Mariam Gogia
Results Doc

1. Design document made implementation easier because it helped me understand the patterns. I struggled to understand how the visitor patter was working and also how I could combine the visitor pattern with composite pattern. Working on UML diagrams made that clear for me and therefore, greatly helped me with implementation.
2. My class diagram was focused on the Authentication module itself and not on the interactions between the modules. For this reason, I did not spend enough time understanding how Authentication module would integrate with the Controller module. As a result, I ended up in few dead ends during the implementations. I believe spending more time in understanding how I would integrate this module with the rest - would have made my design better.
3. Yes, I had a few mistakes in determining relationships between module objects and my teammates helped me correct it.
4. Programming wise I found the integration with other components seamless, however, once I got to the Controller I discovered some shortcomings in logic which was too late to undo. I could have benefitted from more specific or extensive requirements regarding how to integrate the module with the Controller.
5. We had a Zoom session and no written reviews were provided. Comments from the session: Mildred decided to not implement the patterns as per requirements and decided to go about the problem in her way. My biggest comment about her design was on that mostly. I encouraged her to apply the visitor and composite patterns. She also used a factory pattern to create objects in the module - the reason behind it was unclear to me and I told her to ensure whether it was appropriate to use the factory there. Pawel had identified the relationships between some class objects wrongfully which we discussed and fixed. Also, he had created separate classes to create each project (similar to command pattern) and they were implementing Visitor interface. I advised him to change that part of the design, and we spoke about visitor pattern as he had hard time understanding how to apply it to the project.
6. Mildred pointed out that I had identified some dependencies wrongfully and that I had no relationship between the User and the Token. At a time, my diagram did not show lists/maps where I would keep track of defined objects and Pawel helped me understand which objects I may need.
7. I corrected my relationships between the classes and added list/map objects to my implementation. I added a few helper methods and toStrings that are not on the diagram. I

also ended up creating Visitable interface from which I implemented acceptVisitor method for my visitable classes.