

## Use cases

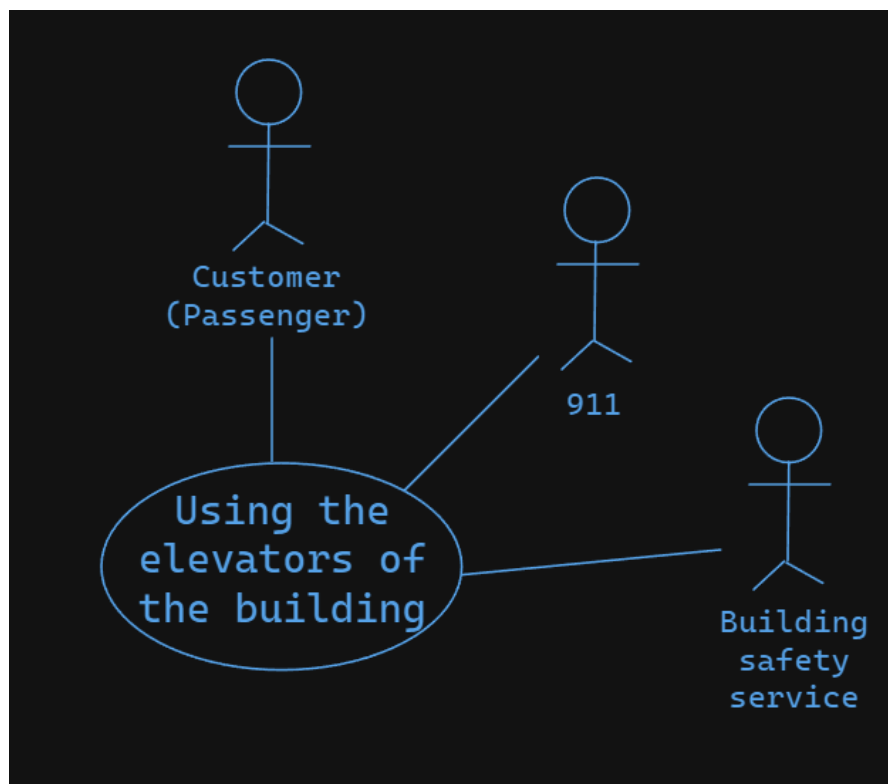
---

- Name: Using the elevators of the building
- Primary Actors: Any number of customers on each floor, each wanting to reach a target floor.
- Stakeholders and Interests: Customers, passengers (customers onboard an elevator), the owner of the building, the building safety service
- Pre-condition(s): All elevators (henceforth synonymous with (elevator) cars) are fully functional, including basic functions, audio systems, display, floor sensors, and safety features. (Extensions for elevator malfunction are beyond the scope of the specification given.)
- Success guarantee (equivalently Post-condition): All customers reach the desired floors.
- Main success scenario:
  - 1: Groups of customers on a floor press the up/down buttons of the floor they are on, depending on whether their target floor is higher or lower. The pressed buttons stay illuminated.
  - 2: The system sends idle cars (cars with no destination and no alarm signal) to the closest floors with pressed buttons, one car per pressed button (up/down). Cars moving to a target floor will make stops on the way for floors.
  - 3: A car arrives at a floor and informs of the current floor via audio and display, and de-illuminates the directional buttons of the floor.
  - 4: The bell is rung and the door to the car opens for 10 seconds so that customers can leave or board the car.
  - 5: The bell is rung and the door to the car closes.
  - 6: The passengers select buttons to go to their destinations.
  - 7: The car moves to the target floor.
  - 8: Once a target floor is reached, a car will behave like step 3. This process is repeated in parallel for all cars until all customers reach their destination (no floors have active buttons).
- Extensions:
  - 3a. The floor had both directions pressed.
    - 3a1. The system should prioritize the closer floor, or the lower floor in case of a tie.
  - 3b. The de-illuminated direction button is pressed again.
    - 3b1. While there is a car on the floor corresponding to the direction, ignore the input, and do not illuminate it.
  - 4a. The passenger or cargo load exceeds capacity (the system receives an "Overload" alarm signal from a specific car).
    - 4a1. Present an audio and text message to inform passengers to reduce the load. Keep the doors open, do not move the car.
    - 4a2. Once load is reduced, resume normal operation.
  - 4b. The "close door" button is pressed.
    - 4b1. Skip the door timeout and go to step 5.
  - 4c. The "open door" button is held.

- 4c1. While it is held, keep doors open, so do not progress to step 5 even after 10 seconds have passed in step 4. Prioritize it over "close door".
- 5a. There is an obstacle detected between the doors.
  - 5a1. If this has happened repeatedly over a short period of time, sound a warning over the audio system and display a text message.
  - 5a2. Stop the door from closing and open it, return to step 4.
- 5b. "open door" is pressed while the door closes.
  - 5b1. Stop the door from closing and open it. Return to step 4.
- 6a. A button that was pressed is pressed again.
  - 6a1. Cancel the floor as a target. The elevator will stop at the closest floor it is safe to do so if there are no other destinations selected in the car.
- 7a. A floor button is newly pressed during movement.
  - 7a1. Honor the new target floor, unless the system identifies that the car cannot stop at the new target without a rapid brake (target is too close while the car was moving too fast).
- 7b. The "open door" or "close door" button of the car is pressed.
  - 7b1. Do not acknowledge their button presses during car movement.
- 7c. The control system receives a "Fire" alarm signal.
  - 7c1. Play an audio and text message informing passengers of the emergency and asking them to disembark once the safe floor is reached.
  - 7c2. For all cars affected by the signal (all cars if signal is from the building; if from a car pressing the "Fire" button, only that car), cancel all selected floors and move car to the nearest safe floor.
  - 7c3. Once safe floor is reached, open doors and maintain that state until "Fire" signal is gone. Do not move the car.
  - 7c4. Once there is no "Fire" signal affecting the car, resume normal operation.
- 7d. The control system receives a "Help" alarm signal from a car. (The "Help" button has been pressed.)
  - 7d1. Connect the passenger to building safety service through a voice connection.
  - 7d2. If there is no response from building safety within 5 seconds, or if there is no response from the passenger after connection, a 911 emergency call is placed.
- 7e. The control system receives a "Power Out" alarm signal.
  - 7e1. Present audio and text messages to all cars, informing them of the power outage.
  - 7e2. Move all cars to safe floors, where each car will open their doors and present audio and text messages asking them to disembark. Keep doors open and maintain that state until the signal is gone. Do not move the car.

## Use case diagram

---



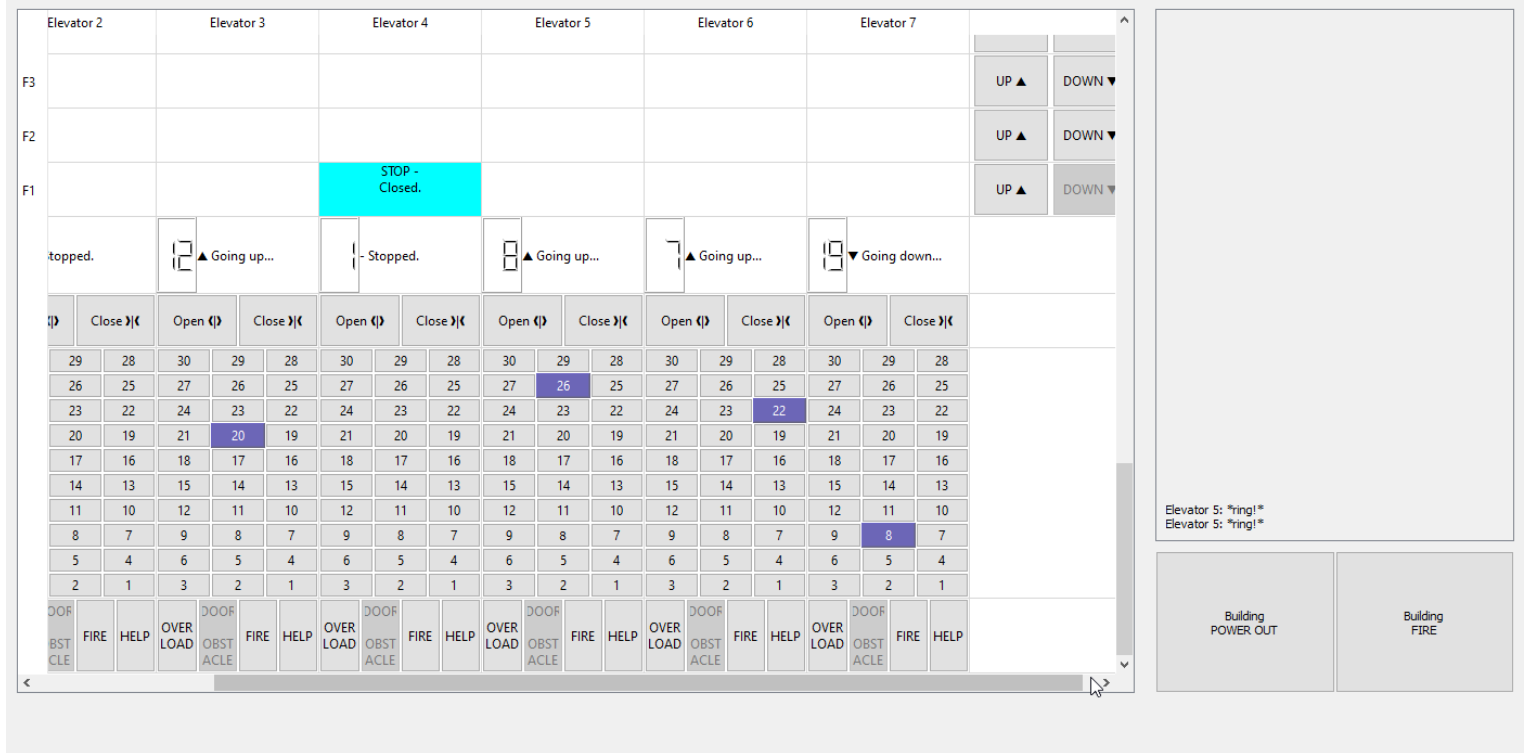
## Design Decisions

---

- This assignment was done with a decentralized elevator system, in which each elevator determines movement on its own, reading the building data and recomputing movement whenever there is a change.
- The Elevator class makes use of the State pattern, reading in the relevant data and setting movement, door, and emergency states, computing different movement based on what state the elevator is in currently.
- The Elevator class also makes use of the Observer pattern, watching for changes in the Building it is a part of by capturing buildingDataChanged() signals.
- The Building class can be considered a Mediator pattern, as it maintains a collection of Elevator objects and offers a common interface for them to read the building's state. The elevators do not interact with each other directly, yet are able to account for the others' movements. (e.g. if an elevator fulfills a floor request before the others, they will immediately stop once the first elevator arrives.)
- For ease of demonstration, the safe floor in emergencies is hardcoded to the first floor. This will make how the elevator ignores queued floors to get to the safe floor apparent.
- Audio feedback like bell ringing or audio messages are displayed in the console in the UI.
- Elevators are placed randomly on startup for ease of testing.

## Extensions

---



- GUI can accommodate any number of elevators and floors. (Image shows 30 floors and 7 elevators)
  - Edit program constants in mainwindow.h.
- Fully asynchronous, signal/slot based concurrency.
- Building-wide fire and power out scenarios can be triggered from the UI. Elevators can also have individual fires.
- Multiple emergency scenarios can be set at once, the state with higher priority will override the others.
- Buttons can be held. (e.g. holding open button will keep door open)

## Sequence Diagrams

### Basic use case 1

- 3 people on F4, P1, P2 go to F7, while other elevator serves P3 to F1.

### Basic use case 2

- P1, P2 on F1, E1 already on F1, other elevators are far away. P1 and P2 want to go to F7, P3 on F2 wants to go to F7 too. They press their respective floor button simultaneously. Other elevators mobilize to serve F2. E1 opens for F1, and stops at F2 to let P3 get on, other elevators stop since F2 has no request anymore, P3 closes the door with the close door button, then E1, carrying all passengers goes to F7.

## Safety scenario: Overload

- An elevator cannot move and ignores all input while it is overloaded, informing the passengers to reduce the load.

## Safety scenario: Door obstacle

- An elevator's door fails to close 3 times, at which point the elevator informs the passengers of an obstacle.

## Safety scenario: Fire

- A passenger onboard an elevator presses its fire alarm button. The elevator informs the ECS, which sends it to the nearest safe floor.

## Safety scenario: Help

- Passenger is connected to the building safety service.

## Safety scenario: Power out

- 1 or more passengers are in an elevator. ECS informs all elevators of a power outage, example elevator informs the passengers via audio and text display, and stops at the closest safe floor designated by the ECS, where the elevator will inform them to exit.

## Traceability Matrix

ID	Requirement	Related Use Case(s)	Fulfilled By	Implemented By	Tested By
1	Each elevator has door light, weight sensors.	Using the elevators of the building (same for all other requirements, there is only one use case)	The UI includes buttons that can be pressed to simulate overload and obstacles, serving as the light and weight sensor triggers.	Elevator.doorSensorSeesObstacle() Elevator.overloadButton Elevator.obstacleButton	On an elevator's panel, open the door with the Open button and press the DOOR OBSTACLE button. The door will fail to close while the obstacle button is active. Pressing the OVERLOAD button will make the elevator think it is overloaded.
2	Each floor has one or more buttons to hail an elevator. Pressing a floor button will create a	-	Buttons for each floor in the UI, can be toggled on and will stay toggled until an elevator stops at that	Building.upButton, Building.downButton DataButton	Press a floor's button and wait until an elevator arrives and deactivates it.

ID	Requirement	Related Use Case(s)	Fulfilled By	Implemented By	Tested By
	(possibly redundant) floor request, and it will stay illuminated until an elevator arrives to transport the customers (regardless of which direction the elevator is servicing).		floor and opens up to admit passengers.		
3	The elevator must ring a bell and open its doors for a fixed time (10 seconds), then ring the bell again and close its doors before proceeding to another floor.	-	The elevator prints a "ring" to the console in the UI when the door starts to open or close. The elevator will start a timer when the door is fully opened, and when expired, will automatically close the door. The timeout period in the simulation has been shortened for ease of testing.	MainWindow.inlineConsoleDisplay() Elevator.ring() Elevator.doorSpeedTimer Elevator.doorWaitTimer Elevator.openDoors() Elevator.closeDoors()	Press an elevator's open button when it is idle at a floor, it will close by itself and print "ring"s to the UI console when opening and closing.
4	Once on-board, passengers can select one or more destination floors using a panel of buttons. Selected floor buttons	-	The UI provides a floor panel view for each elevator. The floor buttons will stay checked until the elevator stops there	DataButton Elevator.destinationButtons	Press a button on the elevator for a floor that is not the current one. The elevator will automatically move to that floor and only

ID	Requirement	Related Use Case(s)	Fulfilled By	Implemented By	Tested By
	must stay illuminated while the floor is enqueued.		and opens up for it.		deactivate the button once the floor is reached and the elevator has opened up for it.
5	The elevator must display the current floor of the elevator to the passengers.	-	Each elevator's display panel has an LCD number display showing the elevator's current floor. It is updated whenever the elevator's floor changes.	QLCDNumber Elevator.currentFloorNum	Send an elevator to another floor and check that its floor label matches the display panel's number.
6	The elevator has a pair of "open door/close door" buttons which can override the default timing of the doors.	-	The elevators observe signals from their door buttons, and when they emit a signal, will either open or close the doors immediately, ignoring any ongoing door timeout timers.	Elevator.openButton Elevator.closeButton Elevator.doorSpeedTimer Elevator.doorWaitTimer Elevator.openDoors() Elevator.closeDoors() DataButton	Open an idle elevator's doors with the open button and immediately press the close doors button.
7	The elevator can display warning messages that are synced to audio warnings.	-	Whenever an elevator enters a new emergency state, it will trigger the UI console text output (in stead of audio) and update the display panel of each elevator with the appropriate text.	MainWindow.textDisplay MainWindow.inlineConsoleDisplay() Elevator.updateEmergency()	Test any of the emergency simulation buttons in the UI and observe the display panels and the UI console.

ID	Requirement	Related Use Case(s)	Fulfilled By	Implemented By	Tested By
8	The elevator has a "Help" safety service which can connect a passenger to building safety service (or 911 if there is no response from the service or the passenger) when the button on the panel is pressed.	-	Whenever the elevator's help button is active, the UI console and elevator display panel will show placeholder text for the act of connecting to safety service or 911.	Elevator.helpButton MainWindow.textDisplay MainWindow.inlineConsoleDisplay()	Press the "HELP" button on an elevator in the UI.
9	The elevator has a "Door obstacle" safety service that stops the door from closing and opens it when the light sensor is interrupted while the door is closing, and it can trigger text and audio warnings in the elevator if it occurs repeatedly.	-	When the elevator's DOOR OBSTACLE button is active, the elevator will not be able to close the doors. If enough failures occur in succession, the elevator will enter a new state and show the user text and audio (console) warnings.	Elevator.obstacleButton Elevator.doorCloseFailures Elevator.doorCloseFailThreshold Elevator.closeDoors() MainWindow.textDisplay MainWindow.inlineConsoleDisplay()	Open an elevator, press the DOOR OBSTACLE button before it closes. The elevator will notify of a door closing failure in the UI console, and once the threshold (3) is reached, will trigger the warnings.
10	The elevator has a "Fire" safety service which, when the signal is given, will cause the elevator to	-	The elevator movement computing function accounts for whether the elevator is in the FIRE state, and if	Elevator.fireButton Elevator.safeFloor Elevator.determineMovement() MainWindow.textDisplay MainWindow.inlineConsoleDisplay() Building.buildingOnFire()	Press the FIRE button of an individual elevator or press the building-wide fire button. All affected



ID	Requirement	Related Use Case(s)	Fulfilled By	Implemented By	Tested By
	go to a safe floor, and trigger text and audio warnings in the elevator informing passengers of the emergency and prompting them to leave at the safe floor.		so, will ignore all queued floors and head straight for the safe floor (floor 1, for ease of testing here).		elevators will immediately head to the first floor (safe floor) and ignore any queued floors or door overrides, and when at the safe floor, will open and inform passengers to disembark.
11	The elevator has an "Overload" safety service that triggers text and audio warnings if the sensors detect passenger or cargo load exceeding capacity, asking them to reduce the load before attempting to move again.	-	The elevator will not move if the OVERLOAD button is active; it will stay open at that floor and prompt passengers to reduce the load.	Elevator.overloadButton Elevator.determineMovement() Elevator.closeDoors()	Press the OVERLOAD button of an elevator and observe how it opens up immediately, informs passengers of the overload, and ignores all door overrides or floor requests until the button is unchecked.
12	The elevator has a "Power out" safety service which, when the signal is given, will present text and audio warnings to passengers informing them of the power outage, and move them	-	The elevator movement computing function accounts for whether the elevator is in the POWER OUT state, and if so, will ignore all queued floors and head straight for the safe floor (floor 1, for	Elevator.safeFloor Elevator.determineMovement() Building.buildingPowerOut() MainWindow.textDisplay MainWindow.inlineConsoleDisplay()	Press the building-wide power out button. All affected elevators will immediately head to the first floor (safe floor) and ignore any queued floors or door overrides, and when at the safe floor,

ID	Requirement	Related Use Case(s)	Fulfilled By	Implemented By	Tested By
	to a safe floor where they will be asked to get off.		ease of testing here).		will open and inform passengers to disembark.