Documentation
Sprint 3

The base accomplishment of this sprint was to get all of the back end the way we wanted it and to get the appropriate style of what we wanted the app to look like. This sprint we had the following tasks to accomplish:

- Abstract for open house
- Retrieve information from the fitbit, so it could be displayed
- Have a way to place the header to all pages
- Finalize the step page
- Get the menu button converted to a sliding one
- Get dashboard connected
- If time get goals page started as well

Scrums

As for the previous sprints, we have met after class on Tuesdays and Thursdays and in Hale, usually after 4pm on Mondays and Wednesdays. During the weekend, we use texts and Skype to talk about the issues we find.

Accomplishments

We have registered for open house and have put in our abstract for the application. There are something we still need to get ready for open house, like the poster, but we have signed up.

One of the biggest accomplishments of this sprint was of the retrieving data from the fitbit. This was done by creating a client. The client is a way to connect directly to the users information. This client contains our token and secret and the token and secret that came back when the user logged in. The user's token and secret got stored in the database and we needed to retrieve that to create the client in the steps controller. This was the struggle that I will talk more about in the struggles section. Once we got the client object setup we were able to connect to the fitgem api.

We finalized the step page, by figuring out how to display forms in the view and how to use the controller to get data from the model. Now in the step page, you can see the history of steps showed in a form.

We have also made a header that will be in all the pages in our app, containing a menu, the name of the user and a link to sign out. The sliding version of the menu will be added the next sprint.

Struggles

One of the biggest struggles was that of getting the data out of the user table and putting it into the steps controller. What was needed was a method in the application controller that allowed us to know what user was using the application currently. This way we could then connect to the user table to get the users token and secret.

Another struggle we had was when we found out we didn't need the devise user. So we got rid of it and had to reconnect just through omniauth. We had to figure out how to link everything thing together that the devise user did automatically, which helped us understand more about how Ruby works.

Burndown Chart