SYSC 3303 Project Report

Figure 1: Elevator Doors [1]

Elevator Simulator

Group: L4 G10

Authors:
Ali Fahd 101107270
Andy Ngo 101132278
Karim Mahrous 101150894
Scharara Islam 101149731

TA: Ola Mohammed

Table of Contents

1 Task Breakdown

Summary of Iterations:

Iteration 0

- Calculate acceleration, maximum speed, and the distance between the floors using the recorded data
    - Use the given recorded data and calculate for acceleration, maximum speed, and the distance between the floors

Iteration 1

- Establishing connections between three subsystem: Scheduler, Elevator and Floor
    - Create the 3 subsystems as threads

Iteration 2

- Adding in state machines for the scheduler and elevator subsystem
    - Create a state machine for the subsystems to organize the functions
    - Create multiple elevators and 22 floors to simulate Dunton tower

Iteration 3

- Establish communication between each subsystem using UDP, to allow the subsystems to run on different computers
    - Make each subsystem run separately and communicate to each other
    - Send packets and receive packets

Iteration 4

- Adding error detection and correction, will require error inputs to cause an elevator to get stuck and will need to be fixed
    - Add errors in for the elevator and fix the error after a certain amount of time

Iteration 5

- Measuring the scheduler, to get the performance of the scheduler subsystem, and have the output of the system displayed as a GUI
    - Establish a GUI to display the elevators movement around the 22 floors of Dunton Tower
    - Get the measurements of the scheduler times

Team Member Roles(Breakdown of Responsibilities):

Iteration 1

- Ali Fahd:
  - Created FloorRequest, FloorSubsystem and Main
  - Created Testing
- Karim Mahrous:
  - UML of Class and sequence Diagram
- Andy Ngo:
  - Created Elevator and Elevator_Motor
- Scharara Islam:
  - Created Scheduler Class and edited by Ali Fahd
  - README

Iteration 2

- Ali Fahd:
  - Updated source code
- Karim Mahrous:
  - Designed State Machine Sequence
  - Updated UML class diagram
- Andy Ngo:
  - Updated source code
- Scharara Islam:
  - README

Iteration 3

- Ali Fahd:
  - Updated source code
- Karim Mahrous:
  - Updated source code
- Andy Ngo:
  - Updated source code
- Scharara Islam:
  - Updated UML Diagrams
  - Created Scheduler State Diagram
  - README

Iteration 4

- Ali Fahd:
  - Updated source code
- Karim Mahrous:
  - Updated source code
- Andy Ngo:

        - Updated source code
    - Scharara Islam:
        - Updated UML Diagrams
        - Created Timing Diagram
        - README

Iteration 5:

- Ali Fahd:
    - Worked on the Report
    - Worked on the Video
    - Worked on GUI Interface
    - Created Testing
    - Presented Final Project
- Karim Mahrous:
    - Worked on the Report
    - Worked on the Video
    - Created Testing
    - Updated Code
    - Presented Final Project
- Andy Ngo:
    - Worked on the Report
    - Worked on the Video
    - Created Testing
    - Updated Code
    - Presented Final Project
- Scharara Islam:
    - Updated UML Diagrams
    - Worked on the Report
    - Worked on the Video
    - Presented Final Project

2 Setup

Running:

Steps to Running the program:

- Import the project file into Eclipse
- Run the Scheduler subsystem class first
- Next the Floor subsystem class
- Then the Elevator subsystem class

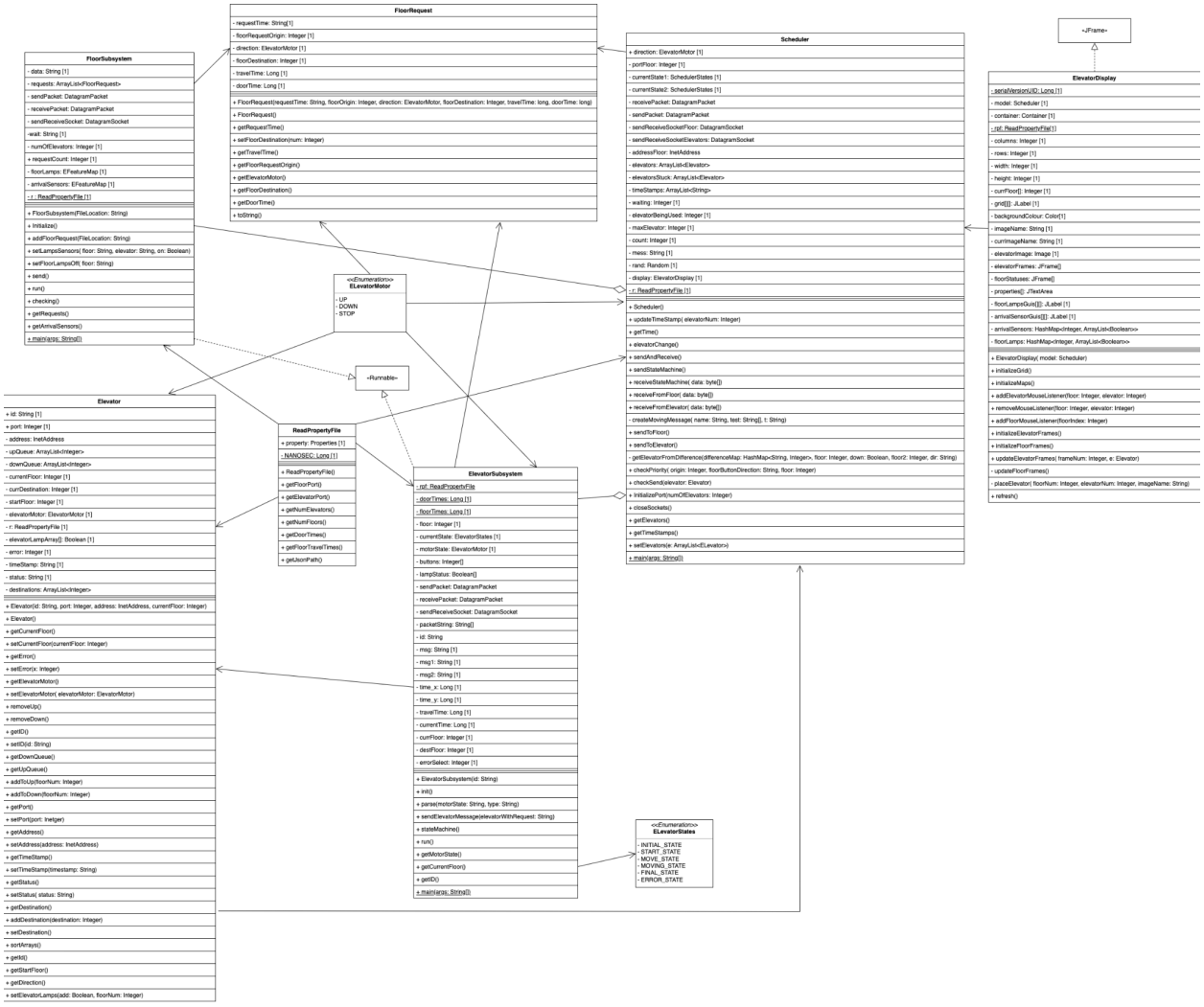
Testing:

Steps to Running the tests:

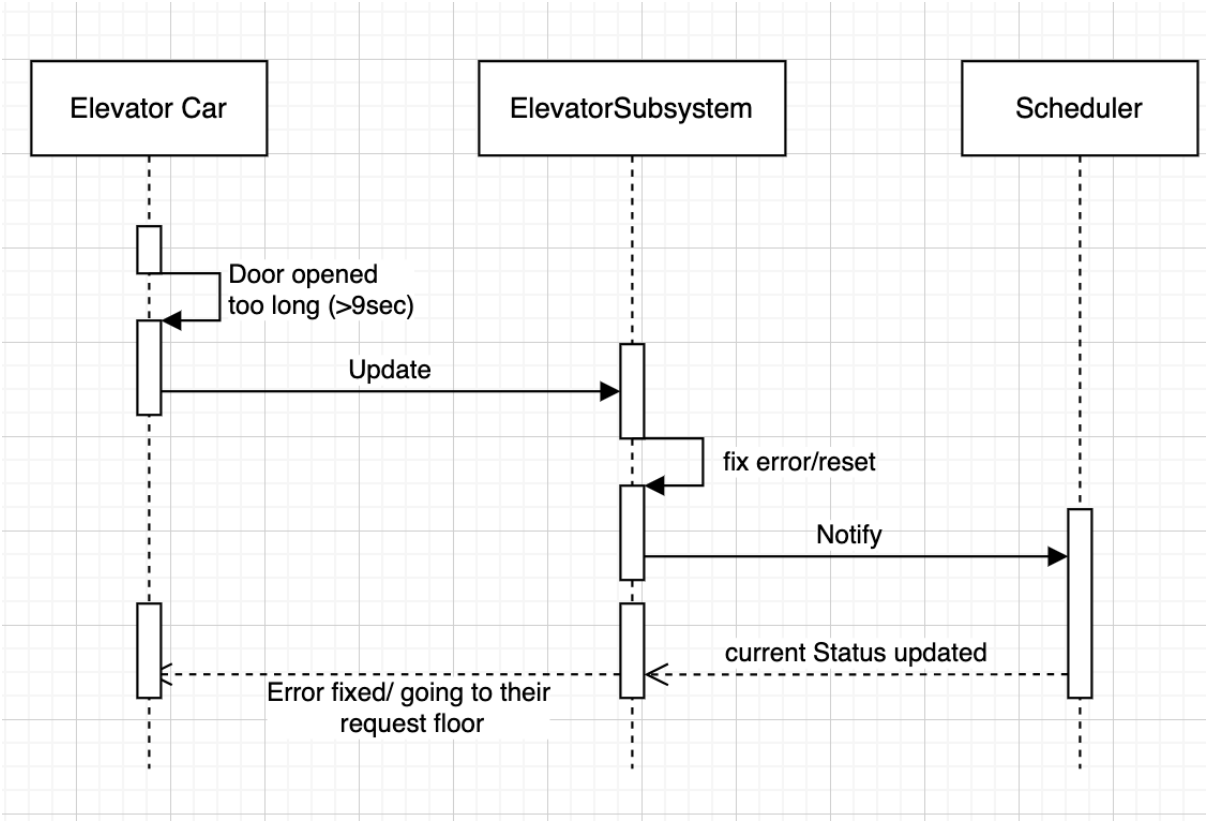
- Import the project file into Eclipse
- Run the test files, SchedulerTest, FloorSubsystemTest, and ElevatorSubsystemTest

Documentation

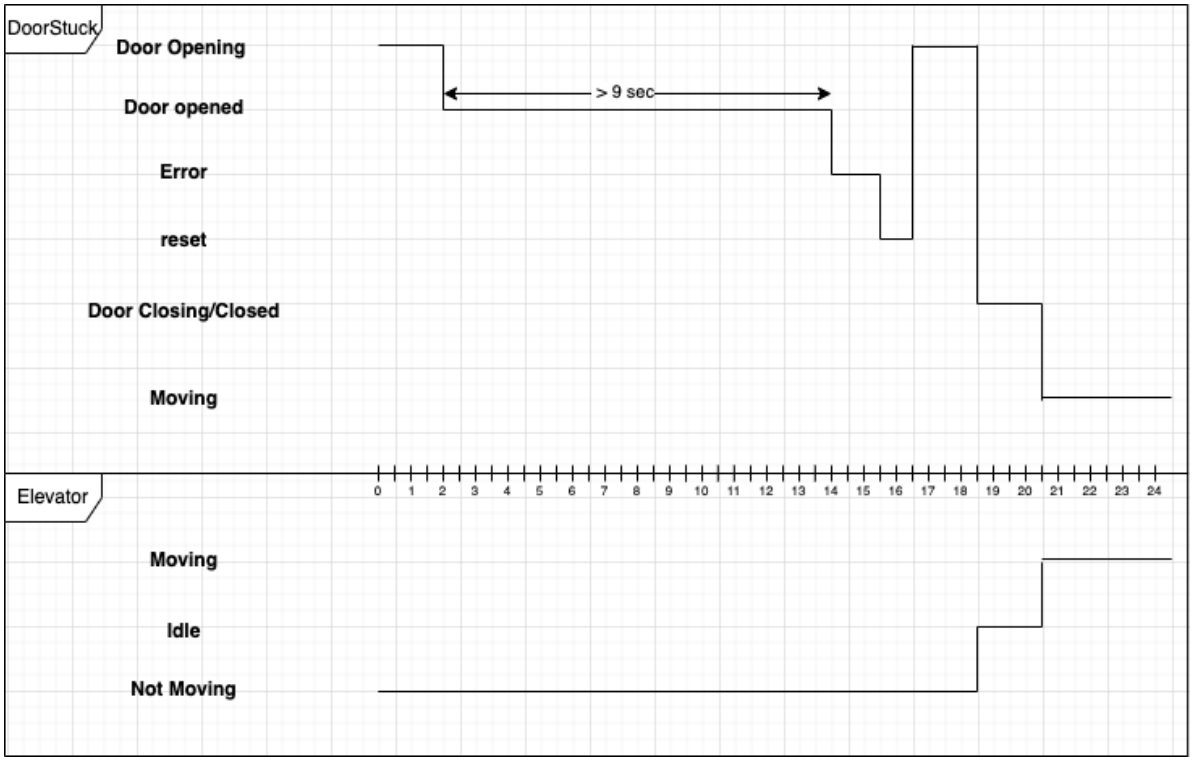
UML Class Diagram



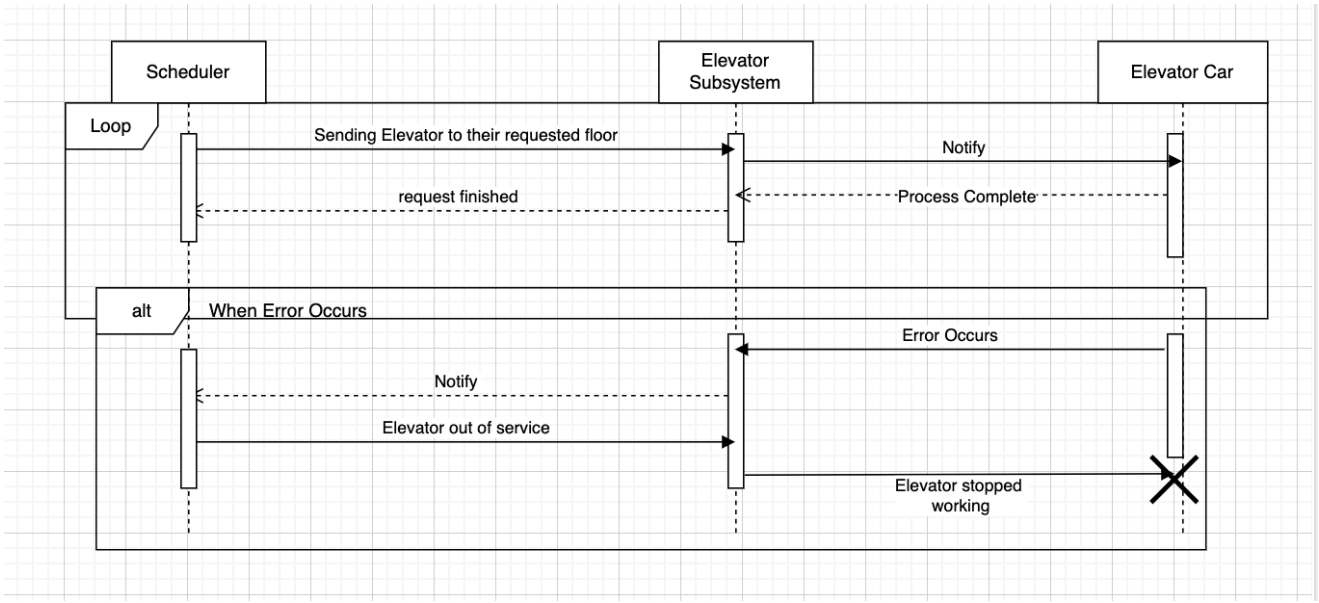
Door Stuck Sequence Diagram



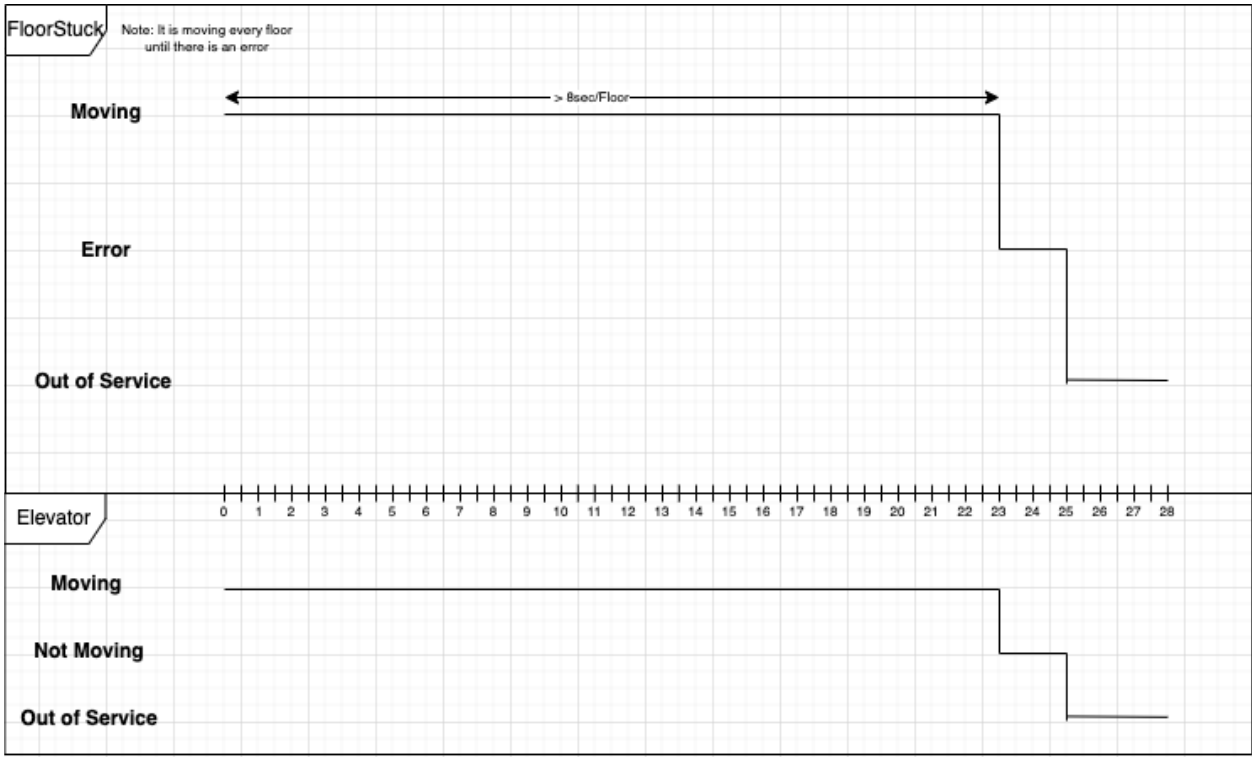
Door Stuck Error Timing Diagram



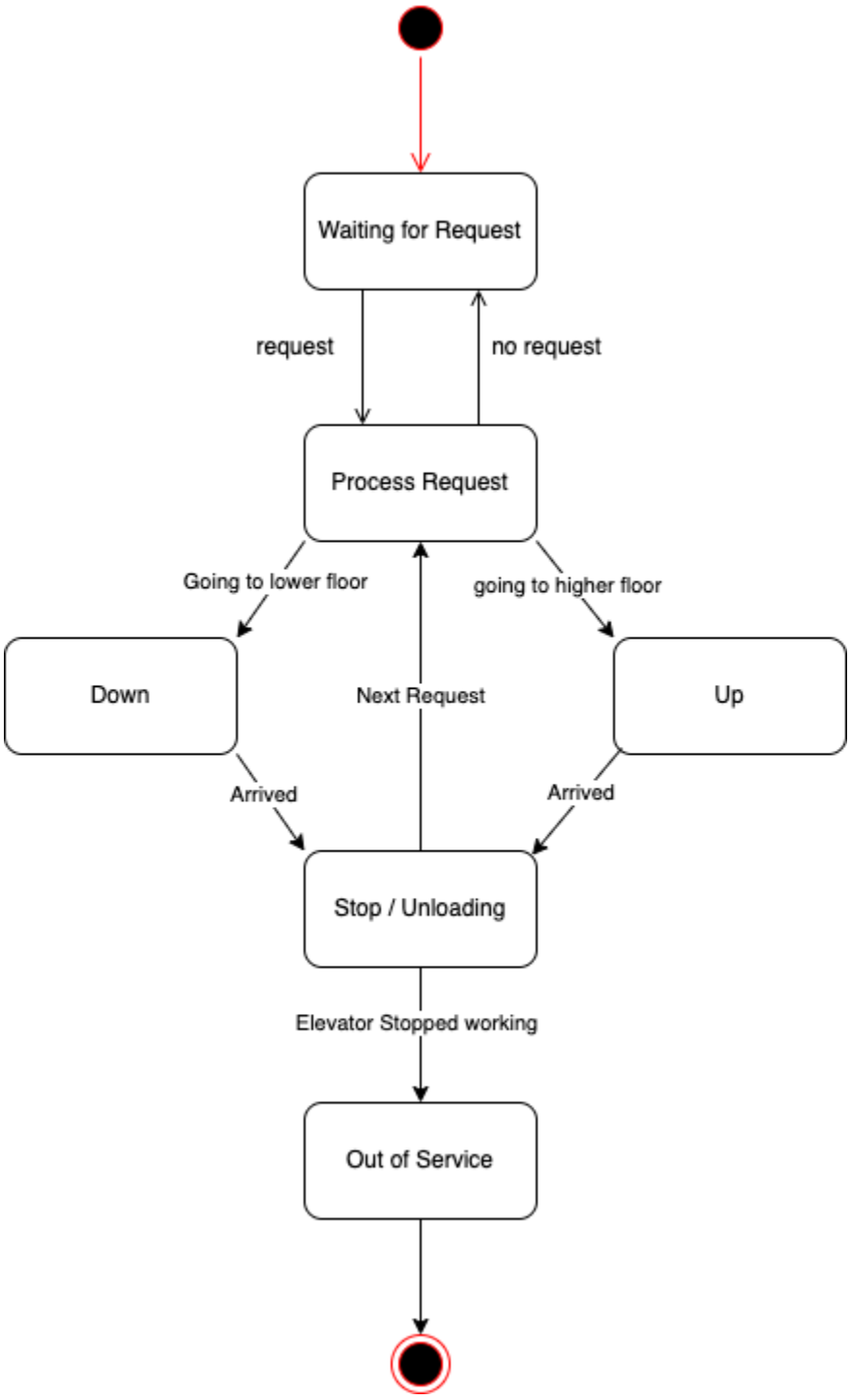
Floor Stuck Sequence Diagram



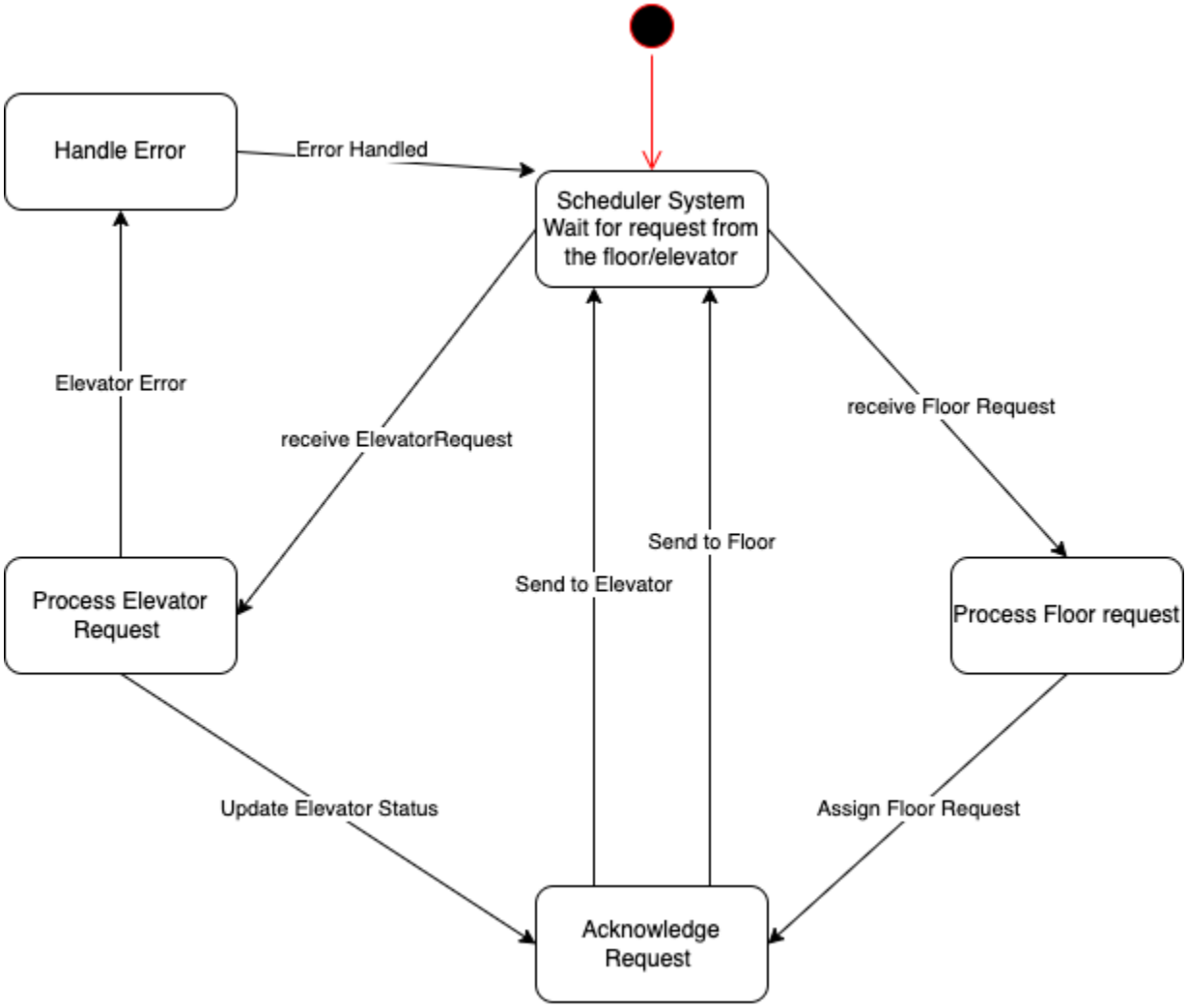
Floor Stuck Error Timing Diagram



Elevator/ Subsystem State Machine Diagram

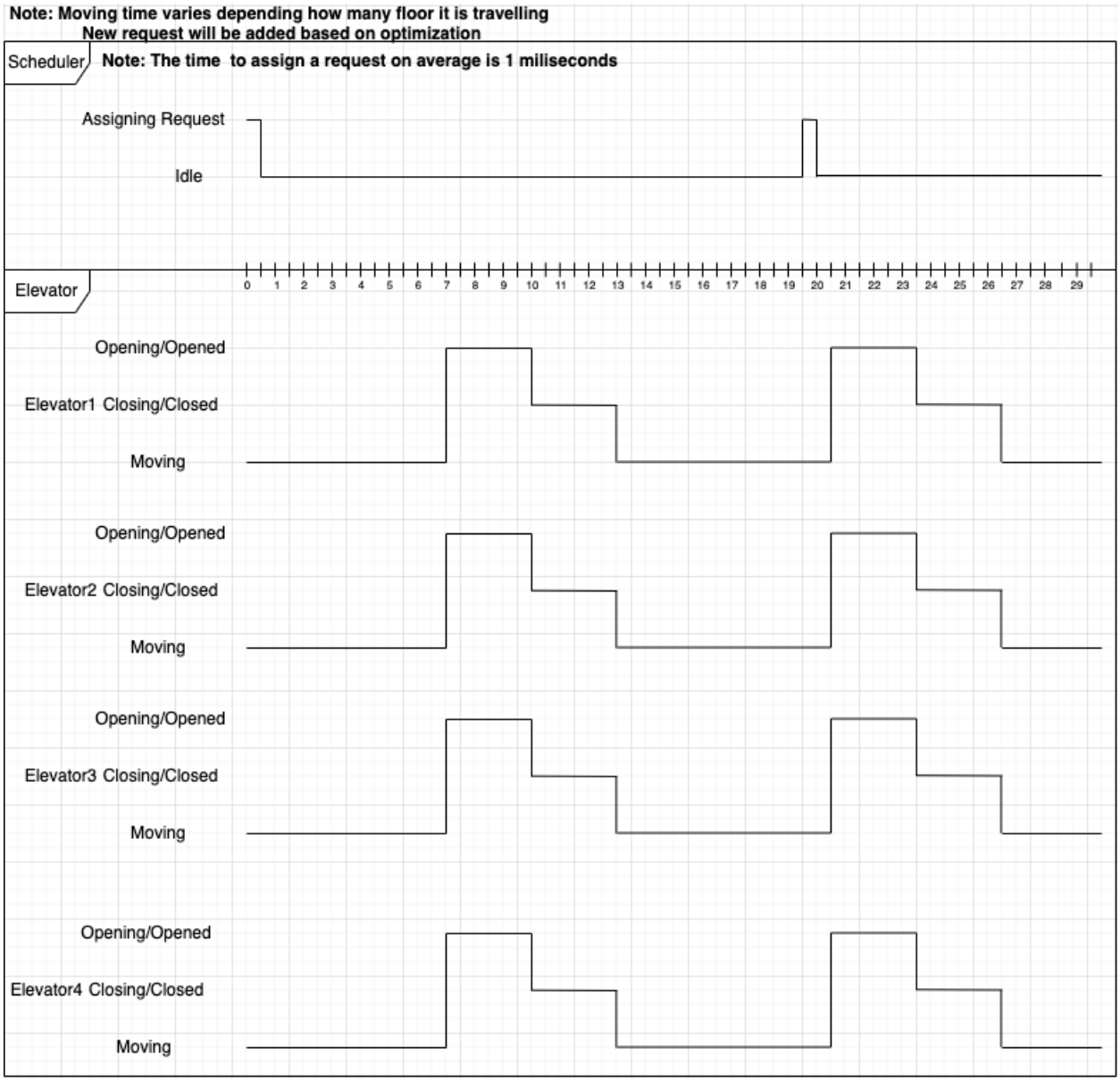


Scheduler Subsystem State Machine Diagram



Performance

Scheduler Timing Diagram



Conclusion

The final code for the elevator simulator will be able to run on their own and communicate to each other through UDP. Each subsystem will have to be run in a certain order to make sure that it runs properly. It starts off the Scheduler, FloorSubsystem, and lastly the ElevatorSubsystem. Afterwards it will create a GUI for the system to make it seem like it's the concierge's point of view of the elevator system. Showing if the elevator is stuck, where the elevator is located, and where the next request is.

Reflections:

This project could have been done better by fulfilling everything each iteration rubric listed. By following the rubric more it would have helped us stay on track more on some iterations. Although we managed to fix up the code for the last few iterations to be suitable with how it should run. If we had more time for the fourth iteration we would have created test cases to supplement the code. A feature that we could have added into the GUI design was the floor lamps, indicating if the elevator is going up or going down. The optimization could not fully be implemented, the elevator optimizes the request by assigning it to the elevator closest to the request floor. But will only process the request once it's completely done from the current request. Instead of stopping at floor 5 since it's passing by it anyways it will finish its request then go to floor 5.

References

[1] Realistic Elevators Opened Closed Halfopen Doors Stock Illustration 1677825796. (2022). Retrieved 6 April 2022, from https://www.shutterstock.com/image-illustration/realistic-elevators-opened-closed-halfopen-doors-1677825796