

Al Abriam

CS 441

1 December 2019

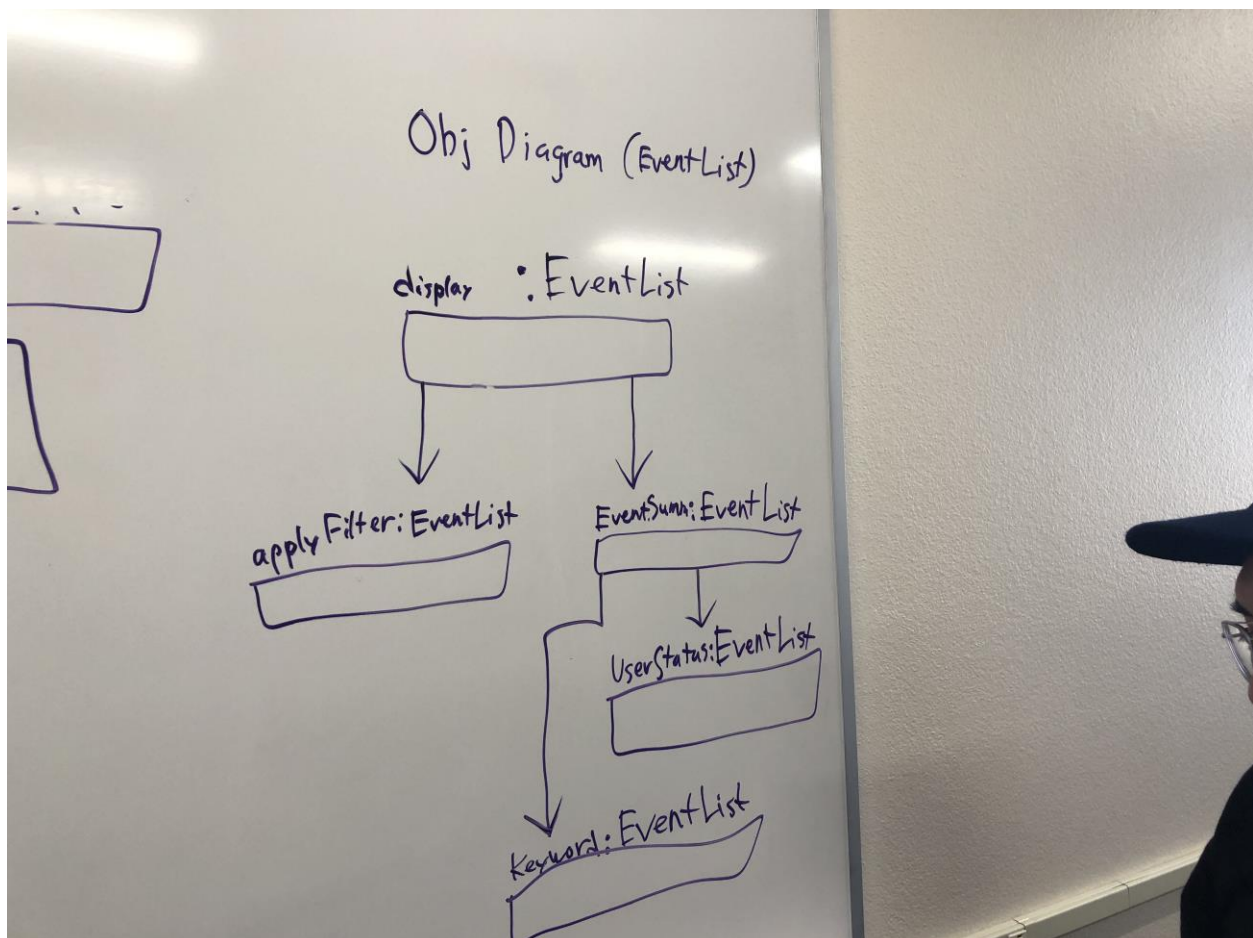
CS 441 Midterm Personal Project Write Up

What I've self-learned are how to break down a proposal into use cases based on user needs, prioritizing them in order of necessity and ability to be implemented. I, along with my groupmates, each formulated a set of use cases for our event manager then compiled our sets into a comprehensive list. We then discussed which use cases took priority based on what we defined our minimum requirements to be. I also learned how to determine the class structures for different aspects of our program and the relationships between those classes. We formed classes based on the four most important use cases for our event manager program: user logging in and out of the system, displaying events to the user, searching and/or filtering displayed events based on user input, and the user navigating the list of displayed events. Once the classes were established, we set about establishing the relationship each class had to one another based on the UML lecture slides. In this case, the User class and EventList class had an aggregate association while UserPreferences had a composite association with the User class. The EventList also held aggregate relationships with both the EventInfo class and the SearchFilters class.

I have contributed to the group proposal the initial use cases of event calendars for users, user-to-user synchronization of event calendars, and management of multiple calendars for a user. I formed the object diagram for the EventList class and the first draft of the sequence diagram for our event manager. I assisted in the coding of event manager class used to display events. The use cases I have developed are the User Calendar, user-to-user synchronization (which we did not implement), and management of multiple calendars (which was also no

implemented). The UML design we chose for our event manager was a structural design primarily made up of aggregate associations; the User class interacted with the AccountPreferences class and the EventList class while the EventList class pulled information from the EventInfo class and adjusted based on information from the AccountPreferences class.

Object Diagram for the EventList class



UML Structure

For Use cases:

13 (login/out)

10 (search/filter)

7 (display event info)

14? (navigate event list)

