Carleton University

Term Project Report

Course: SYSC 3303

Real Time Elevator System

Group 7

Team Members

Dillon Claremont

Thomas Bryk

Jacob Martin

Mustafa Abdulmajeed

Gordon Macdonald

Table of Contents

[Breakdown of Responsibilities 3](#_Toc5737148)

[Iteration 1 3](#_Toc5737149)

[Iteration 2 3](#_Toc5737150)

[Iteration 3 3](#_Toc5737151)

[Iteration 4 4](#_Toc5737152)

[Iteration 5 4](#_Toc5737153)

[Diagrams 5](#_Toc5737154)

[UML Class Diagrams 5](#_Toc5737155)

[Scheduler 5](#_Toc5737156)

[ElevatorSubsystem 6](#_Toc5737157)

[FloorSubsystem 7](#_Toc5737158)

[State Machine Diagrams 8](#_Toc5737159)

[Elevator State Machine 8](#_Toc5737160)

[Cooperating State Machine 9](#_Toc5737161)

[Timing Diagrams 10](#_Toc5737162)

[Error Scenario – “Door Fault” 10](#_Toc5737163)

[Error Scenario – “Motor Fault” 11](#_Toc5737164)

[Scheduler Mean Response Time 12](#_Toc5737165)

[Instructions 13](#_Toc5737166)

[Setup 13](#_Toc5737167)

[Running 13](#_Toc5737168)

[Configuration 14](#_Toc5737169)

[Scheduler Measurements 15](#_Toc5737170)

[Event Descriptions 15](#_Toc5737171)

[Scheduler Response Time Measurements to Various Events 16](#_Toc5737172)

[Scheduler Response Time Analysis 17](#_Toc5737173)

[Reflection on Design 18](#_Toc5737174)

# Breakdown of Responsibilities

## Iteration 1

* Dillon Claremont
  + Scheduler + related classes & State Diagram
* Thomas Bryk
  + FloorSubsystem + related classes
* Jacob Martin
  + ElevatorSubsystem + related classes
* Mustafa Abdulmajeed
  + Request system
* Gordon Macdonald
  + Test framework + UML documentation

## Iteration 2

* Dillon Claremont
  + Update Scheduler to support destination requests directly from elevator, Diagrams.
* Thomas Bryk
  + Update FloorSubsystem to send destination requests to arriving elevators
* Jacob Martin
  + Update ElevatorSubsystem to receive destination requests from FloorSubsystem and send to Scheduler
* Mustafa Abdulmajeed
  + Update Request system with new requests and necessary modifications
* Gordon Macdonald
  + Test Framework, Diagrams, UML documentation

## Iteration 3

* Dillon Claremont
  + Update Scheduler to detect and handle error scenarios (elevator stuck/ door stuck).
  + If an elevator is stuck between floors, it is set as "out of service" and not assigned any more trips.
  + Any pending trips that have not been started yet by this elevator is then reassigned.
  + If an elevator door open/close is interrupted, the scheduler resends the door open/close event to the elevator.
* Thomas Bryk
  + Update FloorSubsystem to allow faults (elevator stuck/ door stuck) to be encoded in input file
  + The input file can have trips encoded with 'Motor' and 'Door' faults
    - 'Motor': simulates an elevator stuck between floors
    - 'Door': simulates a door open/close interruption
  + Timing Diagrams
    - Fault Door Timing Diagram.pdf
    - Fault Motor Timing Diagram.pdf
* Jacob Martin
  + Update ElevatorSubsystem to handle faults (elevator stuck/ door stuck) received from FloorSubsystem
  + Handles both 'Motor' and 'Door' scenarios
* Mustafa Abdulmajeed
  + Update Request system to support the above mentioned error scenarios.
  + GUI planning + prototyping
* Gordon Macdonald
  + GUI planning

## Iteration 4

* Dillon Claremont
  + Update Scheduler to support communications with Floors & Elevators on different hosts
  + Time response times for all events received by the scheduler, provided capability for scheduler to preduce a simple report displaying mean response time and variance
* Thomas Bryk
  + Update FloorSubsystem to support communications with Elevators & Scheduler on different hosts
  + Timing Diagram
    - Scheduler Mean Response Time.pdf
* Jacob Martin
  + Update ElevatorSubsystem to support communications with Floors & Elevators on different hosts
* Mustafa Abdulmajeed
  + GUI planning + prototyping
* Gordon Macdonald
  + GUI planning

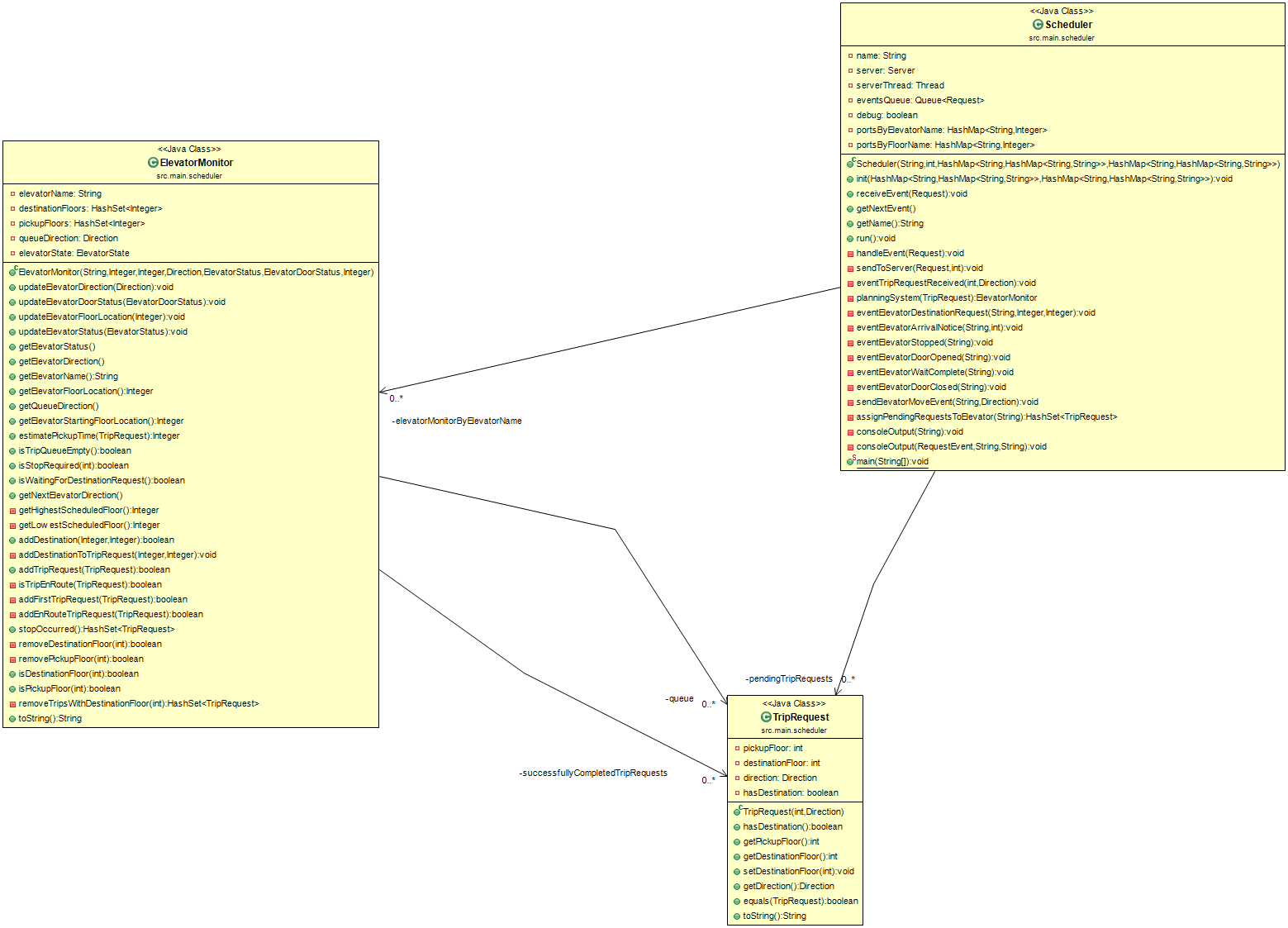
## Iteration 5

* Mustafa Abdulmajeed
  + GUI implementation
* Gordon Macdonald
  + GUI implementation

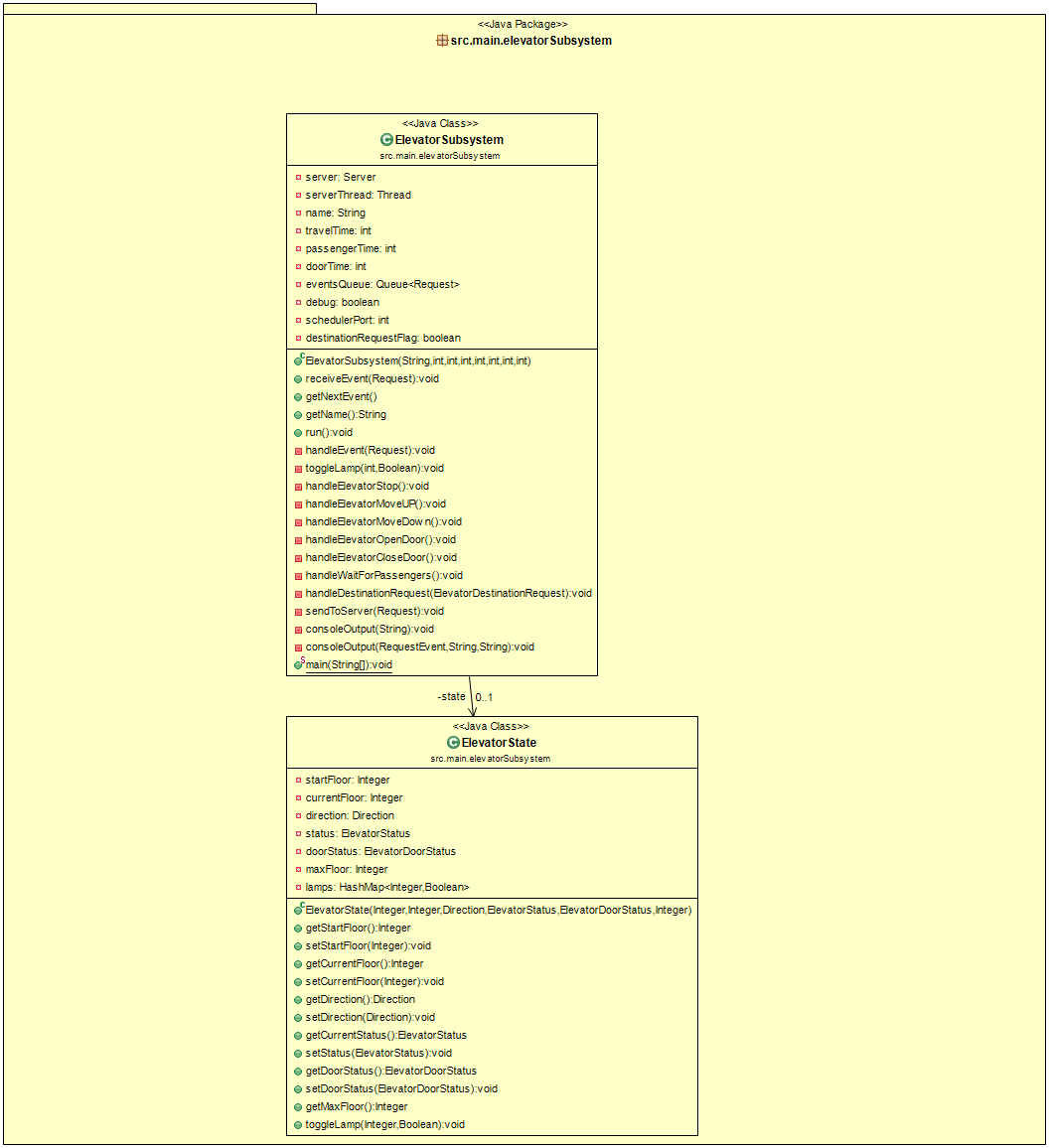
# Diagrams

## UML Class Diagrams

### Scheduler



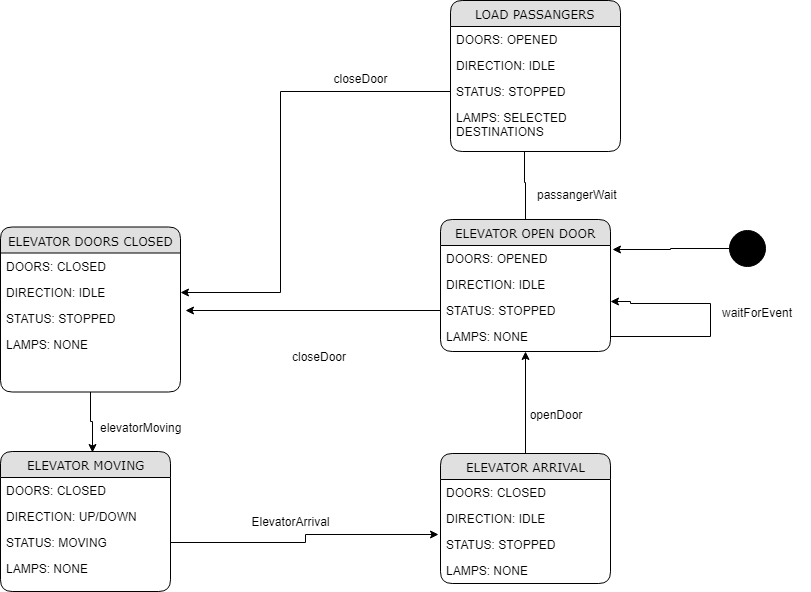
### ElevatorSubsystem



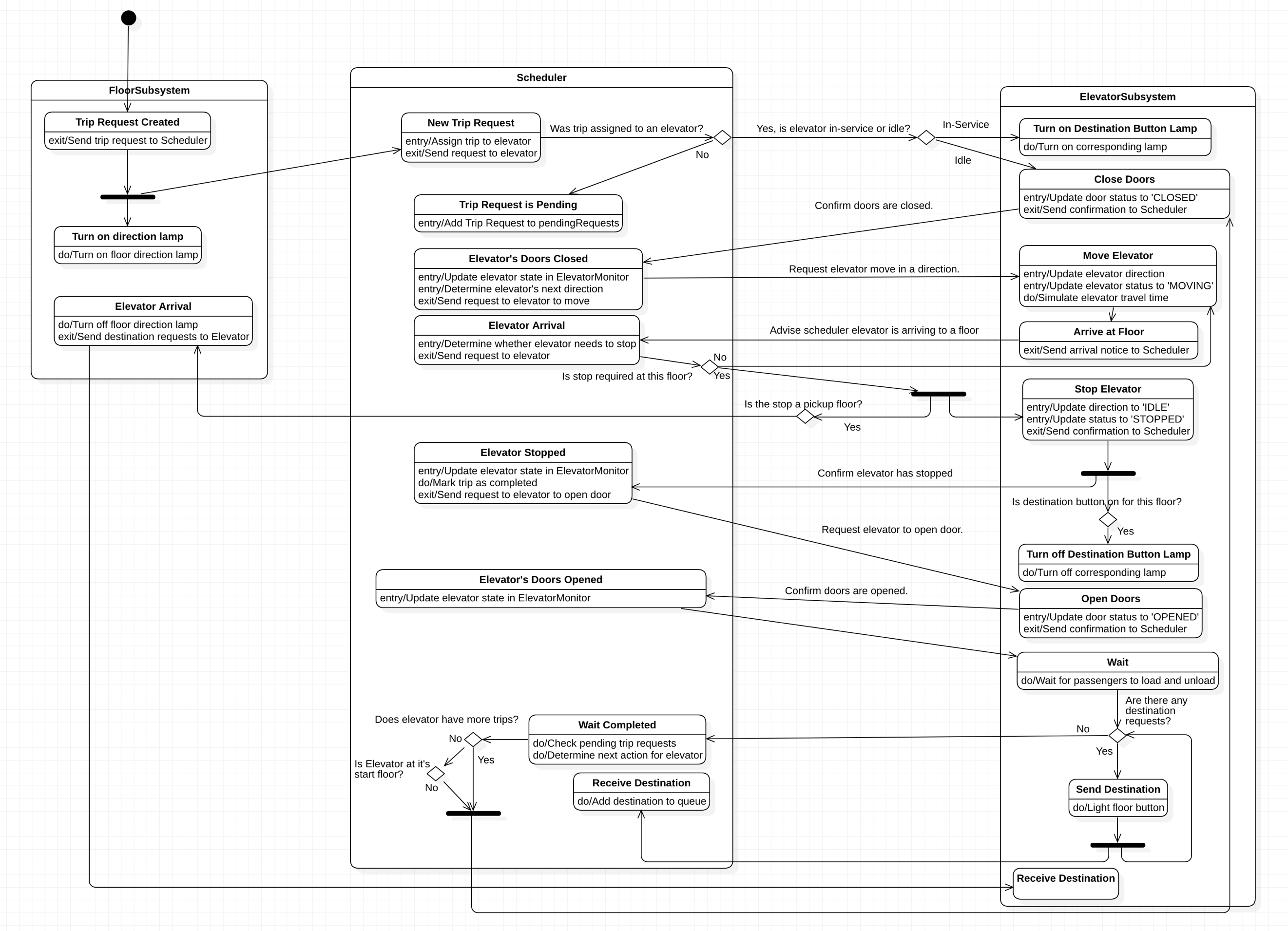
### FloorSubsystem



## State Machine Diagrams

Elevator State Machine

### Cooperating State Machine



## Timing Diagrams

Error Scenario – “Door Fault” SYSC3303%20Workspace/SYSC3303-Group-7/doc/Fault%20Door%20Timing%20Diagram.pdf

### Error Scenario – “Motor Fault”

SYSC3303%20Workspace/SYSC3303-Group-7/doc/Fault%20Motor%20Timing%20Diagram.pdf

Scheduler Mean Response TimeSYSC3303%20Workspace/SYSC3303-Group-7/doc/Timing%20Diagram%20-%20Scheduler%20Mean%20Response%20Time.pdf

# Instructions

## Setup

1. Import project from archive file into Eclipse
2. From Eclipse
   * Select 'File' menu
   * Select 'Import' menu item
   * Under 'General' select 'Projects From File System or Archive File'
   * Select 'Archive...'
   * Locate 'L1G7\_Project\_Iteration4.zip' as Import source.
   * Click 'Finish'
3. From within the project "SYSC3303-Group-7"
   * Run Scheduler.java (located src > main > scheduler)
   * Run ElevatorSubsystem.java (located src > main > elevatorSubsystem)
   * Run FloorSubsystem.java (located src > main > floorSubsystem) last.

## Running

* If an error occurs immediately upon running either the Scheduler, ElevatorSubsystem or FloorSubsystem, the likely cause is due to ports in the local environment that are already in use.
  + To change the configuration the config.xml file needs to be updated. This file is located at src\resources\config.xml
  + Every <Scheduler> <Elevator> and <Floor> element have a port attribute defined. Update the corresponding ports for the Class that has displayed an error (Scheduler / Floor / Elevator)
  + IMPORTANT: All programs MUST be restarted in order for the updated ports to take effect.
* To change the trips scheduled for the elevators, the requests.txt document must be modified.
  + The file is located at src\resources\requests.txt
  + requests must be entered one per line
    - use the following format: relative time from now, pickup floor, trip direction, destination floor, fault [optional]
      * ex. ‘00:00:13.000 2 UP 15 Motor’
      * This means:
        + Send this request using the first value as relative time from now = 00:00:13.000 - this means this request will be sent in 13 seconds
        + pickup floor = 2
        + direction = up
        + destination floor = 15
        + fault = Motor

this can be either 'Motor' or 'Door'

this param is optional.

* From the GUI
  + Trip details can be viewed by double clicking on the trip request in the ‘Trip Requests’ list
    - A trip request is represented as (2, UP, ?)
      * 2 – pickup floor
      * UP – direction of request
      * ? – indicates that a destination request has not yet been entered
        + this will update once a destination button request is simulated (upon pickup).

### Configuration

* There are various parameters that are configurable in this program.
* In config.xml (src\resources\config.xml) it is possible to
  + Specify the number of elevators and floors to be deployed as part of the system
  + Configure the host machine and port for each subsystem (Scheduler, FloorSubsystem(s), ElevatorSubsystem(s))
    - specify the host attribute to desired host IP address (localhost if using local machine)
    - specify the port attribute to desired port
  + Configure various operating specifications for the Elevator
    - startFloor
      * this is the floor the elevator starts on, also when no trips are assigned to an elevator, it returns to its startFloor
    - timeBetweenFloors
      * this sets the travel time for an elevator between floor
    - passengerWaitTime
      * this sets the amount of time an elevator waits for passengers before attempting to close the doors
    - doorOperationTime
      * this sets the amount of time required for an elevator to close/open its doors

Note: All changes to config.xml require a complete program restart to take effect.

# Scheduler Measurements

## Event Descriptions

|  |  |  |
| --- | --- | --- |
| Event Type | Description | Scheduler’s Response |
| ElevatorMotorRequest | This event indicates a confirmation response from an Elevator that it has either stopped or has started moving has been received | IF ELEVATOR HAS STOPPED   1. Determine whether a trip has been completed by this stop 2. Send an “Open Door” request to elevator |
| FloorButtonRequest | This event indicates a request for an elevator from a floor has been received | 1. Determine the most ideal elevator to assign this trip to. 2. If no eligible elevators, save this trip request to a pending queue |
| ElevatorDoorRequest | This event indicates a confirmation response from an Elevator that it has either opened or closed its doors has been | IF ELEVATORS DOORS HAVE OPENED   1. Attempt to assign any pending requests (if there are any) to this elevator (if it is now eligible) 2. Send a notice to Floor that Elevator has arrived 3. Send a “Wait for Passengers” request to the Elevator   IF ELEVATOR DOORS HAVE CLOSED   1. Determine the next direction for this elevator (based on elevator’s current location and its queue status) 2. Send a “Motor” request to the Elevator to move where it needs to go |
| ElevatorDestinationRequest | This event indicates that a destination request has been received from an Elevator. This simulates a passenger pressing a destination button. | 1. Add the destination to the elevator’s queue. |
| ElevatorArrivalRequest | This event indicates that an Elevator is approaching a Floor. This message is sent from the Elevator. | 1. Determine whether the Elevator needs to make a stop at this floor (based on its queues)   IF ELEVATOR MUST STOP   1. Send a “Motor” request to Elevator to stop.   IF NOT   1. Send a “Motor” request to Elevator with direction to go |
| ElevatorWaitRequest | This event indicates that an Elevator has completed its wait for passenger cycle. | 1. Ensure a destination request has been received (if waiting for one ie. At pickup floor)   IF NOT RECEIVED   1. Send another “Wait” to Elevator   IF NOT NEEDED/WAITING   1. Determine next direction of Elevator (based on queue) and send corresponding “Motor” request, send nothing if elevator is at its starting floor |

## Scheduler Response Time Measurements to Various Events

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Elevator  Motor  Request (ms) | Floor  Button  Request (ms) | Elevator  Door  Request (ms) | Elevator  Destination  Request (ms) | Elevator  Arrival  Request (ms) | Elevator  Wait  Request (ms) |
| 3.400046 | 19.367829 | 6.03492 | 4.890016 | 3.58392 | 5.145289 |
| 2.942383 | 9.864774 | 4.17986 | 8.807538 | 3.63236 | 6.272095 |
| 2.600014 | 15.092371 | 2.560901 | 3.682069 | 3.91841 | 1.575891 |
| 1.239201 | 22.447931 | 2.71461 | 2.605861 | 3.497946 | 1.957945 |
| 9.586814 | 4.42836 | 5.050258 | 2.849765 | 3.437933 | 1.47777 |
| 1.611405 | 5.55053 | 4.019763 | 1.361548 | 3.791024 | 1.707711 |
| 2.097647 |  | 3.499843 |  | 3.489768 | 1.64552 |
| 1.021466 |  | 2.948876 |  | 4.175063 | 3.046147 |
| 1.153916 |  | 2.610567 |  | 2.303776 | 1.410362 |
| 1.912994 |  | 3.029737 |  | 4.197081 | 1.605298 |
| 0.774082 |  | 2.431359 |  | 1.427153 | 1.629541 |
|  |  | 1.623775 |  | 1.429444 |  |
|  |  | 1.23257 |  | 1.559381 |  |
|  |  | 1.094482 |  | 1.361918 |  |
|  |  | 1.512486 |  | 1.517065 |  |
|  |  | 0.942463 |  | 1.949722 |  |
|  |  | 1.552131 |  | 1.472411 |  |
|  |  | 1.386641 |  | 1.400522 |  |
|  |  | 1.407087 |  | 1.513344 |  |
|  |  | 1.102472 |  | 1.394781 |  |
|  |  | 1.614879 |  | 1.46182 |  |
|  |  | 1.757014 |  | 1.542198 |  |
|  |  | 1.599176 |  | 1.482517 |  |
|  |  |  |  | 1.581027 |  |
|  |  |  |  | 1.235192 |  |
|  |  |  |  | 1.430425 |  |
|  |  |  |  | 1.429391 |  |
|  |  |  |  | 1.55398 |  |
|  |  |  |  | 1.541221 |  |
|  |  |  |  | 1.586636 |  |
|  |  |  |  | 2.096625 |  |
|  |  |  |  | 1.623907 |  |
|  |  |  |  | 1.44615 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Scheduler Response Time Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Event Type | # of Events | Mean Response (ms) | Variance (ms2) |
| ElevatorMotorRequest | 11 | 2.57636 | 5.54882 |
| FloorButtonRequest | 6 | 12.79197 | 45.45471 |
| ElevatorDoorRequest | 23 | 2.43069 | 1.77184 |
| ElevatorDestinationRequest | 6 | 4.03280 | 5.70454 |
| ElevatorArrivalRequest | 33 | 2.15346 | 1.01250 |
| ElevatorWaitRequest | 11 | 2.49760 | 2.52914 |

# Reflection on Design

Overall as a group we are satisfied with the design choices made throughout the project. One of the observations was that as a result of the original design and planning, enhancements were able to be made relatively easy as the project progressed. This is because a strong emphasis was placed on the single responsibility principle throughout the design process. The modular nature of the design also made it very natural for each team member to find a contributing role.

This design emphasizes configurability and flexibility of the system. From the back-end to the front-end, the system requires minimal modifications to accommodate varying configurations. Multiple elevators or floors can be added by adding simply adding them to a configuration document, when the system is started up, all necessary connections and GUI elements are then generated dynamically. This was a huge benefit because as these requirements have changed throughout the project, the changes to accommodate them were miniscule.

We feel that although the software satisfies the requirements, we would benefit from having automated testing. The testing framework initially developed was eventually removed from the project as it was not updated in accordance with the project.