 # Elevator Simulation

Branch Naming Policy

Name the branch "iteration#-the feature you are working on"

After code review, it can be pushed to the branch "iteration#"

Authors

- Joshua Fryer

- Tanvir Hossain

- Mattias Lightstone

- Yu Yamanaka

- Xinrui Zhang

#Message Queue

We decided to use an open source messaging queue called RabbitMQ..

- We have the message queue running as its own node.

- Our nodes (ElevatorSystem, Scheduler, FloorSystem) are all subscribed to different queues which are named in the config.properties file (eg. "elevator", "scheduler", "floor")

- When you want to send a message, you specify the queue you want to add it to and send that message to the rabbitmq node

- When a message is added to a queue that has a subscriber, that subscriber is notified, and consumes the message in the queue. That message is then removed from the queue.

- The format that messages are sent in is JSON, it allows to use whatever client-side language we wanted to.

```
docker pull rabbitmq
```

```
docker run -d -p 5672:5672 -h myrabbit --name this-rabbit  rabbitmq:3
```

 # Running with Maven (RECOMMENDED, if possible)

Please note that you need to run the RabbitMQ server first.

This project can be run by navigating to the directory where the project is extracted and executing

`./start.sh` in the Linux terminal, or `start.bat` in the Windows command prompt.

If you do not have Maven installed, follow the directions [here](https://maven.apache.org/guides/getting-started/windows-prerequisites.html).

 # Running from Eclipse

Follow the instructions [here](https://www.tutorialspoint.com/maven/maven_eclispe_ide.htm) to import and run the project. Eclipse has native Maven support, so this should work.
Your Import window, before you click Finish, should look like this:

 # Subsystems

This Elevator System Consists of four subsystems:

Floor System

FloorSystem is responsible for simulating passenger actions on the floors of a building, and the operation of lamps. It contains an array of Floor objects, each representing an individual floor. It takes button inputs from each floor (implemented in Iteration 1 as messages from the Simulator) and sends messages to the Scheduler using a class called MessageHandler.
It also controls the lamps on each floor, which indicate the direction in which elevators are moving, and which turn off when an elevator arrives.

Scheduler System

The Scheduler system acts as an intermediary between the Floor and Elevator systems, taking input from both systems and coordinating the elevator.
Messages from the Floor system represent passengers requesting an elevator from a given floor.
Messages from the Elevator system represent either passengers pressing a button from within the elevator to request a destination, or a notification that the elevator has arrived at a floor. This notification occurs upon arrival at _any_ floor, and its purpose is to update the scheduler's representation of the system's state.
It handles the error scenario for door stuck and Elevator Stuck.

Elevator System

The Elevator system represents the physical elevator. It receives a message from the Scheduler, which instructs it what floor to travel to, and will move towards that floor until it reaches its destination, or is interrupted by a new instruction from the Scheduler.

Simulator System

The Simulator system reads from the provided input file and sends messages using the read data to simulate button presses within the Floor and Elevator systems.

 # Directory Structure

- Documentation such as diagrams are located in /doc.
- Each module's source code is located in <packagename>/src/main.java.

From there, the subdirectories follow a structure commonly used in Java programs, and particularly Android applications, wherein the directories spell out a reversed domain name (in this case, com.sysc3303). Within this domain package are the .java source files for that module.

- Additionaly, the Shared-Utils directory contains sources for classes that are used by multiple modules, such as most of the communications code.