6.170 Project 3 – Food Hunter

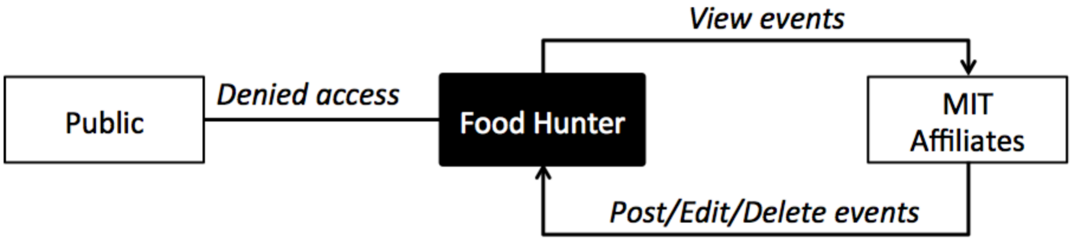
Individual Design Doc
Yachi Chang (changycj)

Design Overview

Food Hunter is a web application that organizes and visualizes the events, specifically those with free food, taking place on MIT campus. MIT organizations, such as student groups, labs, or departments, can post events and indicate the food available on our app, and our app will list these events on a master calendar and also provide a real-time map of where they are. MIT affiliates in search of free food can then look up events and attend accordingly.

The **purpose** of our app is **to organize and present MIT's free food events data**. We feel the current **email spamming** approach that most groups adopt is not effective and creates a vicious cycle of users getting spammed, creating a filter to prevent spam, then groups needing to spam more to get their messages across, then users getting even more spam, and so on. The **free food mailing list** is also hectic at best, and can easily overwhelm one's inbox. Our app aims to visualizes these events in a clean and meaningful way, so **hungry students can easily see where food is available, without having their inboxes explode or having to dig through the spam.**

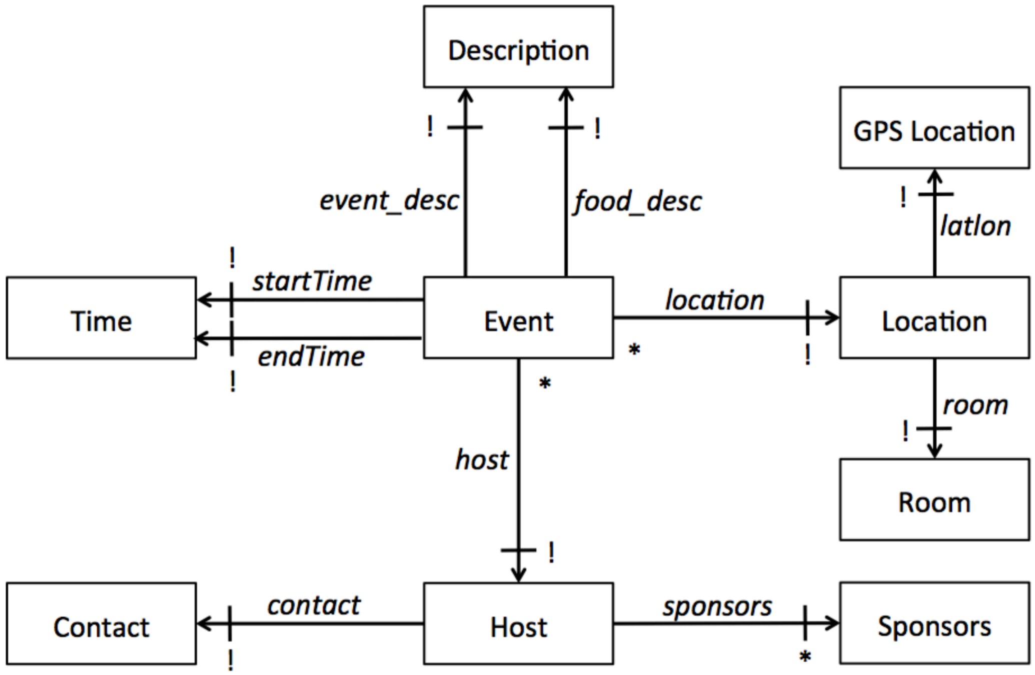
The context diagram for our app is as follows:



To prevent non-MIT affiliates taking advantage of MIT's food resources, the app is strictly limited to MIT affiliates; the user must have MIT certificates to post or view events.

Design Model

Our design model is as the diagram below. We note that it composes of two major elements: Event and Host. An Event contains time, location, and description details about the event, and each Event has a Host, which means a main contact MIT person along with, if any, sponsoring groups, such as clubs, academic departments, or labs. A Host can host multiple events, and also a Location can have multiple Events.



We also define some concepts within this design:

- **Location** links a GPS location with an MIT room number. We plan to visualize all events on a map, and to mark events on the map, we must have a GPS location, but we must also have a room number, so users know where in the building the event is.
- **Event** stores all information about an event: location, time, and descriptions, and also contains a reference to a Host. This arises from the need to document each event, but the possibility that a Host can host multiple events.
- **Host** stores all information about a sponsoring group of an Event; it has a main MIT contact and also a list of sponsors, and is referenced to by an Event. This arises from the fact that a Host can host multiple events, and an Event can have multiple Hosts, so we need to separate hosts and events into separate entities.

Design Challenges

Some design challenges we encountered was:

- **Access control:** We had to decide who can access the event information and also how to control access. Our options included:
    - 1) Everyone can post and/or view and 2) Only MIT affiliates can post and everyone can view and 3) Only MIT affiliates can post and/or view

    - 1) Requiring MIT certificates to post/view and 2) Have people create accounts to access the information
    - We decided that only MIT affiliates can access the information because we don't want to have non-MIT affiliates going to events that weren't intended for them.
    - We also decided to require certificates instead of using accounts, because that's one thing all and only MIT affiliates have, so this way we don't need to store extra data.
- **How to represent hosts**: We had to decide how to represent hosts in our data model, as a host can host many events, an event can have multiple hosts, a person can represent many groups, a group can have multiple contacts, and etc. We had several options:
    - 1) The Host is simply one single MIT affiliate and 2) The Host is one single MIT affiliate plus the organization he/she represents and 3) The Host is only the organization sponsoring
    - We chose the second one, because we want to store who inputted the information into our app as our "main contact" but we also want to keep track of which group is sponsoring the event. We chose Host to contain two references, one to the poster and one to the sponsor.
- **How to store food and event information:** We had to decide how to represent food and event details. We had several options:
    - 1) Store food information by having the poster choose from a list of foods and enter the quantity available, and likewise with events, have users choose from a list of event types and add relevant information and 2) Simply let the poster write short descriptions
    - We thought of the first choice so we could possibly categorize events by food types and quantity, but later we feel that's not really necessary. We eventually chose the second one, because the events at MIT are so diverse that it's hard to capture everything in lists, and forcing the poster to choose from lists of options makes the app harder to use.