CSCI E-97
Assignment 3
Due: Wednesday, 10/28/2015

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 capture 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 sent from the HouseMate System to 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.

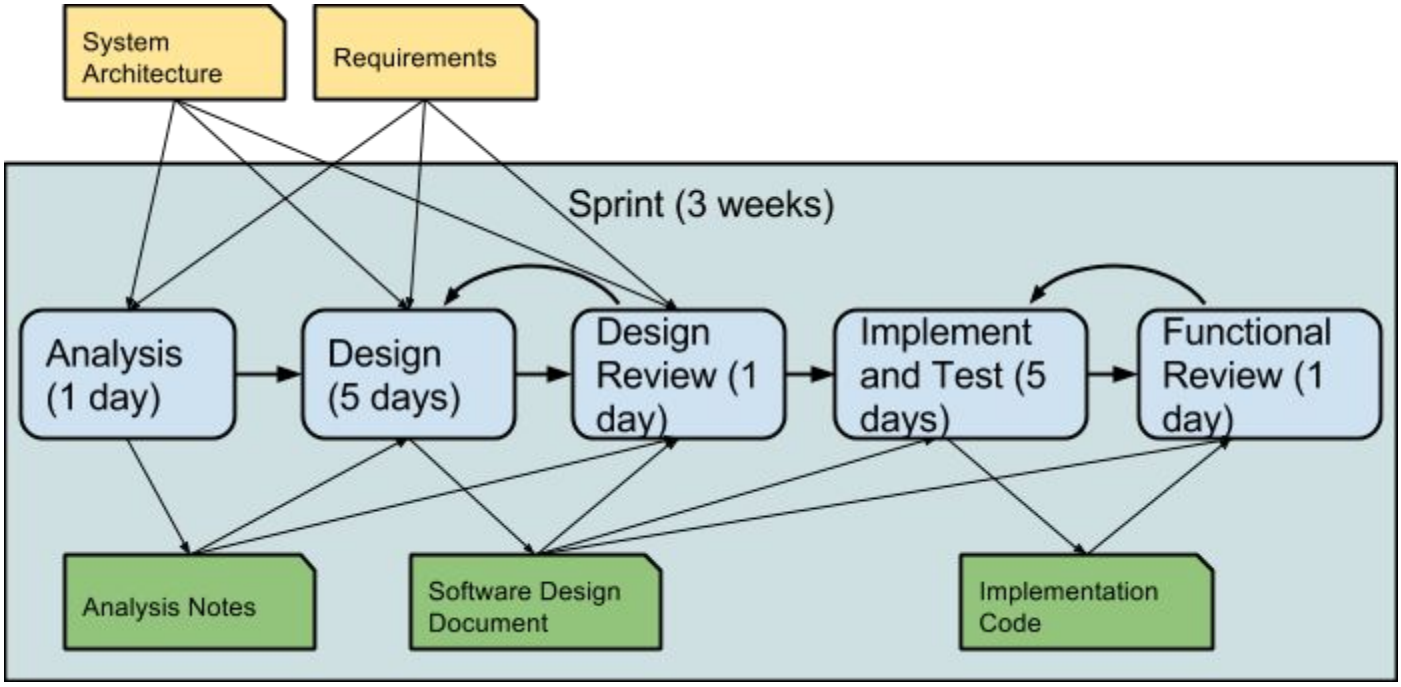
Development Process

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)

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.



The following sections describe each step of the development process.

Analysis

The Analysis step is a team effort to ingest and understand the requirements for the the given task. The product manager is included to help address questions about the requirements. Once the requirements are clear, the team can brainstorm about the best approach for solving the problem. There are usually many ways to solve a problem, and the most obvious approach is generally not the best, so it behoves the team to consider different approaches to solving the problem, including what has worked well or not so well in the past.

Another thing to consider when reviewing the requirements is the scope of work required to address the requirements. If it appears that the scope is too great, then the requirements may need to be further broken down into more digestible pieces (i.e. divide and conquer).

A second input to the Analysis step is the System Architecture. The System Architecture provides a high level overview of how this component fits with other components. The System Architecture should also provide guidance on what technologies to apply in the solution.

The goal of the Analysis step is to understand the customer requirements and to build a consensus on best approach to help guide the next step which is Design. It is good practice to capture the outcome of the Analysis in a brief set of notes.

We will have an in-class analysis discussion to understand the Requirements, System Architecture, and different approaches to solving the problem.

Design

The goal of the Design step is to create a design document that addresses the requirements, fits within the System Architecture, and generally follows the consensus of the previous Analysis step. The Design step requires further analysis of the requirements and drilling down into the details of the solution.

The design should consider and address the following 3 questions:

1) Who is the audience?
2) What is the problem?
3) What do I know about the problem and solution?

The Design step is an opportunity to dive into the problem and explore solutions. Use whiteboards and teammates to test out different solutions. The UML diagramming methods that we are learning in class are especially helpful in capturing these solutions. Use the camera of your mobile phone to capture whiteboard diagrams and reference them when writing your design document.

Remember to consider and support all requirements in your solution. If you have questions about the requirements, reach out to your product manager to resolve. (The teaching staff will act as the product manager, so post questions about requirements to the assignment 2 discussion forum).

The design should also consider the provided System Architecture document. Make sure that the design fits within the guidelines specified by the architecture document, so that the resulting component solution will be able to integrate into the rest of the system.

Use the design template included with the assignment for the basis of your design document. The template includes content descriptions for each section.

Design Review

The design review is a critical step in the software development process as it provides an opportunity early in the process to rid your solution of errors. The design review considers the following questions:

1. What is the problem that is being solved?
2. What are the functional and nonfunctional requirements?
3. What is the proposed solution?
4. Are all of the requirements addressed? Does the solution go beyond the scope of the

requirements, and if so, why?
5. What are the details of the solution?
6. Is the design clearly written and easy to understand?
7. Are there oversights or errors in the proposed solution?
8. Can the solution be simplified? (i.e. apply Occam's razor, Less is More)
9. Are there known risks? If so, what are they? (e.g. performance or scaling issues)
10. Does the design fit within the context of the system architecture?
11. Can the design be implemented in a reasonable amount of time?
12. Does the design include a testing strategy?

The design review is usually conducted with the product manager, and fellow developer and QA team members. The design review meeting usually takes 1 to 2 hours.

For this assignment, we will ask students to work in groups of 3 and conduct peer design reviews. The feedback from the design review should be considered and incorporated into the design document where appropriate. Note that we expect each student to write their own design.

Implementation

Once the design review is complete, and feedback from the design review has been incorporated into the design, it is time to Implement a solution based on the design.

This step incorporates implementation of the design and testing of the solution. So as well as writing the code, the developer should also incorporate working test driver with the solution.

Because of the upfront thought that went into the analysis, design and review, the implementation should go smoothly with a minimum of surprises. The developer, and team, can also expect that the resulting implementation will address the requirements and fit within the overall system architecture. All of this provides a high degree of confidence that the desired solution will be delivered on time.

Functional Review

The final step of the development process is a Functional Review. In a team environment, this is usually conducted with the product manager, developer and QA team members. The functional review takes 1 or 2 hours.

The focus of the Functional Review is to validate that the solutions has been implemented

according to the design, that the implementation addresses the functional and nonfunctional requirements, and there are no obvious errors in the solution. This is a detailed review intended to validate the solution and raise the bar in terms of expectations for quality.

Any issues discovered should be documented, and addressed by the developer.

Once all discovered issues have been addressed, the solution is ready for integration testing and deployment.

For this assignment, the Functional Review will be the grading of your design and implementation by the teaching staff. However, you are welcome to perform your own functional review with your peer from the design review and make adjustments prior to turning in your assignment.

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 peer design review and optionally the functional review
    - Any changes that you made to your design and how they continue to support the requirements
    - 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.