A Not So Smart, Smart-Home.

Guillaume Rousseau V00875834
Scott Hannan V00874232

11/16/2018

Our solution to the Smart-Home assignment was to start by implementing a “thick model”, the hub. The hub was the home of all device registration, user registration, as well as all un-registration and client notification/logging (which is not yet implemented). The 1st step we took was mapping out the UI and how we wanted the system to look and operate and from there we were able to begin writing test cases and the UI. Once we had the framework we decided on how to handle handling events. Inside every Device class (Camera, Thermostat, Lightbulb, SmartPlug) there is a method handle(), each device as well has Buttons associated with them this made handling events device specific and very easy to do. The design is considered a thick model because most of the logic and the control is given to the Hub, we chose a thick model because we found it to be more maintainable, however, this is at the expense of some loose coupling. Our Class diagram shows that we also have controllers linked to UI rather than to devices, we chose this as with such a small application it is easier to handle the events in scenes rather than based solely on their devices.

TO RUN: run main as a java application

To run our program as an admin:
username: scott
password: admin

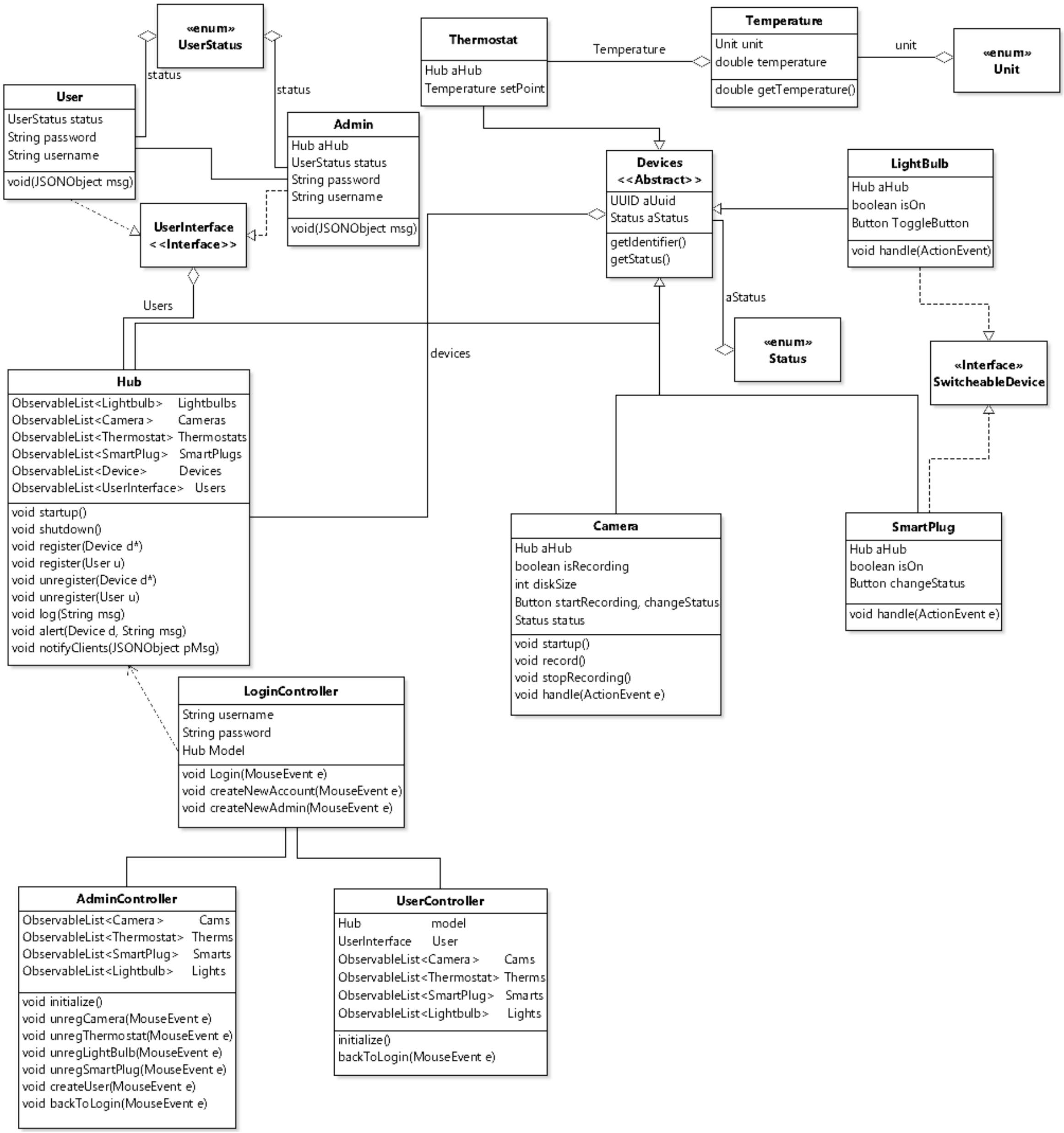
To run our program as a basic user:
username: gui
password: user

To run our Test suite:
Run as a Gradle project (build.gradle)

The tests that we have implemented demonstrate a mix of UI testing with testing avoiding the UI, we did this to make sure the framework of the application was well made as well as the UI.

17(+2) Acceptance Tests:

A1		C1	E1	F1
A2	B2	C2	E2	(F2
A3	B3	C3		F3)
A4	B4			
A5		C5		
	B6			
A7				



A class diagram of the system, demonstrating the thickness of our main model, hub.