CARLETON UNIVERSITY

SYSC 3303 A

SECTION L5

GROUP 4

Final Project Report

Group Members:-

1. Muhammad Tarequzzaman | 100954008
2. Sama Adil Sheikh | 101060020
3. Anannya Bhatia | 100989250
4. Dare Balogun | 101062340
5. Mrunal Patel | 101001748

Table of Contents

Introduction:

In this project, we were tasked with designing and implementing an elevator control system and simulator. The code was written in Java, using Eclipse IDE. There are three major components in the system; Elevator Subsystem which creates different threads of elevator cars (includes buttons, doors and motors), Floor Subsystem for the floor interface and Scheduler is the bridge between the Elevator Subsystem and the Floor Subsystem. It also controls the flow of requests and messages in the system. The programs communicate via DatagramSocket objects.

Breakdown of responsibilities of each team member for each iteration:

- Muhammad Tarequzzaman: -
  - Iteration 1: ElevatorSubSystem.java and ElevatorSubSystemTest.java.
    - Skeleton Construction, UDP connection establishment.
  - Iteration 2: ElevatorSubSystem.java, Elevator.java, ElevatorTest.java and ElevatorSubSystemTest.java.
    - System Implantation using State machine and Junit test.
    - Java Thread Implementation on Elevators.
    - Synchronized method Implementations.
  - Iteration 3: ElevatorSubSystem.java, Elevator.java, ElevatorTest.java and ElevatorSubSystemTest.java., ElevatorInputEvent.txt
    - Elevators Inter level input system implementation.
    - Error case implementation (Door lock, Elevator Stuck).
  - Iteration 4: ElevatorSubSystem.java, Elevator.java, ElevatorTest.java and ElevatorSubSystemTest.java.
    - Elevator Sub System optimization.
  - Iteration 5: ElevatorSubSystem.java, Elevator.java, ElevatorTest.java and ElevatorSubSystemTest.java., ElevatorInputEvent.txt
    - Report Writing about Elevator part and Iteration 5 UML Class Diagram Implementation using UML generator.
- Sama Adil Sheikh: -
  - Iteration 1: Readme.txt, co-author of FloorSubSystem.java and Java doc comments.
  - Iteration 2: Sequence Diagram, Readme.txt, co-author of FloorSubSystem.java and Java doc comments.
  - Iteration 3: UML, Sequence Diagram, Readme.txt, co-author of FloorSubSystem.java and Java doc comments.
  - Iteration 4: Timing Diagrams, Timing analysis and measurement, Readme.txt, co-author of FloorSubSystem.java and Java doc comments.
  - Iteration 5: Final Project report.
- Anannya Bhatia: -
  - Iteration 1: UDP communication for FloorSubSystem.java, Scheduler.java and ElevatorSubSystem, along with UML Diagrams, Set-up instructions.
  - Iteration 2: UDP communication for FloorSubSystem.java, Scheduler.java and ElevatorSubSystem, along with UML Diagrams, Set-up instructions.

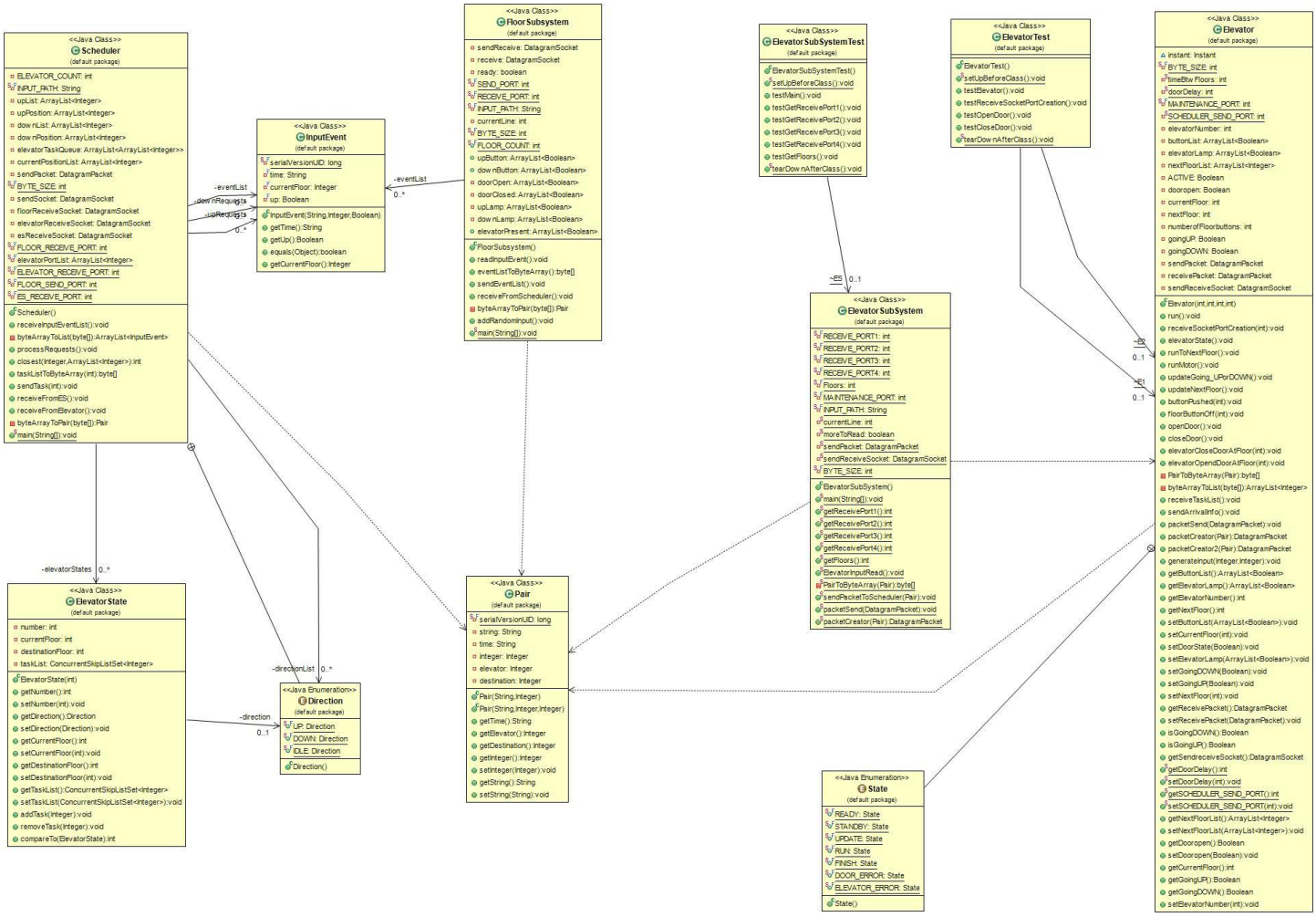
- Iteration 3: UDP communication for FloorSubSystem.java, Scheduler.java and ElevatorSubSystem, along with UML Diagrams, Set-up instructions.
- Iteration 4: UDP communication for FloorSubSystem.java, Scheduler.java and ElevatorSubSystem, UML Diagrams, Timing analysis and measurement.
- Iteration 5: Final Project report, Timing analysis

- Dare Balogun: -
  - Iteration 1: FloorSubSystem.java, InputEvents.java, Pair.java, Scheduler.java and co-author of ElevatorSubSystem.
  - Iteration 2: FloorSubSystem.java, InputEvents.java, Pair.java, Scheduler.java and co-author of methods (PairToByteArray, byteArrayToList and sendArrivalInfo) in Elevator.
  - Iteration 3: FloorSubSystem.java, InputEvents.java, Pair.java, ElevatorState.java, ElevatorInputEvent.txt, Scheduler.java and co-author of method (PairToByteArray, byteArrayToList and sendArrivalInfo) in Elevator.
  - Iteration 4: FloorSubSystem.java, InputEvents.java, Pair.java, ElevatorState.java, ElevatorInputEvent.txt, Scheduler.java and co-author of methods (PairToByteArray, byteArrayToList and sendArrivalInfo) in Elevator
  - Iteration 5: GUI (Graphical User Interface) for the system

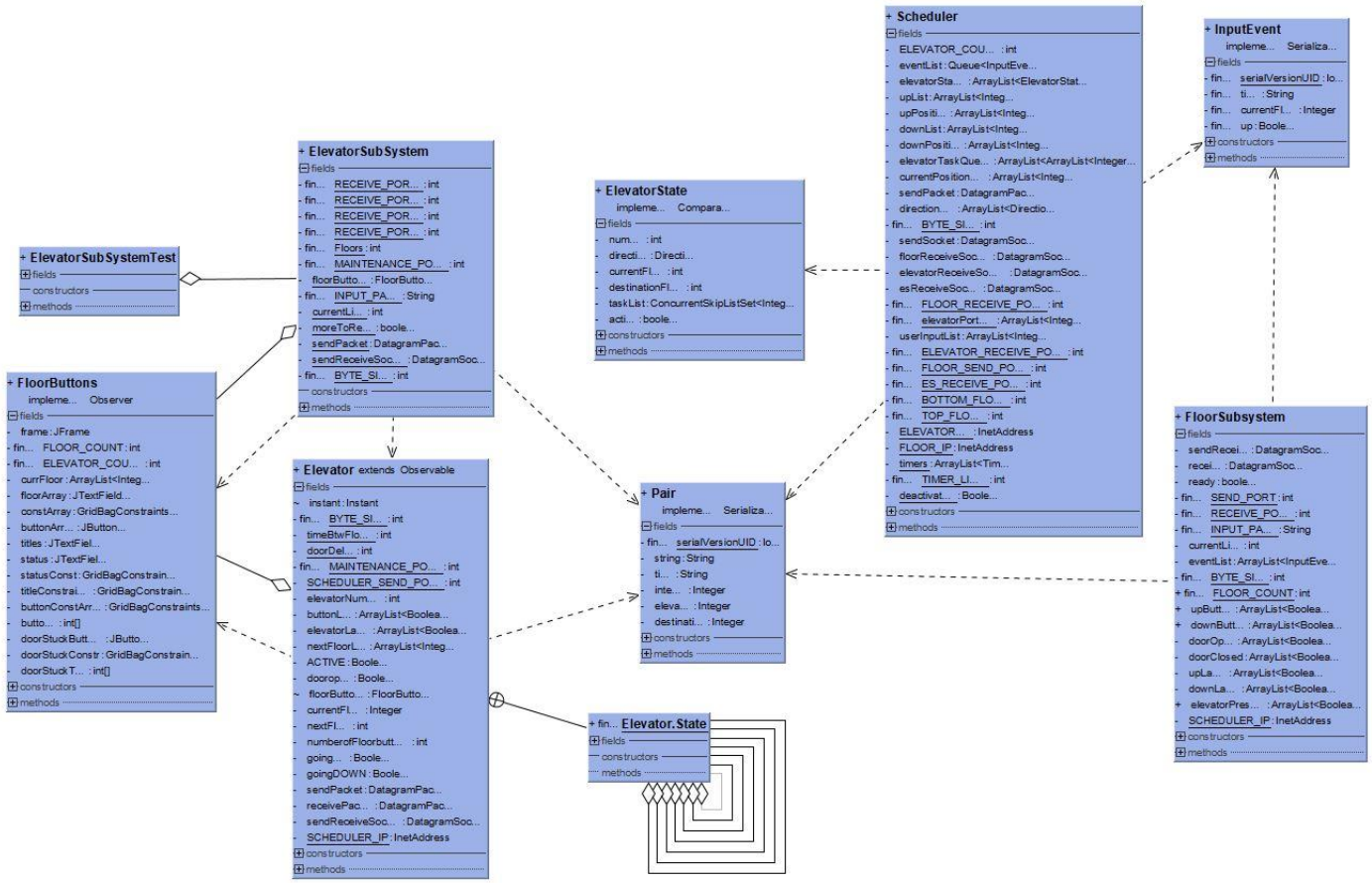
- Mrunal Patel: -
  - Iteration 1: State Machine Diagram
  - Iteration 2: Worked on UML Diagrams and Scheduler.java.
  - Iteration 3: Worked on UML Diagrams and Scheduler.java, ElevaotorSubSystem.java, Elevator.java, ElevatorInputEvent.txt.
  - Iteration 4: Worked on UML Diagrams and Scheduler.java, ElevaotorSubSystem.java, Elevator.java, ElevatorInputEvent.txt, Readme.txt.
  - Iteration 5: State Machine Diagram, Scheduler.java and Graphical User Interface.

Diagrams for All Iterations:

- UML Class Diagram: -

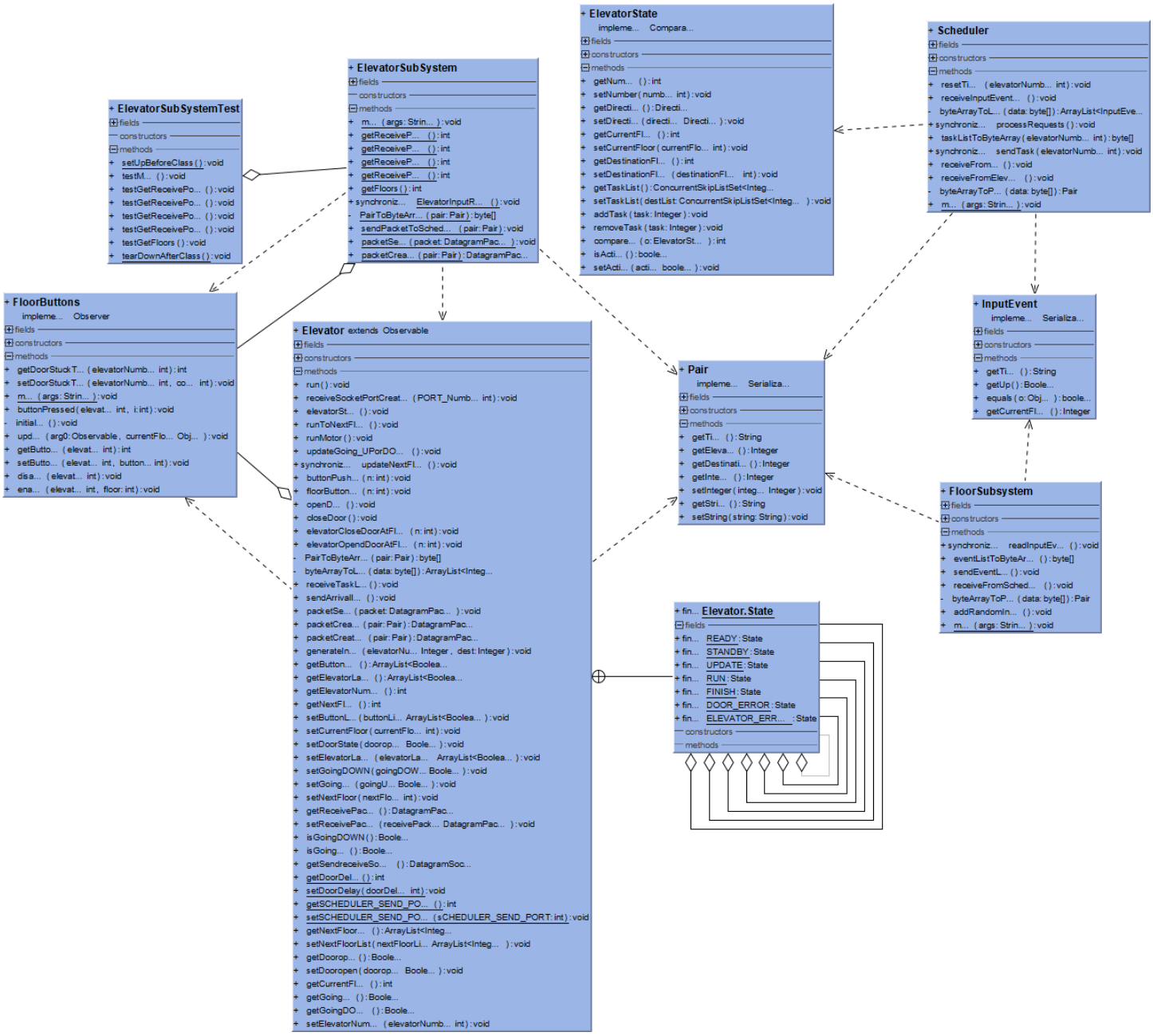


UML 1: Iteration_4 UML



UML 2: Iteration_5 UML Class Diagram (field)

Note: FloorButtons.java is implemented as a GUI for Elevators.



UML 3: Iteration_5 UML Class Diagram (Method)

Note: FloorButtons.java is implemented as a GUI for Elevators.

- State Machine Diagram for Iteration 1: -

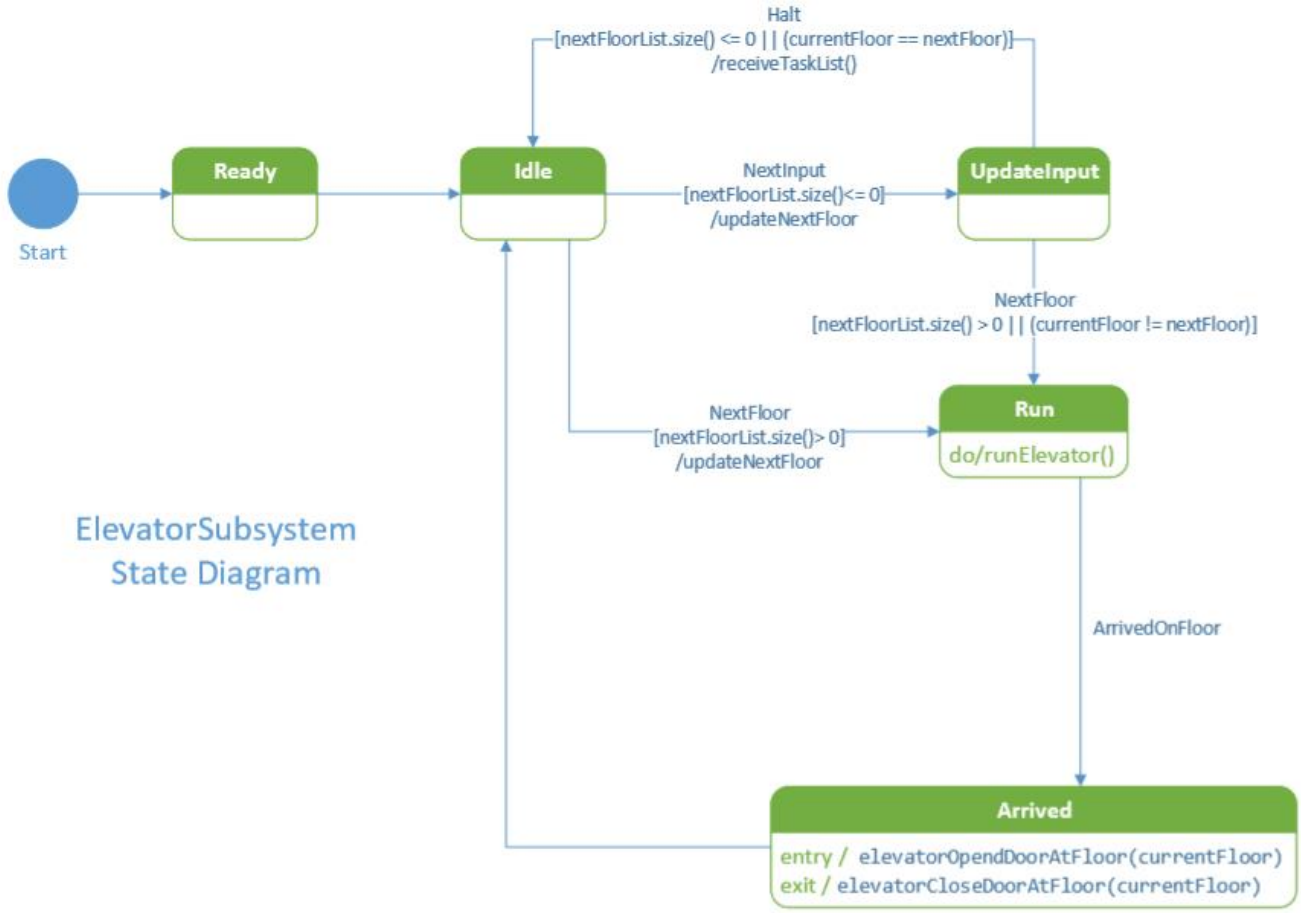


Diagram 1: State machine diagram for elevatorSubsystem

- Sequence Diagram for Iteration 2: -

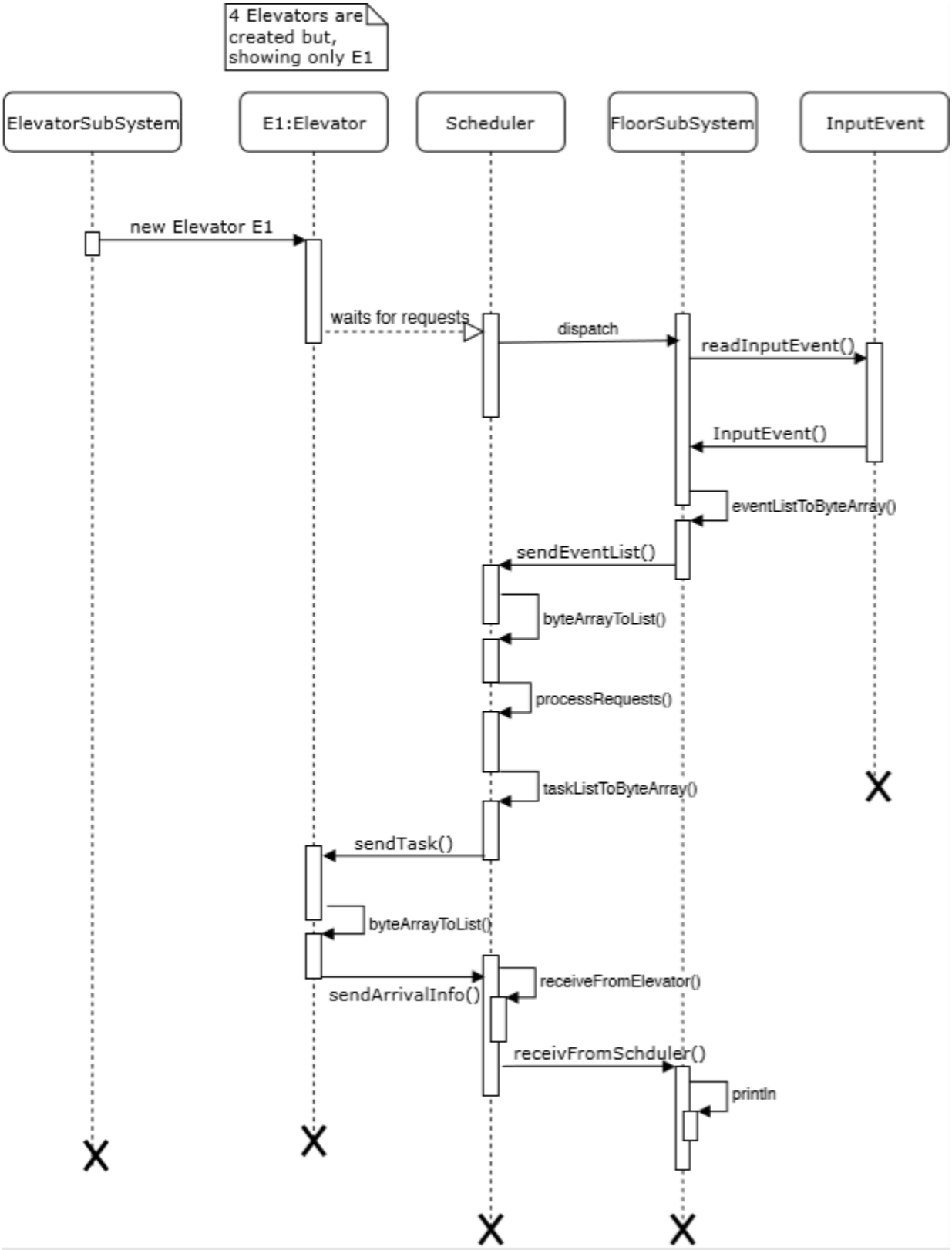


Diagram 2: Sequence diagram for elevator thread creation

- Sequence Diagram_E1 for Iteration 3: -

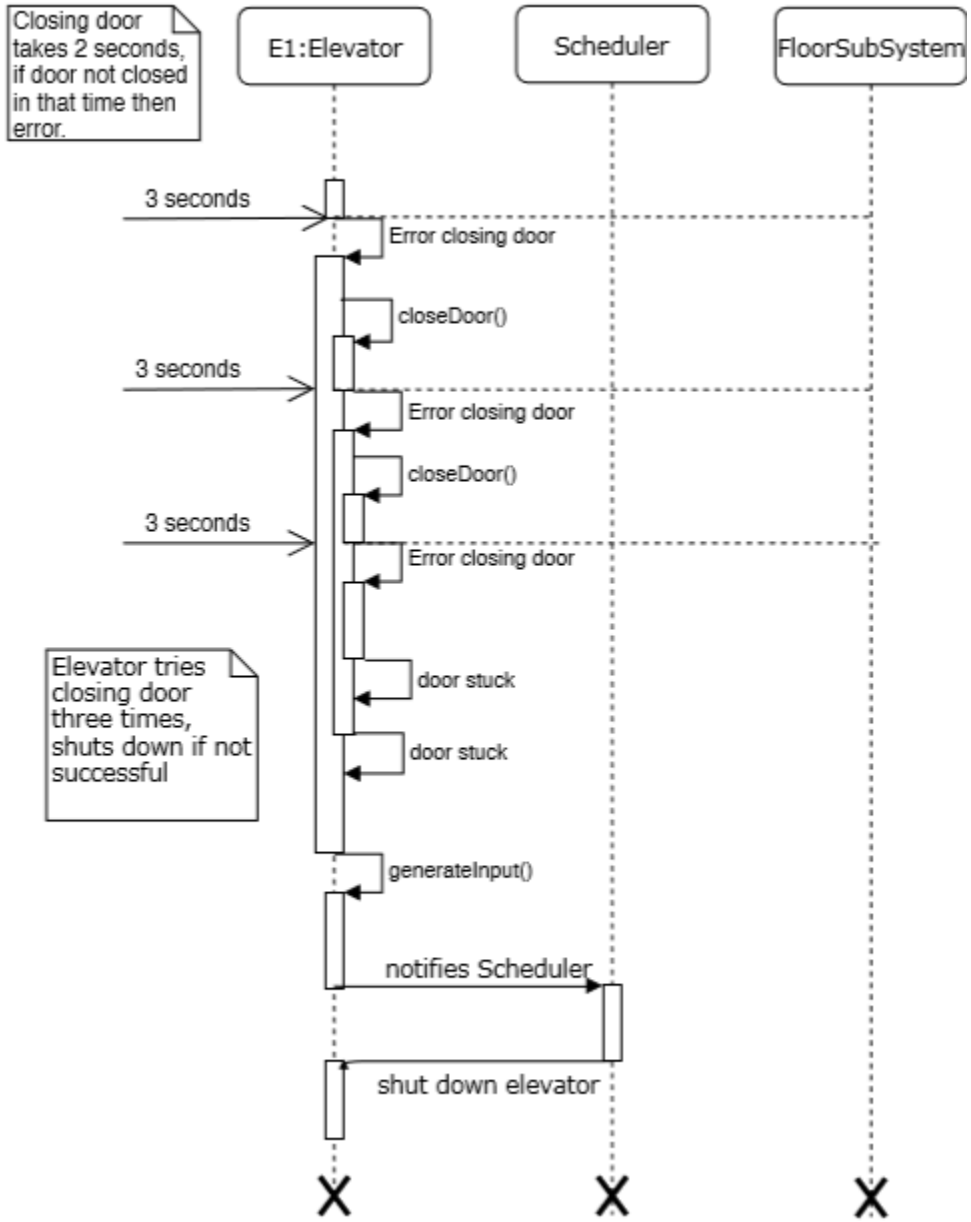


Diagram 3: Sequence diagram for door stuck error scenario

- Sequence Diagram_E2 for Iteration 3: -

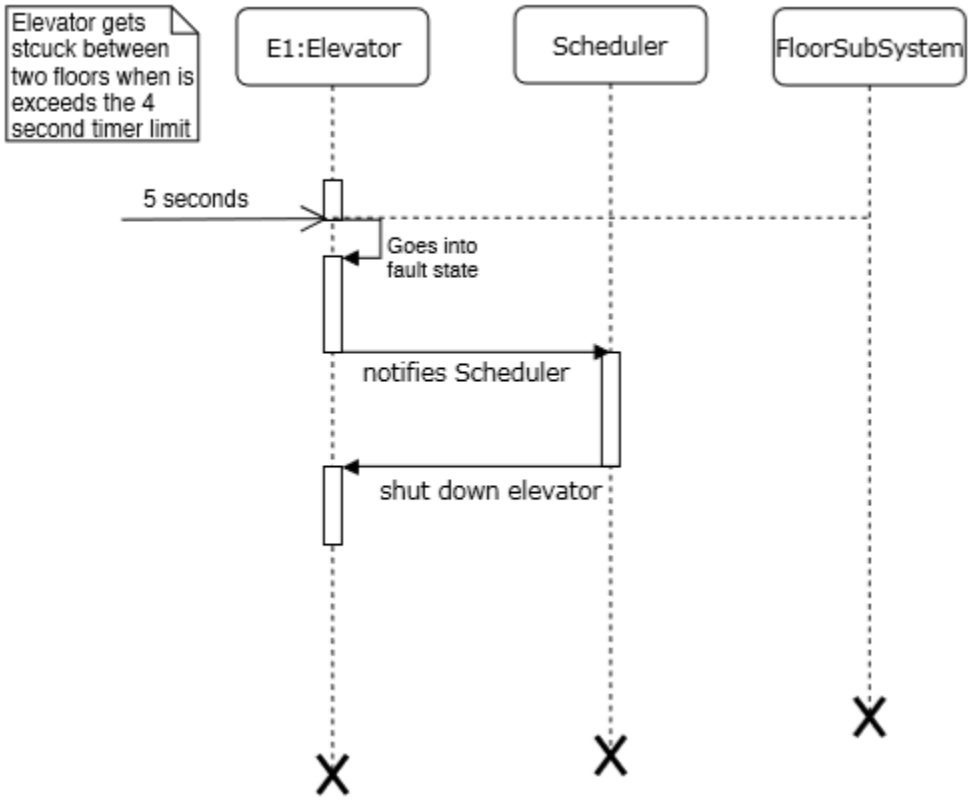


Diagram 4: Sequence diagram for elevator stuck error scenario

- Timing Diagram for Iteration 4: -

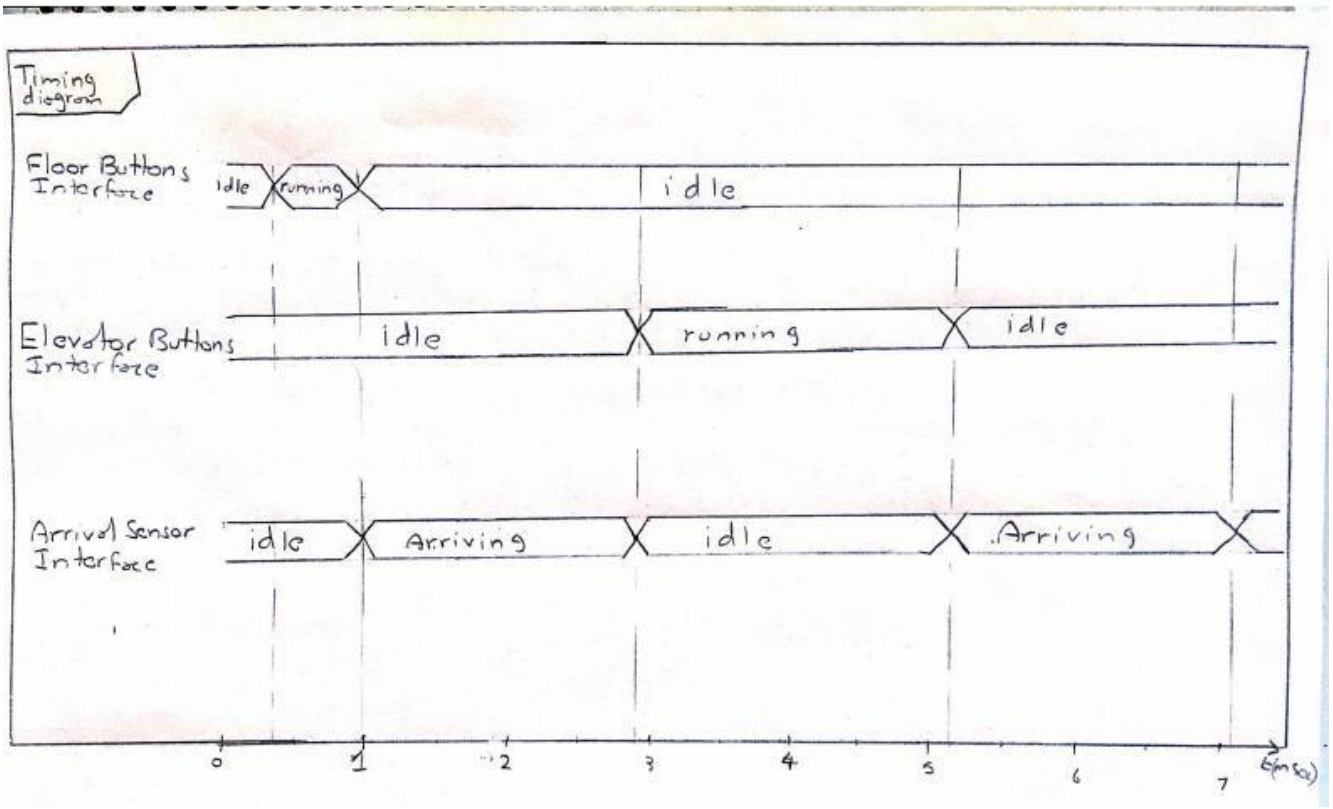


Diagram 5: Timing diagram for iteration 4

- Set up and Test Instructions:

  - **Set up Instructions: -**
    - Open eclipse-java
    - Click File->Open Projects from File System->Archive then select the zip file "L5G4_Final_Project.zip"
    - Click Finish
    - Run ElevatorSubSystem as Java Application
    - Wait for the class to execute fully
    - Open new console
    - Run Scheduler as Java Application
    - Wait for the class to execute fully
    - Open new console
    - Run FloorSubSystem as Java Application
    - Wait for the elevator to reach their desired floor
    - Press the buttons in elevators to direct it to the desired floor or to test error cases

- System Analysis and Measurement Results:

- Arrival sensor interface: - This measured the time (in milliseconds) from right after the scheduler received arrival information from the elevator via UDP to after the scheduler has updated its state with the new information and sent a UDP message to the FloorSubsystem.
- Elevator buttons interface: - This measured the time (in milliseconds) from when the scheduler received a UDP datagram packet with information on which floor was requested using the elevator buttons interface, until the scheduler sends the UDP datagram sending appropriate elevator to the requested floor.
- Floor buttons interface: - This measured the time (in milliseconds) from when the scheduler received the UDP datagram information containing the request from a specific floor till after the scheduler has sent a UDP datagram sending an elevator the requested floor.

Arrival Sensor Interface (milliseconds)	Elevator Buttons Interface (milliseconds)	Floor Buttons Interface (milliseconds)
1.29	1.96	0.80
0.98	2.34	0.49
0.85	2.76	0.67
4.15	2.2	0.54
1.16	2.42	
1.72	2.43	
3.30	2.48	
2.41	2.21	
1.22	2.13	
2.32	2.47	

Table 1: Timing measurements

Task	Computation Time (Mean in milliseconds)	Variance
Arrival Sensor Interface	1.94	1.19
Elevator Buttons Interface	2.34	0.05
Floor Buttons Interface	0.62	0.02

Table 2: Timing analysis

- Design Reflection:

In the project design, the Graphical User Interface is easy to understand and user friendly. The built-in errors make it much simpler to check the performance of the system with detailed display

statements. The system is robust and has enhanced scalability that allows the programmer to add multiple number of elevators with different number of floors with ease.

If the system design had to be redone, we would have reduced the workload from the Scheduler class because the current version of the scheduler does various tasks and if it fails the whole system crashes.

Since JVM shuts down all thread while GC running, that effects overall Hard-Time Deadline, the JVM GC (Garbage Collection) scheduling would have been an apocopate approach in terms of unused Object (Pair.Java) cleaning without thread standby for GC. Another approach could have been done to minimize system load is to use Boolean Array and to sort Heap Array, instead of Pair Object to transmit data in between Elevator – Scheduler – Floor. (M.Tareq 2019)

- Rate Monotonic Schedulability Analysis for the system:

RATE MONOTONIC SCHEDULABILITY ANALYSIS

Process/Task	Period, T (ms)	Priority, P	computation time, C (ms)	utilization, U
Arrival Sensor I/F	50	3	1.94	0.0388
Elevator Buttons I/F	100	2	2.34	0.0234
Floor Buttons I/F	200	1	0.62	0.0031

$$\sum_{i=1}^{N} \left(\frac{C_i}{T_i}\right) = \sum U_i = 0.0388 + 0.0234 + 0.0031$$

$$= 0.0653\%$$

$$N\left(2^{1/N} - 1\right) = 78\%$$

$$\therefore \sum_{i=1}^{N} \left(\frac{C_i}{T_i}\right) \leq N\left(2^{\frac{1}{N}} - 1\right)$$

∴ The Liu-Layland schedulability test passes and the processes are schedulable

- Conclusion:

The project has been completed with successful implementation of the project design as a Graphical User Interface for the Elevator simulator. The diagrams provide information about the

functioning and the different errors scenarios for the system. The scheduler class handles possible faults and failures in the system gracefully. The software has grown incrementally from iteration to iteration and developed into the final system.