

Feature Name: User Form

Developer: Phong Cao

Developer Submitted: 03/27/2024

Peer Reviewer: Jack Pickle

Review Completed: 04/27/2024

Review Approved: 04/27/2024

Major Positives:

1. Normalization of tables makes data models extensible to n many categories, pushing for extensibility.
2. Current design limits I/O intensive operations by going to the database one time for BOTH the create and edit user stories.
3. LLD extensively highlights the use of succinct helper functions - providing structure to code and making functionality far more readable, even at the LLD level.
4. Design makes use of a repository, to ensure that service layer focuses on dealing with business rules, matching Team Peace's proposed High level design.

Major Negatives:

1. Potential high performance cost due to current data model, required to check two tables whenever writing any new category rating to the table, due to the constraint of the categories table. When inserting multiple values for many users, like scores for each of Lifelog's 10 current categories, this could slow down the user experience of the Lifelog application.
2. SQL Column "IsCompleted" is stored numerous times on the database, leading to a large amount of data redundancy.
3. LLD highlights passing the same response object through numerous layers, passing through responses indicates that the layer should not be there.
4. LLD has some instances of activation lines not being visibly hierarchical.



Unmet Requirements

1. LLD does not contain specific logging signatures that correspond with Lifelog BRD

Design Recommendations

1. Prioritize performance as a design goal, denormalization will make development simpler.
2. While normalizing tables is the most traditional SQL approach that we have been trained in, I recommend dropping the categories table entirely, and instead having a single table, with a user hash column and N integer columns for each categories ranking. This would remove the join, dramatically improving performance.
3. Move the “IsCompleted” value from the user form, and attach it to some part of the user’s information, so that the application can decide to prompt the user with the form based on a table it is already accessing.
4. Revise LLD one more time, looking specifically at introducing logging method signatures and ensuring that activation lines are clearly above each other, highlighting ideal “staircase” structure of LLD.
5. Since the user form is so closely related to a Lifelog user's requirements, it may be worth it to put this into the user management web service.

Test Recommendations

1. Success test for Create, should return success.
2. Success test for Edit, should return success.
3. Attempt to create a form with two of the same ratings ie mental health rating 1 and physical health rating 1, should return failure.

4. Attempting to update a form for a user that has not created a form yet, should return failure.
5. Attempts to give a category a negative score, should return failure.
6. Attempt to create / edit for a user that doesn't exist yet, should return failure.
7. Create / edit a form, and check that the log total increased by at least one, should return success.