SYSC 3303 Group Project
Iteration 1

Tristan Demers
Toman Aleksiev
Kenji Isak Laguan
Steven Johnson
Irina Ionescu

Group member name	**Contribution**
Tristan Demers	Coded Scheduler state machine along with helping Elevator state machine and Scheduler state machine interaction.
Toman Aleksiev	Contributed to scheduler state class. Updated README. Created contributions document.
Kenji Isak Laguan	Coded Elevator State class and corresponding Elevator functionality. Contributed to Scheduler State class and communication between both systems. Fixed arising communication/behavioral bugs.
Steven Johnson	Class and sequence diagram. Included detailed setup and test instructions, including test files used.
Irina Ionescu	Code: contributed to elevator state (message sequence) and scheduler state classes, Diagrams: state machine diagram, contributed to scheduler state machine diagram

Note: Set up instructions and testing instructions are located in the “README.txt” file

1. Files included:

```
## List of files
ElevatorSimulator
├── README.txt
├── pom.xml
└── src
    ├── main
    │   ├── java
    │   │   ├── InputFileMaker.java
    │   │   └── primary
    │   │       ├── Elevator.java                    - elevator thread class
    │   │       ├── ElevatorState.java               - elevator state class
    │   │       ├── Floor.java                       - floor thread class
    │   │       ├── Mailbox.java                     - thread-safe singleton class that receives messages, holds them in separate
    │   │       │                                        queues (1 per thread) according to recipient, and releases them to the recipient
    │   │       ├── Message.java                     - message that corresponds to 1 line of input from the input.txt file
    │   │       ├── MessageType.java                 - types of messages that will be sent; used when handling different messages
    │   │       ├── MovingDirection.java             - types of directions; used for handling different cases
    │   │       ├── Scheduler.java                   - scheduler thread class
    │   │       ├── SchedulerState.java              - scheduler state class
    │   │       └── main.java                        - main class
    │   └── resources
    │       ├── input.txt                            - input file for the application with multiple requests
    │       └── inputOneLine.txt                     - input file for the application with one request
    └── test
        ├── java
        │   └── ECSTest.java                         - JUnit test class
        └── resources
            ├── inputMessageSendOnePassSingleFloor.txt   - text files containing one request
            └── inputReadFromFileMultipleFloors.txt      - text file containing multiple requests
```

2. Setup instructions -
   - Open project in Intellij and download any necessary libraries, sdk, etc. when prompted by the IDE
   - In IntelliJ, open ElevatorSimulator as a project
   - select on the main.java file
   - Click the green arrow at the top right; this will build and run the project
   - Output can be seen in the console
3. Testing instructions - find “README.txt”
   Manual testing:
   - run application as described in ## Build and run
   - the floor subsystem class outputs the messages that were read from the input.txt file
   - observe output of the floor subsystem, elevator and scheduler threads as they pass a message back and forth

4. Class diagram - find “UMLClassDiagram.png” in Diagrams folder
5. Sequence diagram - find “UMLSequenceDiagram.png” in Diagrams folder
6. State Machine diagram - find “StateMachineDiagram.pdf” in Diagrams folder