

Comment Anywhere

Week 5 Report, 2/20/2023

Karl Miller

On February 13th, we met to discuss the Gantt chart. Based on the number of implementation pipelines needed and our estimated ability to complete 3-5 per week based on past week performance, we realized we needed to allocate more time for writing all the code. We had allocated too many weeks for deployment on the cloud, so we took a few weeks from that to give us more time to write the code base.

On February 16th, I started writing PageManager and Page. PageManager holds a map of initialized pages. A Page holds a map of cached comments for a particular path and a map of user and guest controllers currently visiting that page. In the future, we can determine which pages to de-allocate by garbage collecting those with no user's visiting them. I also started the route for getting comments by creating the server endpoint method and the controller handler methods.

On February 17th, Luke worked on the back end. He made it so the comments tab is visible for guests. He also created the CafeSettings, CafeCommentSortDisplay, and CafeCommentsWindow classes. He added a "pseudo URL" input that may be removed or made a less visible feature in the future, which allows you to get comments for a page without having to actually visit the page, a useful feature for testing. I also worked on the front end on the 17th, mostly minor tweaks, such as adding the listener and dispatcher methods needed to send requests to the server and parse server responses for getting comments. On that day on the back end, I also generated code using kyleconroy/sqlc for queries needed around comments, domains, and

paths and spent some time creating the regular expression which extracts the domain and path from a URL string.

On February 18th, I created the methods needed to transform comment data in our database to data that can be rendered on the front end and implemented the New Comment pipeline. Later that day, the entire group met and worked on the front end, fleshing out the CafeCommentVote class that Frank had started, which renders a vote dimension (e.g., funny, factual, or agree) and adding CSS for various elements. Then we performed an integration test. There were a couple major bugs that were fixed during the integration test and we documented in our video. The first was that paths were case sensitive, meaning that google.com and Google.com would have different sets of comments. It was an easy fix to convert everything to lowercase. Next was that we were not retrieving comments from the database, due to an accidental copy operation when a reference was required. The final bug was that we were sending empty comments to the front end, which was due to an array accidentally being initialized filled with null data. We fixed these bugs and confirmed that the get comments and new comment pipelines work as expected.