From: **Iida, Takayuki** tiida@fas.harvard.edu 🖉
Subject: RE: Class Diagram
Date: October 6, 2015 at 8:19 PM
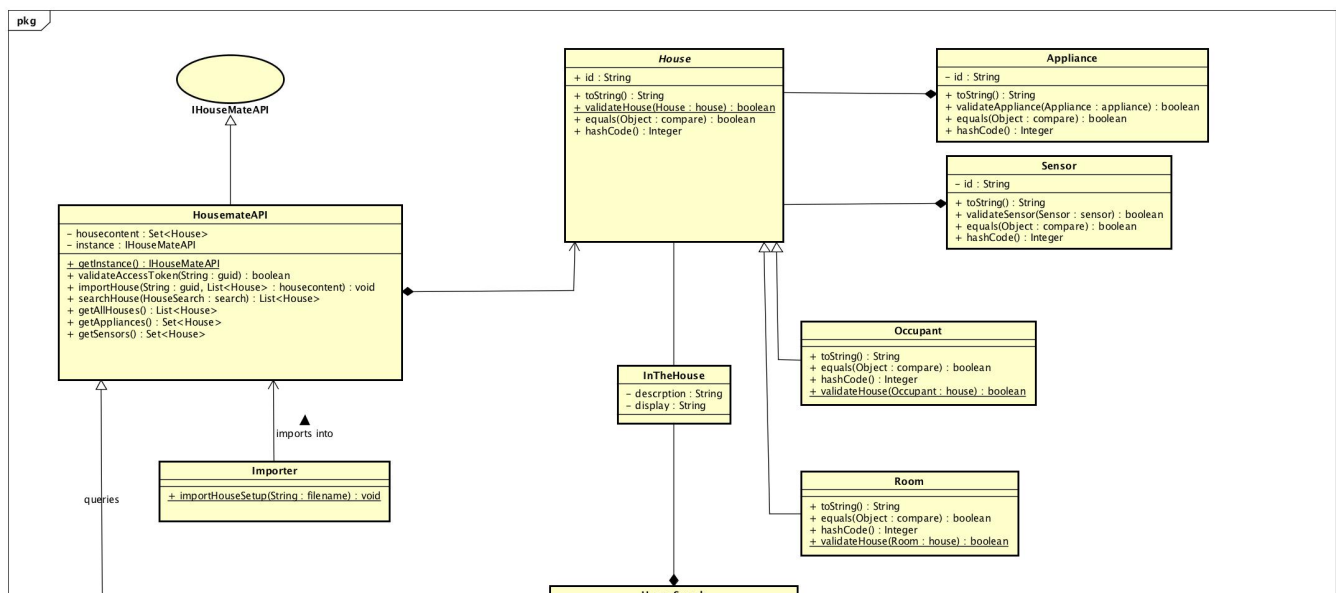**To:** Gerald Trotman geraldtrotman@fas.harvard.edu

IT

Thanks Gerald. Keep in mind that I'm going to try to comment more as a peer reviewer than as faculty for this review. This is so just as with a fellow student reviewer, you have an opportunity to use your own judgment to consider your reviewer's views critically, even skeptically, if necessary.
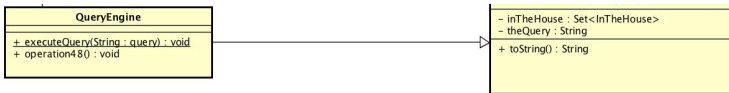
1. I notice that you're using generalization arrows in places where you probably mean to use a different type of arrow. For example, I'd assume that an Occupant doesn't inherit from House.

2. Some of the arrows may be backwards - for example, the arrows on Sensor and Appliance.

3. I don't see a way to tell what room an Appliance is in, or what floor a Room is on. But I do get that you've just started!

4. I don't understand what the "InTheHouse" class is.

5. Right now, the class diagram seems to have a flat set of relationships, where the House and the HousemateAPI holds everything and doesn't delegate ownership to other classes. Maybe it could be more hierarchical, with appliances and sensors belonging to Rooms instead of the entire house. Unless - was there a requirement to have sensors, say, security cameras and things outside of a room, such as mounted outdoors, in in hallways? Guess I'll have to review the requirements again!

6. Appliance and Sensor have a lot in common. They could probably use a common IOT device superclass.

From: Gerald Trotman [geraldtrotman@fas.harvard.edu]
Sent: Tuesday, October 06, 2015 7:55 PM
To: Iida, Takayuki
Subject: Re: Class Diagram

Hey Takayuki,

Whoops! Sorry about that. Here you go, I am sending it as a JPEG:

QueryEngine

+ executeQuery(String : query) : void
+ operation48() : void

− inTheHouse : Set<InTheHouse>
− theQuery : String

+ toString() : String

I'm working on that design document now to send you as well.

Cheers,
Gerald