

CSCI E-97**Assignment 3****Due: Tuesday, 10/24/2017**

Introduction

In this assignment you will create and implement a design related to the implementation of a home automation system, and in doing so explore the Internet of Things (IoT).

The [Internet of Things \(IoT\)](#) is an environment in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Overview

For this assignment, you will design and implement one component of the House Mate System, the House Mate Controller Service.

In the design portion of the assignment, you will create a software design document that satisfies the House Mate Controller Service requirements. You will use a UML class diagram and class dictionary to document the required classes, including the attributes, associations, and methods for each class (similar to the design provided in assignment 1).

Include a use case diagram that shows the use cases supported by the Controller Service.

Provide one or more sequence diagrams that describe the interaction between the Controller Service and the Model Service.

Utilize the Knowledge Graph implemented in assignment 1 for managing the location and state of the occupants.

Utilize the Command Pattern for handling events between the HouseMate System and the Controller. Support queueing and logging using command objects.

In the implementation portion of the assignment, you will implement your design and test your solution.

You will have 2 documents as input to your design:

- Requirements Document describing the functional requirements for the House Mate Controller Service.

- System Architecture document that provides a high level description of how the House Mate Model Service component will fit in the overall House Mate System.

In addition, a software design template is provided for you to use as base for your design.

Development Process

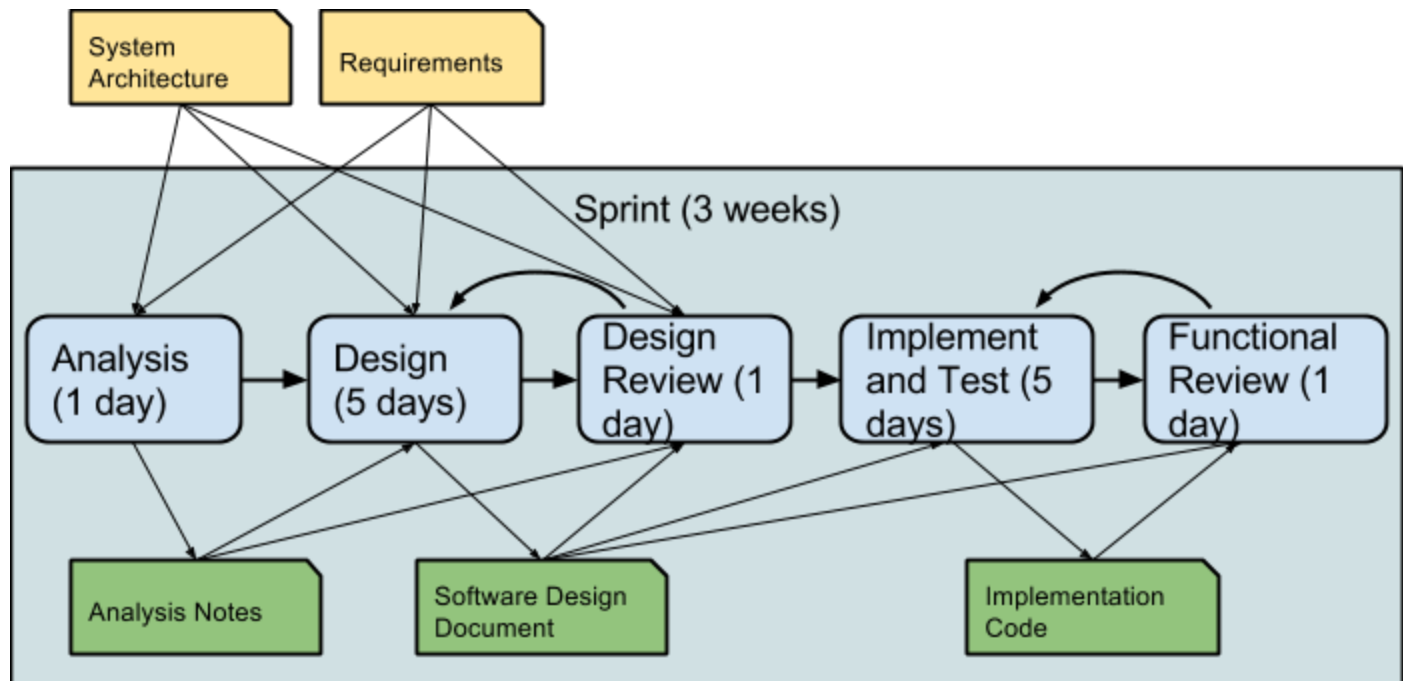
This assignment will simulate a 3 week Sprint. A sprint is a time-boxed unit of development as defined by Scrum. (For more details on Scrum:

[http://en.wikipedia.org/wiki/Scrum_\(software_development\)#Sprint](http://en.wikipedia.org/wiki/Scrum_(software_development)#Sprint))

For this assignment, the class and teaching staff will comprise the Scrum team. Each of the students will be developers assigned the same task (design and implementation of the Product API). Note that normally a Scrum team would be about 8 people, and members would be assigned different tasks to maximize the utility of the team. Also, we will forgo the daily stand ups that normally start the Scrum team's day.

The assignment is framed within the context of a sprint to help demonstrate how formal software design can be successfully applied within a fast-moving agile development environment, ultimately increasing the velocity of the development team and improving their quality of the output.

The following diagram provides an overview of the development process that we will use. This instance of the development process requires 13 days, with 2 days buffer.



Assignment Notes:

The goal of this assignment is to help you become familiar with the process of design and implementation within the context of a collaborative agile development environment.

Another focus of the assignment is to create a design document, leveraging the Object Modeling and Class Diagram techniques recently covered in lecture.

You should implement the classes as defined by the class diagram and class dictionary specified in your design document. All classes except for the TestDriver should be defined within the package `"cscie97.asn3.housemate.controller"`.

Implement a TestDriver class to load in the sample house mate configuration and run the provided sample queries. Print out the results of each of the queries.

When implementing your design, you may vary from the design, but you should document the changes, provide justification for your changes and describe how your changes continue to support the requirements.

Remember to use Java doc to document class and method headers. Add java comments inline where appropriate to explain code logic.

Sample Data

The following input file will provide data for your TestDriver class to load and run.

housesetup.txt house configuration data and commands

What To Turn In

You'll turn in a zip file containing

- Your source code (no .class files)
- Your data files
- Sample output
- Your design document (in pdf format)
- Include a document (in pdf format) describing your results:
 - Comments from your peer design review partners
 - Your comments on your peer's designs
 - Updates to your design based on the peer design review or implementation
 - Did the design document make the implementation easier?
 - How could the design have been better, clearer, or made the implementation easier?
 - Did the design review help improve your design?
 - How did you find the integration of the components

We should be able to unzip your file into a directory, then cd into that directory and compile your program with the command.

- `javac cscie97/asn3/housemate/model/*.java cscie97/asn3/housemate/controller/*.java cscie97/asn3/housemate/test/*.java`

We should be able to run your program with the command

- `java -cp . cscie97.asn3.housemate.test.TestDriver housesetup.txt`

where housesetup.txt is a list of facts used to configure the house and set sensor and appliance status and control values.

Caution: When you believe you're done, try zipping your files, then unzipping them into a totally different directory and following the steps above. In other words, test your packaging before you submit your assignment.

Directions for submitting your solution and a grading sheet specifying the criteria for grading this assignment will be posted on the course website.