

Peer Review

Feature: Lifelog Calendar

Feature Developer: Zarif Shams
Date Developer Submitted: March 28th, 2024

Feature Reviewer: Devin Kothari
Final Date Review Submitted: March 30th, 2024

This Peer Review and Feedback will only involve and pertain to the contents of Lifelog Calendar User Stories 1(LC1), 2(LC2), 3(LC3), 4(LC4), 5(LC5)

Overall Feedback:

Positives

- Design follows the template decided to design all Lifelog LLD's
- All success cases showcase flow of logic and precision of functionality. Most also follow all Lifelog Calendar specific details stated in the BRD.
- Most failure cases/outcomes showcase flow of logic and precision of functionality. Most also follow all Lifelog Calendar specific details stated in the BRD.
- All cases include good technical details and also fully include functions, parameters and variables with proper naming standards to be called in the application, this showcases the developers intuitive-streamline thinking.

Negatives

- The Success case for LC1, LC2, LC3, LC4, LC5 do not contain the specifications for a success log when it is explicitly stated in the BRD that there will be one for each of them.
- Fields where it should be labeled as failure rather than success have not been implemented.
- Most failure cases for LC1, LC2, LC3, LC4, LC5 do not contain the specifications for a failure log when it is explicitly stated in the BRD that there will be one for each of them.

Unmet Requirements:

- For LC1 there are two failure cases that do not have their LLD's showcased; this should be stated somewhere in the LLD doc or should have been changed on the BRD to reflect the developers intentions of not moving forward with them.
- In LC2, there are 14 failure outcomes listed in the BRD, yet only 8 LLDs were documented. It appears that the developer may not have finished implementing the remaining outcomes. Alternatively, the developer might

have realized that some outcomes could be consolidated into a single outcome due to their evolving understanding of development. However, this intention is not explicitly stated in either the LLD doc or the BRD

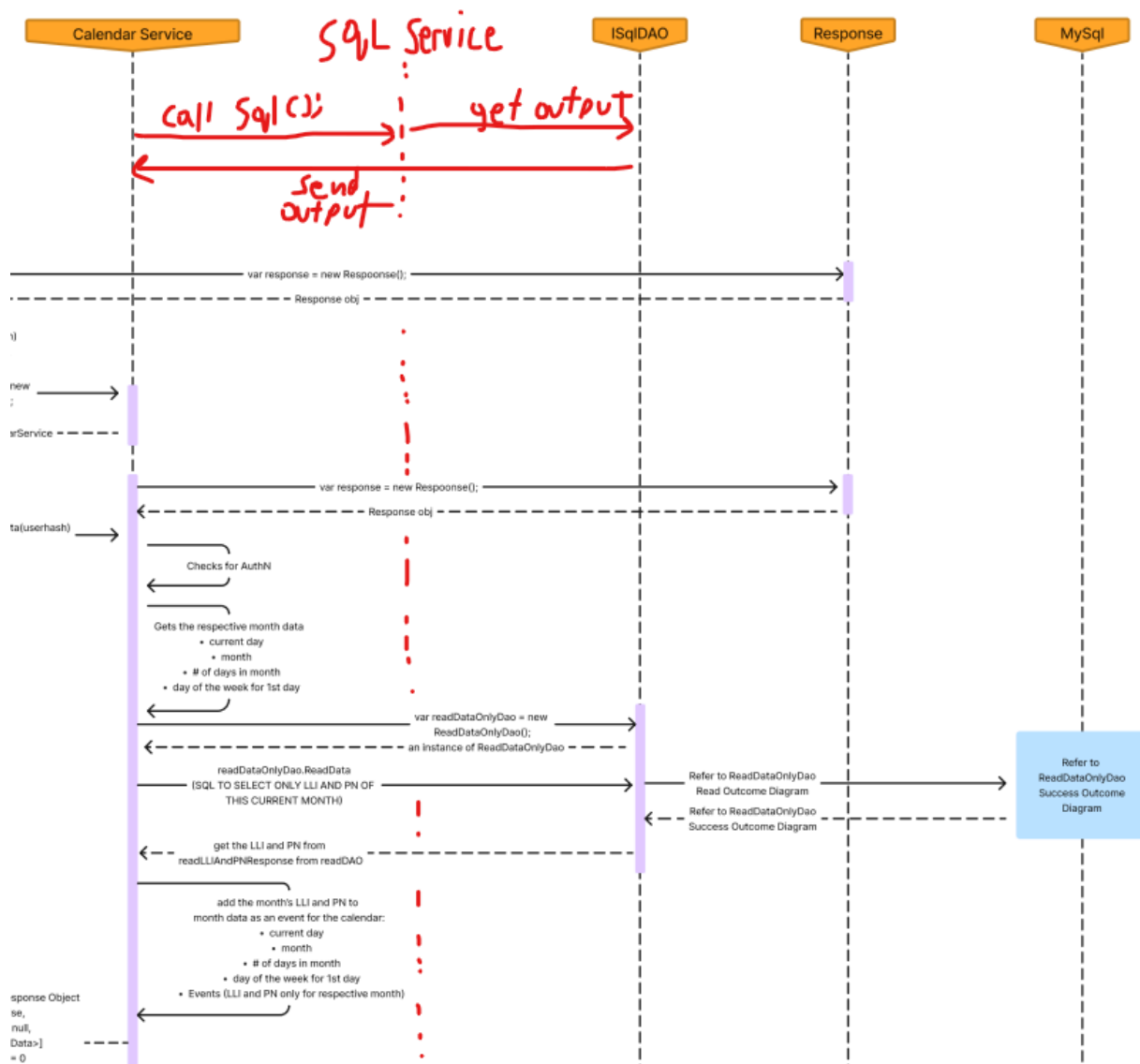
- In LC3, there are 15 failure outcomes listed in the BRD, yet only 9 LLDs were documented. It appears that the developer may not have finished implementing the remaining outcomes. Alternatively, the developer might have realized that some outcomes could be consolidated into a single outcome due to their evolving understanding of development. However, this intention is not explicitly stated in either the LLD doc or the BRD
- In LC4, there are 5 failure outcomes listed in the BRD, yet only 4 LLDs were documented. It appears that the developer may not have finished implementing the remaining outcomes. Alternatively, the developer might have realized that some outcomes could be consolidated into a single outcome due to their evolving understanding of development. However, this intention is not explicitly stated in either the LLD doc or the BRD
- In LC5, there are 5 failure outcomes listed in the BRD, yet only 2 LLDs were documented. It appears that the developer may not have finished implementing the remaining outcomes. Alternatively, the developer might have realized that some outcomes could be consolidated into a single outcome due to their evolving understanding of development. However, this intention is not explicitly stated in either the LLD doc or the BRD

Test Requirements/Recommendations:

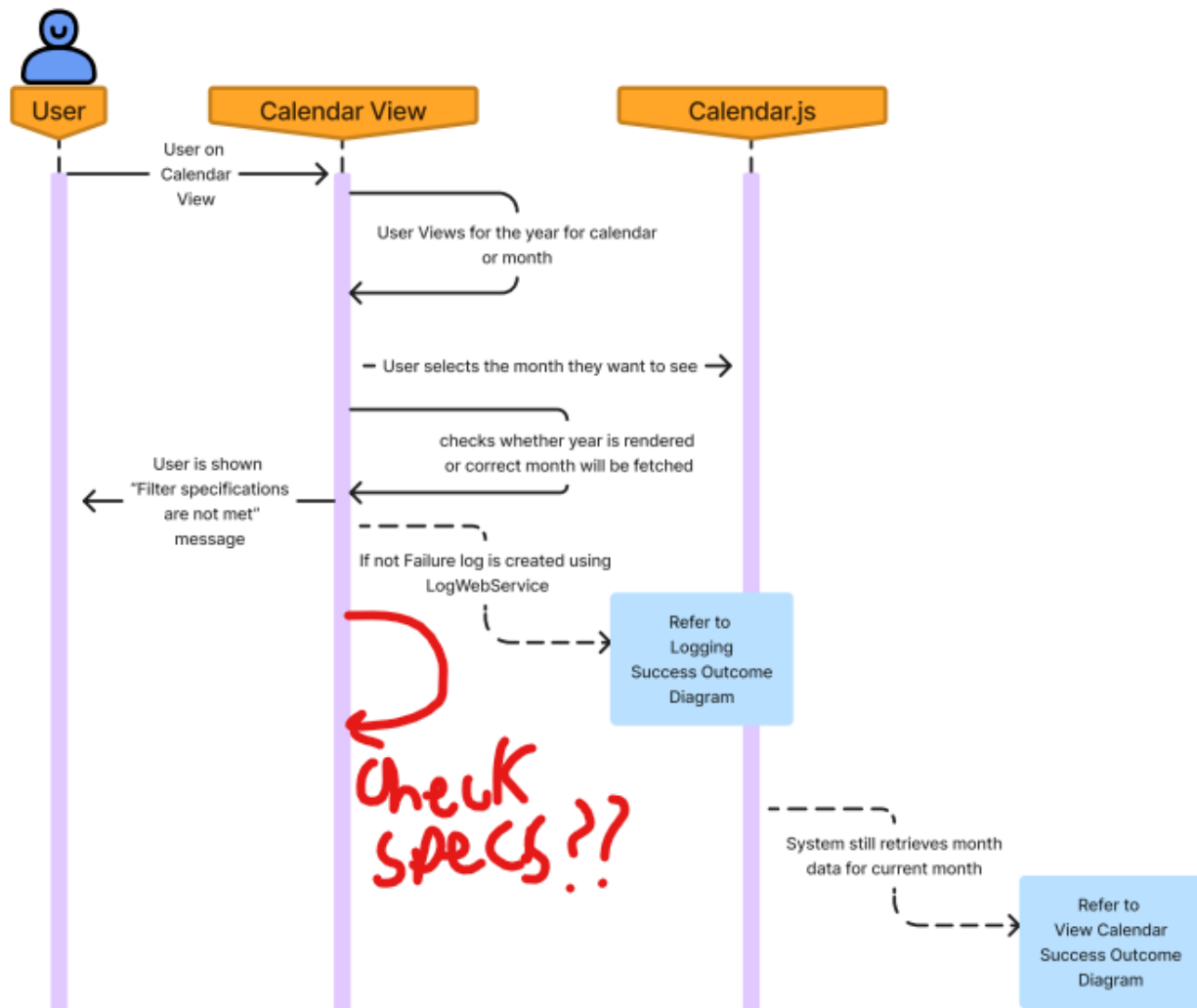
- Based on the test cases that were included in the LLD Doc most were completed and met the requirements.
- Your Feature is a front end heavy feature, due to this you will need to implement front end testing which is not signified anywhere on your LLD document, you will also be required to log any failures in the front end.
 - I would recommend to have a front end test that checks for the amount of events/LLI's in a specific day.
 - Another test would be to see if you are able to access the Personal Note service/view from your calendar view.

Design Recommendations:

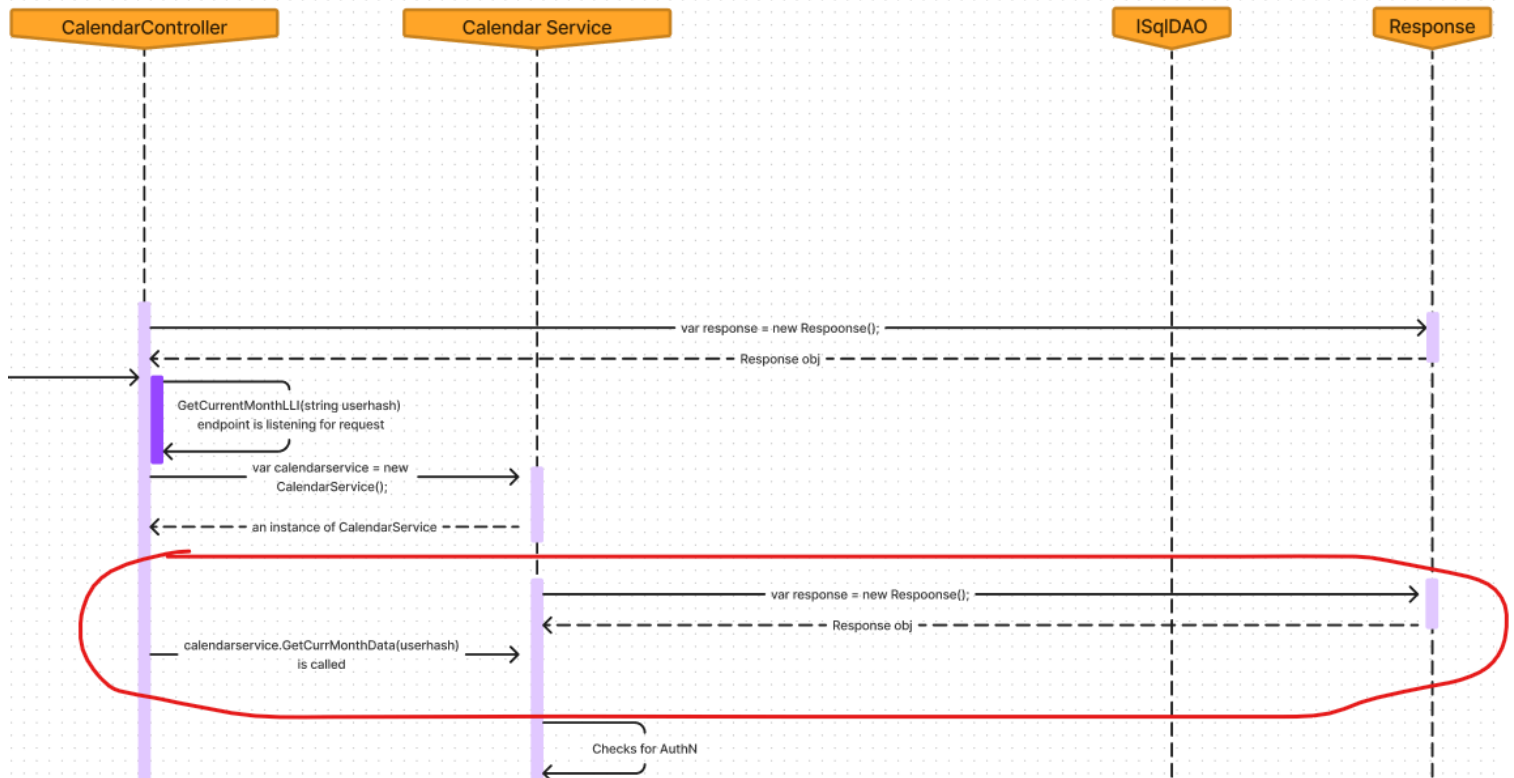
- Overall for all LLD's that include a sql statement within your calendar service, I would recommend to create a separate layer that contains both your dynamic and static sql statements for your service. For example see below.



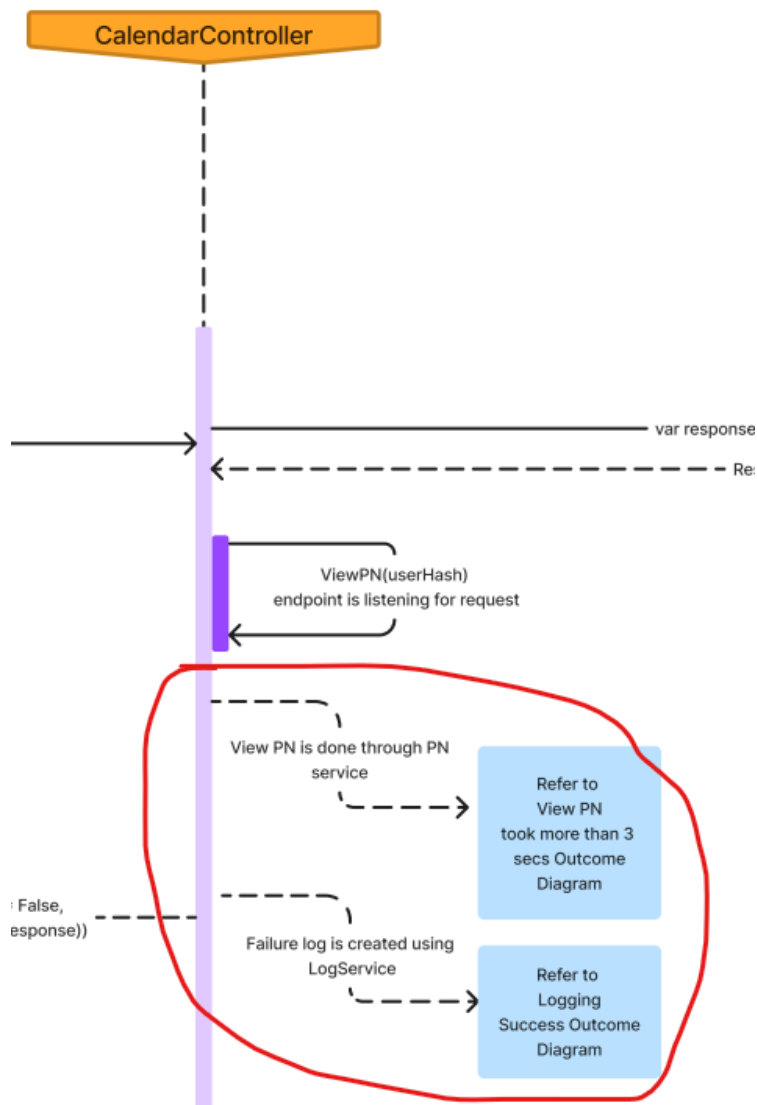
- For the “View Calendar Failure Outcome: Calendar did not meet filter specifications” does not showcase how filter specifications will not be met, does not specify what the filters are, this was not stated in the BRD but it should be stated now IE Date, time, content.



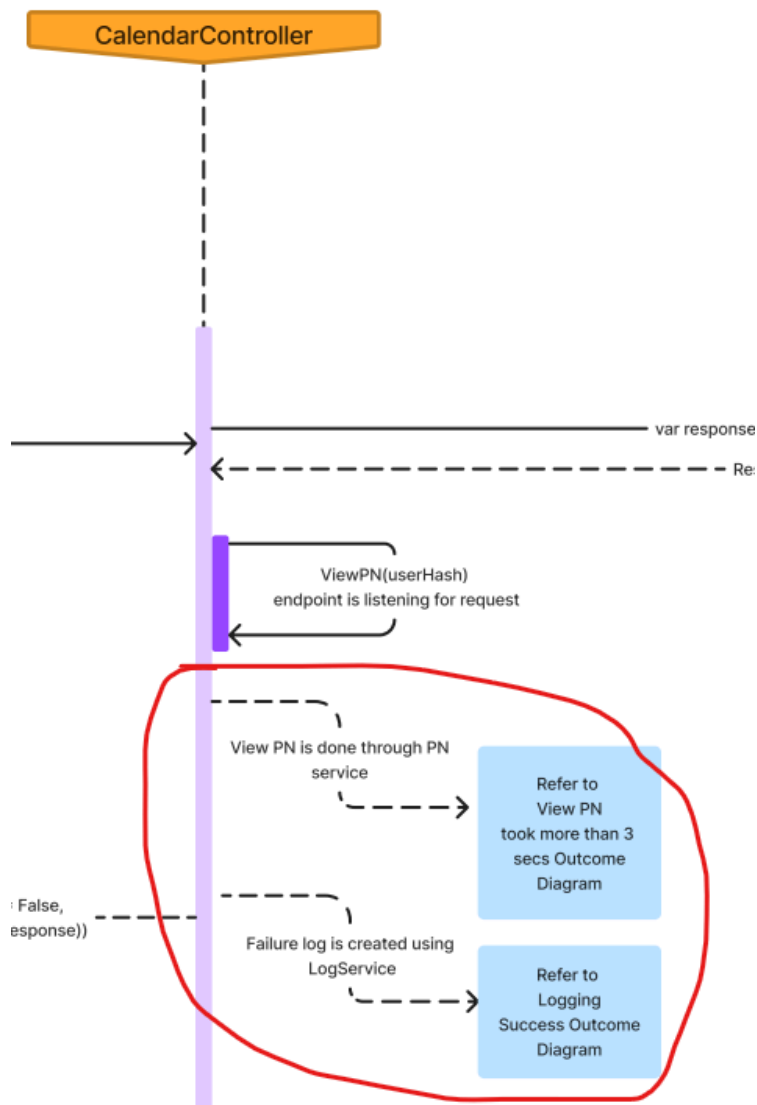
- The “LC2 Failure Outcome: UserHash is empty” & “LC2 Failure Outcome: LLI violates LLI schema” is not stated in the BRD, either remove these failure outcomes as they are not stated or include a note in the LLD stating that these are needed for overall system functionality.
- To improve the design, you should initialize the response object in the service layer below where the function is called from the controller layer, rather than above it. This ensures proper implementation of staircase design, this flaw occurs throughout all the LLD’s within the document.



- To improve the design, you should display a response object being returned from the Personal Note service or LLI service as when you call that service it should be returning a response object and you should be checking that response object's HasError and ErrorMessage, depending on your functionality you should be checking the output as well. This ensures proper implementation, functionality, extensibility and reliability of the feature as a whole, this flaw occurs in LC2, LC3, LC4 and LC5 LLD's as those user stories rely on other services.



- To improve the design, you should initialize the Failure log below the point where the response object is being passed from the service layer to the controller layer. This ensures proper implementation of staircase design; this flaw occurs in LC2, LC3, LC4 and LC5 LLD's as those user stories rely on other services.



- To improve the design, you need to change the arrows from being dashed to solid as this signifies an asynchronous process vs synchronous, especially when the response object is being sent from the service layer all the way to the view. Along with that there are multiple instances of you sending an object from the calendar.js file straight to the user, this does not follow staircase design. Making changes to what was said above will provide a proper implementation of staircase design; this flaw occurs in LC2, LC3, LC4 and LC5 LLD's.

