

Circular Progress Bar

Developer: Allison Austin

Submitted April 11, 2023

Reviewer: Joshua Quibin

Reviewed April 13, 2023

Criteria

Major Positives & Negatives

Positives

- Sequence diagrams include detailed logging, message prompts, database queries, method signatures, axios calls, return values, and HTTP response codes.
- Sequence diagrams follow consistent formatting with cascading request/response structure.
- Design follows MVCS architecture, utilizing a view (GroupMembersTab), model (GroupForm), controller (GroupController), and service (ChoreProgressService).

Negatives

- The design documents fail to address all requirements outlined in the BRD. In the Success [Circular Progress Bar] LLD, we see that the ChoreLogger swimlane is represented, but never utilized, meaning there is no logging being performed upon successful changes to the tracking bar.
- Typo with Axios GET Request method on both Success and Failure LLD - "GroupResult GetGroupMemebr(GroupForm)"

Unmet Requirements

- Changes to the tracking bar will be logged to a database in the form of activity and/or errors (if any). (githelsui/WeCasa: @ Team HAGS JP.)

- The Success LLD fails to represent the logging functionality. Hence, this requirement is not met by the design.

Design Recommendations

- Addressing the unmet requirement of logging a successful query to retrieve the chore progress from the database would entail having the ChoreProgressService calling the ChoreLogger. Here we would expect the Log Level to be 'Info' and the Log Category/Type to be 'Data Store'.
 - Positives: Representing the logging functionality within the Success LLD would be fairly simple, as you would simply replicate the same flow represented within the Failure LLD.
 - Negatives: N/A

Test Recommendations

Integration Tests

- Progress Bar Calculation:
 - Arrange a user with a set number of incomplete and complete tasks.
 - Call GetGroupProgress on the user's groupid to verify the score value for the user.
- Completed Tasks:
 - Arrange a user with a set number of incomplete and complete tasks.
 - Call GetGroupProgress on the user's groupid to verify the initial bar value for the user.
 - Manually mark tasks as completed and call GetGroupProgress on the user's groupid to verify that the bar value has increased as expected.
- Overdue Tasks:
 - Arrange a user with a set number of incomplete and complete tasks.
 - Call GetGroupProgress on the user's groupid to verify the initial bar value for the user.
 - Manually change the due date of one of the incomplete tasks to pass.
 - Call GetGroupProgress on the user's groupid to verify that the penalty is applied appropriately to the progress bar.
- Monthly Progress Bar Refresh:
 - Arrange a user with a set number of incomplete and complete tasks.

- Call GetGroupProgress on the user's groupid to verify the initial bar value for the user.
- Manually change the date to the first of the next month and verify that the bar value has reset to zero.
- Logging Changes to the Tracking Bar:
 - Create a user with a set number of incomplete and complete tasks.
 - Manually mark a task as completed and verify that the activity is logged correctly.
 - Verify that all activity logs are saved to the database.
 - Manually change the due date of one of the incomplete tasks to pass and verify that the activity is logged correctly.
 - Verify that the penalty and error message are logged correctly.

Unit Tests

- Test the validation of a Group ID:
 - Given that the WeCasa application supports the functionality for users to belong to multiple groups, I would attempt to query the database for group progress as a user who has permissions and belongs to multiple groups to ensure the correct values are returned.

References

githelsui/WeCasa: @ Team HAGS JP. (2023, Feb 17) GitHub

<https://github.com/githelsui/WeCasa/blob/main/docs/BRD%20v2.1.pdf>