COMP 3004 Assignment #2

Suhail Mohamed : 101117886

Use case#1 : Using an elevator to travel

Primary Actor : user who wants to travel between floors not using the stairs

Stakeholders and Interest :

- Users
- Owner of the building
- Elevator Installment Company
- Elevator inspectors

Preconditions : User wants to go from one floor to another using the elevator

Minimal guarantee : A user remains safe in the elevator even in the case of an accident

Success guarantee : A user is safely and comfortably moved from current floor to destination floor.

Main success scenario :

1) User presses the up or down button near the floor's elevator entrance, the button pressed lights up and remains illuminated
2) Elevator arrives at floor and rings a bell, the light on the pressed buttons turns off
3) Elevator's door open to allow for the exiting and entering of users into the cab
4) Elevator bell rings again to indicate its departure and closes its door
5) User that has entered the elevator selects their destination floor in the elevator
6) Elevator determines path to drop off users within it
7) User looks at the screen within the elevator to determine what floor the elevator is currently on during travel
8) Elevator travels to users destination floors making stops if need be
9) Elevator arrives at Users destination floor, rings its bell, and opens it door
10) User exits elevator

Extensions :

1)
   a) Button calls elevator but not illuminating
      i) If this is the case the elevator will come to the floor if the user decides to wait.
   b) Button Doesn't call the elevator but lights up
      i) (What do we do here?)
   c) Button doesn't light up and doesn't call elevator
      i) (What do we do here?)

3)
   a) If no user is waiting on floor to enter elevator and not users destination floor
      i) the user may press the close door button
   b) If users try to rush in while the door is closing
      i) Refer to *Obstructing door* use case
   c) If a user needs the door to be held open for an extended period of time (> 10s)
      i) they can hold the open door button to override the closing mechanic
   d) If an obstruction is covering the sensor for an extended period
      i) Refer to *Obstructing door* use case
   e) If elevator weight limit exceeded
      i) Refer to *Weight limit exceeded* use case

COMP 3004 Assignment #2

Suhail Mohamed : 101117886

---

5)
   a) Button of destination floor broken
      i) (What do we do here?)

7)
   a) If the screen in the elevator is broken
      i) (What do we do here?)

8)
   a) There is a fire while the elevator is running
      i) Refer to *Fire occur* use case
   b) Power goes out while elevator is running
      i) Refer to *Power goes out* use case
   c) Other emergencies occur
      i) Refer to *Help in the elevator* use case

9)
   a) Door is stuck when arriving at floor
      i) (What do we do here?)

Use case#2 : Obstructing door

Primary Actor : user who wants to travel between floors not using the stairs, but is obstructing door

Stakeholders and Interest :

- Users
- Owner of the building
- Elevator Installment Company and elevator builders of elevator
- Elevator inspectors

Preconditions : User wants to hold the door open longer than normal

Minimal guarantee : A user is not injured by the elevator closing doors

Success guarantee : A user is safely able to alter door closing mechanic and enter elevator

Main success scenario :

1) the light sensor will detect obstruction and open the door

Extension :

1)
   a) If obstruction occurs repeatedly in a short period of time
      i) A warning is sounded over the audio system and a text message is displayed
   b) Light sensor doesn't detect obstruction
      i) (What do we do here?)

COMP 3004 Assignment #2

Suhail Mohamed : 101117886

Use case#3 : Fire occurs

Primary Actor : user who wants to travel between floors not using the stairs

Stakeholders and Interest :

- Users
- Owner of the building
- Elevator Installment Company
- Elevator inspectors
- Building safety
- Emergency responders

Preconditions : Building on fire and users in elevator

Success guarantee : Elevator acts in a way in which all users are safely put out of harm's way

Main success scenario :

1) The control system receives a fire alarm signal from the building (from either within the elevator or by building staff) and commands all elevators to go to a safe floor.
2) Audio and text are presented to users informing them about their situation.

Extensions :

1)
   a) If no floor is safe
      i) (What to do here?) → *Users are getting cooked like a hot pocket*🔥

Use case#4 : Weight limit exceeded

Primary Actor : user who wants to travel between floors not using the stairs

Stakeholders and Interest :

- Users
- Owner of the building
- Elevator Installment Company
- Elevator inspectors

Preconditions : Weight limit of elevator is exceeded

Success guarantee : Elevator won't move until weight is reduced

Main success scenario :

1) the elevator remains stationary at the current floor and informs users to reduce weight of elevator via text and audio

Extensions :

1)
   a) If users don't leave after warning
      i) (What do we do here?) → *Free-falling elevator?*

COMP 3004 Assignment #2

Suhail Mohamed : 101117886

Use case#5 : Power goes out

Primary Actor : user who wants to travel between floors not using the stairs

Stakeholders and Interest :

- Users
- Owner of the building
- Elevator Installment Company
- Elevator inspectors

Preconditions : Users in elevator while power goes out

Success guarantee : Elevator acts in a way in which all users are safely put out of harm's way

Main success scenario :

1) The control system receives a power out alarm signal, and an audio and text message is sent to the users informing them about their situation.
2) Each elevator is then moved to a safe floor using the battery power backup
3) Inform passengers to exit from elevator when safe floor is reached

Extensions :

1)
    b) If battery power backup breaks or is empty
        i) (What do we do here?) → *Still a Free-falling elevator?*

Use case#6 : Help in the elevator

Primary Actor : user who wants to travel between floors not using the stairs

Stakeholders and Interest :

- Users
- Owner of the building
- Elevator Installment Company
- Elevator inspectors
- Building safety
- Emergency responders

Preconditions : Users in elevator while power goes out

Success guarantee : Elevator acts in a way in which all users are safely put out of harm's way

Main success scenario :

1) The elevator is fitted with a help button which allows users to inform emergency responders about their situation

Extension :

1)
    a) If no response from building safety within 5 seconds
        i) 911 emergency is called
    b) If no response from user within 5 seconds
        i) 911 emergency is called
    c) If 911 emergency call fails
        i) (What do we do here?)

COMP 3004 Assignment #2

Suhail Mohamed : 101117886

Traceability Matrix:

ID	Requirement	Related Use Case	Fulfilled by	Test	Description
1	Building of N floors is serviced by a group of M elevators	N/A	Ecs	Running the simulation you can observe the different elevators servicing the different floor requests	The *Ecs* class ensures all requests are handled as when the *createPassenger()* function is called, if a requests hasn't been made to the floor the created passenger is on, the Ecs calls *allocateElevator()* to allocate 1 of the M elevators to the requested floor
2	Each Elevator has a display and audio system	Using an elevator to travel **(UC1)**	Ecs	The simulation has a screen for each elevator, which displays messages	Using *QTTextBrowsers* our simulation is able to allow for elevator displays to be viewed.
3	Pause simulation button	N/A	Ecs	Checking the pause simulation checkbox pauses the simulation	Using the *QTCheckBox isChecked()* function if the checkbox for the pauseSimBox is checked when *runSimulation()* is called it returns right away, not doing any computations, effectively pausing the simulation
4	User presses the up or down button near the floors elevator entrance	Using an elevator to travel **(UC1)**	Ecs, Elevator	Running the simulation you can see floor displays change to reflect what button a passenger has pressed on a floor	When creating a passenger using the createPassenger() function, the function also returns information in regards to what direction (up or down) the passenger wants to go. This information is captured by the *clicked* variable in runSimulation. The floor display is then updated to reflect this change.
5	Elevator arrives at floor and rings bell	Using an elevator to travel **(UC1)**	Ecs, Elevator	Running the simulation you can see elevators ringing their bell on arrival	The *Elevator* class returns a *QString* every time the *travel()* function is called, which allows the *Ecs* to update the display for the elevator in the simulation. When an elevator reaches a destination floor it returns *arrivalMsg*, which returns the message: *"BELL RINGS! Doors open"*

COMP 3004 Assignment #2

Suhail Mohamed : 101117886

6	Elevator doors open allowing for passengers to leave and enter	Using an elevator to travel **(UC1)**	Elevator, Sensor	Running the simulation you can see that the elevators stay on the floor for a period of time to allows passengers to leave	When an elevator arrives at a destination floor it calls its sensors' *arriveAtFloor*() function which starts the sensor's timer, this timer forces the elevator to wait on the floor for some time.
7	Elevator bell rings again to indicate its departure and closes its door	Using an elevator to travel **(UC1)**	Ecs, Elevator	Running the simulation you can see elevators ringing their bell on departure	Similar to how the elevator returns a *QString* on arrival it also return a *QString* on departure, that string being: *"BELL RINGS! Doors close"*
8	Elevator determines path to drop off users within it	Using an elevator to travel **(UC1)**	Elevator	Running the simulation you can see users get dropped off at their desired floor	The elevator uses a simple algorithm to move passengers, the elevator simply moves to its highest floor, then once the highest floor is reached it moves down to its lowest floor. This allows all passengers in the elevator to get dropped off at some point in time
9	User looks at the screen within the elevator to determine what floor the elevator is currently on during travel	Using an elevator to travel **(UC1)**	Ecs, Elevator	Running the simulation the elevator display always displays this information while travelling	When the *Ecs* calls an elevator's *travel()* function it returns a *QString*. The *QString* returned always has information from *showTravelInfo()*, this information includes the current floor and future destination floors.
10	Pair of buttons which open and close door	Using an elevator to travel **(UC1 extension)**	Ecs, Elevator, Sensor	Running the simulation occasionally a passenger presses the hold or close buttons	When the *Ecs* calls the *pickupPassengers()* it checks if a user has pressed the hold or close button. If they have pressed hold, the elevator calls its sensors' *waitOnFloor()* function. If the passenger pressed close it calls the *leaveFloor()* function
11	Sensor will detect an obstruction and open the door	Obstructing door **(UC2)**	Ecs, Elevator	When a user obstructs the elevators door a message appears in the elevator display	When the *Ecs* calls the *pickupPassengers()* function the Ecs checks if any passengers have the *coveringSensor* boolean value as true, if they do it displays the message: *"Passenger covering elevator door, please move out of way"*

COMP 3004 Assignment #2

Suhail Mohamed : 101117886

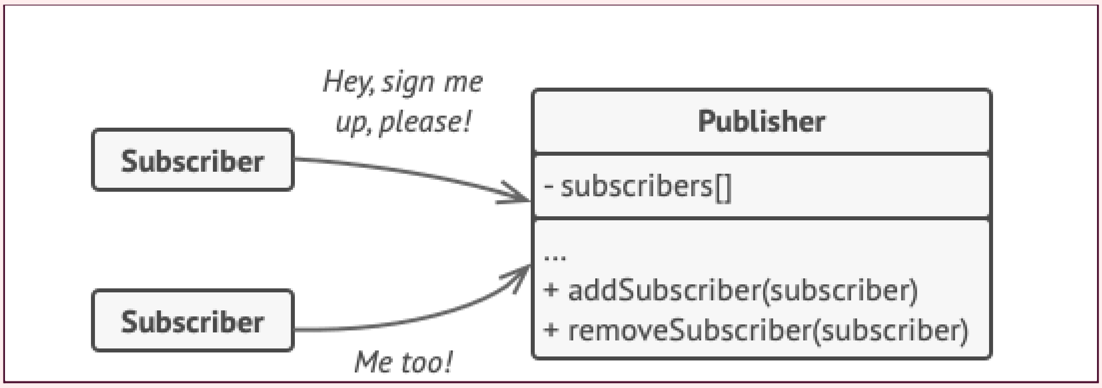
12	The control system receives a fire alarm signal from the building and commands all elevators to go to a safe floor.	Fire Occurs **(UC3)**	Ecs, Elevator	When the user of the simulation presses the fire button the Elevator system acts accordingly	When a user presses the fire button, the *emergencyButtonsClicked()* is signalled, the function then determines what emergency has occurred and sets the corresponding flags for the emergency. For fire, this would be the *fireEmergency* flag. Since the flag has been set when *Ecs* calls its *runSimulation()* function it will be notified of the fire emergency and call *runSafeMode()*, run safe mode clears the building of passengers and requests all elevators to travel to the *SAFE_FLOOR = 1* and updates the elevator displays. Once the elevators have travelled to the safe floor passengers are released and the simulation starts again.
13	the elevator remains stationary at the current floor and informs users to reduce the weight of the elevator via text and audio	Weight limit exceeded **(UC4)**	Ecs, Elevator, Sensor	When the elevator's weight limit is exceeded, the elevator stops taking in passengers and moves along.	When the *Ecs* calls the *pickupPassenger()* the elevator calls its *addPassenger(Passenger)* function, which using the elevator's sensor *checkOverload()* function checks if the passenger can be added to the elevator. If they can't, the elevator moves on and another elevator is allocated for the floor.
14	The control system receives a power out an alarm signal, and an audio and text message is sent to the users informing them about their situation.	Power goes out **(UC5)**	Ecs, Elevator	When the user of the simulation presses the power out button the Elevator system acts accordingly	This works similarly to the fire out case, except the message displayed on the elevator system changes to *"POWER OUT IN BUILDING"*
15	The elevator is fitted with a help button which allows users to inform emergency responders about their situation	Help in the elevator **(UC6)**	Ecs, Elevator	When the user of the simulation presses the help button the Elevator system acts accordingly	This works similarly to the fire out case, except the message displayed on the elevator system changes to *"HELP REQUESTED, System Reset".*

COMP 3004 Assignment #2

Suhail Mohamed : 101117886

Design Decisions:

The design of the elevator system is based on the observer pattern. With the *Elevator* class being interested in listening to when the *runSimulation()* function is called in the *Ecs*. Using the proper jargon the *Ecs* class would be considered a publisher and the *Elevator* class would be considered a subscriber.



From : https://refactoring.guru/design-patterns/state

When the *runSimulation()* function is called *Elevators* call their *travel()* method which updates their state. After this, the *Ecs* can use the updated state of the *Elevators* to do other work, such as potentially adding passengers to the elevator. The reason this design decision was made was that it was the easiest way to implement this interaction between *Elevators* and the *Ecs*, as the *Ecs* controls the flow of the program it made sense to make *Elevator* subscribers to it as events are driven by the code within it.