Team:

Ryan Rouleau
Ryan Turner
Jacob Crawford
Brent Dagdagan

Title: Clean Community

Project Summary: A website that allows volunteers to report issues and potential volunteer opportunities around their city that other volunteers can take on and get credit for their good deed.

Summary of Work Done: This iteration we worked to connect the database to the controllers in order to start serving up data on the api, as well as to create a rudimentary front end. We also wrote unit tests so that any further development in User/Posting classes conform to a specific format.

Breakdown:

Ryan Rouleau - Worked on initializing our Vue.js frontend app and began preliminary steps to start interacting with our REST API.

Ryan Turner - Worked on implementing additional functionality in the user DAO and updating the user posting statistics using the user DAO.

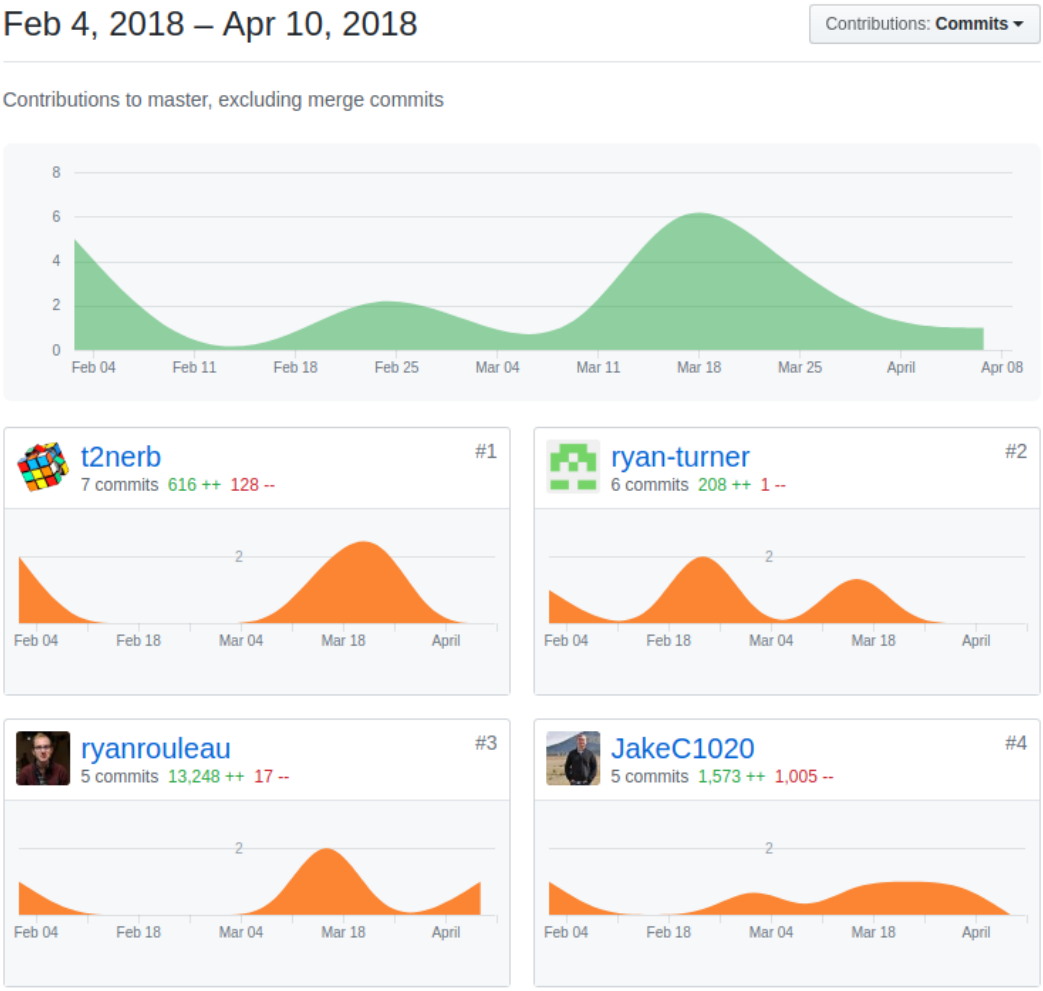
Jacob Crawford - Completed Spring setup, integrated controllers with existing components and worked toward completing the intermediate components.

Brent Dagdagan - Added more functionality to DAO objects. Completed UserFactory for permission levels we've established thus far. Updated DB schema. Added unit tests for data access objects so any further development in these classes must conform to the format in the tests.

Final Iteration:

For our final iteration, we intend to patch up the last few places in our backend that are needed for the controllers to be able to have a path to the database, and connect the front end to the backend via the defined REST API.

Github Graph:



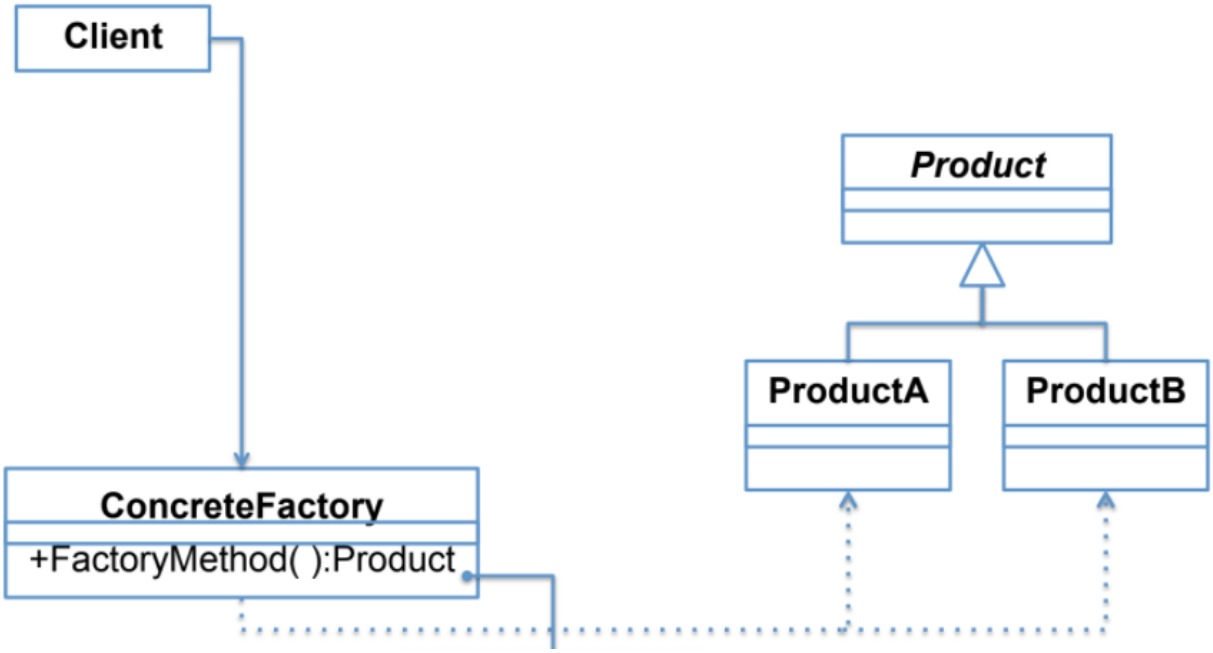
Remaining Effort:
We estimate we have about 8 hours of work left.

As we have developed more this iteration, we have discovered that a lot of our design has been redundant to what Spring provides for us. As a result, we have slowly been trying to simplify the design to reduce complexity and increase maintainability.

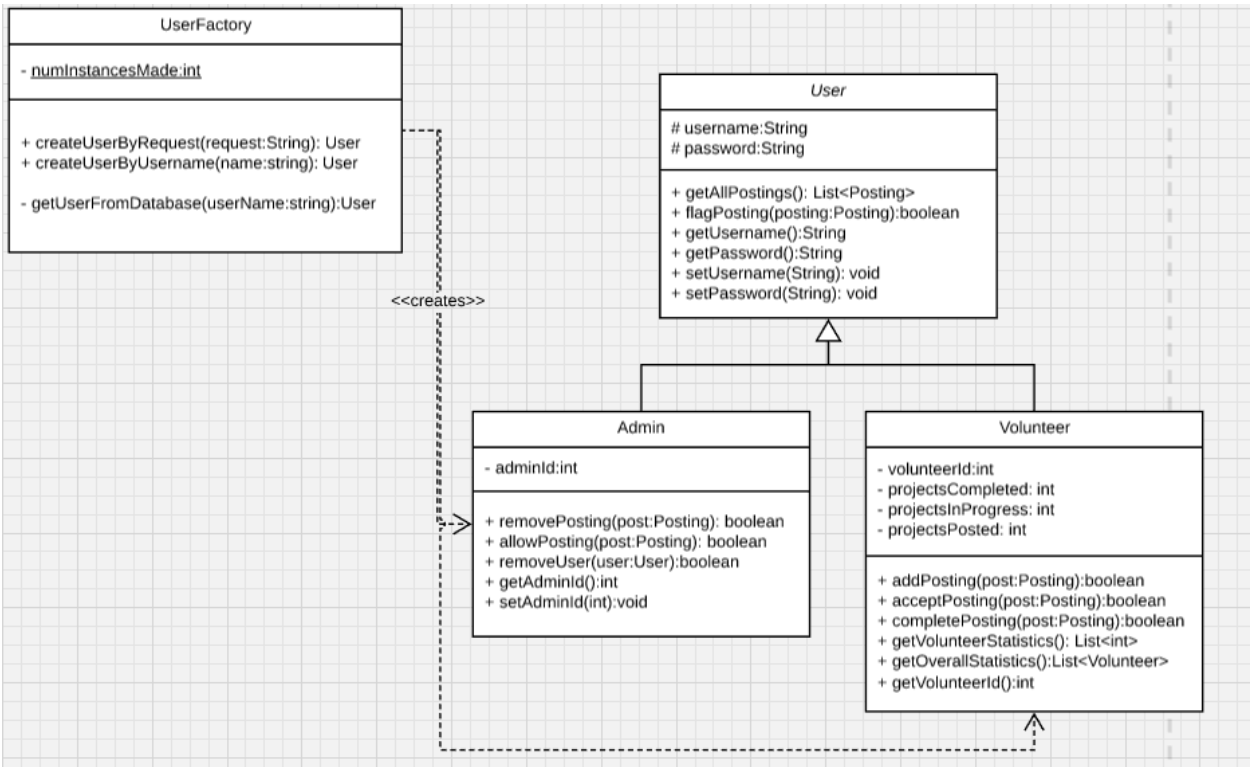
Design Pattern Description:
We decided to use the Factory design pattern in our project. We use this in order to encapsulate the logic surrounding User creation and whether they should be an Admin or a Volunteer. The advantage of this, is it provides more flexibility in the event we have more types of users in the future, such as a Moderator or Organization.

Design Pattern Class Diagram:

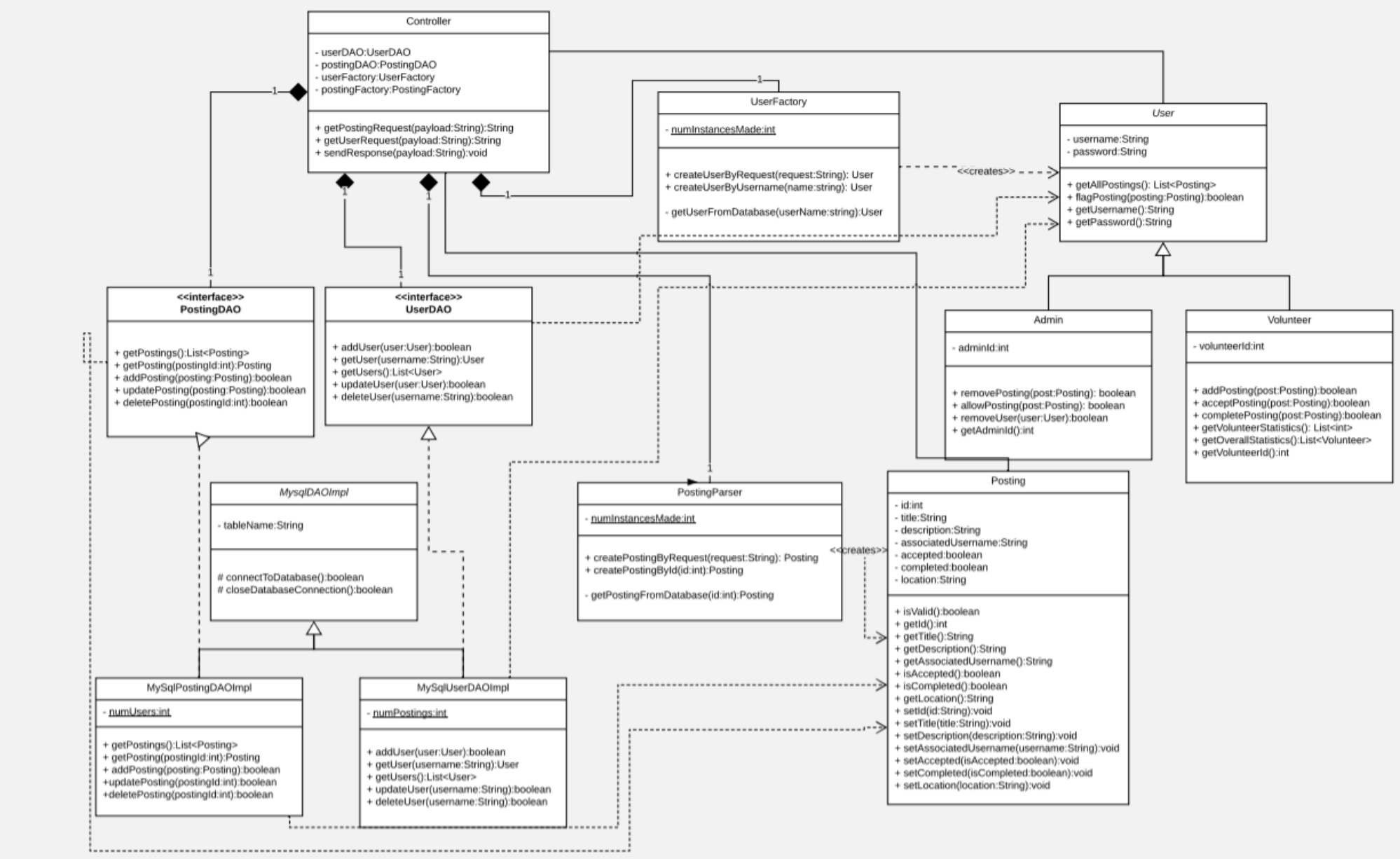
Actual:



Ours:



Previous Class Diagram:



Completed Class Diagram