# Part 1 – Use Cases

## Use Case 1: Administrator Configures and Runs Simulation

**Primary Actor:** Elevator Administrator
**Scope:** Elevator System Simulator
**Level:** User goal
**Stakeholders and Interests:**

- Elevator Administrator: Wants a properly working elevator system simulator available for usage in the real world.
- Building Owner: Wants a simulated elevator system that meets operational needs, testing, and any other necessary qualifications.
- Raven Elevators Inc. (REI): Wants a working, reliable elevator system simulator to validate the planned system design and functionality.

**Precondition**: The simulation software is using Qt C++, ready to be configured.
**Minimal Guarantees:** Basic simulation specifications are met.
**Success Guarantees:** Fully functioning elevator system simulator with all required use cases handling passenger requests and safety events.
**Trigger:** The Elevator Administrator launches the simulator.
**Main Success Scenario:**

1. The Elevator Administrator selects the number of floors and number of elevators, in accordance with the specifications.
2. The administrator specifies the number of passengers and their behavior, with specific time steps.
3. The Administrator defines safety events with specific time steps.
4. Once everything has been set up, the Elevator Administrator starts the simulator by pressing the “Start” button, and the elevator system simulator begins.
5. While the simulator is running, the GUI log displays the simulation time and elevator states (e.g., Idle, Doors Open). It also shows active safety conditions, if applicable.
6. The simulation advances one time step at a time, handling all passenger requests and safety events as they occur. The Elevator Administrator observes the simulation during this time.
7. Once all requests and safety events are handled, the simulation ends, and all elevators return to an idle state.

**Extensions:**
5a. The Elevator Administrator pauses the simulator:

a. Administrator presses the “Pause” button, stopping the simulator in its current state.
b. Administrator presses the “Continue” button to resume the simulator.
c. Resume at step 5.

5b. The Elevator Administrator stops the simulator:

a. Administrator presses the “Stop” button, ending the simulation early.
b. All elevators return to idle, and the simulation terminates. Everything is reset back to original state.

## Use Case 2: Passenger Requests Elevator and Travels to Destination

**Primary Actor:** Passenger(s) in the simulation
**Scope:** Elevator System in the Elevator System Simulator

**Level:** User goal

**Stakeholders and Interests:**

- Passenger(s) in the simulation**:** Want to reach their destination floor efficiently, quickly, and safely.
- Elevator Administrator**:** Wants to observe the functionality of the system in response to passenger-initiated actions and safety scenarios.
- Raven Elevators Inc. (REI): Wants proof that the elevator system simulator can efficiently handle requests from passengers.
- Building Owner**:** Wants reliable elevator service for passengers.

**Precondition:** Simulation is set up by Elevator Administrator with the number of floors, elevators, and passengers.

**Minimal Guarantees:** Elevator responds to every floor request.

**Success Guarantees:** Passenger is transported from their starting floor to their destination floor, and the elevator returns to an idle state following completion.

**Trigger:** A passenger presses the “up” or “down” button to travel upward or downward in the building.

**Main Success Scenario:**

1. The passenger presses the “up” or “down” button on their current floor and waits for the elevator.
2. The pressed button illuminates and remains lit until an elevator arrives.
3. The elevator control system assigns an available elevator to the request.
4. The assigned elevator moves to the passenger’s current floor, with its location and state displayed the GUI as it moves through each intermediate floor.
5. Once the elevator arrives at the correct floor, it rings a bell, opens its doors, and the pressed “up” or “down” button becomes unlit.
6. The passenger enters the elevator and selects their destination floor using the number panel inside the elevator.
7. The elevator doors close after 10 seconds.
8. The elevator moves to the destination floor, with its location and state displayed the GUI as it moves through each intermediate floor.
9. Once the elevator arrives at the destination floor, it rings the bell, opens its doors, and the passenger exits.
10. The doors close after 10 seconds, and the elevator returns to an idle state, as shown in the log console.

**Extensions:**

3a. No elevators available:

a. The request is queued until an elevator becomes available.

b. Resume at step 3 once an elevator is available and assigned to the request.

7a. Passenger presses the “open door” button:

a. The doors remain open as long as the button is held beyond the default 10 seconds.

b. Once the button is released and the doors close after 10 seconds, resume at step 7.

7b. Passenger presses the “close door” button:

a. The doors close immediately.

b. Resume at step 8.

8a. Safety event occurs during travel:

a. The elevator follows the relevant safety procedure (either Use Case 3 or Use Case 5).

b. After the safety event is resolved, resume at step 10.

## Use Case 3: Fire Alarm Triggers Emergency Procedure

**Primary Actor:** Elevator Control System
**Scope:** Elevator System in the Elevator System Simulator
**Level:** System goal
**Stakeholders and Interests:**

- Passenger(s) in the simulation**:** Want to be safely evacuated during an emergency.
- Elevator Administrator**:** Wants to verify that safety features function correctly.
- Raven Elevators Inc. (REI)**:** Wants proof of proper safety procedures taken during a fire.
- Building Owner**:** Wants all passengers to have the assurance of safety during emergencies.

**Precondition:** Simulation is running properly (Case 1 is fulfilled).
**Minimal Guarantees:** Elevators respond to the fire alarm signal.
**Success Guarantees:** All elevators move to a safe floor, passengers leave the elevator(s), and elevators remain idle until the emergency is resolved.
**Trigger:** The control system receives an alarm signal.
**Main Success Scenario:**

1. The control system detects the “Fire” alarm signal at a predefined simulation time step.
2. The control system commands all elevators to move to a designated safe floor.
3. The elevator’s display shows a text message, and the audio system announces: “Emergency detected! We are moving to a safe floor; please disembark when the doors open.”
4. As each elevator moves, its location and state (moving) are displayed in the log on the GUI.
5. Upon arrival at the safe floor, the elevator rings a bell and opens its doors.
6. Passengers exit the elevator.
7. The doors close after 10 seconds, and the elevator remains idle, with its state logged as “idle” and the safety condition marked as active.
8. The simulation continues until the fire event is resolved or the administrator stops it.

**Extensions:**

2a. Elevator is already on the safe floor:
    a. The elevator skips movement
    b. Simulator proceeds to step 5.

## Use Case 4: Door Obstacle Delays Elevator Closing

**Primary Actor:** Elevator Control System
**Scope:** Elevator System in the Elevator System Simulator
**Level:** System goal
**Stakeholders and Interests:**

- Passenger(s) in the simulation**:** Want uninterrupted use of the elevator.
- Elevator Administrator**:** Wants to ensure safety mechanisms function correctly.
- Raven Elevators Inc. (REI)**:** Wants proof that an elevator does not close until there are no obstacles.
- Building Owner**:** Wants safety and system reliability for all passengers.

**Precondition:** An elevator has arrived at a floor, and its doors are open.
**Minimal Guarantees:** The doors do not close on an obstacle.
**Success Guarantees:** The doors reopen when an obstacle is detected, and a warning is issued if the issue persists.
**Trigger:** The elevator fails to close after the default 10 seconds after an attempt to close.
**Main Success Scenario:**

1. The elevator doors are open for 10 seconds after passengers have entered or exited.
2. After 10 seconds, the doors begin to close.
3. The elevator sensor detects an interruption.
4. The control system stops the closing action and fully reopens the doors.
5. The doors attempt to close again after 10 seconds.
6. The doors close successfully, and the elevator proceeds to the next request.

**Extensions:**

3a. The sensor is interrupted repeatedly (i.e. three times):

a. The control system triggers a warning with the audio system: “Please clear the doorway for the doors to close” with a similar display message explaining the detected door obstruction

b. The doors remain open until there are no more obstacles

c. Resume at step 5.

### Use Case 5: Passenger Presses Help Button

**Primary Actor:** Passenger(s) in the simulation
**Scope:** Elevator System in the Elevator System Simulator
**Level:** User goal
**Stakeholders and Interests:**

- Passenger(s) in the simulation**:** Want assistance in an emergency or issue.
- Elevator Administrator**:** Wants to verify the help feature works as specified.
- Raven Elevators Inc. (REI)**:** Wants proof of robust safety features.
- Building Owner**:** Wants accessible safety and support for elevator passengers.

**Precondition:** Simulation is running, and at least one passenger is inside an elevator.
**Minimal Guarantees:** The help button triggers a response from the system.
**Success Guarantees:** The passenger is connected to building safety or emergency services.
**Trigger:** A passenger presses the “Help” button inside the elevator.
**Main Success Scenario:**

1. The passenger presses the “Help” button.
2. The control system receives the “Help” alarm signal and logs it as an active safety condition.
3. The system initiates a voice connection to the building safety service.
4. Within 5 seconds, building safety responds, and the passenger communicates their issue.

**Extensions:**

4a. No response from building safety within 5 seconds:

a. The system places a 911 emergency call.

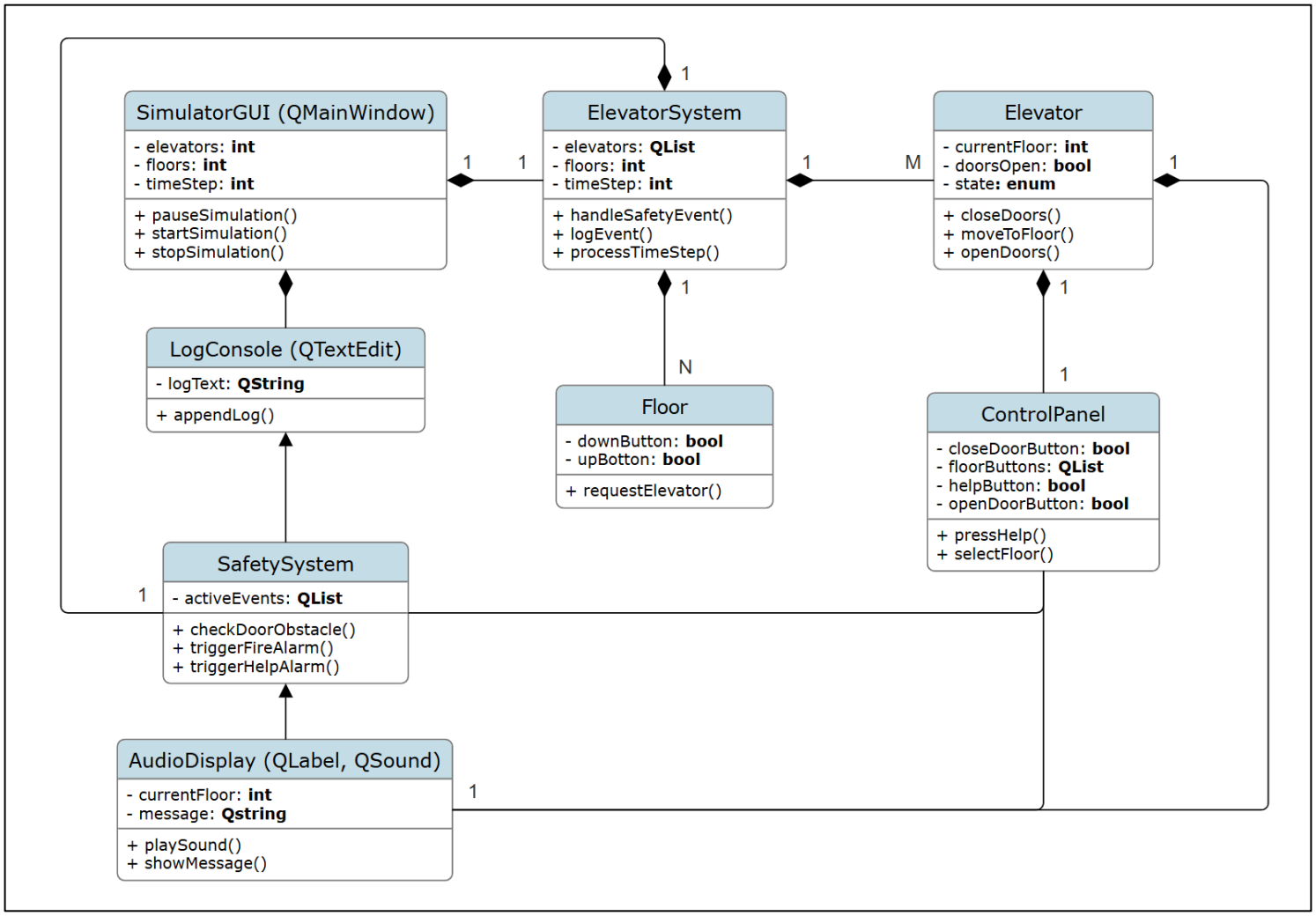
b. The passenger is connected to emergency services.

4b. No response from the passenger after connection:

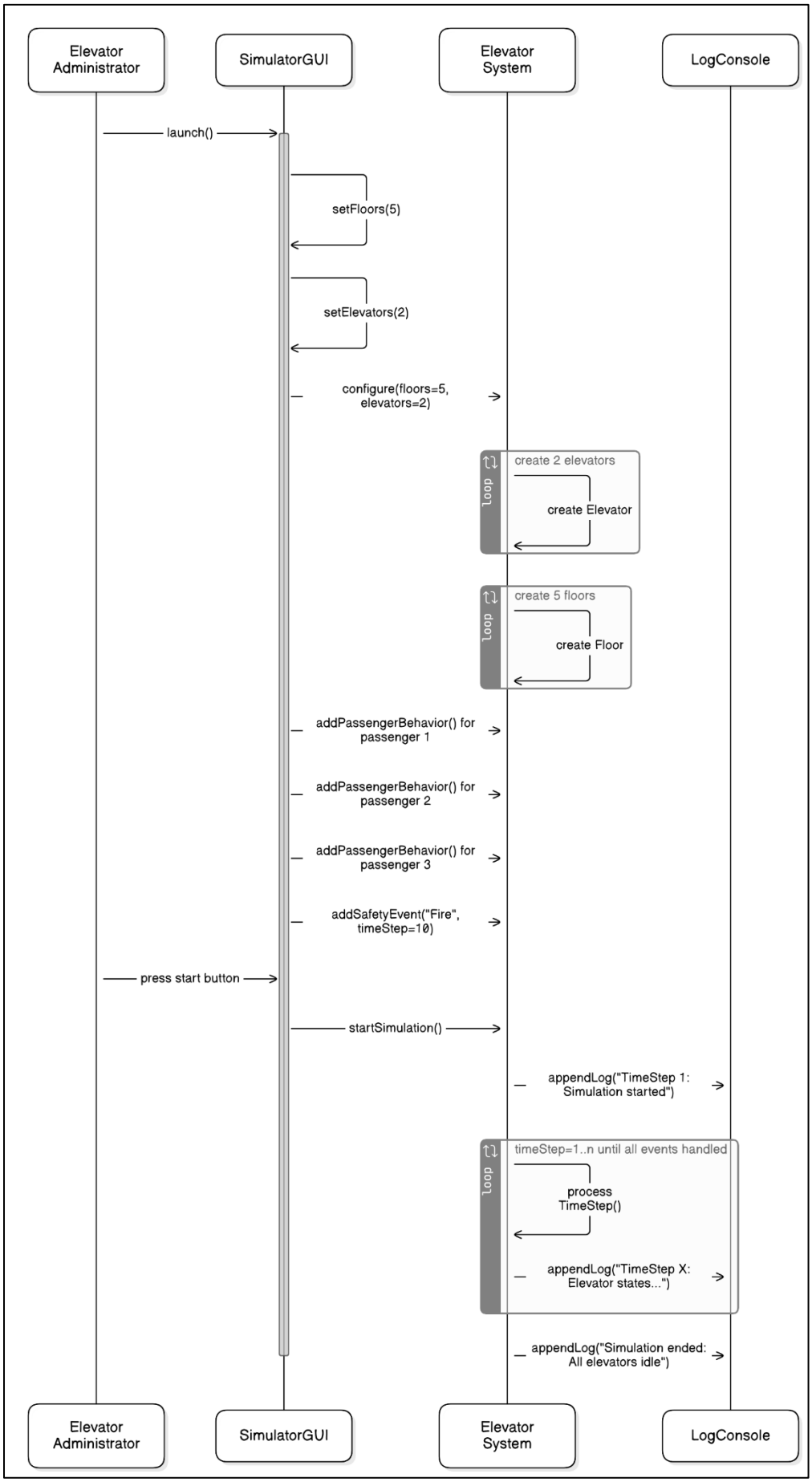
a. The system places a 911 emergency call after a brief timeout.

# Part 2 – Design Documentation

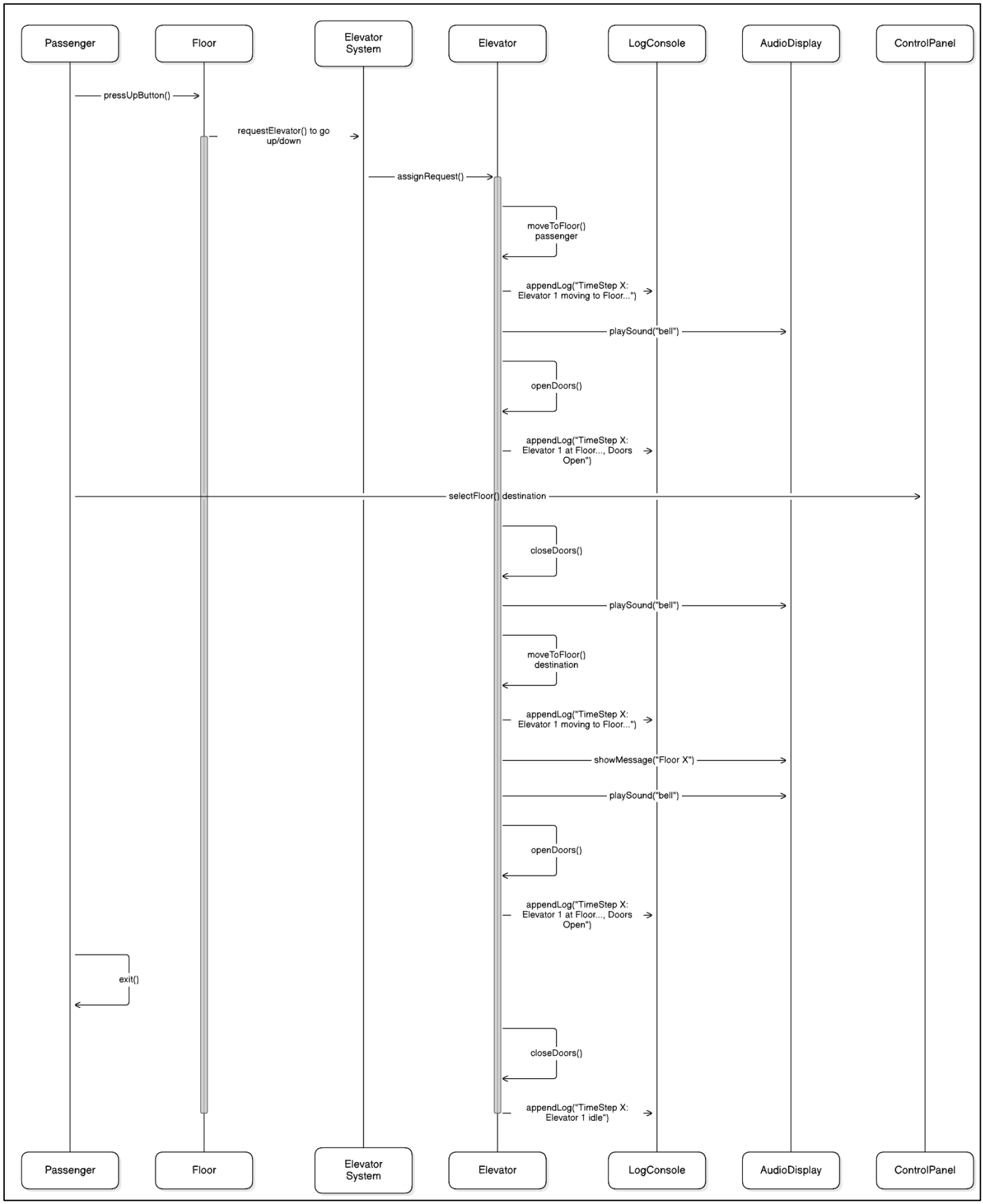
## UML Class Diagram



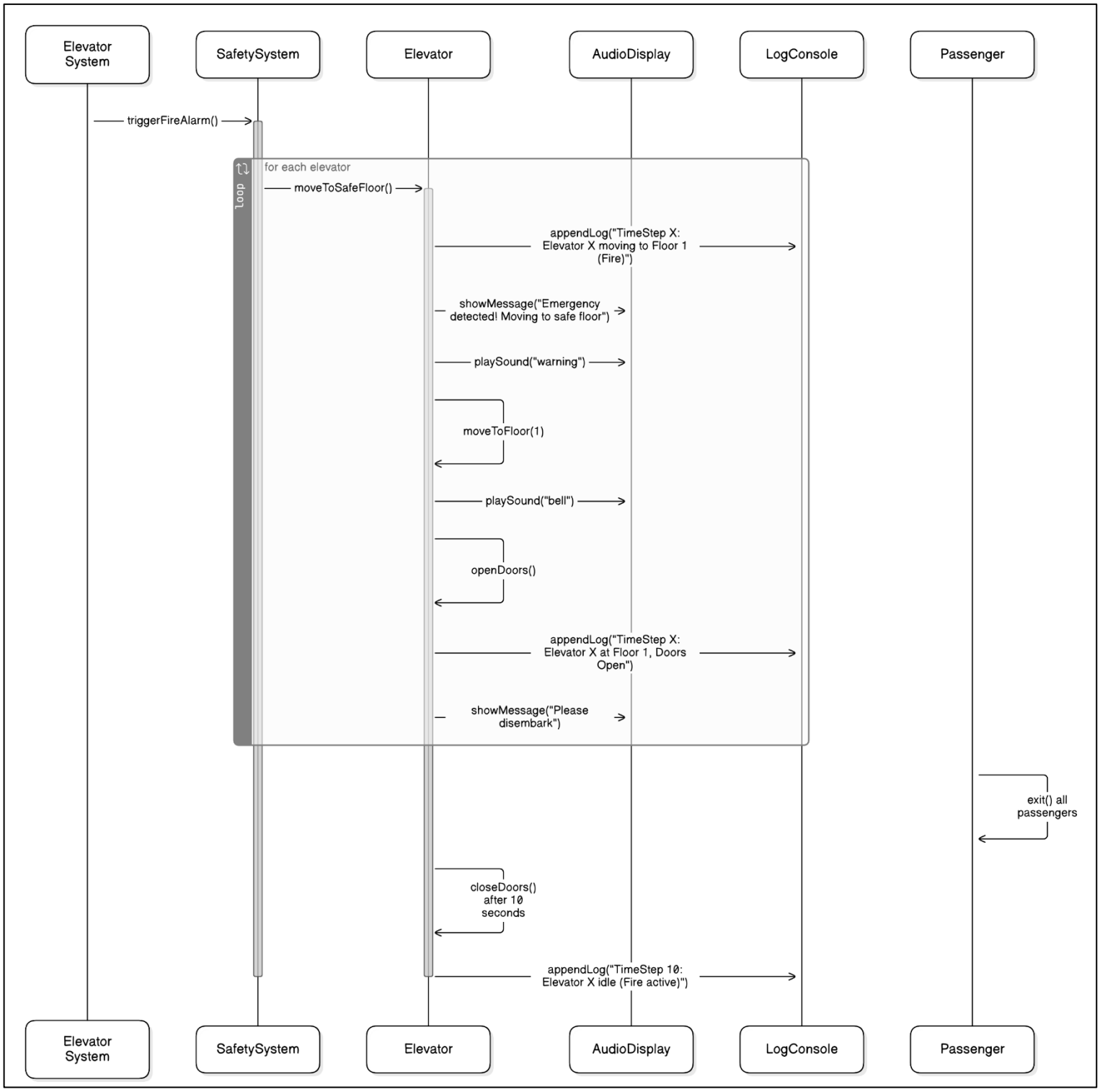
## Success Scenario 1: Administrator Configures and Runs Simulation - Use Case 1



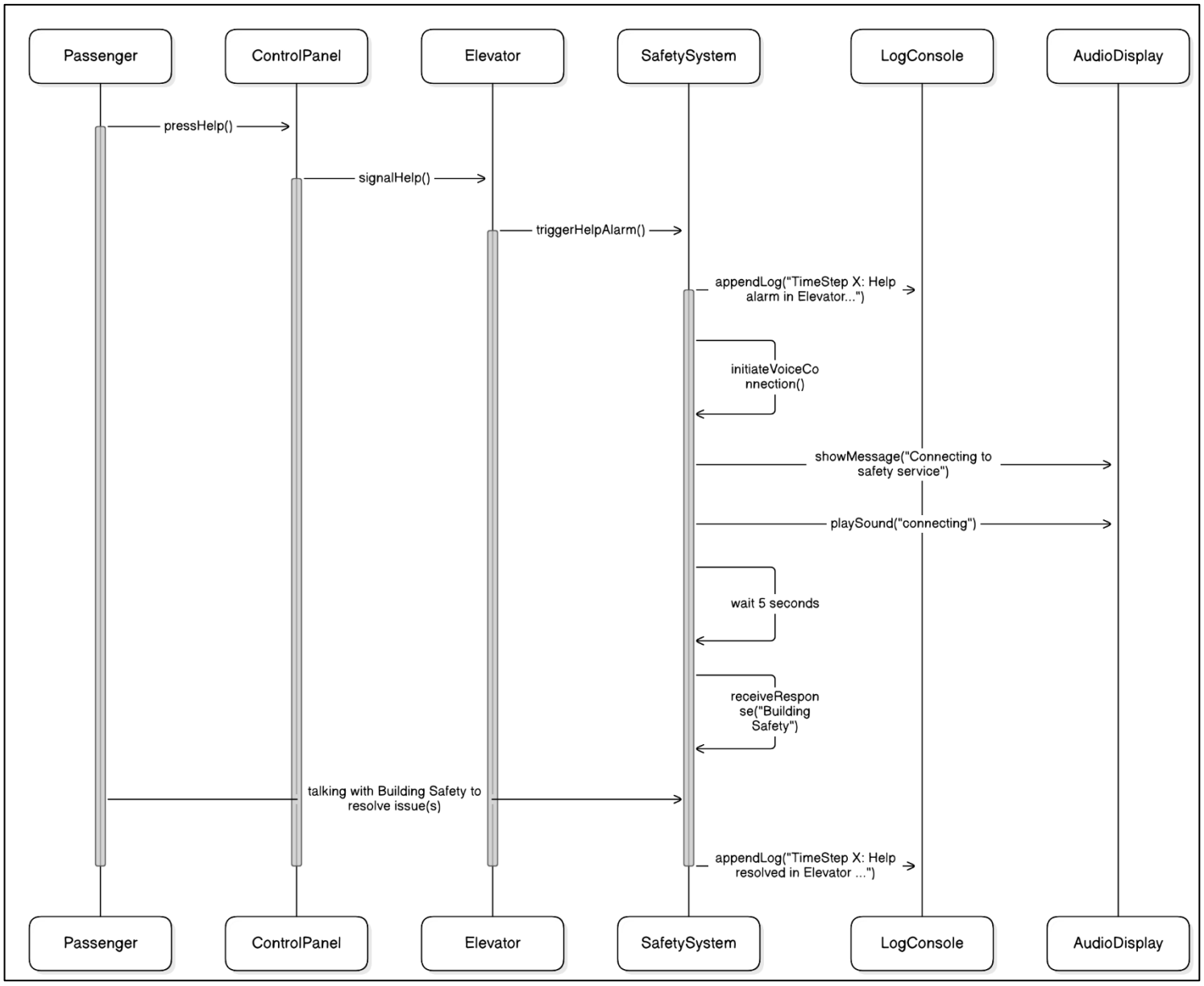
## Success Scenario 2: Passenger Requests Elevator and Travels - Use Case 2



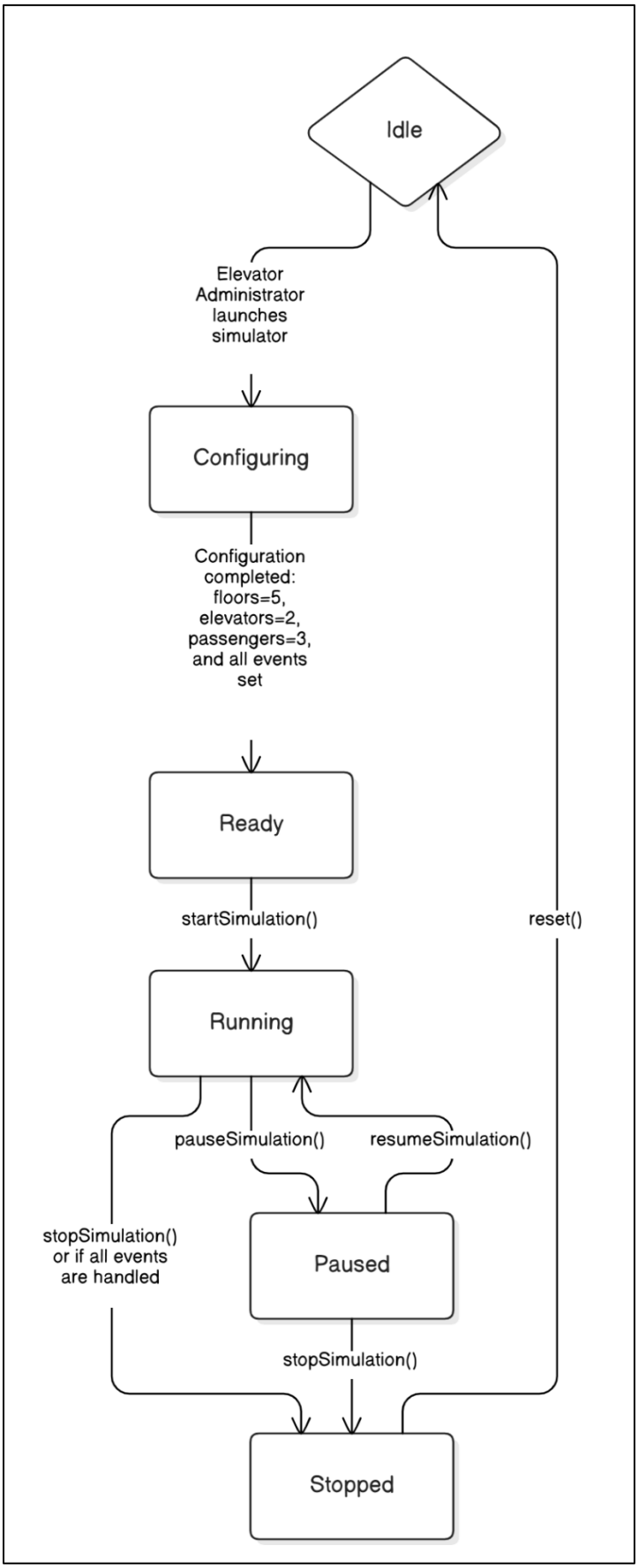
## Safety Scenario 1: Fire Alarm Triggers Emergency - Use Case 3



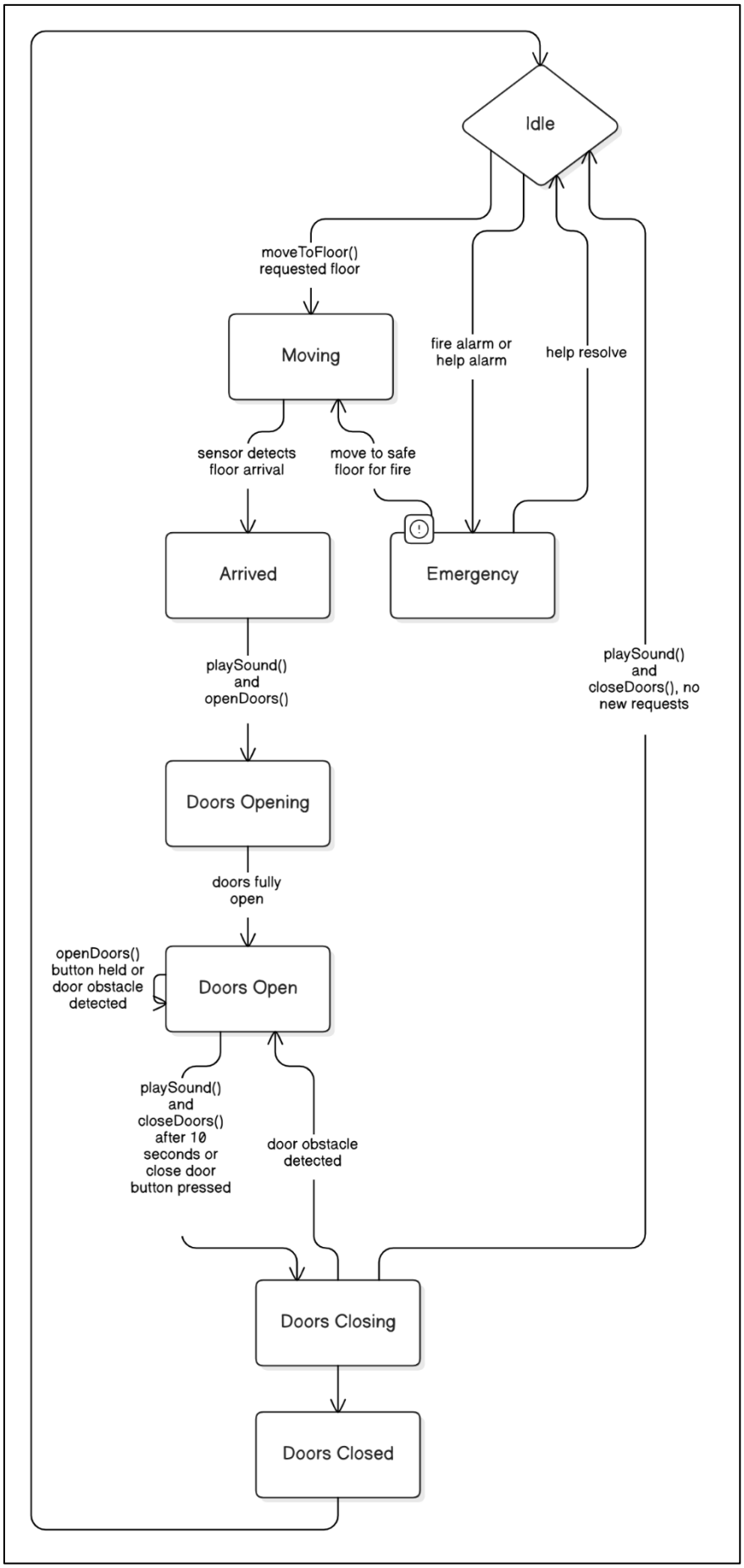
## Safety Scenario 2: Passenger Presses Help Button - Use Case 5



## UML State Machine Diagram for Simulation Controller (ElevatorSystem)



## UML State Machine Diagram for an Elevator



## Textual Explanation of Design Decisions

I designed the system to have all the main parts outlined in the instructions:

- SimulatorGUI: The Elevator Administrator can see the number of floors, elevators and passengers. There are also buttons to control the simulator.
- ElevatorSystem: This keeps track of all the elevators and floors, movement upward/downward and passenger requests or safety issues.
- Elevator: Passengers use this to travel through the building
- LogConsole: Text box on the screen to show the logs, which lets the Elevator Administrator know what is happening in real time.
- Floor: Contains passenger buttons to request an elevator.
- ControlPanel: Buttons inside the elevator that passengers may press (i.e., open door, help)
- SafetySystem: This system handles emergencies defined in some use cases and success scenarios above.
- AudioDisplay: Shows the floor number, important messages and plays sounds.

The SimulatorGUI starts everything and is constantly communicating with the ElevatorSystem. The elevator system controls each of the elevators and up/down buttons on the floors. The LogConsole gets real time updates as the simulation is occurring, which lets the administrator know what is going on. These messages come from the elevator system. In emergency situations or when the help button is pressed, the SafetySystem gives information and helps passengers in the elevators so that they know what to do. Any sounds or visual messages the passengers need (according to the specifications) use the AudioDisplay. The GUI overall is simple and straight to the point.

I assumed one thing not addressed in the instructions. For emergencies, I chose Floor 1 as the safe floor since it was most logical and would make the most sense in the case of a fire for example. Additionally, when someone presses the help button, I assumed the building safety team answers in 5 seconds as the success case.

# Part 3 – Implementation

The code is on GitHub, found at this link: https://github.com/oluwadara03/Elevator-System-Simulator

# Part 4 – Video

Here is the link to the video: https://youtu.be/bzHdC2nz8qI

# Part 5 – Traceability Matrix

| ID | Requirement/Feature | Design Element | Related Use Case | Tested By | Implemented By |
|---|---|---|---|---|---|
| 1 | Elevator Administrator selecting the number of floors, elevators | GUI | <Paragraph 1> and Use Case 1 Step 1 | Drop down boxes for both parameters, then pressing “Initialize” | SimulatorGUI |
| 2 | Elevator Administrator selecting the number of passengers with their behaviour and time step | GUI | <Paragraph 1> and Use Case 1 Step 2 | A combination of sliders, drop down boxes, check boxes and text fields for the parameters, then pressing "Set Passenger" | SimulatorGUI |
| 3 | Elevator Administrator selecting safety events and their time steps | GUI | <Paragraph 1> and Use Case 1 Step 3 | Pressing any of the following buttons on the GUI: “Fire Alert”, “Power Failure”, “Help” or “Overload” | SimulatorGUI |
| 4 | Simulation runs properly, handling the “Stop”, “Pause” and “Start” events | GUI | <Paragraph 1>, Use Case 1 Step 4 to Step 7, UML State Machine Diagram for Simulation Controller | Pressing any of the following buttons on the GUI: “Stop”, “Pause” or “Start” | SimulatorGUI |
| 5 | Log console displaying important information to the Elevator Administrator | GUI | <Paragraph 1> and Use Case 1 Step 5 | Viewing logs on the GUI | SimulatorGUI |
| 6 | Display current timestep during simulation | GUI | <Paragraph 1>, Use Case 1 Step 6 | Viewing logs on the GUI | SimulatorGUI |
| 7 | Display elevator location and state, and any possible active safety conditions | GUI | < Paragraph 1>, Use Case 1 Step 6 | Viewing logs on the GUI | SimulatorGUI |
| 8 | Building having M elevators, in a building with N floors, with up/down buttons | Simulation Controller, Elevator | <Paragraph 2>, UML Class Diagram | Initialized by the Elevator Administrator, and viewing the simulator response | SimulatorGUI |
| 9 | The elevator arrives, rings its bell, opens doors for 10 seconds, allowing for passenger entry/exit and then finally closes once finished | Elevator | <Paragraph 2>, Use Case 2 Steps 5-10 | Viewing logs on the GUI | SimulatorGUI |
| 10 | Passengers inside the elevator select a destination floor using the number panel | GUI, Elevator | <Paragraph 2>, Use Case 2 Step 6 | Entering the floors into the GUI and waiting for the output by the console log | SimulatorGUI |

| | | | | | |
|---|---|---|---|---|---|
| 11 | Pressing the open door or close door button, causing the default timing to extend | GUI, Elevator | <Paragraph 2>, Use Case 2 Extensions 7a and 7b | Checking the “Open Door” or “Close Door” button on the GUI | SimulatorGUI |
| 12 | Help button links the passengers to building safety | GUI, Safety System | <Paragraph 2>, Use Case 5 | Pressing "Help" button and seeing a connection within five seconds according to the console log | SimulatorGUI |
| 13 | Elevator arrival at the destination floor | Elevator | <Paragraph 3>, UML State Diagram | Viewing logs on the GUI | SimulatorGUI |
| 14 | Warning passengers using the audio and textual feedback on the elevator | Elevator | <Paragraph 4> | Viewing any logs relating to a warning on the GUI | SimulatorGUI |
| 15 | Door obstacles cause the elevator to reopen, with a warning if repeated multiple times | Safety System, Elevator | <Paragraph 6>, Use Case 4 Extension 3a | Checking the “Obstruct” button on the GUI | SimulatorGUI |
| 16 | A fire causes the fire safety scenario/condition, where the elevator helps passengers to evacuate safely. | Safety System, Elevator | <Paragraph 7>, Use Case 3, Safety Scenario 1 | Pressing the "Fire Alert" button and viewing the log for its reaction | SimulatorGUI |
| 17 | An overload of too many passengers triggers the overload safety scenario/condition, where the elevator does not move and alerts the passengers of the issue | Safety System, Elevator | <Paragraph 8> | Pressing the "Overload" button and viewing the log for its reaction | SimulatorGUI |
| 18 | Losing power would cause the power outage scenario/condition, where the elevator would move to safe floor and help passengers evacuate safely. | Safety System, Elevator | <Paragraph 9> | Pressing the "Power Failure" button and viewing the log for its reaction | SimulatorGUI |