Use Case Diagrams

Name:
Use Case 1: Normal Elevator Use

Primary Actor(s):
Passenger

Stakeholders and Interests:
Passenger - Wants to reach the desired floor

Precondition(s):
The passenger knows how to operate an elevator

Success guarantee(s):
The elevator takes the passenger to the floor they selected

Main success scenario:

1. Passenger presses either the “up” or “down” floor button
2. Elevator control system selects the best-suited elevator to service the request
3. Appropriate floor button illuminates
4. Elevator arrives at the passenger’s floor
5. Appropriate floor button stops illuminating
6. Elevator rings bell
7. Elevator and floor doors open
8. Passenger boards the elevator
9. Passenger selects a destination floor from the panel of buttons
10. Elevator closes the elevator and floor doors after 10 seconds have passed since the doors were opened
11. Elevator begins to move to the selected floor
12. Display shows the current floor of the elevator
13. Elevator arrives at the destination floor
14. Elevator rings bell
15. Elevator and floor doors open
16. Passenger leaves the elevator

Extensions:
10a. Passenger holds the “close door” button on the panel

    10a1. Elevator closes the elevator and floor doors prematurely

10b. Passenger holds the “open door” button on the panel

10b1. Elevator keeps the elevator and floor doors open beyond its default period

10c. The light sensor is interrupted when the door is closing

10c1. Elevator stops the elevator and floor doors from closing and opens them

10c2. Elevator waits and attempts to close the doors

10c2a. The light sensor is interrupted repeatedly over a short period of time

10c2a1. Elevator issues a warning over the audio system and a text message is displayed

Name:

Use Case 2: Passenger boards the elevator and presses the Help button

Primary Actor(s):

Passenger

Stakeholders and Interests:

Passenger - Needs help on the elevator

Safety Service - Wants to help the passenger and ensure that they are okay

911 - If called, must respond to the call and dispatch assistance immediately

Precondition(s):

The passenger is on the elevator and is having issues that require help

Success guarantee(s):

The help button on the elevator functions accordingly and connects the passenger with the safety service

Main success scenario:

1. Passenger presses the “help” button
2. Elevator control system receives a “Help” alarm signal from the elevator indicating that the “Help” button has been pressed
3. Passenger is connected to the building safety service through a voice connection

Extensions:

3a. There is no response from building safety within 5 seconds

3a1. 911 emergency call is placed

3b. There is no response from the passenger

3b1. 911 emergency call is placed

Name:
Use Case 3: The control system receives a “Fire” alarm signal from the building while a passenger is on board an elevator

Primary Actor(s):
Passenger

Stakeholders and Interests:
Passenger - Needs to be delivered to a safe floor

Precondition(s):
The passenger is on the elevator

Success guarantee(s):
The elevator begins to move to a safe floor and notifies the passenger of an emergency using the audio and text message systems

Main success scenario:

1. Elevator control system commands all elevators to move to a safe floor
2. Elevator begins to move to a safe floor
3. Elevator presents an audio and text message to the passenger, informing them of an emergency and asking them to disembark once the safe floor is reached
4. Elevator reaches a safe floor
5. Elevator opens the elevator and floor doors
6. Passenger leave the elevator

Name:
Use Case 4: A Passenger enters an elevator with cargo that exceeds the carrying capacity

Primary Actor(s):
Passenger

Stakeholders and Interests:
Passenger - Wants to reach the desired floor

Precondition(s):
The passenger has called an elevator

Success guarantee(s):
The elevator sensor detects that the passenger and cargo exceed the carrying capacity and sends an “Overload” alarm signal to the elevator control system

Main success scenario:

1. Elevator arrives at the passenger's floor
2. Appropriate floor button stops illuminating
3. Elevator rings bell
4. Elevator and floor doors open
5. Passenger boards the elevator with cargo
6. Elevator sends an "Overlaod" signal to the elevator control system
7. Elevator presents an audio and text message to the passenger asking for the load to be reduced before attempting to move

Extensions:

7a. The passenger presses one of the floor buttons on the panel

    7a1. Elevator does not move and continues to present the message

Name:

Use Case 5: The control system receives a "Power Out" alarm signal from the building while a passenger is on board an elevator

Primary Actor(s):

Passenger

Stakeholders and Interests:

Passenger - Needs to be delivered to a safe floor

Precondition(s):

The passenger is on the elevator

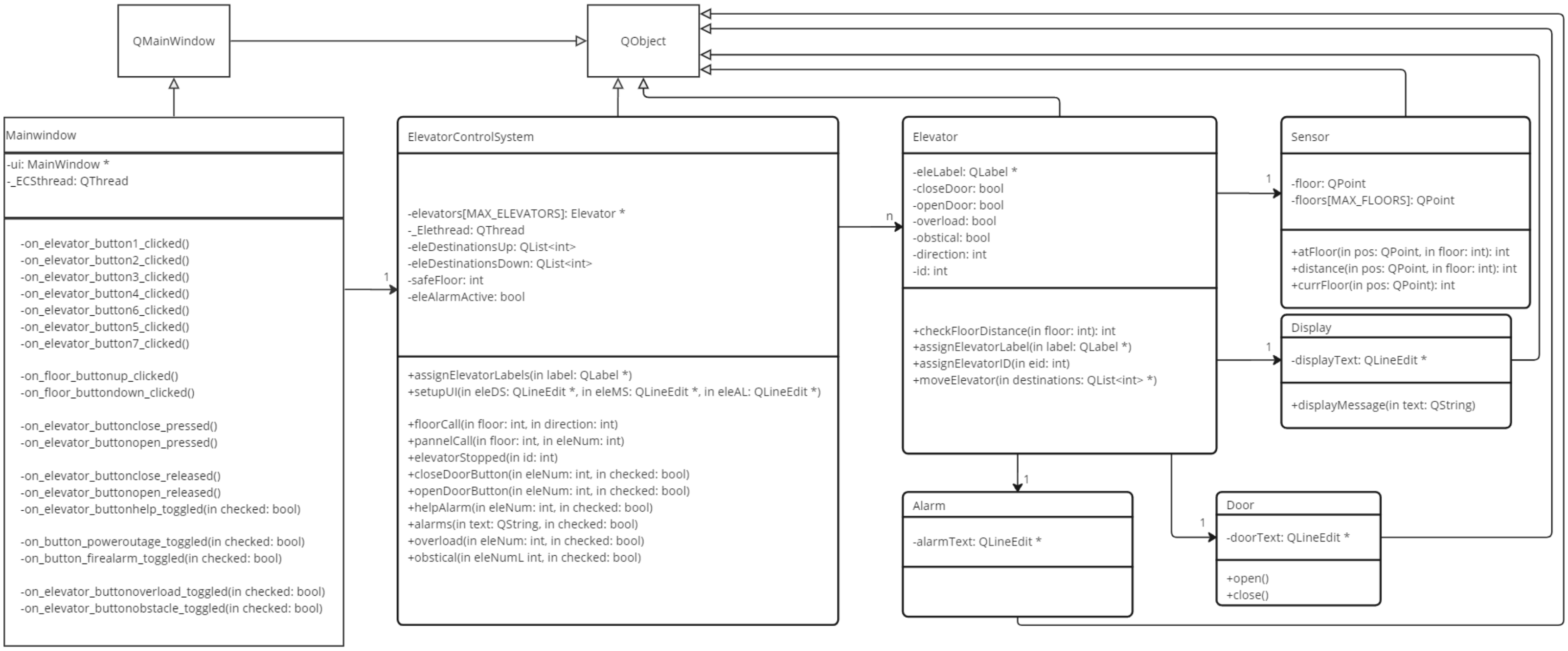
Success guarantee(s):

The elevator begins to move to a safe floor and notifies the passenger of an emergency using the audio and text message systems

Main success scenario:

1. Elevator begins to move to a safe floor
2. Elevator presents an audio and text message to the passenger, informing them of an emergency and asking them to disembark once the safe floor is reached
3. Elevator reaches a safe floor
4. Elevator opens the elevator and floor doors
5. Passenger leave the elevator

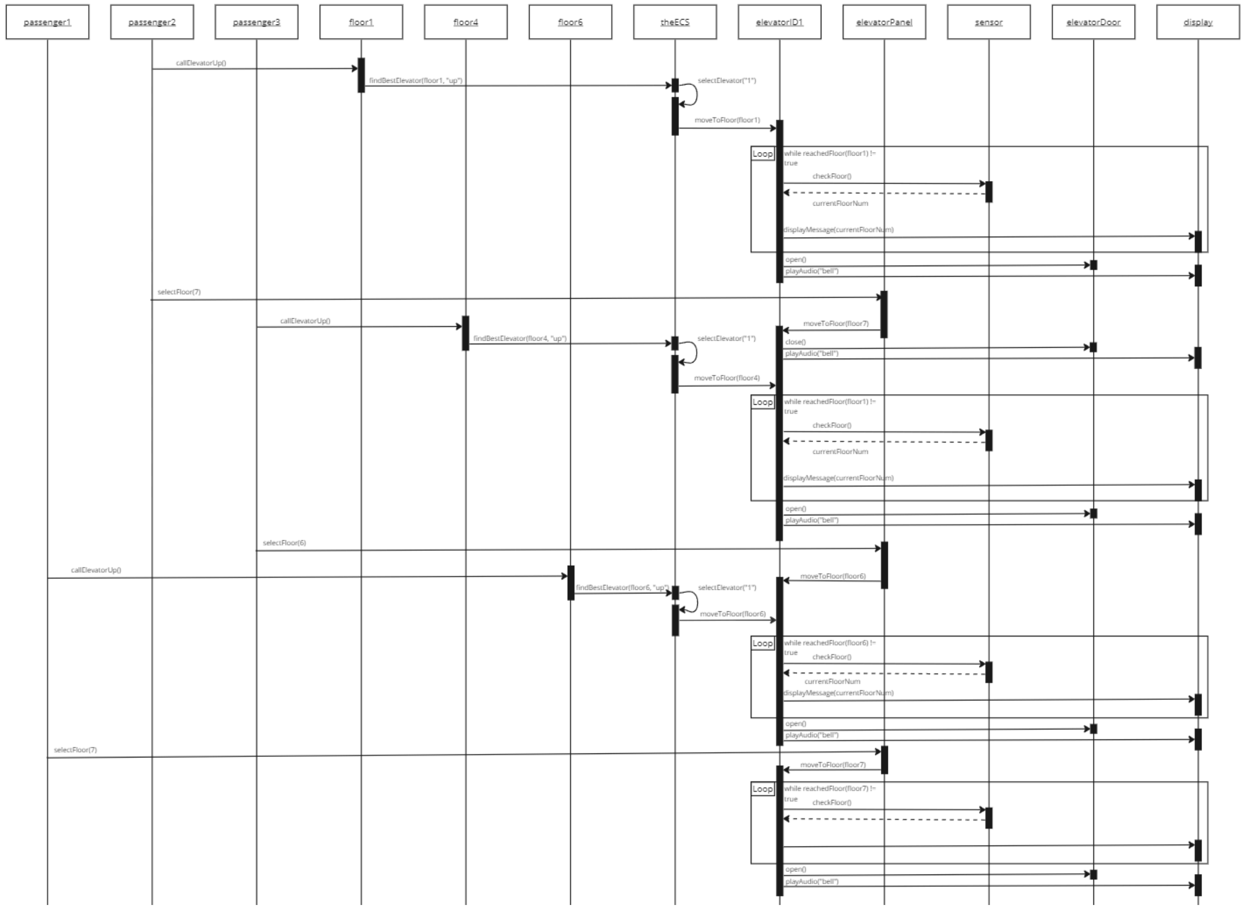
UML Class diagrams


QMainWindow
QObject
Mainwindow
-ui: MainWindow *
-_ECSthread: QThread
-on_elevator_button1_clicked()
-on_elevator_button2_clicked()
-on_elevator_button3_clicked()
-on_elevator_button4_clicked()
-on_elevator_button6_clicked()
-on_elevator_button5_clicked()
-on_elevator_button7_clicked()
-on_floor_buttonup_clicked()
-on_floor_buttondown_clicked()
-on_elevator_buttonclose_pressed()
-on_elevator_buttonopen_pressed()
-on_elevator_buttonclose_released()
-on_elevator_buttonopen_released()
-on_elevator_buttonhelp_toggled(in checked: bool)
-on_button_poweroutage_toggled(in checked: bool)
-on_button_firealarm_toggled(in checked: bool)
-on_elevator_buttonoverload_toggled(in checked: bool)
-on_elevator_buttonobstacle_toggled(in checked: bool)
1
ElevatorControlSystem
-elevators[MAX_ELEVATORS]: Elevator *
-_Elethread: QThread
-eleDestinationsUp: QList<int>
-eleDestinationsDown: QList<int>
-safeFloor: int
-eleAlarmActive: bool
+assignElevatorLabels(in label: QLabel *)
+setupUI(in eleDS: QLineEdit *, in eleMS: QLineEdit *, in eleAL: QLineEdit *)
+floorCall(in floor: int, in direction: int)
+pannelCall(in floor: int, in eleNum: int)
+elevatorStopped(in id: int)
+closeDoorButton(in eleNum: int, in checked: bool)
+openDoorButton(in eleNum: int, in checked: bool)
+helpAlarm(in eleNum: int, in checked: bool)
+alarms(in text: QString, in checked: bool)
+overload(in eleNum: int, in checked: bool)
+obstical(in eleNumL int, in checked: bool)
n
Elevator
-eleLabel: QLabel *
-closeDoor: bool
-openDoor: bool
-overload: bool
-obstical: bool
-direction: int
-id: int
+checkFloorDistance(in floor: int): int
+assignElevatorLabel(in label: QLabel *)
+assignElevatorID(in eid: int)
+moveElevator(in destinations: QList<int> *)
1
Sensor
-floor: QPoint
-floors[MAX_FLOORS]: QPoint
+atFloor(in pos: QPoint, in floor: int): int
+distance(in pos: QPoint, in floor: int): int
+currFloor(in pos: QPoint): int
1
Display
-displayText: QLineEdit *
+displayMessage(in text: QString)
1
Alarm
-alarmText: QLineEdit *
1
Door
-doorText: QLineEdit *
+open()
+close()

The Mainwindow is responsible for generating, setting up, and updating the GUI as well as connecting the GUI to the back-end. The ElevatorControlSystem is responsible for all the logical calculations that are necessary for deciding which elevator is to service which floor call and how to best approach the internal panel calls. The elevator panels and floor buttons are now all simulated in the GUI and are demonstrated in the Mainwindow rather than being separate classes. The Elevator is only responsible for its movement and controlling its internal components. The Elevator keeps track of 2 routes, the "up" route and the "down" route. Each time a call is made, the ElevatorControlSystem assesses the call and then adds the call, if necessary, to the appropriate route. The Elevator reads from the routes and then removes the floors as it services them. The user of the simulation system can act as any number of passengers simulating any amount of calls at a time. The UML Class Diagram above does not include getter functions, setter functions, signals, and duplicate functions dependent on the number of elevators or floors. Any functions not represented in the UML Class Diagram can be accessed in the source code.

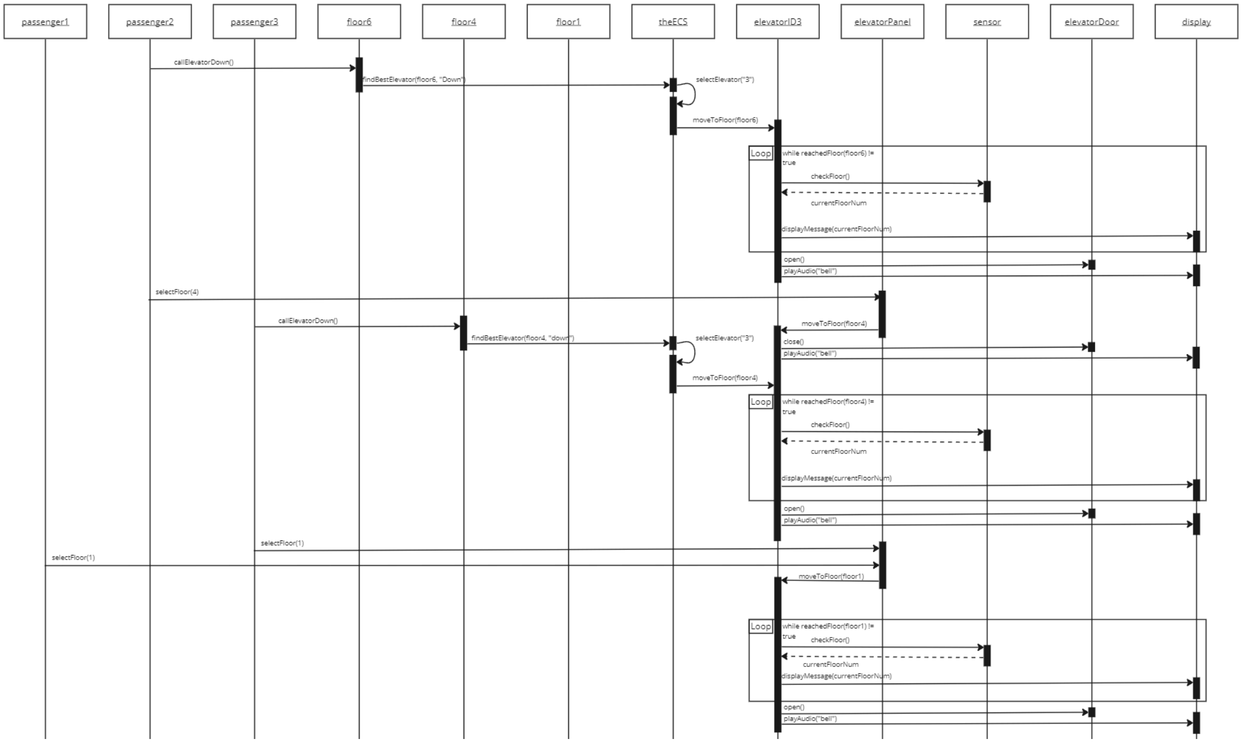
Sequence Diagrams

Success Scenario 1:



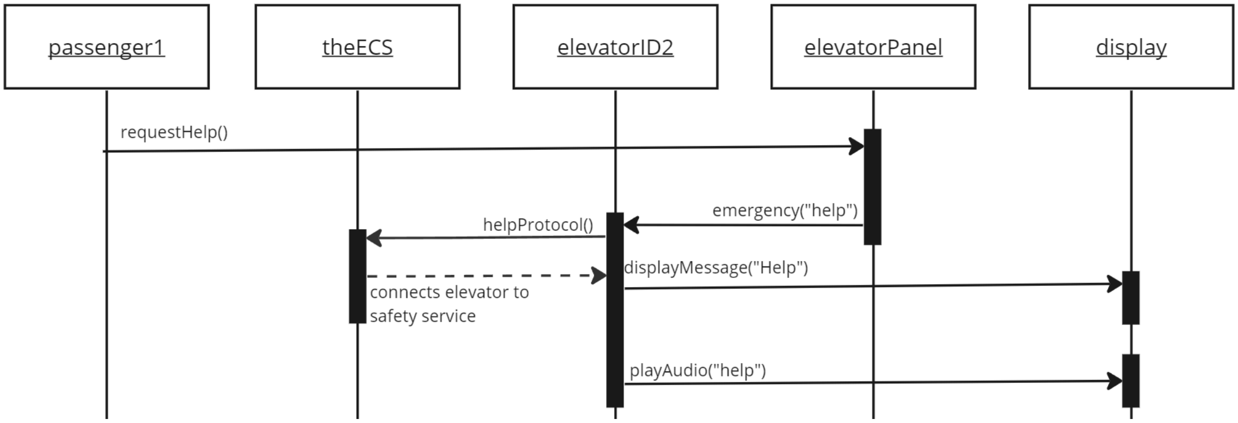
The success scenario sequence diagrams follow the same flow of logic presented in *Use Case 1: Normal Elevator Use* for the centralized design with the utilization of the functionality presented in the class diagrams. The passengers interact with the floor buttons informing the ECS of a service request. The ECS selects the best-suited elevator which then causes the elevator to move towards the passenger's floor. The elevator loops while moving through the floors, updating its display and checking the floor it is passing every time. Once the destination is reached, the elevator proceeds to open its doors, display the corresponding message and ring it's bell to notify the passengers of its arrival. If another passenger requests a floor that is different from the currently selected destination, it will rearrange its destinations depending on the order of the floors along its path, prioritizing floors that are along its direction (up or down) and orders them by how close the floors are. If the same floor is selected again, the duplicate request is ignored.

Success Scenario 2:

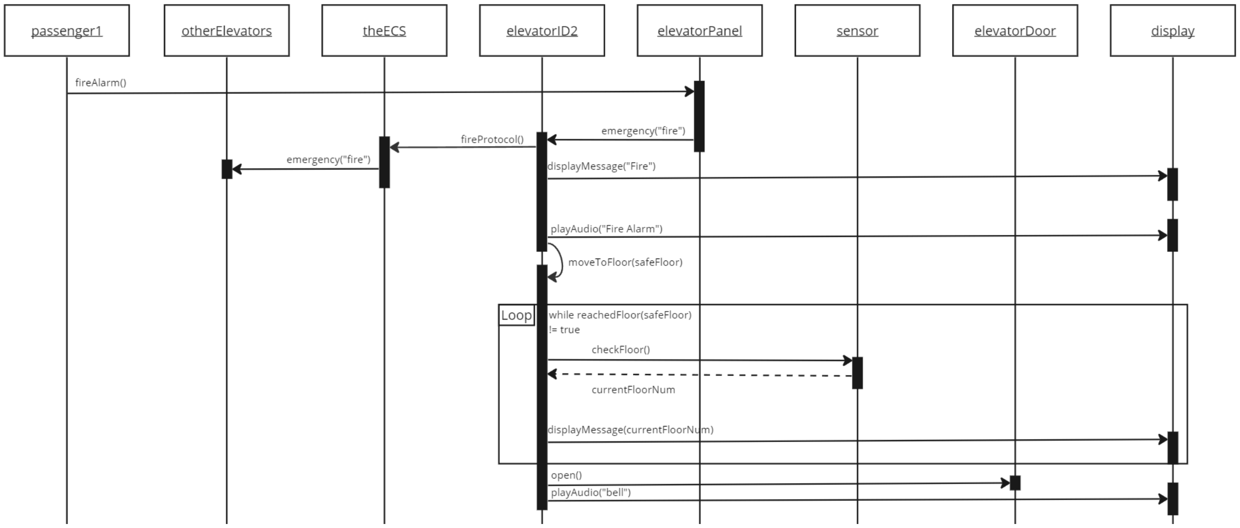


The main difference between the two success scenario sequence diagrams is the order in which the passengers utilize the elevator and the approach the elevator control system takes when handling multiple calls. The first scenario demonstrates how the elevator navigates to the higher floors and ignores any internal panel calls that may result in duplicates within the routes. The second scenario demonstrates the elevator's downward route and how the elevator control system handles multiple simultaneous calls to the same and different floors. Both represent a different route the elevator can take and how the elevator control system can handle multiple simultaneous calls throughout the elevator's routes.

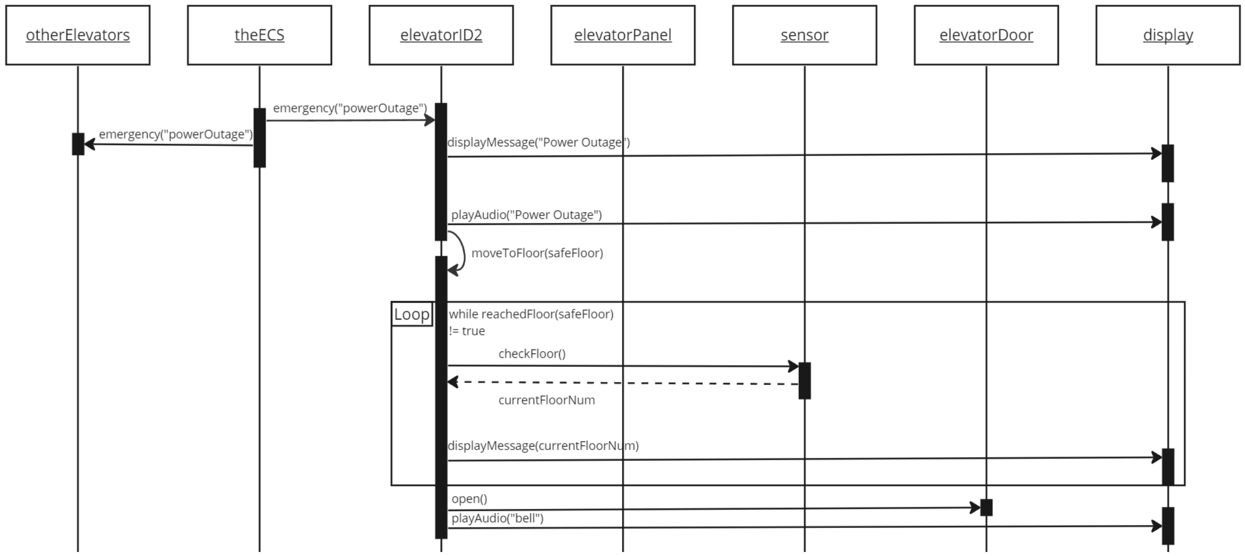
Safety Scenario Help:



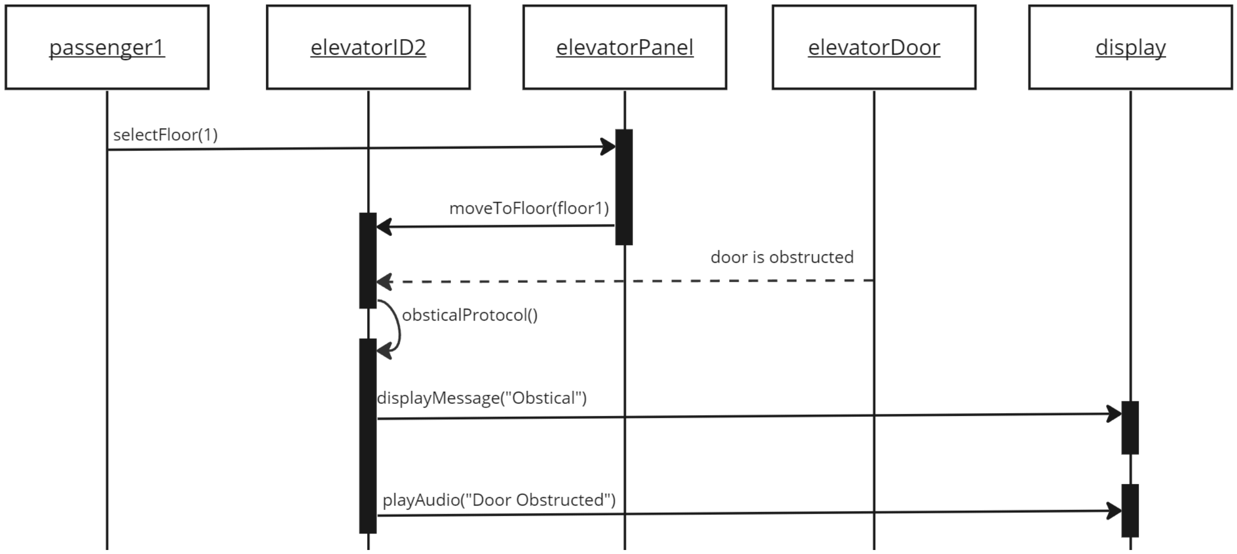
Safety Scenario Fire:



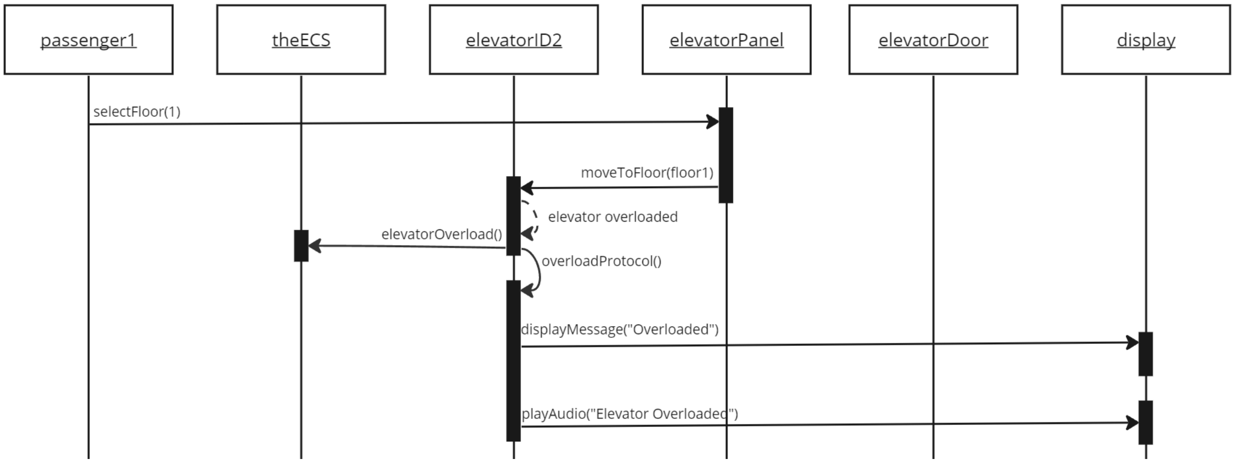
Safety Scenario Power Outage:



Safety Scenario Door Obstacles:

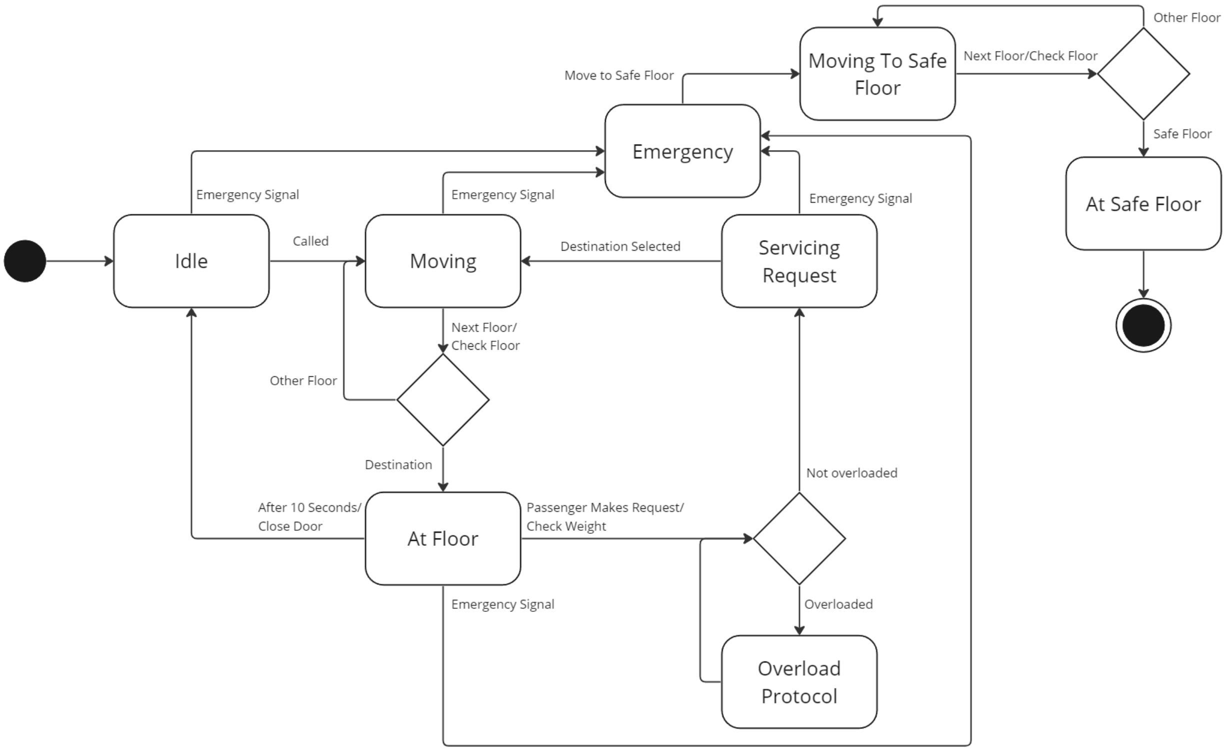


Safety Scenario Overload:

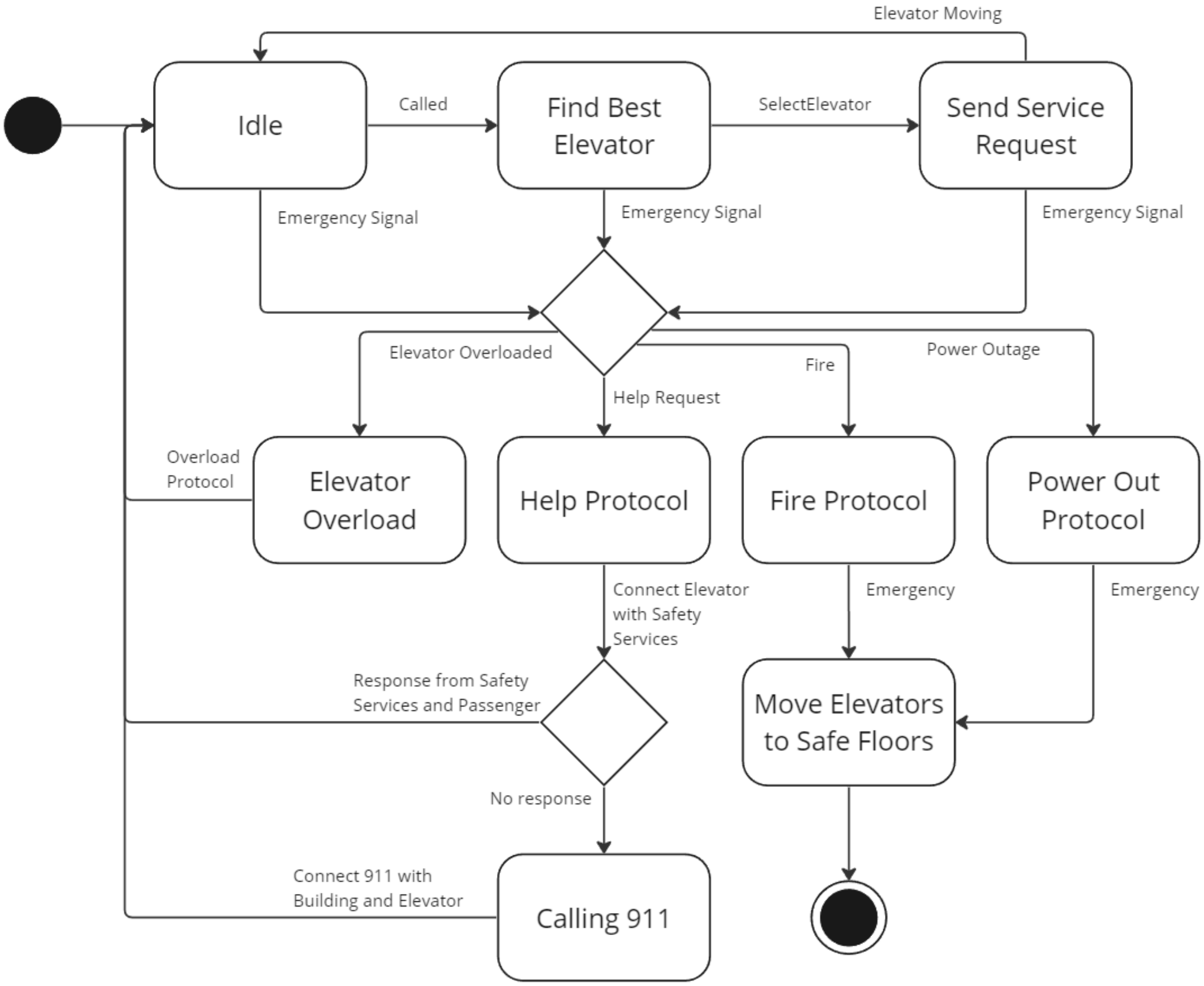


State Machine Diagrams

Elevator:



Elevator Control System:



Discussion

This is an implementation of the centralized design of the elevator system that utilizes an Elevator Control System to perform the logic necessary to decide which elevator should be selected to service a call. The ECS requests all elevators to return the distance between them and the called floor as well as the direction they must travel to reach that floor. The ECS then considers the direction that the elevators are currently travelling and adds the requested floor to the best-suited elevator's route. The selection process prioritizes stationary elevators first and then proceeds to assess the conditions of all other elevators. If at any point there is an equivalence between the elevators, the ECS selects the leftmost elevator.

The ECS also handles all signal and alarm communication to the elevators including any floor or panel button presses, any safety or alarm requests, and any building-wide emergencies. The ECS would communicate the signal type to all elevators and recall all of them to the designated safe floor.

The simulator GUI represents a system for testing and assessing the movement of elevators throughout the floors by providing controls to reproduce passenger inputs. The three elevator panels on the left allow for simulation of the controls inside the elevator as well as information the elevator is processing. The middle panel shows the floor requests and the elevators' current positions. The right panel is for testing emergency systems and safety scenarios.

Traceability Matrix

ID	Requirement	Related Use Case	Fulfilled By	Test	Description
1	A building is serviced by M elevators (also called cars)	N/A	Mainwindow, ElevatorControl System, defs.h	Run the simulator and observe the UI	Using QT's built-in user interface framework, the user has an interface to simulate elevator calls and alarms. All buttons are clickable with the mouse. The left side the the screen presents internal elevator panels and their component states. The center panel presents the floors, floor buttons, and elevators. The right panel presents toggles for scenarios and emergencies.
2	On each of the N floors is a pair of buttons marked "up" and "down"	N/A	Mainwindow, defs.h	Run the simulator and observe the UI	The center panel presents the floors, floor buttons, and elevators.
3	When a button is pressed it illuminates, and remains illuminated, until an elevator arrives to transport the customers who, at this floor, have requested an elevator going in a certain direction	Use Case 1: Normal Elevator Use	Mainwindow, ElevatorControl System	Run the simulator and observe the UI	When the user interacts with the buttons, they dim when clicked to show a response to the user though the simulation does not account for the physical illumination of those buttons with a building.

4	When the elevator arrives, it rings a bell, opens its doors (the elevator and floor doors) for a fixed time (10 seconds) allowing people to exit or board, rings the bell again, closes its doors and proceeds to another floor	Use Case 1: Normal Elevator Use	Elevator, Sensor, Display, Door, Alarm	Request an elevator to move either through a floor button or a panel button. Observe the elevator's audio state and the console output inside Qt creator.	The user can observe the elevator component states at the top of each elevator panel. These state changes are documented in the console output of Qt creator. The elevator will open its doors, perform an audio cue, and wait 10 seconds before closing the doors and ending the audio cue.
5	Once on-board passengers select one or more destination floors using a panel of buttons; there is one button for every floor.	Use Case 1: Normal Elevator Use	Mainwindow, ElevatorControl System, Elevator	Run the simulator and observe the UI. Interact with any panel button and observe the elevator movement through the floors.	The user can press any of the internal elevator buttons that correlate to the building floors to request the elevator to move to that floor. The elevator will add the floor to its route in the direction that it needs to travel to reach the floor. The center panel will update with the elevator's current position. The user can observe the ECS documentation in the output console of Qt creator.
6	The elevator has a display which shows passengers the current floor of the elevator	Use Case 1: Normal Elevator Use	Elevator, Display	Request an elevator to move either through a floor button or a panel button. Observe the elevator's display state and the console output inside Qt creator.	While the elevator is moving through the floors, the display state will update to represent the elevator's current floor. The elevator queries the sensor on which floor it is currently at and communicates that information to the display accordingly.
7	There is also a pair of buttons on the elevator control panel marked "open door" and "close door". These buttons can be used by a passenger to override the default timing of the doors. The	Use Case 1: Normal Elevator Use	Mainwindow, ElevatorControl System, Elevator	Request an elevator to move either through a floor button or a panel button. Once the elevator	When the elevator door is set to the open state, the elevator checks the state of the open and close door buttons. If the close button is held down, the elevator will set its state to close, skipping the 10-second timer. If the open button is held beyond the 10-second timer, the door state will

	door will remain open beyond its default period if the “open door” button is held depressed; the doors can be closed prematurely by pressing the “door close” button			arrives, hold either the open or the close buttons on the elevator panel. Observe the elevator’s door state.	remain open until the button is released. The elevator checks the state of these buttons every half second while the door is open.
8	Inside the elevator there is also a help button linked to building safety service	Use Case 2: Passenger boards the elevator and presses the Help button	Mainwindow, ElevatorControl System, Elevator	Request an elevator to move either through a floor button or a panel button. Presse the help button on the elevator panel. Observe the elevator’s audio state, display state, and the console output inside Qt creator.	When the user toggles the help button, the elevator first finishes its current service of the floor and then proceeds down to the ground floor, which is designated as the safe floor in the ECS. The connects the elevator with the operators immediately. The ECS documents this process in the console output of the Qt creator.
9	Each elevator has a sensor that notifies it when it arrives at a floor	Use Case 1: Normal Elevator Use	Sensor	Request an elevator to move either through a floor button or a panel button. Observe the elevator’s movement and the console output inside Qt creator.	While the elevator is moving, it queries the sensor on if it has reached the requested floor. The sensor then informs the elevator if it must move up or down, or if it has reached the desired floor.
10	The elevator control system should ensure that the group of elevators services all (floor and on-board) requests expeditiously	Use Case 1: Normal Elevator Use	ElevatorControl System	Request multiple elevators to move to random floors. Observe the elevators’	The ECS documents its decision-making and its updates to the elevators’ routes in the console output of the Qt creator. The ECS selects the elevators based on a criteria described in the *Discussion* segment. The ECS updates the

				movements and the consol output inside Qt creator.	elevators' routes accordingly and signals them to move. The elevators service the floors depending on the ECS criteria.
11	Each elevator has a display and an audio system. The display shows the current floor number and warning messages that are synced with audio warnings	Use Case 1: Normal Elevator Use	Elevator, Display, Alarm	Request an elevator to move either through a floor button or a panel button. Observe the elevator’s display state, audio state, and the console output inside Qt creator.	While the elevator is moving through the floors, the display state will update to represent the elevator's current floor. The elevator queries the sensor on which floor it is currently at and communicates that information to the display accordingly. The audio is determined by the Alarm which the elevator updates as it reaches a floor or signals for an emergency.
12	The control system receives a “Help” alarm signal from an elevator indicating that the “Help” button has been pressed. In that case, the passenger is connected to building safety service through a voice connection. If there is no response from building safety within 5 seconds or if there is no response from a passenger a 911 emergency call is placed	Use Case 2: Passenger boards the elevator and presses the Help button	Mainwindow, ElevatorControl System, Elevator, Display, Alarm	Request an elevator to move either through a floor button or a panel button. Presse the help button on the elevator panel. Observe the elevator’s audio state, display state, and the console output inside Qt creator.	When the user toggles the help button, the elevator first finishes its current service of the floor and then proceeds down to the ground floor, which is designated as the safe floor in the ECS. The connects the elevator with the operators immediately. The ECS documents this process in the console output of the Qt creator.
13	If the light sensor is interrupted when the door is closing, the control system stops the door from closing and opens it. If this occurs repeatedly over a short period of time, a warning is sounded over the audio system and a text	Use Case 1: Normal Elevator Use	ElevatorControl System, Elevator, Door, Display, Alarm	Request an elevator to move either through a floor button or a panel button. Presse the elevator’s obstacle button on the right	When the user toggles the obstacle button, the elevator does not close the door and updates the display and audio states to show that there is an obstacle blocking the doors. The ECS documents these events in the console output of the Qt creator.

	message is displayed			panel. Observe the elevator’s audio state, display state, and the console output inside Qt creator.	
14	The control system receives a “Fire” alarm signal from the building and commands all elevators to move to a safe floor. Similarly, a “Fire” alarm signal from the elevator itself will cause that elevator to go to a safe floor. In both cases an audio and text message are presented to passengers informing them of an emergency and asking them to disembark once the safe floor is reached	Use Case 3: The control system receives a “Fire” alarm signal from the building while a passenger is on board an elevator	ElevatorControl System, Elevator, Display, Alarm	Request multiple elevators to move to random floors. Toggle the fire alarm button on the right panel. Observe the elevators’ movements and the consol output inside Qt creator.	When the user toggles the fire alarm button, the elevator first finishes its current service of the floor and then proceeds down to the ground floor, which is designated as the safe floor in the ECS. The ECS documents this process in the console output of the Qt creator.
15	The control system receives an “Overload” alarm signal from an elevator if the sensors indicate that the passenger or cargo load exceeds the carrying capacity. In that case, the elevator does not move and an audio and a text messages are presented to passengers asking for the load to be reduced before attempting to move again	Use Case 4: A Passenger enters an elevator with cargo that exceeds the carrying capacity	ElevatorControl System, Elevator, Display, Alarm	Request an elevator to move either through a floor button or a panel button. Presse the elevator’s overlaod button on the right panel. Observe the elevator’s audio state, display state, and the console output inside Qt creator.	When the user toggles the overload button, the elevator does not close the door and updates the display and audio states to show that there is a load that exceeds the elevator's weight limitations. The ECS documents these events in the console output of the Qt creator.

16	The control system receives a “Power Out” alarm signal. In that case, an audio and a text messages are presented to passengers informing them of the power outage. Each elevator is then moved to a safe floor and passengers are asked to disembark via audio and text messages. The battery backup power is sufficient to do all of this	Use Case 5: The control system receives a “Power Out” alarm signal from the building while a passenger is on board an elevator	ElevatorControl System, Elevator, Display, Alarm	Request multiple elevators to move to random floors. Toggle the power outage alarm button on the right panel. Observe the elevators’ movements and the consol output inside Qt creator.	When the user toggles the power outage button, the elevator first finishes its current service of the floor and then proceeds down to the ground floor, which is designated as the safe floor in the ECS. The ECS documents this process in the console output of the Qt creator.