Metro State University
ICS 372 Object-Oriented Design and Implementation
Spring 2023

Group Project 2

Due: 11:59 PM on April 28, 2023

Points: 100

Goals:

1. Perform finite-state modeling techniques to come up with the state transition table and diagram.
2. Identify the classes in an FSM-based system with minimal conditionals.
3. Use the Unified Modeling Language to document work.
4. Work in small groups.
5. Implement a design utilizing structures such as classes and interfaces.
6. Employ Java coding standards.

The Problem

You need to use the FSM approach to model, design, and implement a train that goes in a loop around the terminals of an airport. Each terminal has a station at which the train stops. As you will see, the number of stations is irrelevant. The system works as follows.

After starting from a station, the train accelerates for 6 seconds, before it attains full speed. At some point, perhaps after attaining full speed, the train keeps running until it gets a signal that it is approaching a station. At that time, it decelerates. The signal that the train is approaching a station might come even before the 6-second acceleration period ends; if that happens, the train starts decelerating.

At some point after decelerating, the train receives a signal that it has arrived at a station. The train immediately stops.

1 second after the train stops, the doors of the train start opening. It takes 4 seconds for the doors to open fully. The doors remain in the fully-open position for 30 seconds. After that the doors start closing. It takes 5 seconds for the doors to close.

As the doors are closing, there could be obstructions, such as travelers walking through. On detection of such obstructions, the doors reopen: the reopening time is dependent on how far the door had closed. For example, if 3 seconds had elapsed after the closing process started, the time to fully reopen would be 3 * 4 / 5, that is roughly 2 seconds. (We will drop the fractions.) After fully reopening, the doors will reclose after 8 seconds. The doors may again encounter obstruction and the process could repeat.

After the doors are fully closed, the train starts moving (accelerating) after 3 seconds.

You must implement a controller for the above system using the FSM approach. The design and implementation must be consistent with the approach taken in the Microwave case study. There should be no conditionals in your implementation, except for unavoidable situations such as for implementing singletons.

Things to be Submitted

You need to submit two files: one for analysis and design, and a second one for implementation.

Analysis and Design

Analyze and design the system using the FSM approach. Submit a single PDF document that contains all of the following.

1. The state transition table. This must be in the standard format, with states appearing in the rows and events shown in the columns. The states and events must be properly named in the table, reflecting their role in the application. Difficult-to-read tables will not get much credit and hand-written tables will be ignored.
2. The state transition diagram. Draw the diagram showing all the states and transitions. For simplicity, do not show self-loops: that is, do not show transitions that take the state to itself. The diagram must use the standard notations. The states and events must be properly shown in the diagram. Difficult-to-understand diagrams will not get much credit and hand-drawn diagrams will be ignored.

You can talk to me to get any clarifications you need, especially with respect to the requirements.

I will also ignore any document that is in any other format (Word, GIF, JPEG, etc.) If you submit multiple PDF documents, I will choose one of the PDF files and ignore everything else. Be sure to ensure that the information is all in the portrait mode if at all possible. But it is more important to not have any part of the information cut off or unviewable in any way.

Implementation

Implement the design following sound object-oriented principles. Submit the entire application as an Eclipse Java modular project using JavaFX. All code (classes, interfaces, constructors, and methods) must be properly formatted and documented. Use packages to clearly group your classes. The documentation, naming conventions, and code formatting must follow the coding conventions we have discussed before. Document and lay out your code properly as specified under the CodingStandards.pdf document.

I will provide no debugging support. You must resolve such issues within the group.

Grading

Your assignment will be graded as written in this section.

Design:

The states and events must correctly reflect the behavior of the application and be such that they promote clean implementation that adheres to sound object-oriented principles.

Correctness of states and events and state transitions will count as below. Note that if you don't have the correct states and events, you will also lose points for correctness of state transitions.

Correctness of States	10
Correctness of Events	10
Correctness of State Transitions	20

State Transition Table (10 points)

State Transition Diagram (10 points)

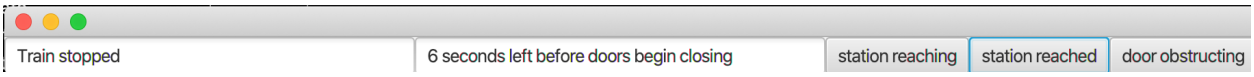
Implementation (40 points)

Grading Issues Related to Group Work

Usually, all members of a group get the same grade. But I have had multiple groups disputes, which have often made grading a somewhat difficult process. I am forced to reserve the right to give students within a group different grades. This will not be done capriciously; I will be meeting with every group every week, so will have at least weekly updates on how things are going on within groups, which might obviate the need to assign different grades. I will inform the student involved before I give him/her a different grade.

The GUI and Some Support Code

The GUI should look as below. There should be three buttons as indicated, an area for indicating the status of the doors, and an area for the train status.



Every student has to submit an evaluation of the group work within a week after project submission. Depending on how well they wrote the evaluation, a student will receive points between 0 and 5. Suppose that score is x. Then (5 – x) will be subtracted from the student's group project score.

Also, the group project evaluation will be a basis for assigning individual scores to group members. Blaming group members at the last minute will have less consequence than consistent and early reporting of uncooperative and irresponsible attitudes.