2017FA COSC 1320 001 Class Project

Contents

[The Events System 3](#_Toc499768697)

[Introduction 3](#_Toc499768698)

[A “simple” Events Specification 3](#_Toc499768699)

[Use Cases 4](#_Toc499768700)

[Actors 4](#_Toc499768701)

[Views 5](#_Toc499768702)

[Registrations 5](#_Toc499768703)

[Sets Registration Rules 5](#_Toc499768704)

[Manages Participants 5](#_Toc499768705)

[Manages Participation in Event 5](#_Toc499768706)

[Additional Data 5](#_Toc499768707)

# The Events System

## Introduction

This document is a specification for an “Events” System. “Events” covers a wide functionality and many different Event contexts. From an engineering perspective “Events” can be seen as a functionality “primitive” meaning something that happens attached to a time frame. For end-users Events could be anything from an item in a personal schedule, something that happened in their life (perhaps the subject of a diary or blog post), an informal gathering, a formal gathering or a formal gathering with limited participation (seats) and/or payment.

In addition to the myriad of possible definitions for “Events” there is also an attendant proliferation of features. For example beyond the basic ability to schedule an event (time and description) there are also requirements for functionality like event registration (automatic, role based and requiring approval), multiple event “administrators” (with different levels of permission), “private” events, and recurring events. To make matters even more complex there are multiple “states” that have to be considered in the context of event flow – for example “pending registrations”, “canceled events”, “approved registrations”.

This all means that when we talk about an “Events” system we are talking about a complex and potentially confusing set of requirements, definitions and feature sets. The purpose of this document is to reduce this confusion by “putting a stake in the ground” around what we believe to be an important collection of “Events” functionality, defining that in terms of how this functionality may interrelate with a larger, more complex set of Events functionality and providing enough detail so that an engineer could produce a reasonable system based on these requirements.

## A “simple” Events Specification

We propose the requirements for a “simple”, yet powerful Events system. This set of requirements could be used to manage:

* “Semi-formal” Events like House parties, small meetings, social events, small organizational events (speaker presentations, phone conferences, club meetings, trainings).
* Events where registration can be “open” or “closed”
* Events managed primarily via a Windows GUI. This specification does not attempt to define the technical implementation of the solution.

## Use Cases

“Use Cases” are a specification notation and methodology that capture requirements from an end-user perspective. This is particularly useful because the requirements can be documented in a language that is meaningful to non-technical stake holders as well as Engineers. Furthermore, because Use Cases document complete sets of functionality they are useful in establishing benchmarks and metrics against which engineering progress can be measured.

There are many different definitions of Use Cases, their notation and best practices. For the purpose of this document we capture the Titles of the Use Cases and don't attempt to list out the actual Use Cases in any detail. The justification for this is that the functionality is obvious enough so that the level of specificity achieved by actually creating the Use Cases is not justified. In cases where functionality is more complex or “non-trivial” there will be attendant specification to alleviate ambiguity.

## Actors

Actors are the real people who are expected to interact with the system. Actors execute Use Cases. **Event Administrator** – A user permitted to create and manage one or more events. Might be better to call them an Event Owner, with Event Administrator reserved for someone who can change other people's events.

**Participant** – Someone participating in the event.

The System should be able to create, read, update and delete (CRUD) events and Participants.

Events should have:

* A title
* A description
* A Category
* Beginning and ending dates and times
* Maximum Attendees
* Location
* Participants
* Status (active, pending, canceled, etc.)
* Notes (other text to describe the Event)

Participants should have:

* First and last names
* Age
* The capacity to be registered for an Event

In addition to the above an Event Administrator will also need to “filter” by his “owned” events and “event status” - pending, canceled, active etc. “View My Events” is a search result. The results should be available in a grid, a short description and complete text.

## Views

When Events are presented they should have options for:

* Full View – All Event details
* “Teaser View” - A short detail view
* “List View” - A table view of the events

“Teaser View” should be configurable on a per-event system (per category?) basis.

“List View” should be sortable (with per list individual sorts). What columns show should be selectable (administratively). The list view should allow edits (state, info etc.)

## Registrations

Registrations occur through the Event Administrator over the phone or in person.

### Sets Registration Rules

* Registration on/off
* Private on/off
* Wait List

### Manages Participants

The Event Administrator can view a list of participants. Information about each participant is shown (registration state etc.) Depending on Registration Rules the Participants state can be changed.

### Manages Participation in Event

This is a “My Events” screen and probably can be similar to the Administrative screens.

### Additional Data

* Location
* Event Details (Directions, gift instructions, additional links (with tracking)
* Taxonomy/Categorization