

# Reports

Our task on the reports team was to replicate the reports page from the original Caliber 1.7 into Salesforce to the best of our ability. We decided early on that this would have to be a custom feature written in Lightning, and Apex, along with the javascript charting library, ChartJS. A combination of time constraints, unfamiliarity with Lightning, and the difficulty of the algorithms to retrieve and process data from the Database. Versioning was not enforced during the project, which cause at no less than 3 points, a loss of data. These loses ranged from small setbacks to full blown 30 man-hour rollbacks. It is for these reasons that we only implemented what we implemented for charts.

In retrospect, I would have tried to implement standard reports alongside the custom ones seamlessly to get a more cohesive product out given our limited time and experience. Standard reports were used in the community portal, but I would have liked to cut down on development overhead by using them, and I recommend that future implementations try to integrate them wherever possible even it it means replacing existing custom reports.

I will now discuss specifically what implementation we made and what ones were still pending. This portion assumes the reader has access to the original caliber 1.7 for reference. In addition, here is our wireframe modeled after the original caliber 1.7: <https://www.lucidchart.com/invitations/accept/90c6c89c-e52b-4936-be72-ef456568365f>

You'll notice that we had planned to split off our charts and tables into separate components, which if you look at our actual implementation you'll notice that it did not happen like that. We instead put all our viewstate elements into a single component to make development a bit easier, at the cost of keeping our code more clean and readable.

So far in terms of custom reports, we have created a filter that allows users to drill down by year, quarter, batch, week, and associate. These, as you can see in the lucidchart, inform the view to choose a certain view state: batch, batch by week, associate, and associate by week. We have created the subcomponent A batch and batch by week views with backend logic for the associate view mostly written out. For subcomponent B, we have created the batch view and the batch by week view, but have not implemented the latter into the front end view. We haven't implemented anything from subcomponent C. We implemented the batch view for the D subcomponent, and we didn't get around to the batch by week subcomponent E edge case.

# Portal

## Overall

Mistakes:

Didn't enforce the use of versioning

Slow to react to problems given project duration

Didn't enforce strict adherence to updating documentation