

SYSC 3303: Iteration 4

Breakdown of Responsibilities – Iteration 1:

Ali Alvi:

Developed the Scheduler.java class.

Took part in code review sessions and approved outstanding pull requests.

Akaash Kapoor:

Developed the FloorSubsystem.java class.

Took part in code review sessions and approved outstanding pull requests.

Areeb Ul Haq:

Developed the ElevatorSubSystem.java class.

Took part in code review sessions and approved outstanding pull requests.

Hassan Jallad:

Developed unit tests to show the program reading the input file and passing the data between the different subsystems.

Took part in code review sessions and approved outstanding pull requests.

Raj Sandhu:

Developed UML diagram.

Developed Sequence diagram.

Wrote the README file.

Took part in code review sessions and approved outstanding pull requests.

Breakdown of Responsibilities – Iteration 2:

Ali Alvi:

Extended development for classes within the scheduler folder.

Developed the scheduler state machine.

Worked as a primary developer on scheduler and state machine diagram alongside Raj Sandhu.

Assisting in debugging general issues with the scheduler classes.

Took part in code review sessions and approved outstanding pull requests.

Akaash Kapoor:

Redesigned codebase to take advantage of the blocking queue data structure.

Performed debugging of code issues.

Assisted Raj with UML class diagram

Revised unit tests.

Took part in code review sessions and approved outstanding pull requests.

Areeb Ul Haq:

Extended unit test development.

Assisted in general debugging of the program.

Assisted Raj in designing sequence diagram.

Took part in code review sessions and approved outstanding pull requests.

Hassan Jallad:

Extended development for the classes within the elevatorSubsystem folder.

Developed the elevator state machine.

Performed debugging of code issues.

Developed elevator state machine diagram.

Took part in code review sessions and approved outstanding pull requests.

Raj Sandhu:

Developed UML diagram.

Developed Sequence diagram.

Worked with Ali to develop state machine diagram.

Wrote the README file.

Assisted Ali with the classes within the scheduler folder, performing code cleanups and assisting in debugging.

Took part in code review sessions and approved outstanding pull requests.

Breakdown of Responsibilities – Iteration 3:

Ali Alvi:

Extended development for classes within the scheduler folder such that the scheduler can now coordinate the movement of cars.

Assisted in debugging the program such that it worked successfully.

Revised scheduler state machine diagram to address feedback from iteration 2.

Took part in code review sessions and approved outstanding pull requests.

Raj Sandhu:

Developed UML diagram.

Developed Sequence diagram.

Wrote README.

Assisted in debugging the program such that it worked successfully.

Assisted lead scheduler developer Ali Alvi in code cleanup.

Akaash Kapoor:

Extended development for the classes pertaining to the floor subsystem.

Developed the timer class used to measure elevator arrival timings.

Led in debugging the program such that it worked successfully.

Took part in code review sessions and approved outstanding pull requests.

Hassan Jallad:

Extended development for the classes pertaining to the elevator subsystem.

Led in debugging the program such that it worked successfully.

Took part in code review sessions and approved outstanding pull requests.

Revised elevator state machine diagram to address feedback from iteration 2.

Areeb Ul Haq:

Extended development for the test classes related to the program.

Assisted in debugging the program such that it worked successfully.

Took part in code review sessions and approved outstanding pull requests.

Breakdown of Responsibilities - Iteration 4:

Ali Alvi:

Updated UML diagram to incorporate changes made in iteration 4.

Updated elevator state machine diagram to incorporate changes made in iteration 4.

Took part in code review sessions and approved outstanding pull requests.

Raj Sandhu:

Developed timing diagrams to showcase the various faults considered in iteration 4 (doors stuck open, elevator stuck between floors), as well as a successful run.

Wrote README.

Took part in code review sessions and approved outstanding pull requests.

Akaash Kapoor:

Extended development of the timer class used to measure elevator arrival timings.

Extended development for the classes pertaining to the elevator subsystem and the scheduler to detect and handle faults.

Took part in code review sessions and approved outstanding pull requests.

Hassan Jallad:

Extended development for the classes pertaining to the elevator subsystem and the scheduler to detect and handle faults.

Took part in code review sessions and approved outstanding pull requests.

Areeb Ul Haq:

Extended development for the test classes related to the program.

Took part in code review sessions and approved outstanding pull requests.

File Names:

The source code to successfully run this project is located within the src folder.

The classes representing the floor subsystem are located within the floorSubsystem folder (src/floorSubsystem) and are called FloorSubsystem.java, FloorAttributes.java, and FloorRequest.java.

The classes representing the elevator subsystem are located within the elevatorSubsystem folder (src/elevatorSubsystem) and are called ElevatorSubsystem.java, Elevator.java, and ElevatorRequest.java.

The classes representing the elevator subsystem states are located within the elevatorStates folder (src/elevatorStates) and are called ElevatorState.java, MovingDown.java,

MovingUp.java, Stationary.java, and Failure.java.

The classes which store and receive text data and read text input are stored in the input folder (src/input) and are called inputBuffer.java, and Reader.java respectively.

The classes representing the scheduler subsystem are located within the scheduler folder (src/scheduler) and are called Scheduler.java, ElevHandler.java, FloorHandler.java, SchedulerState.java, SchedulerStateMachine.java.

The class which is used to measure the elevator arrival timings is stored in the utils folder (src/utils) and is called Timer.java.

The input file that contains the test data to be read by the floor subsystem is located within the Passengers folder and is called testFile.txt.

The classes that contain the unit tests used to test the functionality of the program are located in the tests folder (src/tests) and are called InputTests.java, StateTests.java, SchedulerTests.java, and TimerAndFailureTests.java.

The UML class diagram that shows the class hierarchy and relationships is located within the Documentation folder and is called uml-class-diagram.png.

The Sequence diagram that shows the interactions of the threads between the three subsystems is located within the Documentation folder and is called sequence-diagram.png.

The scheduler state machine diagram which shows the different states that it may take on is located within the Documentation folder and is called scheduler-state-machine-diagram.png.

The elevator state machine diagram which shows the different states that it may take on is located within the Documentation folder and is called elevator-state-machine-diagram.jpg.

The timing diagrams which show the stuck between floors and doors stuck open faults, as well as a successful run is located within the Documentation folder and is called timing-diagrams.png.

This README file which outlines the individual responsibilities of the group members, information about file names, set up information, and test information is located within the Documentation folder and is called README.txt.

Note: the LaunchElevator.java class (src/launchElevator) is used for testing the program and is not to be used for running the code.

Set up Instructions:

Download the zip file submission from brightspace to a location of your choice.

Open up the Eclipse IDE.

Click File, Import, General, Projects from Folder or Archive, Next, then click Archive in the top right of the screen, find where you stored the zip file submission, select it, then hit finish.

Ensure Eclipse is correctly configured to run multiple main methods.

Run the Scheduler.java main method in its own console window.

Pin the current console window.

Create a new console window.

Run the ElevatorSubsystem.java main method.

Pin the second console window.

Create a new console window.

Run the FloorSubsystem.java main method.

Pin the third console window.

Testing Instructions:

Download the zip file submission from brightspace to a location of your choice.

Open up the Eclipse IDE.

Click File, Import, General, Projects from Folder or Archive, Next, then click Archive in the top right of the screen, find where you stored the zip file submission, select it, then hit finish.

Navigate to the InputTests.java file (src/tests/InputTests.java) and hit run.

Navigate to the StateTests.java file (src/tests/StateTests.java) and hit run.

Navigate to the SchedulerTests.java file (src/tests/SchedulerTests.java) and hit run.

Navigate to the TimerAndFailureTests.java file (src/tests/TimerAndFailureTests.java) and hit run.