Elevator

# Table of contents

[1. Table of contents 2](#_Toc1)

[2. Table of figures 2](#_Toc2)

[3. Description 3](#_Toc3)

[4. Use cases 3](#_Toc4)

[5. Use case diagram 4](#_Toc5)

[6. Class 5](#_Toc6)

[6.1 Class list: 5](#_Toc7)

[6.2 Object diagram: 6](#_Toc8)

[6.3 Class diagram: 6](#_Toc9)

# 2. Table of figures

[Figure 1. Usecase diagram 4](#_Toc1)

[Figure 2. Object diagram 5](#_Toc2)

# 3. Description

* System controls 8 elevators
* Each elevator has 20 ‘request floor’ buttons and a current floor indicator
* Each floor has 2 ‘request elevator’ buttons (up, down) and a floor indicator (on the highest and lowest floor there is only one ‘request elevator’ button)
* Both the elevator and the floor has a sliding door that open at the same time
* The floor dock has both pressure and optical sensors to detect obstructions (doors are opened if obstructed and closed again after 5 seconds)
* Speaker on each floor dock that announces requested elevator
* The system only send an idle elevator or an elevator going in the direction of the request (if none are available the request stays pending)
* When any request button is pressed a back-light is turned on until request is satisfied

# 4. Use cases

* Control 8 elevators to be able to move to each floor
* Make sure that you can safely load and unload the elevator
* The system makes moving between floors more efficient and easier compared to stairs
* User, Technician

|  |  |
| --- | --- |
| Name | Request Floor |
| Description | Elevator is requested to move to a floor |
| Actor | User, Technician |
| Flow | 1. Actor presses the button for the floor the want to request  2. Doors are closed (if not already) on both the floor and elevator  3. The system moves the elevator to the requested floor  4. Sound is played on the destination floor announcing elevator  5. Doors are opened on both the floor and elevator  6. Doors are closed on both the floor and elevator after 5 seconds |
| Exception | 1. Door is obstructed when closing, in this case the door is opened again and waits 5 seconds to retry.  2. Elevator is requested between the current floor and the requested floor, in this case the elevator makes a stop at the floor that requested the elevator. |
| Result | The elevator is at floor B with the doors closed |

|  |  |
| --- | --- |
| Name | Request Elevator |
| Description | Elevator is requested to move to a floor |
| Actor | User, Technician |
| Flow | 1. Actor presses the ‘Request Elevator’ button for the direction they want to go in  2. The system checks if there are any elevators that are idle or moving in the direction of the floor that is requesting an elevator  3. If the system finds a elevator with these criteria the system tells that elevator to go to / stop at the requesting floor  4. Sound is played on the requesting floor announcing elevator  5. Doors are opened on the both the floor and elevator |
| Exception | 1. If the system finds no elevator with the criteria from step 3 the system will wait until the criteria is met and continue the steps. |
| Result | The elevator is at the requested floor with the doors opened |

|  |  |
| --- | --- |
| Name | Catch Clamp |
| Description | Clamps that catch the elevator when falling |
| Actor | Tension\_Sensor, Technician |
| Flow | 1. A fall is detected  2. Clamps are initiated |
| Exception | 1. When the technician is testing the system it will activate regardless of fall detection. |
| Result | Catch clamps initiated |

|  |  |
| --- | --- |
| Name | Update floor indicator |
| Description | Update indicator on each floor and inside each elevator displaying what floor the elevator is currently on. |
| Actor | Elevator\_Sensor, Technician |
| Flow | 1. An elevator is detected at a floor  2. Floor indicators are updated |
| Exception | 1. When the technician is testing the system it will update to any value the Technician provides without the need of detection |
| Result | floor indicator are updated to the floor at which the elevator is detected |

# 5. Use case diagram

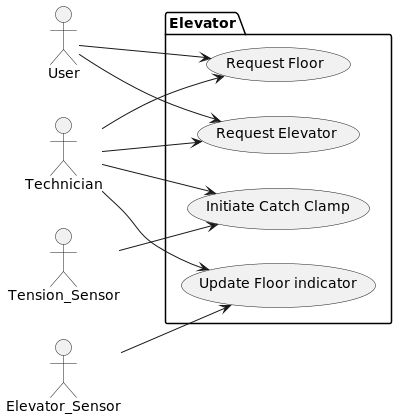


Figure 1. Usecase diagram

# 6. Classes

## 6.1 Class list:

|  |  |
| --- | --- |
| **Class** | **Responsibility** |
| Controller | Central class managing all activities inside the other classes. |
| Elevator | Class managing the hardware and all other activities inside an elevator. |
| IDoor | Interface class for the doors found on each floor and elevator. |
| Door | Class that manages the door hardware. (implements IDoor) |
| Floor | Class managing the hardware and all other activities on a floor. |
| Button | Class that uses the hardware to detect button presses. |
| Elevator\_Sensor | Class that uses the hardware to detect elevators on each floor. |
| Floor\_Indicator | Class that manages a screen that displays the floor on which the elevator was detected. |
| IObstruction\_Sensor | Interface class for the various sensors used to detect obstructions. |
| Pressure\_Sensor | Class that uses the hardware to detect pressures. (implements IObstruction\_Sensor) |
| Optical\_Sensor | Class that uses the hardware to detect optical obstructions. (implements IObstruction\_Sensor) |
| Tension\_Sensor | Class that uses the hardware to detect changes in tension. |
| Catch\_Clamp | Class that manages the safety clamp hardware. |

## 6.2 Object diagram:

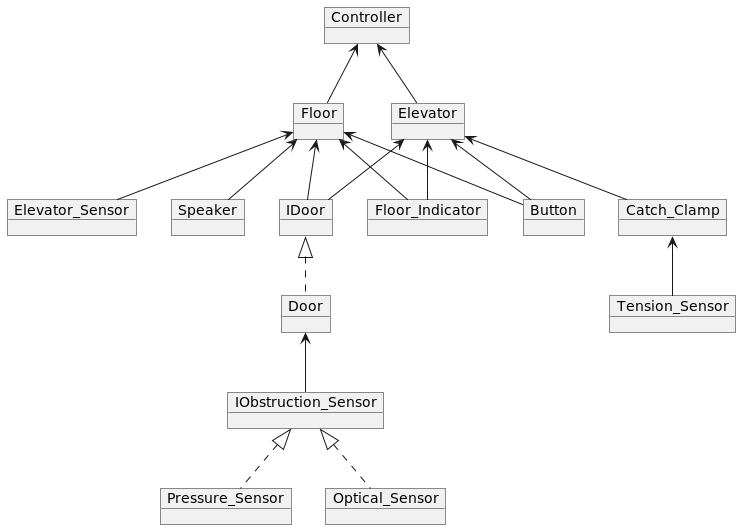


Figure 2. Object diagram

## 6.3 Class diagram:

