Peer Review

Feature Name: Location Recommendation

Developer Name: Devin Kothari

Reviewed by: Zarif Shams

Date Developer Submitted: April 17th

Date Review Completed: April 21st

Date Review Approved:

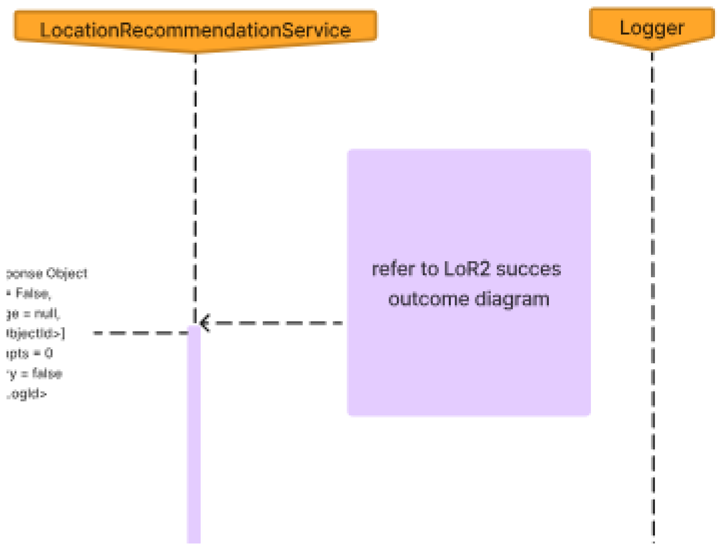
Major Positives and Negatives

- Positives:
  - Very nice flow of steps, each step is coherent to the sequence diagram and steps are not out of order and in sequence
  - Excellent separation of concerns with the inclusion of the MapRepo, this alone makes code more extensible and reusable by other features that need API code from MapRepo.
  - Correct Security/Authorization procedures are being done within the service controller. Validation of Token with correctly being done with Security Service
  - Mentioned which failure cases that are specified in the BRD to be untestable because of over reliance on the API

- Negatives:
  - For LoR1, LoR2, and LoR3, Developer does not need to use POST requests as there seems to be no user input specified in the BRD and it is not specified exactly what data is being sent to the backend server. JWTtoken should be sent no matter what type of http request is being sent. Recommended to use a GET request instead.
  - For LoR4, Developer needs to specify what kind of HTTP requests he is doing here. For the frontend to interact with controllers there

needs to be a HTTP request done; it is not being specified in this user story.

- For LoR4, Developer seems to go to the backend just for the authorization of the token. There should be no need to go to the backend just for authorization as correct security procedures should already be placed in the frontend.
- For LoR4, ViewIGM() method in LoR service only authorizes the appPrincipal, it does not do what the method name implies.
- Developer needs to mention for all User stories that the failure outcomes for logging that are stated in the BRD are taken care of within the logging service.
- Developer needs to mention where up until when in the sequence diagram do we refer to in LoR failure 4, since reference is in the middle of a sequence diagram.

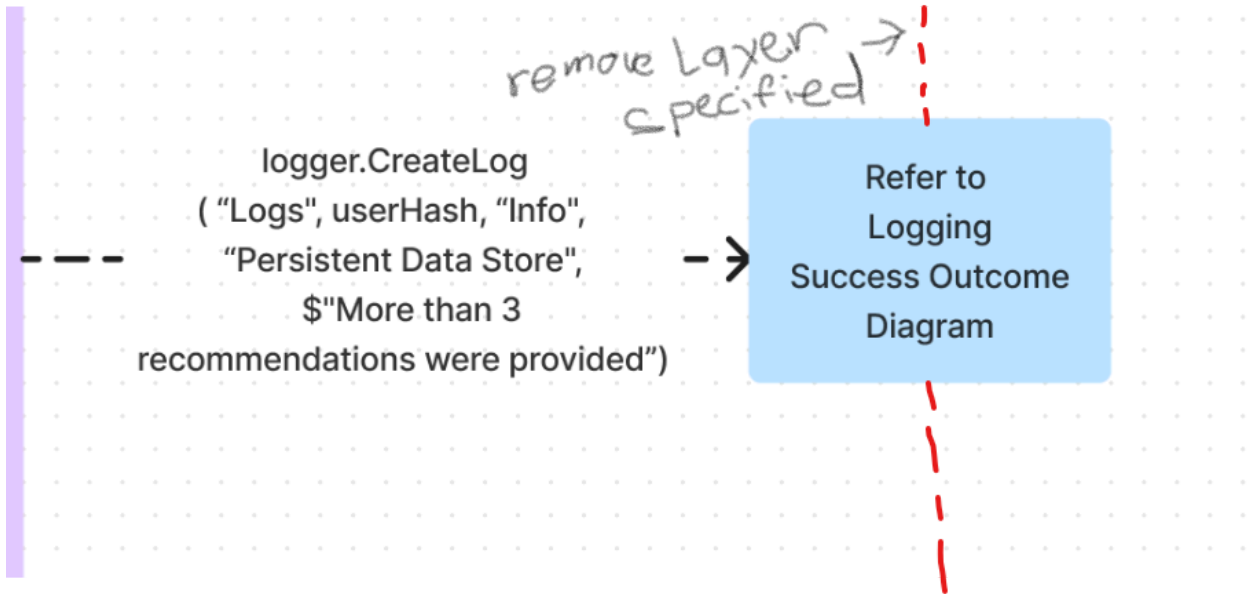


Unmet Requirements:

- No mention of failure outcomes that are failures to store data in a database or failures to correctly log information. Developer should mention at the beginning of the user stories where these failure outcomes are taking place.
- In the BRD for LoR3 Developer seems to use the same sequence diagram for "The view containing the following is not displayed" and "The pin's corresponding LLI details are not shown" failure outcomes. Developer needed to explain why this is the case.

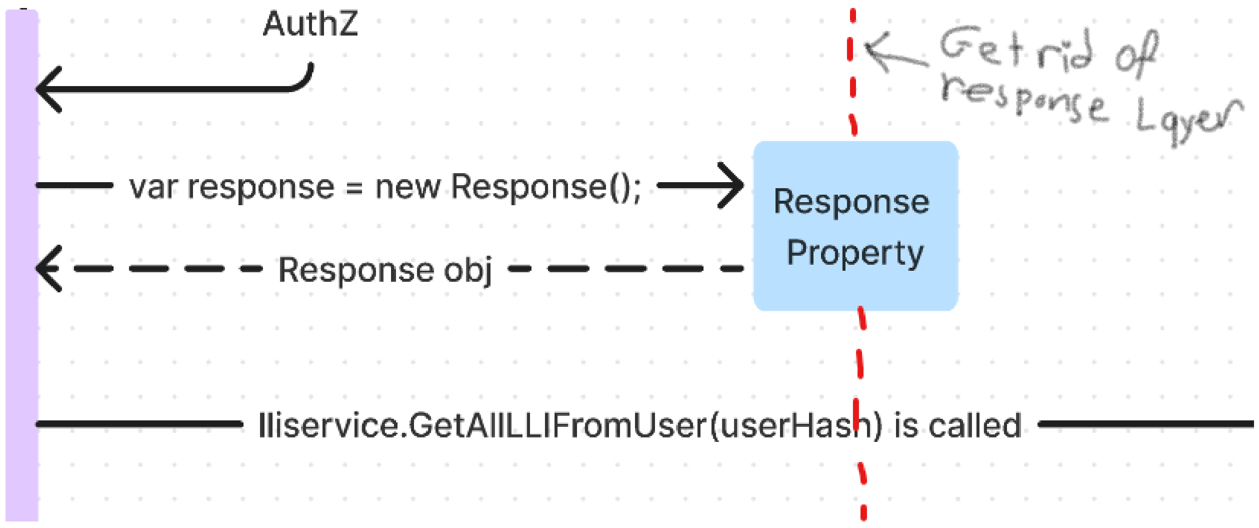
Design Recommendations

- Developer should consider removing the logging layer in sequence diagram. When doing logging just have a block that references the logging being done within the diagram.


remove Layer specified
logger.CreateLog ("Logs", userHash, "Info", "Persistent Data Store", $"More than 3 recommendations were provided")
Refer to Logging Success Outcome Diagram

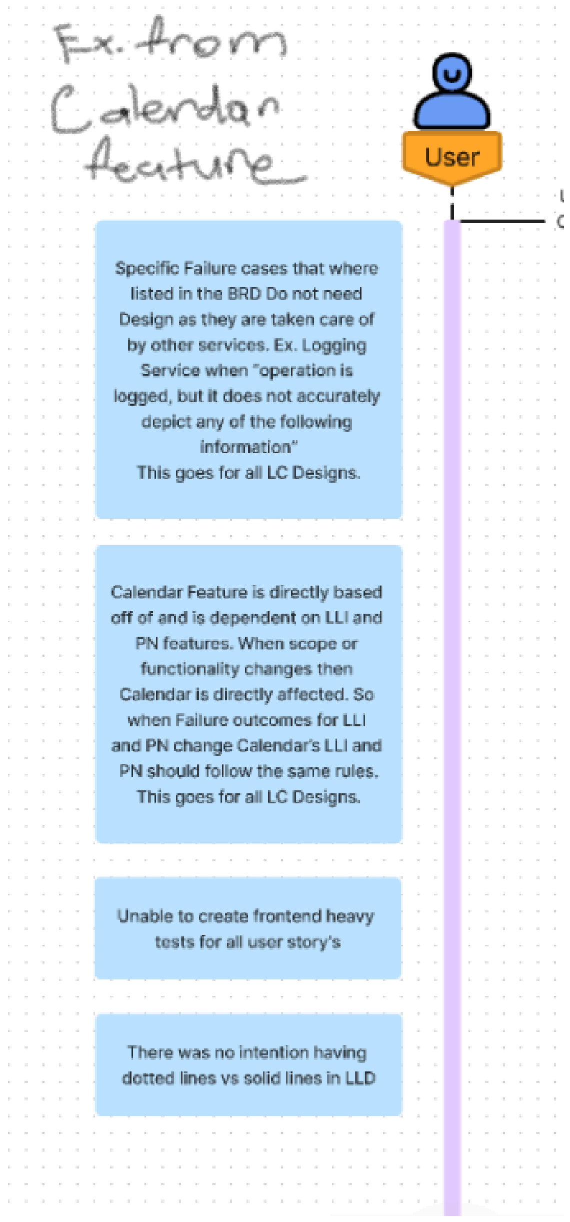
This improves readability of the sequence diagram as there is no "skipping" any layers in the sequence diagram anymore.

- Developer should consider removing the Response layer as well in the sequence diagram. Whenever creating a response obj just has a block that references the creation of a response obj being done within the diagram.


AuthZ
Get rid of response Layer
var response = new Response();
Response Property
Response obj
lliservice.GetAllLLIFromUser(userHash) is called

This improves readability of the sequence diagram as there is no “skipping” any layers in the sequence diagram anymore and also contains this operation between the two layers where it is supposed to happen.

- There should be any extra notes said about the overall design of your user stories/feature. This is where you should specify what parts of the design you couldn’t do due to time constraints or it being done somewhere else.



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.

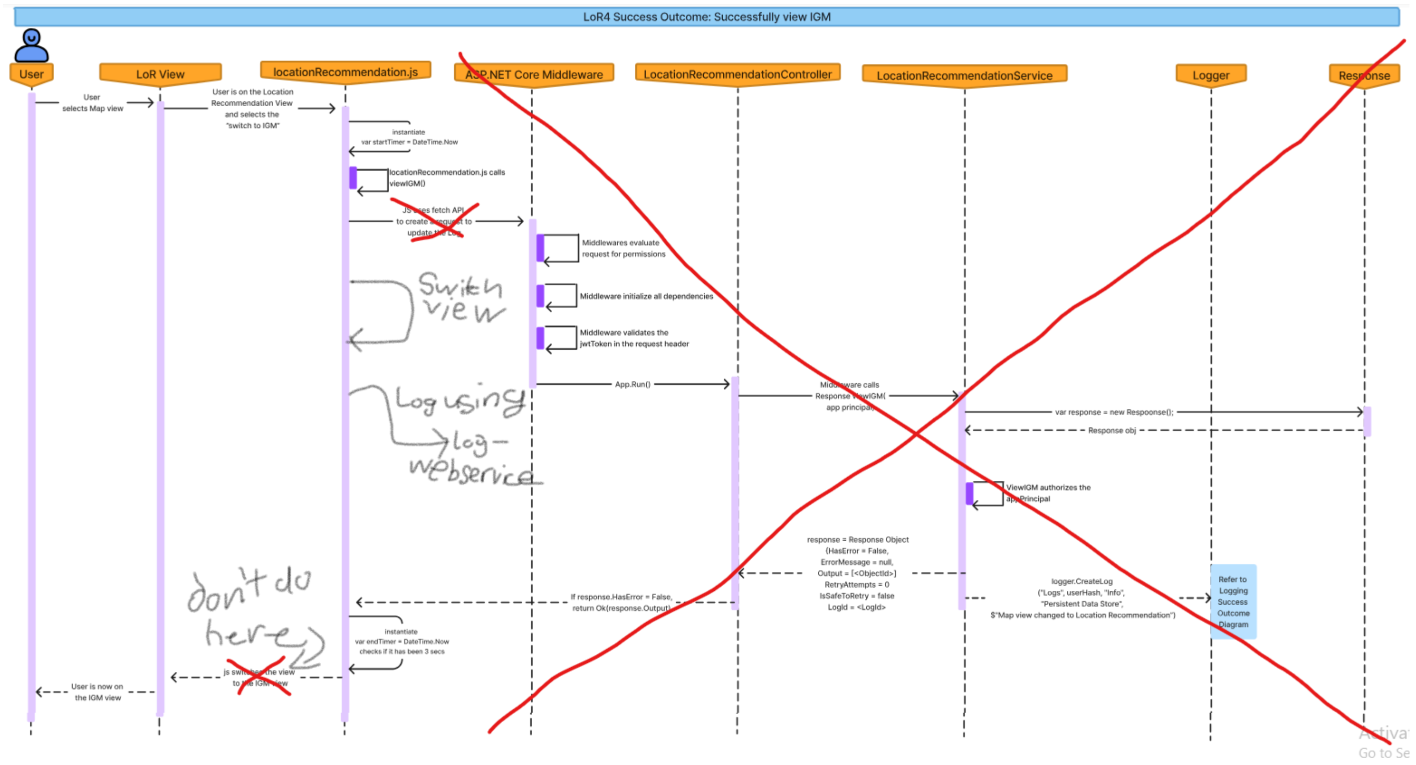
Calendar Feature is directly based off of and is dependent on LLI and PN features. When scope or functionality changes then Calendar is directly affected. So when Failure outcomes for LLI and PN change Calendar's LLI and PN should follow the same rules.
This goes for all LC Designs.

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

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

This provides people who are viewing your sequence diagrams more context onto why you designed it how it is.

- LoR4 should be a feature that should have only been done in the frontend assuming correct security procedures of authorization is being done in the frontend. There is no need to go to the backend especially since the only operation being done is authorization and logging.


LoR4 Success Outcome: Successfully view IGM
User
LoR View
locationRecommendation.js
ASP.NET Core Middleware
LocationRecommendationController
LocationRecommendationService
Logger
Response
User selects Map view
User is on the Location Recommendation View and selects the "switch to IGM"
instantiate var startTimer = DateTime.Now
locationRecommendation.js calls viewIGM()
~~JS uses fetch API to create a request to update the Log~~
Middlewares evaluate request for permissions
Middleware initialize all dependencies
Middleware validates the jwtToken in the request header
App.Run()
Middleware calls Response ViewIGM(app principal)
var response = new Respoonse();
Response obj
ViewIGM authorizes the appPrincipal
response = Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
logger.CreateLog ("Logs", userHash, "Info", "Persistent Data Store", $"Map view changed to Location Recommendation")
Refer to Logging Success Outcome Diagram
If response.HasError = False, return Ok(response.Output)
instantiate var endTimer = DateTime.Now checks if it has been 3 secs
~~js switches the view to the IGM view~~
User is now on the IGM view
Switch view
Log using log-webservice
don't do here

Test Recommendations

- Any Frontend tests being done should use logwebservice to successfully log all tests.
- When reading a pin in the MapRepo make sure there is error checking whether the pin passed in actually exists or is not null.

- Additionally in the MapRepo make sure that the pin is actually a valid correct pin. You don’t want to pass in the incorrect pin or a mock pin that is meant to break MapRepo. Would recommend comparing the hashes of the pin.