

Events System Specification and Use Cases

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), email notification of changes to events, 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:

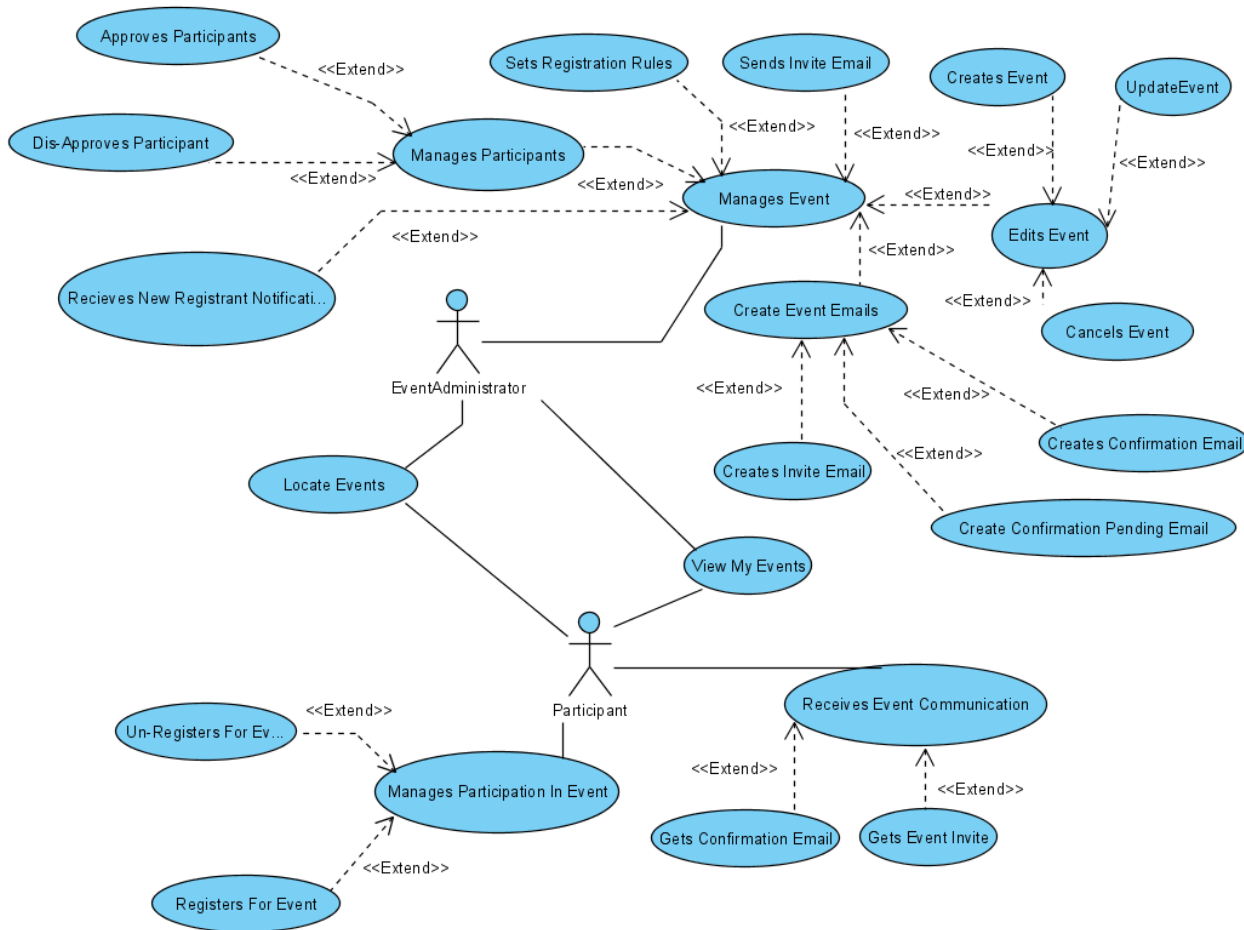
- “Semi-formal” Events like Houseparties, small meetings, social events, small organizational events (speaker presentations, phone conferences, club meetings, trainings).
- Events where registration can be “open” or “closed”
- Events managed primarily via email and the web but with an interface to handle non-web registrations.

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.



Master Use Case Diagram

Actors

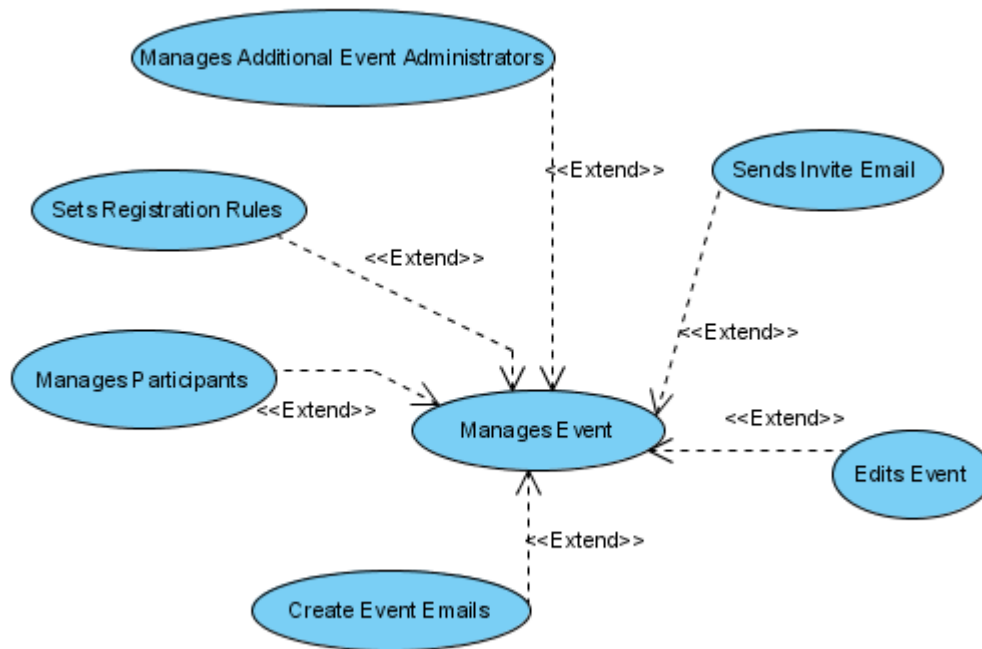
Actors are the real people who are expected to interact with the system. Actors execute Use Cases.

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.

Use Case Details

Manage Events



This is a hub for an “Event Administrator” from here they CRUD events, manage event participants and send event emails. Note that the event illustrated below includes a location – which, as implied by the object model snippet is another associated object “owned” by the event. There are also options to “publish location” (???) and “Show Map” - the map would be an “adornment” of the location. The “instructions” field would also be an adornment (of the Event or the Location?). This is an example of common functionality of an event system that may not be strictly part of an “event”.

Also illustrated is a typical rendered “composite” event.

My Wonderful Event

We will be hosting another opportunity to attend a “My Wonderful Event” Please come!

May 20th, 2006
From 6:00pm to 10:00pm

My Wonderful House
260 Main Street
My Town, AK, 59990



Take a left, another left, a right
and then you'll be there

[XML](#)

Rendered Event “composite”

Add Event

Title

Description

Start
 May 20 2006 calendar
 12 :00 PM

End
 May 20 2006 calendar
 12 :00 PM

Max. Attendees

☐ All Day ☐ No Time ☒ Private

Location
☒ Publish Location ☒ Show Map

Place Name

Street Address 1

Street Address 2

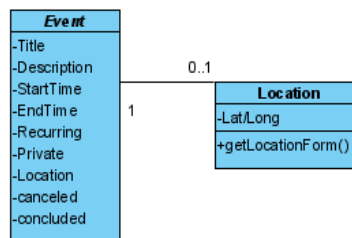
City State Zip

Latitude Longitude

Instructions

- Need a good date chooser
- Default dates should be relevant
- “Instructions” shown on this form would be a CCK data field and are not *required* for events (could be part of location?)
- Once an event has been published it would have a “cancel” attribute

Location should be defined through an interface – not part of event per se .



Add Event Screen

Locate Event

Location of Events is something that both Event Administrators as well as Participants need to do. Participants (or potential participants) will search based on -

- Content
- Location (taxonomy?, keyword?)
- Type (taxonomy)
- Distance from a geographic point (zip code centroid?)

Find Event

Title

Keywords

☒ Title
☐ Description
☐ Title and Description

☒ Contains all words
☐ Contains any words

Date Ranges
 Start May 20 2006 calendar
 12 :00 PM
 End May 20 2006 calendar
 12 :00 PM

Radius
 Zip Distance
 5 miles

Location
 Place Name

City State Zip

Locate Events

- Event Administrator
- Attendees (show me all the events that “Andy” is attending)
- Date and Time

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.

Create Event Emails

During the lifecycle of an Event a number of emails will be generated by the Event Administrator. These include:

Invites

Updates

Cancellation Notices

It is assumed that for this iteration that the system will assist the Administrator in the creation of these emails – but they are not sent “automatically”. For example the Administrator may change the date of an event or the location and need to create an Update email. However the email will not get automatically sent by the system as a result of an edit. (This scenario could be handled later within the context of a “participant (or observer – see “upcoming.org”) subscription” to an Event.)

Event Emails should have templates – templates should be available to all events (site-wide) and it should be possible for an Administrator to create an Event specific template based on a “site wide” template. For this iteration there need be no mandate of the use of a template or template elements (logos, heads, footers, disclaimers etc.).

- Sent Emails should be archived with a record of to whom they were sent.
- A standard set of embeddable links should be available -
 - View the event
 - Register for the event
 - Change your registration
 - Forward to a friend (for “open” events).
- Open rates and click-throughs should be tracked for events

Auto-generated Emails

As the result of some user actions emails will be automatically generated -

- Upon registration either a “successful registration” or a “registration pending” email will be sent to the registrant (what do with registrants who don't have emails?).
- An approved pending registration will result in a notification email to the registrant.
- A cancellation or change of information regarding an event will result in a notification email to all registered users as well as all registration pending users.
- A new registration or pending registration will result in an email being sent to the Event

Administration.

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 via an outbound email as well as a web URL. The outbound email should have a link that brings someone back to a semi-completed registration page.

Forwards

Open events should have forwards built-in. Forwards should be trackable.

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. If a Registrant's state is changed then notifications will be sent to the Registrant.

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