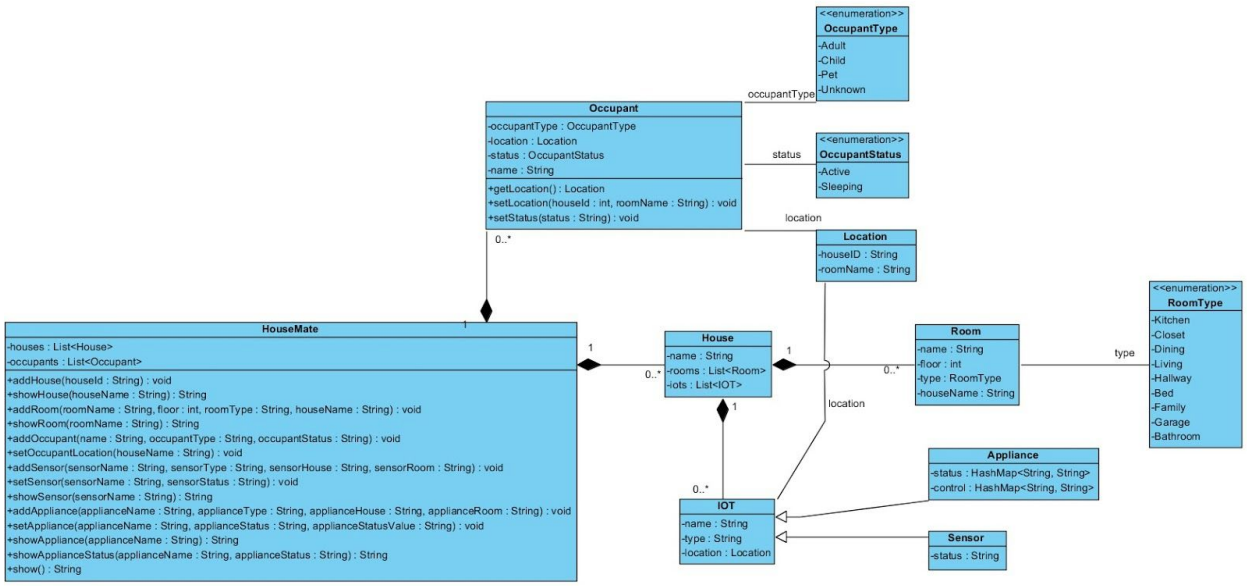
Kush Patel

Assignment 2 Analysis

Below is my first attempt at the design document, the one I received feedback on.



Some feedback I received:

- Pull some methods out of HMMS, or at least have the HMMS be a pass through for objects like room / sensor. Then these methods can properly manage the attributes that belong to them.
- Also, that way you can error handle at the House level or lower and pass the error back up instead looking for where the error arises from in the HMMS.
- You don't need to include all getters/setters in a UML diagram.
- For printing the output, it makes sense for the HMMS to do it since the method is called "show"; it would be weird if it returned a value and didn't actually print it out.

Changes from peer feedback and my own design analysis

- Removed the enums, too much to maintain + if they increase complexity, the overhead work increases from just 4 OccupantTypes to 10 or however it changes. Same for the other enums.
- I liked extracting the addRoom feature to house as rooms are an attribute of the house. I however didnt continue that idea for the IOTs because I wanted to see the difference in

design / implementation and see if either method was actually better for the system / code.

- I added in the Controller to the design.
- I also added custom toString() methods to all the main objects which allowed me to just toString() and concat each component when doing “show” methods. Example:
    - showHouse() = house.toString + each(room.toString) + each(IOT.toString).
    - or just showSensor(smoke_detector) = get(smoke_detector).toString

Design Document vs Implementation

Yes, the design document definitely made a bulk of the code much easier. Once I had updated my design from the feedback, I simply went class by class in the design essentially “copying+pasting” the design diagram into Java code. However, the document didn’t account for all the logic for the command parsing or exception handling and which error to account for. This definitely did take a large part of the development process, which makes it feel like the design didn’t help as much as it should have. I think the design is just a high level skeleton of the code you are going to develop, and I feel like I wrote it twice; once for the design document, once actually implementing the code.

So I definitely think the design needs to include exceptions, at least a list of them; it will simply help creating all of them when building the skeleton of the code. Also, detail logic / design could be beneficial to help the implementation of the methods. The method description says what you want to accomplish, but not necessarily how it should be accomplished. If that was included, I could have gotten some feedback on my code implementation; though the logic is interesting, not sure if it’s the best implementation.

The design review definitely helped a lot in the development process. Both reviewing others and seeing how they are solving a similar problem, as well as getting feedback on your own. In my current professional career, my company does pair programming, so I’m constantly getting feedback and between 2 people we always seem to cover our basis and not miss anything. So peer review is always important to help account for all discrepancies and finding design flaws.