2 Heads Are Better Than 1
Iteration 3
Security Document

Security in our project is based largely around the rails gem Devise and OmniAuth using the Twitter API, which handles much of the secure data in Event Social. Devise is a rails gem that provides a flexible authentication solution. It is composed of ten modules, and we use seven of these. One of these modules is Database Authenticatable, which hashes and stores a password in the database to validate the authenticity of a user while signing in. The authentication in Event Social is done through POST requests. This means that we can never see users' passwords in plaintext in the database and it is sent over as a hash to keep it more secure. Another one is Trackable, which tracks sign in count, timestamps, and IP address. Even though we didn't implement all ten of the Devise modules, it would be easy to implement them in the future if we were to continue to work on the project. It is as simple as turning the modules on in Devise and then adding the required code to the project to implement the modules. Because of the scope of the semester-long project, we were not able it implement everything that we would have liked with Devise, but we made sure that there were easy solutions to the problems so that we could implement them in the future if we so choose. OmniAuth handles the authentication with Twitter, and since Twitter doesn't allow non-whitelisted applications like Event Social to view Twitter users' email addresses, we can only access their usernames and can use queries to pull tweets that we need. Twitter uses HTTPS to authenticate users via their Twitter login credentials to allow them to use the Twitter features of Event Social. That means that users' information is always secure and is not being sent over the air in plaintext. Event Social is hosted on Heroku, which is a free, but limited, hosting service for rails that offers basic security for small-scale applications. You can pay for larger applications to receive more benefits.