ElevatorSimulator_SYSC3303

This project will aim to implement a multithreaded simulation of an elevator system

To Run

ElevatorSim is an eclipse project. To Run, import the project into Eclipse Java and run from eleveatorsim.Main. The Unit Tests for the project are implemented in JUnit5. Run them by right clicking on the project and selecting Run As -> JUnit Test.

Authors

- Rahul Anilkumar (101038785) - Worked on Message Classes, File Parser, JUnit Tests
    - Classes: MessageRequest, FileParser, MessageRequestUtil, FileParserTest, MessageRequestUtilTest
- David Wang (101032271) - Worked on Scheduler, Startup Logic, Sequence Diagram, Documentation
    - Classes: Main, MessageReciever, Scheduler
- Michael Patsula - Worked on Floor Subsystem
    - Classes: Direction, Floor, FloorButtons, FloorController
- Thomas Leung (101043255) - Worked on Scheduler Class, JUnit Tests, UML Class Diagram
    - Classes: Scheduler, SchedulerTest
- Trevor Bivi (101045460) - Worked on Elevator Subsystem, JUnit Tests
    - Classes: Elevator, MessageDestination, FloorTest

Classes

elevatorsim.Main - This class serves as the program entry point. It creates all the necessary threads and starts all the system.

elevatorsim.common.MessageRequest - This structure contains the data that will be sent between the floor subsystem and elevator subsystem.

elevatorsim.common.MessageReciever - This interface contains a method stub that all classes that are meant to recieve MessageRequests is expected to implement

elevatorsim.elevator.Elevator - This class models the brains of the elevator system. In the current iteration all it does is send back all messages it recieves elevatorsim.enums.Direction - This enumeration contains the directions that an elevator can go. Currently has UP, DOWN, and INVALID directions.

elevatorsim.enums.MessageDestination - This enumeration contains the destinations that a MessageRequest can go to. Currently has ELEVATORS and FLOORS.

elevatorsim.floor.Floor - This class models a floor in the building. It keeps track of what requests have been raised by the floor.

elevatorsim.floor.FloorButtons - This class models the buttons on the floor, specifically the up and down elevator request buttons. They are set and unset as the Floor reads and sends requests.

elevatorsim.floor.FloorController - This class models the logic of the floor system. It sends and recieves requests from the scheduler.

elevatorsim.scheduler.Scheduler - This class models the scheduler system of the elevator. Currently allows sending of messages back and forth between the elevators and floor subsystem.

elevatorsim.test.FileParserTest - Junit test class to test the FileParser class methods

elevatorsim.test.FloorTest - Junit test class to test the Floor class methods

elevatorsim.test.MessageRequestUtilTest - Junit test class to test the MessageRequestUtil class' static methods

elevatorsim.test.SchedulerTest - Junit test class to test the Scheduler class methods

elevatorsim.util.FileParser - Utility class that helps us parse MessageRequests from text files

elevatorsim.util.MessageRequestUtil - Utility class that helps us manipulate the map of MessageRequests that is read in from the FileParser

UML Class and Sequence Diagrams

The UML diagrams can be found in the Diagrams/Iteration 1 directory

Iteration1_Sequence.PNG - The sequence diagram depicting the interactions between threads

Iteration1_Class_Diagram.PNG - The class diagram depicting the structure of the ElevatorSim project