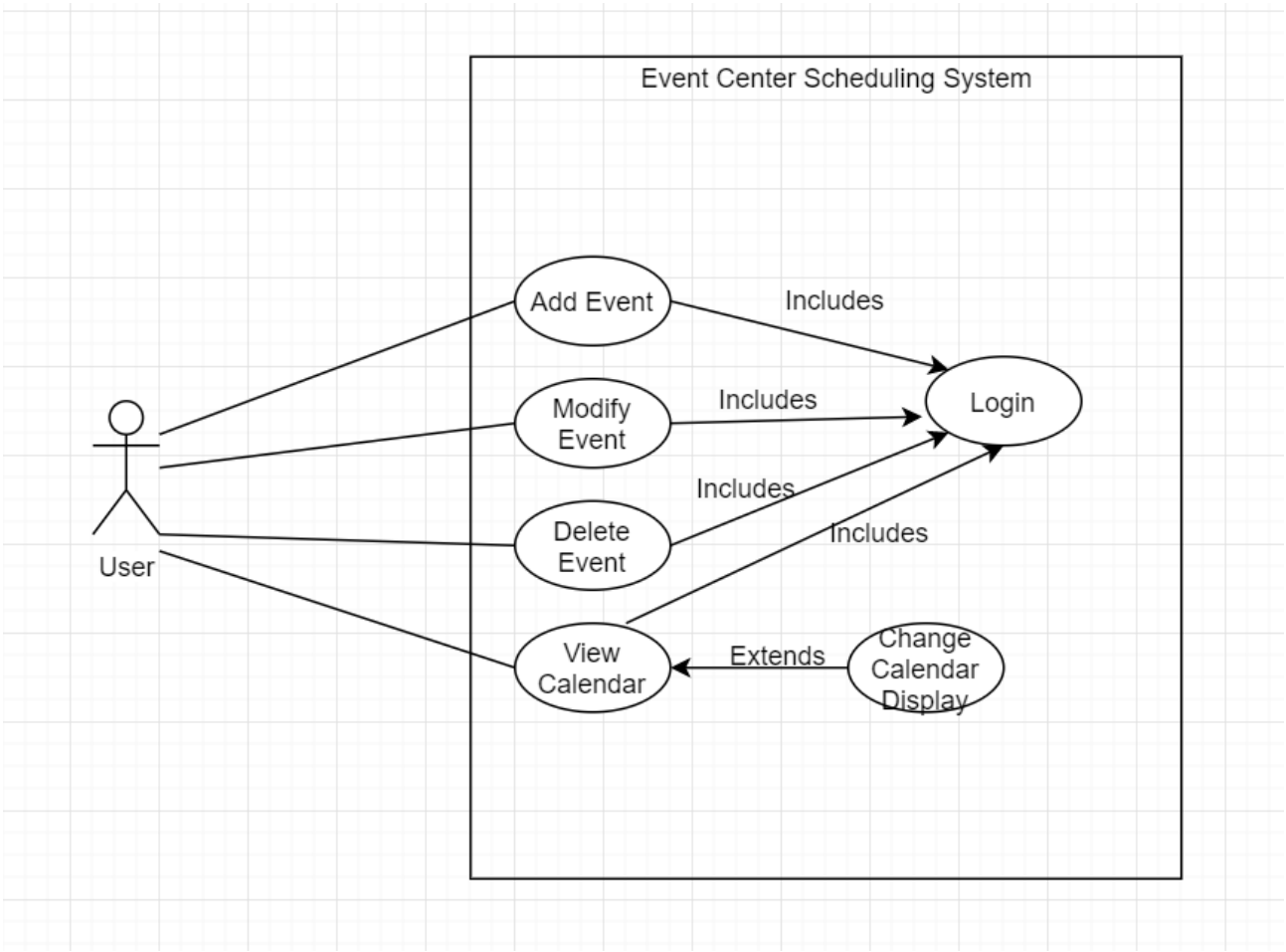
This product requires the user to be using a semi-current version of either Firefox or Chrome. The pages in the product will be using html with JavaScript running the product's main functionality. The pages will communicate with each other via the use of electronic forms utilizing HTTP GET requests.

3.2 Functional Requirements

- **Event Creation**
  - The user will be able to create an event with the following event information:
    - Room Name
    - Date and Time
    - Event Name
  - Upon submitting this input, the program will check that the event can be created without conflict. If no conflict exists, it will save it to the external memory and take the user back to the display screen.
- **Event Modification**
  - The user will be able to modify a displayed event with the following information change:
    - Room Name
    - Date and Time
    - Event Name
  - Upon submitting these modifications, the program will check that the modified event does not conflict with any of the current events in the system. If no conflict exists, it will overwrite the previous event with the new one in external memory and take the user back to the display screen.
- **Event Deletion**
  - The user will be able to delete events from the display screen. If the user does this, the program will delete the event from the display screen and from the external memory.
- **Event Display**
  - The user will be able to select a start date and an end date of the events they wish to display.
- **User Login**
  - The user should be able to login with their username and password and then be sent to the display screen.
  - A new user will be able to register a username and password if they need one.

3.3 Behavior Requirements

3.3.1 Use Case View



4 Other Non-functional Requirements

4.1 Performance Requirements

- Actions started by a user shall take no more than 5 seconds to perform or provide the user with an indication that the program is working.
    - This is to prevent the user from thinking the program has crashed.

4.2 Safety and Security Requirements

Safety Requirements:

- Data will be saved upon the following events to prevent data loss:
    - Event Creation
    - Event Modification
    - Event Deletion
    - User Registration

Security Requirements:

- The Product will be hosted on a private company web server.
- Users will have a username and password.

4.3 Software Quality Attributes

4.3.1 Flexibility

- Code should be written in a way that no function is working with information that does not match its function. This also helps achieve maintainability.

4.3.2 Maintainability

- Code should be written in a readable and modifiable way so that changes to a function do not require excessive changes to the whole program to modify one function.

4.3.3 Robustness

- Customers should not run into any defects that cause the program to crash and lose data.
- Multiple customers should be able to add events at the same time without event conflicts occurring.

4.3.4 Usability

- Software should be easy to understand so that the client can learn how to do their job in a timely manner without frustrations and confusion.
- Tutorials and manuals will be supplied to the customer to easily guide new users on the program’s functionality.

5 Other Requirements

Not Applicable

Appendix A – Data Dictionary

Not Applicable

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 the Teaching Assistant to determine the effort put forth to produce this document>

10/3 Group formed, idea for event center scheduling system decided on.

10/8 Discussed Requirements, Different types of users…

10/10 Started work on SRS document

10/15 Decided that there should only be one level of user, more work done on SRS

10/17 Solidified ideas about UI, more work done on SRS

10/22 Work on SRS rough draft concluded

10/24 Final Revisions Completed