User stories:

As a user, I want to create an event, so I can get together with others.

As a user, I want to edit my event, as the venue I booked was pushed back a week.

As a user, I want to cancel an event, so that no one shows up alone when the event can no longer take place.

As a user, I want to find an event based on a simple category, so I can get together with others.

As a user, I want to search for an event in a simple category, so I can get together with others.

As a user, I want to search by specific things that may be at an event, so I can find an event with just the right conditions.

As a user, I want to sign up for an event, so I can get together with others.

As a user, I want to be able to pull out of an event, so if I change my mind, I can leave a space open for others.

As a user, I want to look at a map with events near me, so I can get together with others.

As an administrator, I want to remove unwanted and duplicate events from the site, so I can control spam and prevent confusion for the users of the site.

As an administrator, I want to be able to approve when events are edited and added, so I can prevent overlap and other confusion.

Tasks:

Make the **Meetup-Data** database. - 3-5h

Make **User** model. 1h

Make **User-Role** model. 1h

Make **Role** model. 1h

Make **Location/Facility** model. 1.5h

Make **Event** model. 1h

Make **Resources** model.

Make **User-Event** model. 1h

Make **UserRepostitory** repo. 2-4h(Trever)

Make **EventRepostiory** repo. 2-5h(Nic)

Make **LocationRepostiory** repo. 2-4h(Spencer)

Make **User-EventRepostitory** repo. 2-4h

Make **ResourcesRepostitory** repo. 2-4h (Spencer)

Make Exceptions under data package. 0.5h

Extract Interfaces from each repo. 0.5h

Make **UserService** service. 1.5h(Trever)

Make **EventService** service. 1.5h(Nic)

Make **LocationService** service. 1.5h(Spencer)

Make **ResourcesService** service. 1.5h(Spencer)

Make **Result** class. 1h

Make **UserController** controller. 2h(Trever)

Make **EventController** controller. 2h(Nic)

Make **LocationController** controller. 2h

Make **ResourcesController** controller. 2h

Make **UserEventController** controller. 2h

Make **AuthController** controller. 2h

Make **GlobalErrorHandling** class. 2h(Spencer)

(stuff in security package?) ?h

Update our application.properties files (in test too). 0.5h

Update our pom.xml file. 0.5h

Make .dml files for population of database for testing. 1.5h(Nic)

Make **EventRepostioryTest** class. 2.5h

Make **EventServiceTest** class. 2.5h

Make **ResourcesRepositoryTest** class. 2.5h

Make **ResourcesServiceTest** class. 2.5h

Make **LocationRepostioryTest** class. 2.5h

Make **LocationServiceTest** class. 2.5h

Possible helper classes. ?h

Update the **App** class. 0.5h

Create React app and delete default files. 0h

App.js

Event.js

EventForm.js

LoginForm.js

ConfirmDelete.js

RegisterForm.js

ChangePasswordForm.js

Event-data.js

EventList.js

Index.js

ErrorPage.js

Nav.js

NotFound.js

UserEdit.js

User.js

AuthContext.js

Api.js

Auth-api.js

User-api.js

(.test.js for a variety of different ones)

Probably a .js for the map api.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

^

Sum = 10-20h

Make the AWS environment. - 3-5h

Stretch:

Page for users to see all the events they are signed up for.

Page for users to see all the events they organized and ones that got rejected.

Map has clickable points for the event listed there.

Main page (Nic)

Create form page (Spencer)

Find event page (Nic)

Event info page (Spencer)

My events page (Trever)

All events page (admin) (Trever)

Register User Form Page (Spencer)

Login page (Spencer)

Delete Confirm page (Trever)

404 Page (Nic)