Jerome Andaya  
Abraham Cohen  
Clayton Ezzell  
Alfredo Garcia

CS411

Project Assignment 3

The UML Diagram is based on Story 2:

This user wants to go to a concert, but only wants to be surrounded by people his age. He creates an account and signs in. He filters the searches to 21+. He also suffers from low income, so he wants to keep the costs low, so he searches for places that have $20 as the maximum cost for a ticket. He goes to the concert, but leaves dissatisfied. He leaves a ‘bad’ rating on the venue and vows to never return.

This establishes the need for more filters like age and price range. There also should be a way to leave feedback on the concerts the users go to, in case they have a bad time. The reviews and ratings should be visible when the user is searching for a concert.

The UML Diagram is located here:

<https://github.com/agarcia64/FirstGroupBestGroup/tree/master/UML%20Diagram>

The happy path is as follows:

|  |  |
| --- | --- |
| Happy | What could go wrong… |
| User logs in using Bands in Town | Delayed login prevents advancement |
| Search for shows successfully | Incorrect API calls |
| Filter by Age, Cost successfully | Incorrect API calls |
| Purchase Ticket (Link to TicketMaster or on-site) | Wrong links, no way to return to site |
| Rates Band/Venue successfully | Unable to post to database |

Framework of Choice

Express.js is our framework of choice (perhaps even MEAN Stack). Express is very lightweight, requiring only two files to run a barebones site. This will allow us to quickly bring up a simple site to use as a baseline for our application. Node allows for simultaneous processing due to it being an event driven server, limiting the number of blocking calls, and speeding up our site significantly. To use in conjunction with Express will be MongoDB to try and homogenize our code to use JavaScript technologies as much as we can. Depending on our needs, we may also use Angular.js to complete the MEAN stack, but that is yet to be determined.

Ruby on Rails would be a framework not entirely suited to our needs. This is our first web application and the barrier of learning Ruby seems out of scope, whereas we could just as easily write a web application in Python or JavaScript. Similarly, for the future scalability of the site, Rails favors “convention over configuration” which prioritizes the “rails way”, applying patterns to every aspect of the site. Though in the long run this may favor us, this is another learning barrier that can be avoided entirely by using a more familiar language. Additionally, Ruby on Rails is “automagical” which adds mystery to us developers. Though Express also comes with quite a bit out of the box, we will have more control over the actual flow.