**To:** Mrs. Boss Lady

**From:** Team 12

**Date:** 4/02/2014

**Subject:** User Authentication

**User Authentication Considerations:**

* **Devise**
  + https://github.com/plataformatec/devise
* Authlogic
  + http://rubydoc.info/gems/authlogic/frames
* Sorcery
  + http://rubydoc.info/gems/sorcery

**Criteria Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Package** | **Devise** | **Authlogic** | **Sorcery** |
| **Rails Version** | 3.2+ | 3.2 | 3.2+ |
| **User Types** | customizable | customizable | customizable |
| **Add/Remove Users** | yes | yes | yes |
| **Sessions** | yes | yes | yes |
| **Password Encryption** | yes | yes | yes |
| **Email Activation** | yes | yes | yes |
| **Logging Info** | yes | yes | yes |
| **User Update** | yes | yes | yes |
| **Installation** | medium | n/a | easy |
| **Integration** | medium-hard | n/a | easy |

All of the user authentication packages that we explored seemed to meet all of our criteria, Authlogic's Rails compatibility being the only exception. In order to use Authlogic for any Rails edition after 3.2, confusing workarounds were necessary. This allowed us to eliminate Authlogic as a choice and decide between Sorcery and Devise.

After eliminating Authlogic, we decided as a group to leave the decision of user authentication package up to how easy each was able to be implemented. In order to test each package, we used the existing depot application from previous assignments. Instead of using the user authentication from the book, we obvious just switched it out with the test packages. While implementing Devise, we ran into many problems integrating the package into the existing code. This was mainly due to inexperience with Rails. However, Sorcery was very easily integrated. We were able to successfully create users, pages that required a user to be logged in, and the ability to update user information. We did not create two different sets of users(basic and admin), but we can easily implement this feature in our Pets application in the future.

As a group we decided that for our application it is best to use Sorcery as our user authentication package. Devise appeared to be the overall best option, however, given our limited experience of Rails and the lack of good support, our group turned to Sorcery instead. Sorcery met all of our criteria, was easy to implement, easy to install, and had a lot of supporting documents throughout the internet.