SE4950 – Fall 2020
Project 2 – Secure Design Patterns

Background

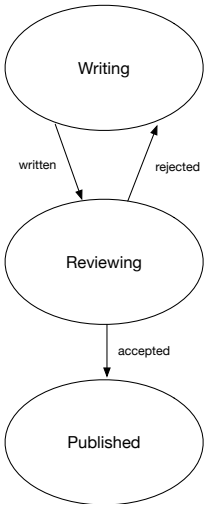
The purpose of this project is to provide you hands on experience in developing a design using secure design patterns. The project will cover using two different secure design patterns that we have covered in lecture. We will have time in class on Friday 10/2 and Monday 10/6 to work on the project.

In doing the project, you will be producing UML class diagrams (you are free to produce other types of UML diagrams if you believe they will help document your design). To produce these, there are several options:

- Draw.io – http://draw.io - Can be used via the web or downloaded to your computer. Seems to be the most versatile of the editors.
- Violet UML Editor - https://sourceforge.net/projects/violet/files/violetumleditor/ . Works but is less intuitive.
- Umlet UML Editor – http://umlet.com - Least intuitive to use of the 3.

Part 1 - State Machine

For this portion of the project, we are going to use the basic State pattern, just to get out feet wet with the basic patterns first. We will implement a system that has documents that conform to the following state machine:



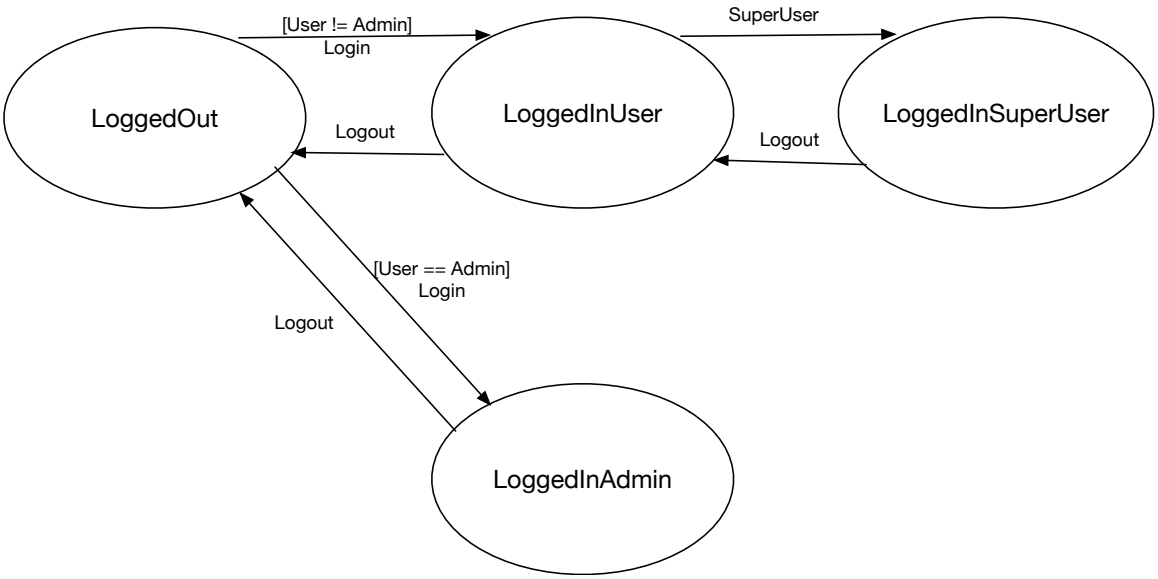
Part 1 - Assignment

1. Implement the system using the State machine pattern we talked about in class, drawing the UML class diagram for the state machines portion of the system.
2. Code the pattern objects in Java. As we will not actually be running the system, I'm will not be grading based on a clean compile, but instead on an understanding of how the pattern and the state machine works, looking for logic in the code methods.

Part 2 - Secure State Machine

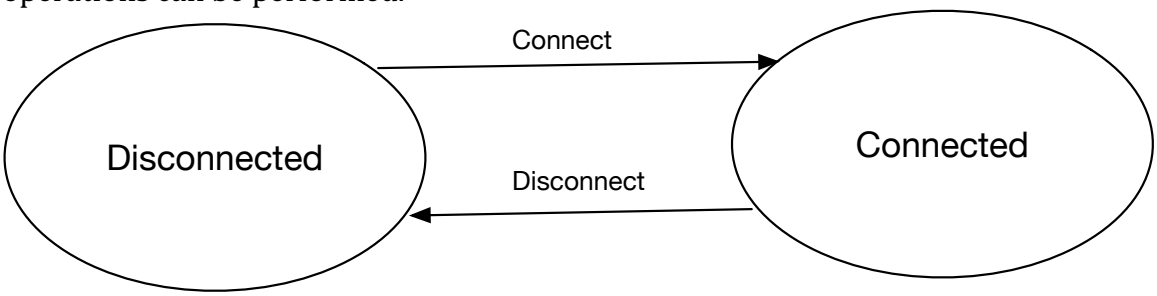
For this portion of the project, we are going to use the Secure State pattern to implement security and user state machines to implement a system with the following state machines:

Security State Machine:



User Level State Machine:

The system controls an industrial robot, and must connect to the robot before other operations can be performed.



There are 9 commands that the user can give to the system:

login password
logout
superuser password
connect
disconnect
op1
op2
op3
op4

The security restrictions placed on the system say that the connect event should only be forwarded to the user state machine if we are in the LoggedInAdmin or LoggedInSuperUser states. Additionally, op3 and op4 should only be performed if the User Level state machine is connected.

Part 2 - Assignment

1. Implement the system using the Secure State machine pattern from the SEI Secure Design Patterns report, drawing the UML class diagram for the state machines portion of the system.
2. Code the pattern objects in Java. As we will not actually be running the system, I'm will not be grading based on a clean compile, but instead on an understanding of how the pattern and the state machines work, looking for logic in the codes methods.

Part 3 - Secure Strategy Factory

Imagine a system that handles medical patient records and bills. The records are compressed and encrypted, using different algorithms and different keys for different sections of the record. The system is designed to allow different users to extract copies of patient records and patient bills based on their user credentials.

In this system, there are four classes of users:

1. Patient
2. Primary Care Physician
3. Consulting Physician
4. Administrative Professional

There are three levels of patient record that can be produced (extracted and decrypted from one stored record):

1. Patient level (available to the Patient)
2. Primary Care level (available to the Primary Care Physician)
3. Consulting level (available to the Consulting Physician)

There are two levels of patient bills that can be produced (extracted and decrypted from one stored record):

1. Patient level (available to the Patient and Administrative Professional)
2. Medical Staff level (available to the Primary Care Physician, Consulting Physician, and Administrative Professional)

Part 3 - Assignment

1. Implement the system using the Secure Strategy Factory design pattern from the SEI Secure Design Patterns report, drawing the UML class diagram for the pattern portion of the system. (Hint, you are going to have to have two strategies, one for patient records and one for patient bills).
2. Code the pattern objects in Java. As we will not actually be running the system, I'm will not be grading based on a clean compile, but instead on an understanding of how the pattern and the security logic work, , looking for logic in the code methods.

Files

Submit a version of your UML diagrams exported from your tool of choice to PDF/PNG format, along with any Java files to Blackboard.

Grading

The project will be worth 50 points, and graded as follows:

- 5 points – UML Diagram of State system.
- 5 points – Code of State objects.
- 10 points – UML Diagram of Secure State system.
- 10 points – Code of Secure State objects.
- 10 points – UML Diagram of Secure Strategy Factory system.
- 10 points – Code of Secure Strategy Factory objects.