Davis Yoo

7/20/2018

Report 5

For the past week and a half I have continued working on the Pricing Dashboard project that I started at the beginning of the summer. This project is an Angular application that will be used by the Pricing team. The Pricing team, which is a part of the Financial Services Group, is responsible for documenting changes in prices of shares of Mutual Funds. Right now, the Pricing team begins most of their work when the Stock Exchange closes at 3:00pm Central Standard Time because the prices of Mutual Funds cannot be determined until the market is closed. The job of the Pricing team is to make sure every fund that should have a price does have a price. This is done by email and is updated in the Microsoft Access database. Because of this, the process is a lot slower than it could be. The Pricing Dashboard is an application that will help speed up the process of getting prices by automating some of the work while also making it look better and easier to read.

I started by creating the actual projects in GitLab using templates for the Static and API projects. I began working on the UI portion while another intern worked on setting up the database we would eventually use. With the UI, I started by making a multi-select dropdown menu that would allow the user to select which system they wanted to show up on the page. Each system selected would be displayed in its own box and would have the name of the system along with two circular graphs showing the percentage of prices and rates that have been completed. Each box has two buttons in the middle of the circular graph, one labeled “(x)% Prices” and the other “(x)% Rates” with the (x) being a number between 0-100. When clicked, the Prices button takes the user to the details page of that system which contains a more detailed graph and a data table showing the details of the funds within that system.

Recently, we also started looking into integrating the database into the project. We have been testing out Spring Data JPA. With Spring Data JPA, we are able to query the database for specific information we need, such as getting pricing information based on the system selected. A neat feature of Spring Data JPA is that there are multiple ways you can query the database. For example, one method is to write a really long method name that starts like this: findPricesByID\_EffEndTSAndLastMntDateTimeIsBetweenAndFund\_ID…. and so on. Another method is to define a String variable to a SQL statement and call that variable when needed. This is my first time using Spring Data JPA, so I am still not very knowledgeable with it, but I feel like I am learning and I look forward to looking more into it over the next week. I will only be working full time for one more week before I take a couple weeks off and then go back to part time.