Textual Explanation

Class Diagram – The elevator class has all member vars needed to keep track of its location, weight, status of idle, and displays and audio systems as strings which are always on. Other parts such as bell and doors are represented as functions along with their actions. The buttons for the elevator is represented in the MainWindow UI elevator panel such as push,help,open,close for the doors. As well as the destination floor buttons which is the drop down selection for floors. All necessary fucntions to facilitate the elevators movement, displaying of safety msgs, adding or removing passengers are implemented as well.

The ECS contains the array of elevators of the building itself, as well as passengers for testing purposes. Currstrat to know which allocation strategy were using atm, as well as number of elev,floors,passengers(for testing) in the building. All the functions are used to take in requests such as floor,elev, and safety issues. And those will utilize other functions in ECS as needed such as alloc strats implemented. The passwalkin and out are in this class since elev cant modify data in the ECS directly. ECS manages everything about the elevator and floors in this design like ECS tells elev doors to open, not the elev itself then tells its door to open to simplify implementation.

The MainWindow UI represents the passenger interacting with the system, with floors and elevators etc. as well includes simulation inputs and outputs onto a textbox to be able to see actions occuring in the program instead of QT creator. The drop down selection for floor num, elev num, pass num are used to keep track of actions and are used as parameters for requests. All the buttons are represent the buttons for the elevators and floor buttons. As well as the safety ones for just testing purposes.

Basic Use case Diagrams:

Basic use case Sequence Diagrams – The case starts with a passenger clicking up or downbtn() on the floor panel in the UI. This will call the flrreq(direction,serveflrnum,passnum) in the ECS to let it know its requesting an elevator to the serveflrnum. Current time is also passed in as a parameter to check if the current times during a rush hour or not (7-9am,12-2pm,4-7pm). If it is then the allocation strategy used will be elevcenteredstrat(serveflrnum) which finds an elevator that is closest by least number of floors between it and the serveflrnum. Otherwise if it’s a normal time then loadbalancestrat() is used to prolong the elevators life, and distributes the requests between all elevators by its currweight as even as it can.

When the ECS finds an elevator(if it doesn’t in testing then just returns all elevators are in use.) it commands elevator to move by 1 floor each time in a direction depending if the serveflrnum is above or below the elevator and display its currflrnum, until it reaches the serveflrnum. The ECS then turns off the up or down btn of the floorpanel, and rings() and openDoor() of the elevator. This allows the passwalkin(passnum,elevnum) which the ECS will tell the elevator to addpassengers(passweight). The passenger has the option to press the open or close button in the elevator. Otherwise the ECS just rings() and closeDoor() of the elevator after 5s. In the UI the floor is selected through the dropdown in elevator panel, and the passenger hits the push button to simulate pressing a destflrbtn. The elevator then sends a elevreq(destflrnum,elevnum,passnum) to ECS and handles it by moving the elevator until its currflr is == to destflrnum given. Once it is the ECS rings the bell and openDoor() of the elevator again, At this point the passenger walks out and their weight is removed from the elevator. Then the ECS just rings() and closeDoor() of the elevator after 5s.

Safety features Sequence Diagrams:

Help – As the Passenger pressed the help button in the elevator, the elevsafetyreq(help,elevnum,passnum) is called in ECS to let it know the help btn was pressed and which elev it came from. The ECS then connects the elevnum using bssconnection(elevnum,passnum) and waits for a response. If the people from BSS or the passenger doesn't respond(I have it as a random generator between 0 and 1 to detect a response). Then the ECS will call911().

Door Obstacle – The case starts with the ECS telling the elev to ring and close its door(), if it detects an obstacle while its closing, then the elevator will call elevsafetyreq(obstacle,elevnum,passnum) to let it know theres an obstacle issue, now if its been < 5 times theres been an obstacle, the ECS will tell the elevator to interuptclose() which the elevator will ring its bell and openDoor(), and ring its bell again and then closeDoor() and the elevator can proceed to move on. Or if its 5 times or more, then the ECS will ring its bell and holdOpenDoor() and the ECS sets the safetymsg for each elevator as "theres an obstacle….." , and displays and plays in the audio system the obstacle safety message in the elevator and doesn't move.

Fire – The case starts with either the Building calls a safetyreq(fire) or a passenger presses the fire button in an elevator then that also calls safetyreq(fire) to the ECS. In both cases the ECS sets the safetymsg for each elevator as "theres a fire ….." , and displays and plays in the audio system the fire safety message in the elevator, and commands all the elevators to move to the set safefloor(which is 1 in my design) and updates the current floor as it travels as well. Once the elevator reaches the safe floor the ECS rings their bells and holds open their doors. At this point the passengers can walk out and their weight is removed from the elevator.

Overload – The case starts with the passwalkin(passnum,elevnum) into an elevator. Now the ECS detects if that passengers weight + currweight of the elev will exceed the maxweight allowed of the elev. If it does then elevsafetyreq(overload,elevnum,passnum) will be called as it's a separate safety issue for an elevator and not a general one for all elevs. The ECS sets the safetymsg for each elevator as “theres an overload …..” , and displays and plays in the audio system the overload safety message in the elevator, which the passenger will halve to walk out otherwise elev wont close the doors. Otherwise if the currweight doesn’t exceed maxweight then well update the elevators currweight to add the passengers weight. In both cases either the pass walks out after an overload is raised or it doesn’t exceed, the ECS rings and closes the elevs door.

Power out –The case starts with either the Building calls a safetyreq(powerout) to the ECS. In both cases the ECS sets the safetymsg for each elevator as “theres a powerout …..” , and displays and plays in the audio system the powerout safety message in the elevator, and commands all the elevators to move to the set safefloor(which is 1 in my design) and updates the current floor as it travels as well. Once the elevator reaches the safe floor the ECS rings their bells and holds open their doors. At the point the passengers can walk out and their weight is removed from the elevator.

Basic Use case Traceability Matrix:

ID	Requirement	Related Use Case	Fulfilled by	Test	Description
1	On each of the N floors is a pair of buttons marked “up” and “down”. 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.	Elevator Use case (section 1)	MainWindow, ECS, Elevator	Start Sim in QT, and press up or down btn and observe output	a passenger clicking up or downbtn() on the floor panel in the UI. This will call the flrreq(direction,serveflrnum,passnum) in the ECS to let it know its requesting an elevator to the serveflrnum. When the ECS finds an elevator(if it doesn’t in testing then just returns all elevators are in use.) it commands elevator to move by 1 floor each time in a direction depending if the serveflrnum is above or below the elevator and display its currflrnum, until it reaches the serveflrnum. The ECS then turns off the up or down btn of the floorpanel.
2	When the elevator arrives, it rings a bell, opens its	Elevator Use case (section 2 - 7)	MainWindow, ECS, Elevator , Passenger	Observe output of UI from when up or down	When the ECS finds an elevator(if it doesn’t in testing then just returns all elevators are in use.) it commands elevator to move by 1 floor each time in a direction depending if

	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.			btn is pressed	the serveflrnum is above or below the elevator and display its currflrnum, until it reaches the serveflrnum. The ECS then turns off the up or down btn of the floorpanel, and rings() and openDoor() of the elevator. the ECS just rings() and closeDoor() of the elevator after 10s
3	Once on-board passengers select one or more destination floors using a panel of buttons; there is one button for every floor.	Elevator Use case (section 6)	MainWindow, ECS, Elevator, Passenger	select a passenger, and a destinaton floor on elevator panel. And press push to observe the output	In the UI the floor is selected through the dropdown in elevator panel, and the passenger hits the push button to simulate pressing a destflrbtn. The elevator then sends a elevreq(destflrnum,elevnum,passnum) to ECS.
4	Each elevator has a sensor that notifies it when it arrives at a floor. The elevator has a display which shows passengers the current floor of the elevator.	Elevator Use case (section 8)	MainWindow, ECS, Elevator,	Observe output of UI after floor or elevator push button are pressed. To see Display of elev movement	The display is always on with its currflrnum, as the elevator moves between floors, its updated through the move(direction) in elevator class
5	There is also a pair of buttons on the elevator control panel marked “open door” and “close door”. These buttons can be used by	Elevator Use case (section 5)	MainWindow, Elevator	Press Open and close buttons on UI and observe output.	On the UI, the The passenger has the option to press the open or close button in the elevator. Which it will call the Elevators openDoor() that rings and opens the door or closeDoor() that rings and closes the door.

	a passenger to override the default timing of the doors. The 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.				
6	Inside the elevator there is also a help button linked to building safety service.	Elevator Use case (section 7c)	Mainwindow, ECS, Elevator	Press help button and observe output in UI	As the Passenger pressed the help button in the elevator, the elevsafetyreq(help,elevnum,passnum) is called in ECS to let it know the help btn was pressed and which elev it came from. The ECS then connects the elevnum using bssconnection(elevnum,passnum) and waits for a response.
7	The elevator control system should ensure that the group of elevators services all (floor and on-board) requests expeditiously.	Elevator Use case	MainWindow, ECS, Elevator,	Observe output of the UI after pressing a floor button, on how it chooses its strategy	This will call the flrreq(direction,serveflrnum,passnum) in the ECS to let it know its requesting an elevator to the serveflrnum. Current time is also passed in as a parameter to check if the current times during a rush hour or not (7-9am,12-2pm,4-7pm). If it is then the allocation strategy used will be elevcenteredstrat(serveflrnum) which finds an elevator that is closest by least number of floors between it and the serveflrnum. Otherwise if it’s a normal time then loadbalancestrat() is used to prolong the elevators life, and distributes the requests between all elevators by its currweight as even as it can.
8	Each elevator has a display and an audio system. The	Elevator Use case	MainWindow, ECS, Elevator	Press any of the safety buttons,	elevsafetyreq(safetyissue,elevnum,passnum) will be called as it’s a separate safety issue for an elevator and not a general one for all elevs. The ECS sets the safetymsg for each

	display shows the current floor number and warning messages that are synced with audio warnings.			and observe output in UI	elevator as “theres an safetyissue …..” , and displays and plays in the audio system the safety message in the elevator. The display is always on with its currflrnum, as the elevator moves between floors, its updated through the move(direction) in elevator class

<u>Safety Feature Traceability Matrix:</u>

ID	Requirement	Related Use Case	Fulfilled by	Test	Description
1	The control system receives a “Help” alarm signal from an elevator indicating that the “Help” button has been pressed.	Elevator Use case (section 7c)	MainWindow, ECS, Elevator, Passenger	Press Help button and observe output in UI	As the Passenger pressed the help button in the elevator, the elevsafetyreq(help,elevnum,passnum) is called in ECS to let it know the help btn was pressed and which elev it came from
2	the passenger is connected to building safety service through a voice connection.	Elevator Use case (section 7c)	ECS, Elevator, Passenger	Press Help button and observe output in UI	The ECS then connects the elevnum using bssconnection(elevnum,passnum) and waits for a response.
3	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.	Elevator Use case (section 7c)	ECS, Elevator, Passenger	Press Help button and observe output in UI	bssconnection(elevnum,passnum) will wait for a response. If the people from BSS or the passenger doesn’t respond(I have it as a random generator between 0 and 1 to detect a response). Then the ECS will call911().
4	If the light sensor is interrupted when the door is closing, the control system stops the door from	Elevator Use case (section 5a)	MainWindow, ECS, Elevator	Press Door Block button and observe output in UI	if it detects an obstacle while its closing, then the elevator will call elevsafetyreq(obstacle,elevnum,passnum) to let it know theres an obstacle issue. The ECS will tell the elevator to interuptclose() which the elevator will ring its bell and openDoor()

	closing and opens it.				
5	If this occurs repeatedly over a short period of time, a warning is sounded over the audio system and a text message is displayed.	Elevator Use case (section 5a)	MainWindow, ECS, Elevator	Press Door Block button and observe output in UI	if its detected 5 times or more, then the ECS will ring its bell and holdOpenDoor() and the ECS sets the safetymsg for each elevator as “theres an obstacle…..” , and displays and plays in the audio system the obstacle safety message in the elevator and doesn’t move.
6	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.	Elevator Use case (section 7a)	MainWindow, ECS, Elevator	Press Fire button and observe output in UI	the Building calls a safetyreq(fire) or a passenger presses the fire button in an elevator then that also calls safetyreq(fire) to the ECS. Then the ECS commands all the elevators to move to the set safefloor(which is 1 in my design) and updates the current floor as it travels as well. Once the elevator reaches the safe floor the ECS rings their bells and holds open their doors.
7	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.	Elevator Use case (section 7a)	ECS, Elevator	Press Fire button and observe output in UI	In both cases the ECS sets the safetymsg for each elevator as “theres a fire …..” , and displays and plays in the audio system the fire safety message in the elevator. The messages are played/displayed the whole time till passengers leave (from activation of fire mode).
8	The control system receives an “Overload” alarm signal from an elevator if the sensors	Elevator Use case (section 4b)	MainWindow, ECS, Elevator, Passenger	Press Overload button, or floor button until you	the passwalkin(passnum,elevnum) into an elevator. Now the ECS detects if that passengers weight + currweight of the elev will exceed the maxweight allowed of the elev. If it does then

	indicate that the passenger or cargo load exceeds the carrying capacity.			observe the change in output in UI.	elevsafetyreq(overload,elevnum,passnum) will be called.
9	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.	Elevator Use case (section 4b)	ECS, Elevator, Passenger	Press Overload button, or floor button until you observe the change in output in UI.	The ECS sets the safetymsg for each elevator as “theres an overload …..” , and displays and plays in the audio system the overload safety message in the elevator, which the passenger will have to walk out otherwise elev wont close the doors. (Assuming the elevator is stationary as ECS will not command elevator to move until passenger leaves)
10	The control system receives a “Power Out” alarm signal.	Elevator Use case (section 7b)	MainWindow, ECS, Elevator	Press Power out button and observe output in UI	the Building calls a safetyreq(powerout) to the ECS.
11	In that case, an audio and a text messages are presented to passengers informing them of the power outage.	Elevator Use case (section 7b)	ECS, Elevator	Press Power out button and observe output in UI	The ECS sets the safetymsg for each elevator as “theres a powerout …..” , and displays and plays in the audio system the powerout safety message in the elevator
12	Each elevator is then moved to a safe floor and passengers are asked to disembark via audio and text messages.	Elevator Use case (section 7b)	ECS, Elevator	Press Power out button and observe output in UI	The ECS commands all the elevators to move to the set safefloor(which is 1 in my design) and updates the current floor as it travels as well. Once the elevator reaches the safe floor the ECS rings their bells and holds open their doors. The messages are played/displayed the whole time till passengers leave (from activation of power out mode).