Assignment #4 Writeup
Brandon T. Wood
CSCI E97
November 17th 2018

Did application of the design patterns help or hinder your design and implementation? Please explain how.

They greatly hindered my ability to solve the problems at hand. They mixed implementation details with required specifications. A better approach would have been to solve a problem the best way I could with my knowledge of Design Patterns instead of being forced to work around them like constraints.

How could the design have been better, clearer, or made the implementation easier?

Well spreading the requirements over three documents, multiple lesson videos and not including a test file for this assignment greatly increased the challenge of coming up with a decent design. I would say the test file is the single most important file to include for rendering a good design. For each assignment a student must first pull the actually useful information out of these documents, then follow their restraints while trying to create a program that will ingest and output the desired test file.

During the previous three assignment I would design something that seemed as if it would work great, but then while trying to implement it would basically have to rewrite my entire design to make it fit the test file. This time I decided to wait until the test file was released for it before starting to save on this rewrite and design with the full specifications. But to my confusion no test file was ever produced, instead we were to reuse the last assignments file which did not really prove anything about the new functionality that I could understand.

Anyway, I think without the forced need to use the Design Patterns I could have used a lot of the same code that went into the model with the knowledge engine and produced the same functionality in half the lines of code and in half the time it took me.

Any implementation changes that you made to your design and how they continue to support the requirements.

Though it was one of the main focuses of this assignment, I found that after I completed my design that there was no room to have any function using the Visitor object. It was requested that we use it for credentials or permissions, but since these were both Composites I didn't see what a visitor could do. One simply traversed the composites to find out if any of the credentials and permissions matched what you had been handed.

Though let us consider it a trade for another pattern, in order to implement the tokens in front of the methods of the HouseMateModel I used a Decorator Pattern like behavior of the SecuredHouseMateModel class to eliminate changing the underlying HouseMateModel that it inherits from. Another slight was implementing the SecuredHouseMateController which I believe you wished to implement by having each rule check with the entitler before acting.

But doing so would have caused me to have to rewrite my entire controller service and since the Controller has admin permissions I decided to just check the perms before running the rules and let the rules use the underlying insecure methods of the HouseMateModel to operate. Additionally there are a handful of features that I left stubs for but could not complete fully as without a meaningful test file could only guess at the implementation you would take. This being the VoicePrints for authentication with the Ava service. I assumed based off earlier lectures that the test file would include voice print information, but since no test file materialized I felt at a loss to produce anything.

Is the design process getting easier?

I think this class is actually driving me away from design-first as a whole. Everytime I design something before I build it I encounter massive problems with what looks like a solid implementation. I would relate it like a cartographer trying to draw a map before he's walked the land. I think I will stick heavily to an iteration based approach for solving problems in the future.

Did the design review help improve your design?

Not really, most of the reviews given to me in any of my code reviews are mainly about technical aspects and formatting of my document. From such a high level it's almost impossible to make any meaningful statements about the user's implementation. All of the paragraphs or words and concise looking diagrams are just not as telling as opening someone's source and playing with

it. Code is where the truth is because it's what actually runs, but the design document is more like managerial and marketer fluff.

Your comments for your review partners

My Review of Nick's DD:
1. Are you allowed to use diagrams directly from the requirements sheet?
2. Your UML, class dictionary and Sequence diagrams look great! The alternating dark and light columns greatly improve readability.
3. Why is everything in italics, makes it hard to read somewhat?
4. I would be very interested to read your other sections once they are filled out.

My Review of Drew's DD:
(He never got back to us with a document before I submitted this.)

Comments from peer design review and optionally the functional review

Nick's Comments on my DD:
1. The introduction, use case and requirements sections are well thought out and have sufficient detail.
2. The UML seems to depict usage of the Abstract Factory but only the Security portion is captured. Do you have plans to abstract this further to include other entities?
3. I don't see usage of the Visitor pattern in the UML. How will you determine composition/mixing of roles and permissions.
4. HouseMatecontroller should be depicted as a package, not a class.
5. HouseMateModel should be depicted as a package, not a class.

Drew's comments on my DD:
Introduction: Might want to leave out "This service monitors the activity of sensors and appliances" refers to the Controller Service.
Overview: Drop in the House Mate Architecture diagram, if you haven't done so already.
Requirements: Maybe mention something about the ability to login/logout using voiceprint or username passwords?
Use Cases: (I am not totally sure on this) but, should you have the arrow pointed to the Entitlement Service box?
Where is the point of entry for the admin into the system? The CLI? – do you have on in the entitlement service as well?
Class Diagram: Nice – good inclusion of the Composite and Factory patterns.

Class Dictionary: Good
Exception Handling: Good
Testing: Just one test : FunctionalityTestDriver? If so, maybe throw in one more for good measure. Possibly one that highlights extensibility of service?
Risks: Good.