

CSCI E-97**Assignment 2****Due: Wednesday, 10/5/2020**

Introduction

In this assignment, you will create and implement a design related to the implementation of a city 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 Smart City System, the Smart City Model Service.

In the design portion of the assignment, you will create a software design document that satisfies the Smart City Model System 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).

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:

- The Smart City Requirements Document describing the functional requirements for the Smart City Model Service.
- The System Architecture document provides a high-level description of how the Smart City Model Service component will fit in the overall Smart City System.

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

Development Process

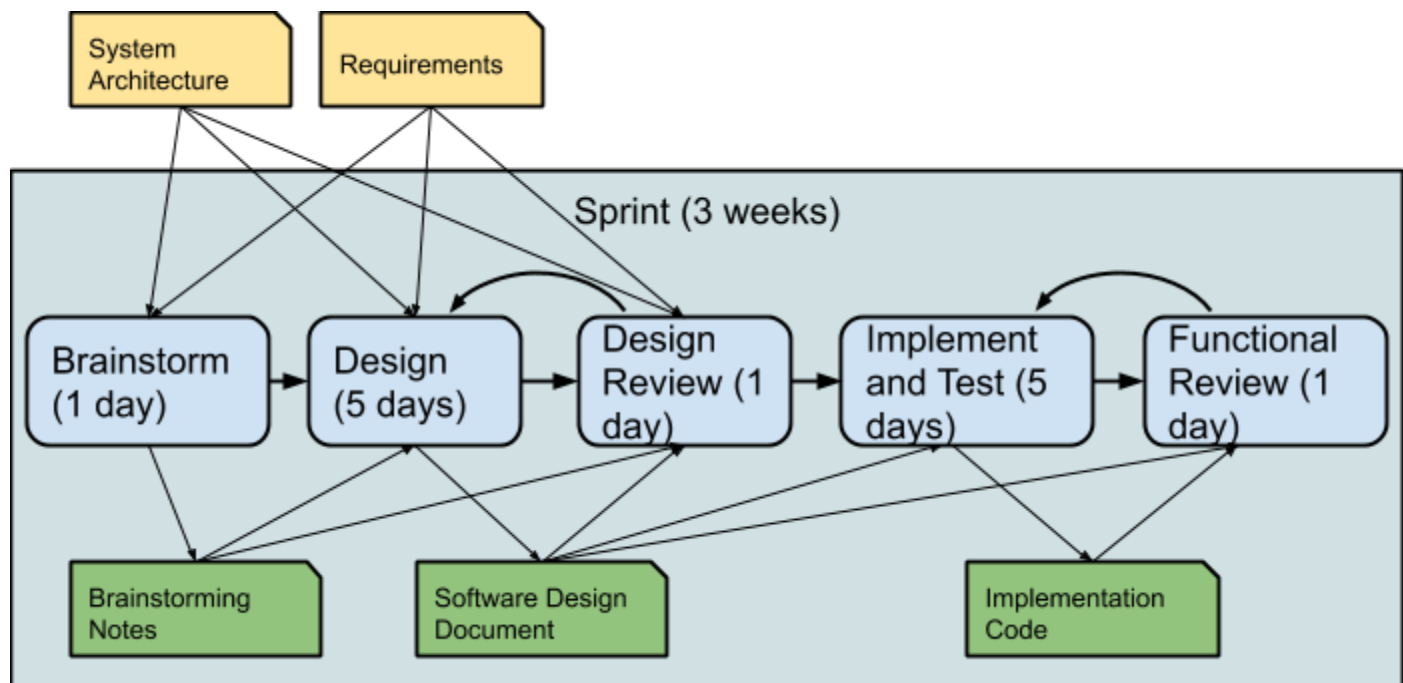
This assignment will simulate a 3-week Sprint. A sprint is a time-boxed unit of development as defined by Scrum. (For more details on Scrum:

[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 team's productivity. 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 the quality of their 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.

Brainstorming

Brainstorming is a team effort to review and understand the requirements for 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 consider different approaches. Use the experience of the team to help choose the best solution.

Consider 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 Brainstorming step is the System Architecture. The System Architecture provides a high-level overview of how this component fits with other components. The System Architecture also provides guidance on what technologies to apply.

The goal of the Brainstorming step is to understand the customer requirements and to build a consensus on the best approach to help guide the next step which is Design. It is good practice to document the outcome of the brainstorming session.

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 Brainstorming session. The Design step requires further analysis of the requirements and fleshing out the details of the 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. When writing the design document, 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?

Design Review

The design review is a critical step in the software development process as it provides an opportunity early in the process to correct errors and oversights. The design review considers the following questions:

1. What is the problem that is being solved?
2. What are the functional and non-functional 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, students will work in groups of 3 to 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 the solution based on the design.

This step incorporates the implementation of the design and testing of the solution. So as well as writing the code, the developer should also incorporate a 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 to 2 hours.

The focus of the Functional Review is to validate that the solution has been implemented according to the design, that the implementation addresses the functional and non-functional 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 peers 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 the 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.smartcity.model”

Implement a TestDriver class to load in the sample city 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 Javadoc 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.

smartcity_model.script city model 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 results document (in pdf format) describing your results:
 - Did creating the design help make the implementation easier?
 - How could the design have been better, clearer, or made the implementation easier?
 - Describe implementation changes that you made to your design and how they continue to support the requirements
 - Did the design review help improve your design?
 - Comments about your design from your peer design review partners
 - Comments provided by you for each of your peer design review partners

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/smartcity/model/*.java cscie97/smartcity/test/*.java`

We should be able to run your program with the command

- `java -cp . cscie97.smartcity.test.TestDriver smartcity_model.script`

where smartcity_model.script is a list of CLI commands used to configure one or more cities.

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.

Submit your Assignment 2 solution online through Canvas.