Peer Review

Feature Name: Media Mementos

Developer Name: Jack Pickle

Reviewed by: Zarif Shams

Date Developer Submitted: April 17th

Date Review Completed: April 27th

Date Review Approved: April 28th

Major Positives and Negatives

- Positives:

  - Each step is coherent to the next step, Operations are done correctly in a sequence

  - Excellent separation of concerns with the MediaMementosRepo. This makes this repo be reusable and extensible by allowing other features that may want to upload media do so easily by using this repo

  - Nice use of notes on the sides to indicate what types of users are allowed for each user story

  - Design is understandable and coherent, ie. Any one from our team can read this design and know how to make a MVP of the feature.

- Negatives:

  - This doesn't seem too heavy on security, the Controllers should have some security mechanism and so should the beginning of your service.

  - Random symbols and letters on the design document itself, this just looks messier and it seems to make the design less important as developer misses attention to detail.

User selects LLI they want to upload media for, and selects a .jpg or .png to upload
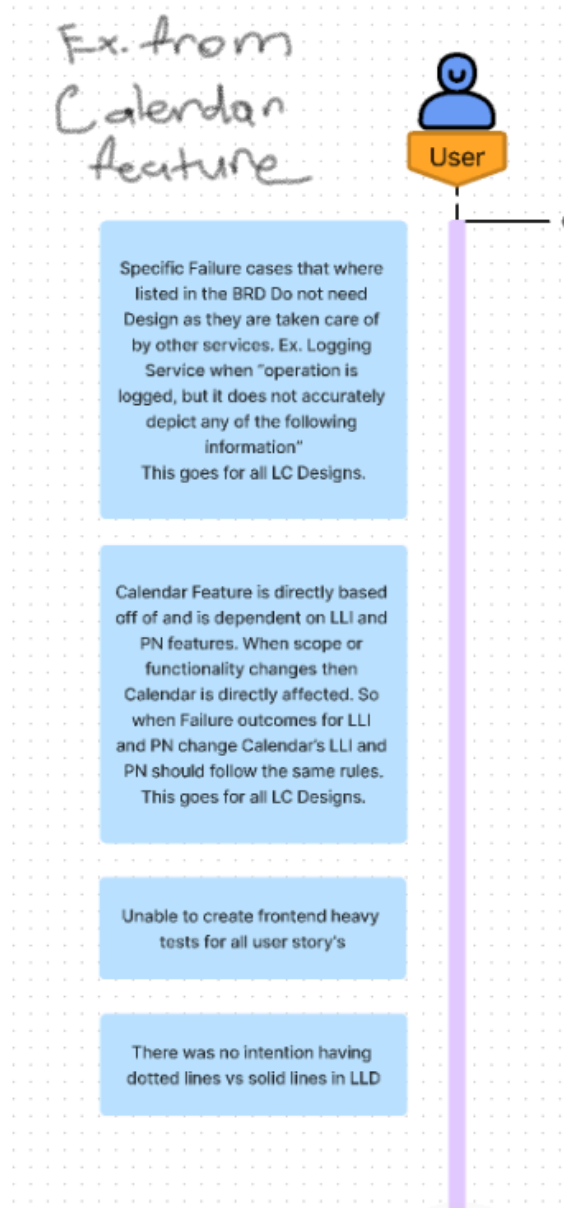
d
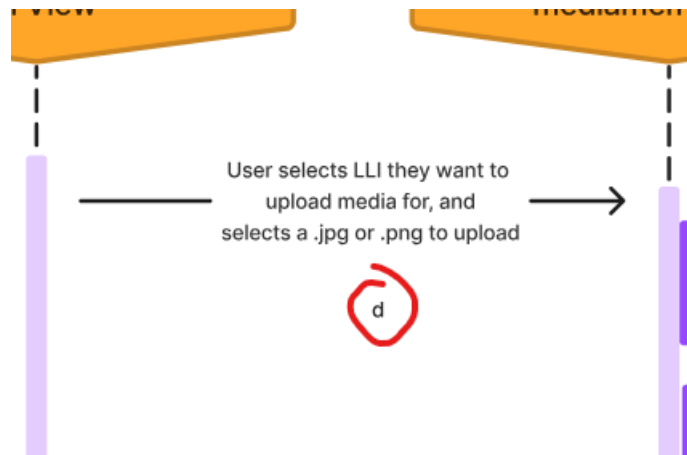
- For all of MM2 failure outcomes Since there is a reference to LLI Failure outcomes Developer needs to mention where up until when in the LLI Failure outcomes do we refer to

- For MM2 Failure outcomes there seems to be 4 main failure outcomes from the BRD but when referring to LLI failure outcomes there are only 2 mentioned.

- For MM2 Failure outcomes the 2 main failures referred to in LLI Failure outcomes don't seem to correlate with the 4 main failure outcomes that are mentioned within the BRD

- For MM3 failure outcomes the "The media file is not unrelated to the LLI." Failure outcome that is stated in the BRD is not mentioned anywhere in the design document.

Unmet Requirements:

- In the BRD for MM2 failure outcomes, the Developer refers to LLI sequence diagrams. In the LLI sequence diagrams referred to I do not see the correlation to MM2 failure outcomes. Developer needs to explain why this is the case otherwise this is considered unmet.
- In the BRD for MM3, failure outcomes for "The media file is not unrelated to the LLI." is unmet. A failure outcome for this is not stated or mentioned in the design document.

Design Recommendations

- Developer should Heavily consider adding some notes about the overall design of your LLDs. Not only will this make your design more coherent and raise transparency, but it will show whomever is looking at your design your thought process, etc. Below is what I did for Calendar.

Ex. from
Calendar
feature

User

Specific Failure cases that where listed in the BRD Do not need Design as they are taken care of by other services. Ex. Logging Service when "operation is logged, but it does not accurately depict any of the following information"
This goes for all LC Designs.

Calendar Feature is directly based off of and is dependent on LLI and PN features. When scope or functionality changes then Calendar is directly affected. So when Failure outcomes for LLI and PN change Calendar's LLI and PN should follow the same rules.
This goes for all LC Designs.

Unable to create frontend heavy tests for all user story's

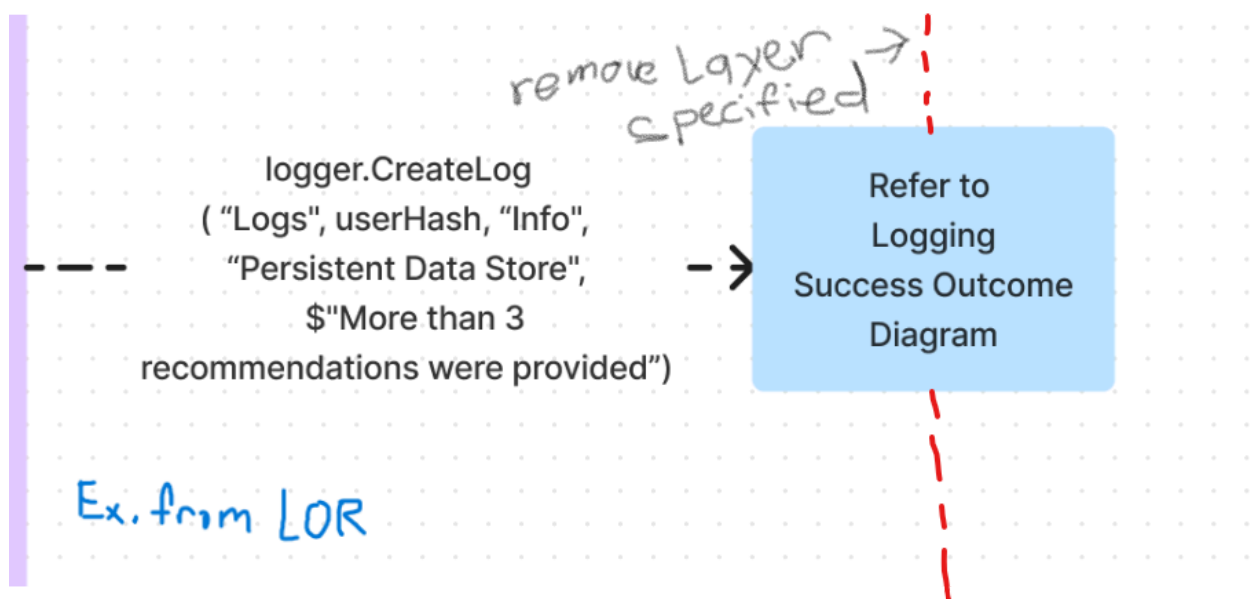There was no intention having dotted lines vs solid lines in LLD

You should put in any explanations for any unmet requirements as notes

- Developer should perform security at points where security should be done. I would try to always put security checks when first going into the controller, and first accessing your feature service.

- Developer should try to choose one main design goal and rank the rest in order of what the developer thinks is the most important. For example,

mention somewhere in your LLDs that the main goal is security, and extensibility is 2nd and performance is last.

- Developer should be able to point out where exactly these design goals are being met and mention that in the LLD.
- Developer should consider removing the logging layer in sequence diagram. When you do log events, have a pointer to a block like so

logger.CreateLog
("Logs", userHash, "Info",
"Persistent Data Store",
$"More than 3
recommendations were provided")

*remove Layer →
specified*

Refer to
Logging
Success Outcome
Diagram

Ex. from LOR

Remove the layer specified and have this pointer to a block be obtained between the two layers. This improves readability and it doesn't look like you are "skipping" layers in your sequence layer

- Developer should also consider removing the Response layer as well in the sequence diagram. The same should be done like with logging as mentioned before
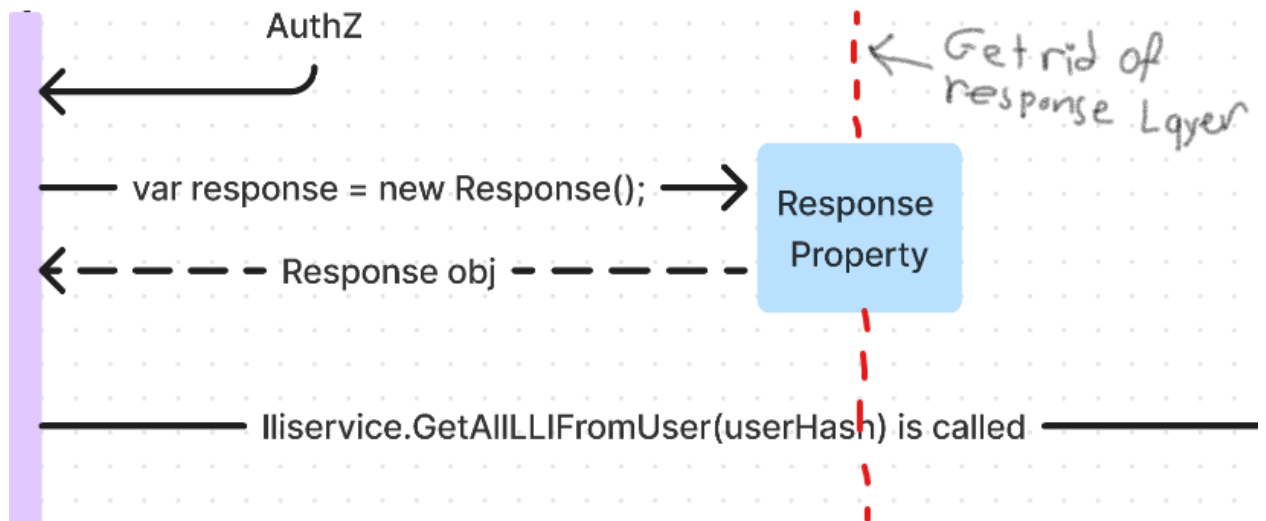
This improves readability and there is no "skipping" any layers in the sequence diagram.

Test Recommendations

- Any Frontend tests being done should use logwebservice to successfully log all tests.
- Frontend testing should be mentioned if it is being done within your feature.
- Make sure your validateBinaryFile() is consulted on. What if someone sends in binary that is meant to break something. This could possibly be a security issue.