Feature : Calendar
Developed by : Anh Huynh
Submission date : 4/17/2023
Reviewed by : Aster Lee
Review date : 4/18/2023

Major Positives
- The design follows the HLD properly.
- The design shows a knowledgeable use of asynchronous methods to be used for each layer.

Major Negatives
- Layers from Service to Storage is all pass through methods.
- There does not appear to be any security checks.
- There are no designs made for failure cases.

Unmet Requirements
- Add Event (Insert Appointment)
  - There does not seem to contain location of the appointment.
  - There is no notification for finishing Insert Appointment story.
  - There is no limitations of 5 appointments per day.
  - Success case mentions sidebar which is not mentioned in the design.
  - There are no designs for failure cases.
  - There are no business rules being checked.
- Modify Event (Update Appointment)
  - There is no notification for finishing Update Appointment story.
  - The design lacks sidebar which the success case mentions.
  - There are no designs for failure cases.
  - There are no business rules being applied to the design.
- Delete Event (Delete Appointment)
  - There are no notifications being sent to the user or other parties upon deletion of appointment.
  - There are no design for a case where the user deletes a project.
  - There are no designs for failure cases.
- Get All Appointments from a user and display on Calendar
  - There are no mentioning of this in the BDR and, thus, cannot be checked for unmet requirements.

Design Recommendations

- Do security check in the manager layer.
- Stop doing pass through functions on service and DAO layer.
- Use try catch blocks in order to check for unexpected error as there are no failure case designs.
- On Get All Appointments from a user design, returning a boolean from the data storage does not seem a great idea.
- The data gained from the Data Storage should be filtered in Service or DAO to only get the specific data of the Appointments users will need.
- In the Insert Appointment design, make sure to add location variable as it is a part of the appointment along with time and date.
- Specify what is going on between DAO and Data Storage when the DAO is executing "await command.ExecuteWriter()"

Test Recommendations

- Backend
  - Make sure to test Manager layer as there are multiple commands happening in the Manager.
  - Try to make sure the boolean returns are proper on each layer.
  - If Service and DAO are staying the same as the design, there are no real reason to test as they are pass through functions.
- Frontend
  - Do a end-to-end testing to see the updates being properly done on the calendar whenever the user inserts, updates, and deletes appointments.