## **Problem Statement**

There are many niche and small communities which would benefit from a single platform where users can plan events to share with others in their area. This would help foster a sense of community and sharing a love for their hobbies at a time where meeting new people is becoming more difficult than ever. We aim to build that platform.

## **User Experience**

For most users visiting the site, they will be searching for a new event to attend to socialize, learn, and practice their hobbies. Upon initially visiting the site, visitors will see currently existing (future) events near their current location. These visitors may browse at will or alternatively search by tags. If an event interests them, they must sign-up to the service, create a brief profile, and then sign-up to the individual events by selecting the “attend” button. They are then enrolled into that event. Following attending the event, they will rate the event, and leave a description so others can gauge the event quality.

Some members will come to the site and become hosts of events. Any member of the site can choose to become the host of a site and create a new event. These hosts, from the main page, may create an event, where they are taken to a page in which they define the event details such as the event location, time and date, capacity they are able and willing to host, and a description to include various details of their event.

As an additional note, an important aspect to the service is the ability to rate the events users attend. The rating system is crucial for quality control, and is in effect the way the site admins ensure the events posted are presented as defined in the description and are not abused.

## **Proposal**

Create an application for hosting or signing up for an event of interest.

### **Scenario 1**

As an introvert, Josh finds it difficult to meet others. Josh goes to our app and browses through community events and is delighted to find a lego building club in his area. He signs up to one of the available slots and can now attend the meeting.

### **Scenario 2**

Kelly loves to cook. She decided to help one of her friends cook for a big party. Kelly was informed by many of the attendees they are terrible cooks and wish they could cook as well as Kelly does. Kelly knows of our app and decides to offer free cooking lessons to all her friends and to anyone else in the community who would like to join.

## **Vocabulary**

**Amature Hour**

Amature Hobby Enthusiast group events. The website allows for people to view and add events of their own.

**User**

Users can host and join events.

**Anonymous User**

Can view events and comments but can’t interact with them.

**Event**

User hosted gathering where the user specifies the activity to be performed in the specified time frame.

## **Technical Requirements**

1. Relational database
2. Spring Boot, MVC, JdbcTemplate, Testing
3. React UI and Bootstrap
4. Sensible layering and pattern choices
5. A full test suite that covers the domain and data layers
6. Deployment to AWS.
7. Integration with google maps api

### **Security Requirements**

Amature Hobby Enthusiast group has two roles: MEMBER, ADMIN.

Users must register before they can make or join events.

There are three levels of authorization.

Actions that are available to:

* everyone
* the MEMBER role
* the ADMIN role

## **High-level Requirements**

* Create an event (Logged in, ADMIN approval).
* Edit an event (ADMIN).
* Browse events (anyone).
* Join an event (logged in).
* Schedule an event (logged in) (Needs ADMIN approval before being displayed).
* Cancel upcoming event (ADMIN).
* View any event location by category (anyone)

## **Scenarios**

**Add an event**

Data:

* Brief description
* Category
* Discussion topics
* Suggestions
* Potential cost?
* Date/Time

Precondition: User must be logged in.

Post-condition: If the user is a MEMBER, the event is not automatically posted. It must be approved by an ADMIN. If the user is an ADMIN, they can choose to post it immediately or keep it in a pending status.

### **Edit an event**

Precondition: User must be logged in.

Creator can edit own post, admin can edit

Edit is allowed to add more info to the event or fix mistakes.

### **Cancel an event**

Precondition: User must be logged in with the ADMIN role. Event must be in the future.

Will put the event to canceled status

### **Browse event**

* Text-based: Users filter by name. Display results as HTML.

Precondition: None

Post-condition: None

### **Edit/Cancel Event**

Precondition: User must be logged in.

Members can edit/cancel their own events, Admins can cancel any future event if it doesn’t pass the approval standards.

### **View Location for specific events**

Decide how to display location for an event to anyone who uses the application.

* Map-based: Users filter by city. Display results as a map.

Precondition: None

Post-condition: None

## **Other stuff**

* A User can join/create many events.
* Events have a maximum capacity determined at creation. Could potentially be changed later.
* User created events require administrator approval.

## **Technical Learning Goal Candidates**

* Lombok: It's a java plugin that automatically generates getters and setters or equals methods for the class.
* Google api for incorporating maps.

## **Deployment**

The Amatuer Hour application will be deployed with AWS including utilization of AWS RDS, and for both the server (our API) and our React client load balancers, EC2 instances, etc via docker images uploaded to the Elastic Beanstalk and secured via security groups.

## **Approach**

MYSQL Database:

Build out the sql scripts to build tables:

Login

Roles

-Member

-Admin

Event

-Host

-Description

-Date

-Requirements

-Location

Rsvp

-EventId

-MemberId

Ratings/ Comments

Tags/ Categories?

Users

-username

-password

Notifications/Messaging? If time allows

Backend:

Controllers:

Data: CRUD methods for each table

Domain: CRUD validation for each table

Models:

Utilizing Lombok.

Event

Ratings/ Comments

Tags/ Categories?

Users (To attach event to user)

Notifications/Messaging? If time allows

Security:

Testing

React Client:

Nice looking UI xD

Login

Sign-up

Main page w/ list of locations to filter by

View events by category/tag

Link to see events user already signed up for

Individual event page

Link to ability to rate/comment for that event?

User page? View users past and future events