Elevator project - TTK4135

System documentation

Asgeir Hunshamar
Olav Vassbotn

April 2019

1 Class diagram

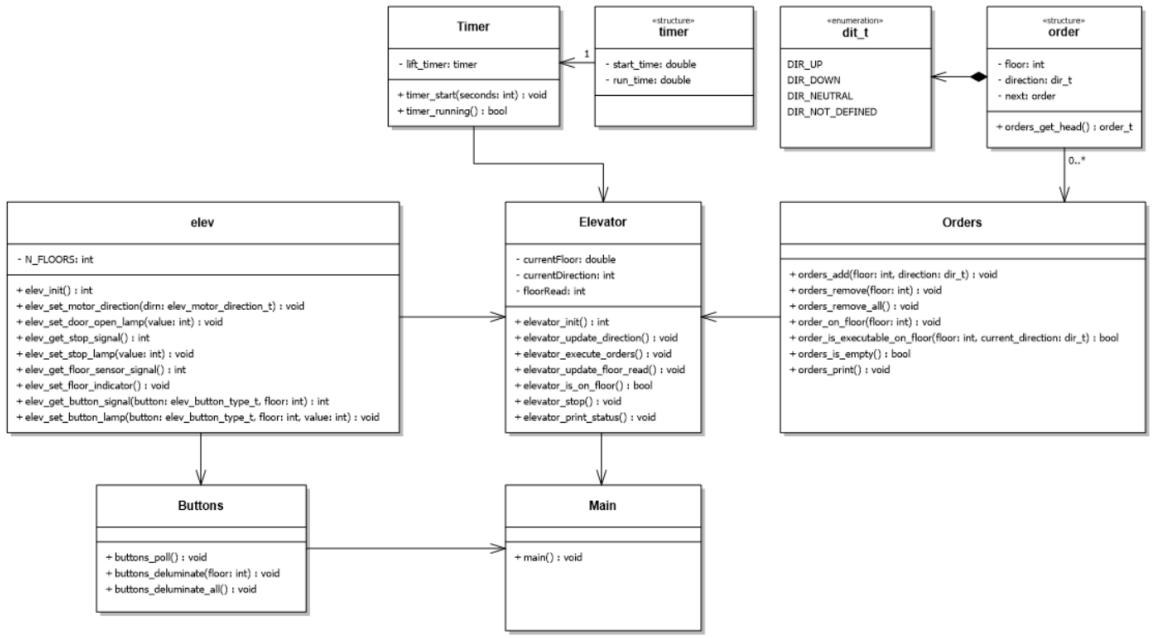

Figure 1.1: Class diagram for the elevator project

2 Sequence Diagram

Here are sequence diagram of scenario 2.1 the use case 'Elevator passenger placing an order', this should be the same regardless of where the elevator is prior to the button press.

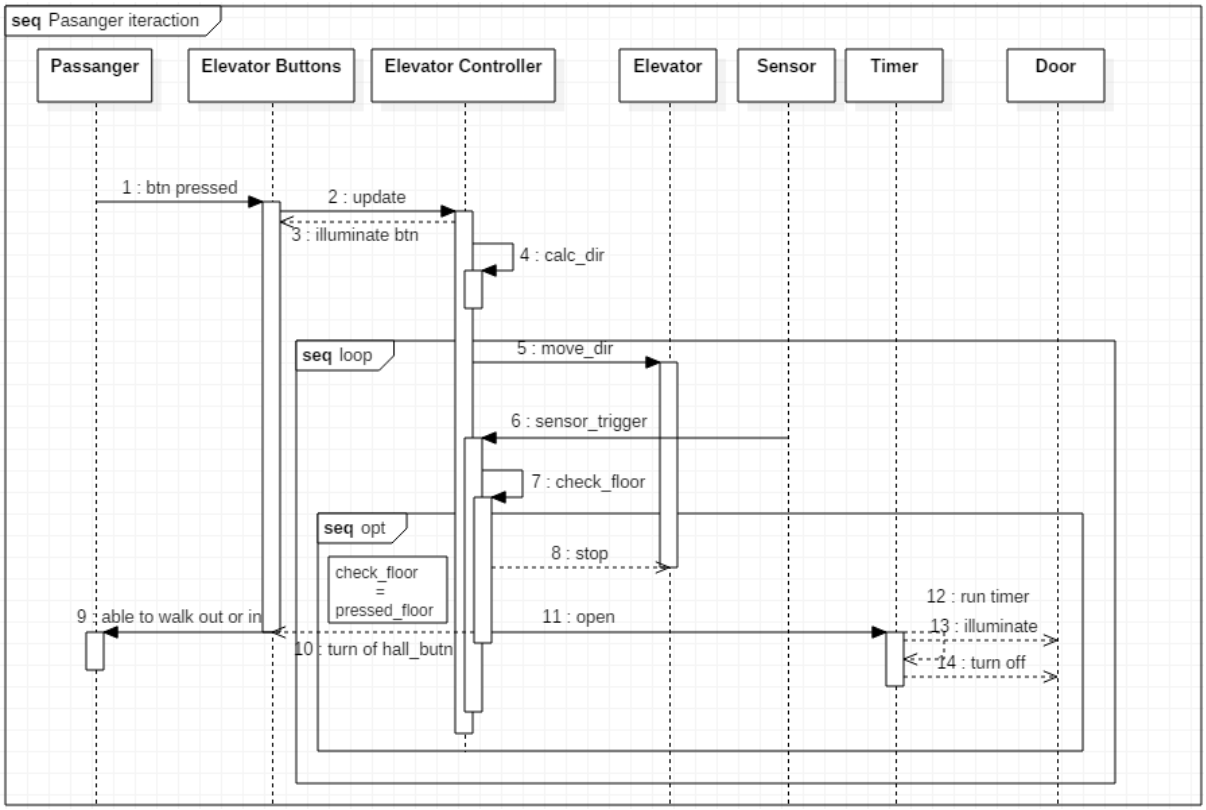

Figure 2.1: Sequence diagram of 'Elevator passenger placing an order'

2.1 Elevator passenger placing an order

Main Success Scenario:

1. Elevator passenger places an order from inside the elevator
2. The elevator continually takes care of orders already in the system
3. The elevator deletes orders that have been completed
4. The elevator only changes direction if it does not have more orders in the direction it previously had
5. The elevator stops upon arrival at desired floor
6. The order is removed and the door is opened
7. The door closes after a timer is triggered after 3 seconds
8. The elevator takes no new actions until a new order is placed

Extensions:

2a. The elevator arrives at the ordered floor

    1. Jump to step 5

2.2 Elevator passenger pressing stop button

Main Success Scenario:

1. Elevator passenger presses stop button
2. The elevator stops
3. All orders are removed from list
4. The door is opened and a timer is started
5. The door closes after timer has reached 3 seconds
6. The elevator remains still until a new order is placed

Extensions:

3a. The elevator is between floors

    1. Jump to step 6

3 State machine diagram

Figure 3.1 shows the state machine diagram for the elevator.

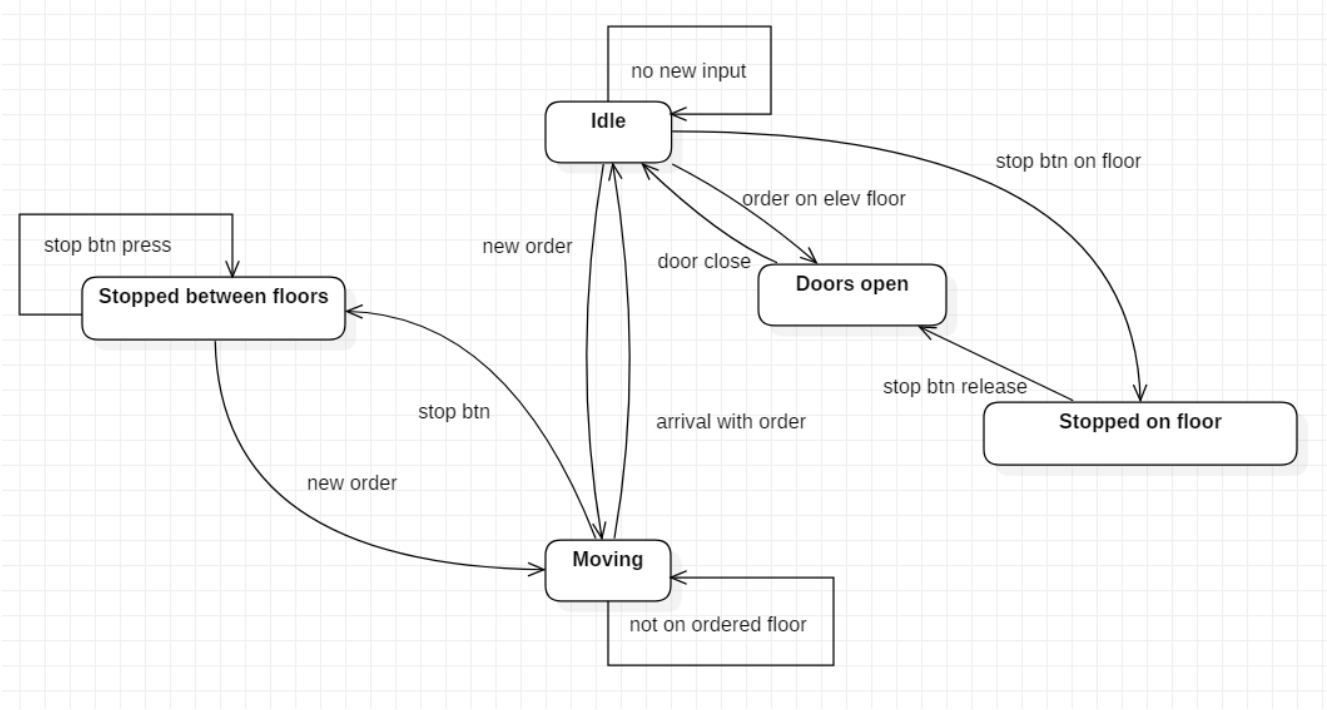

Figure 3.1: State machine diagram for the elevator

4 Activity diagrams

This is the activity diagram for the use case 2.2 called 'Elevator passenger pressing stop button'.

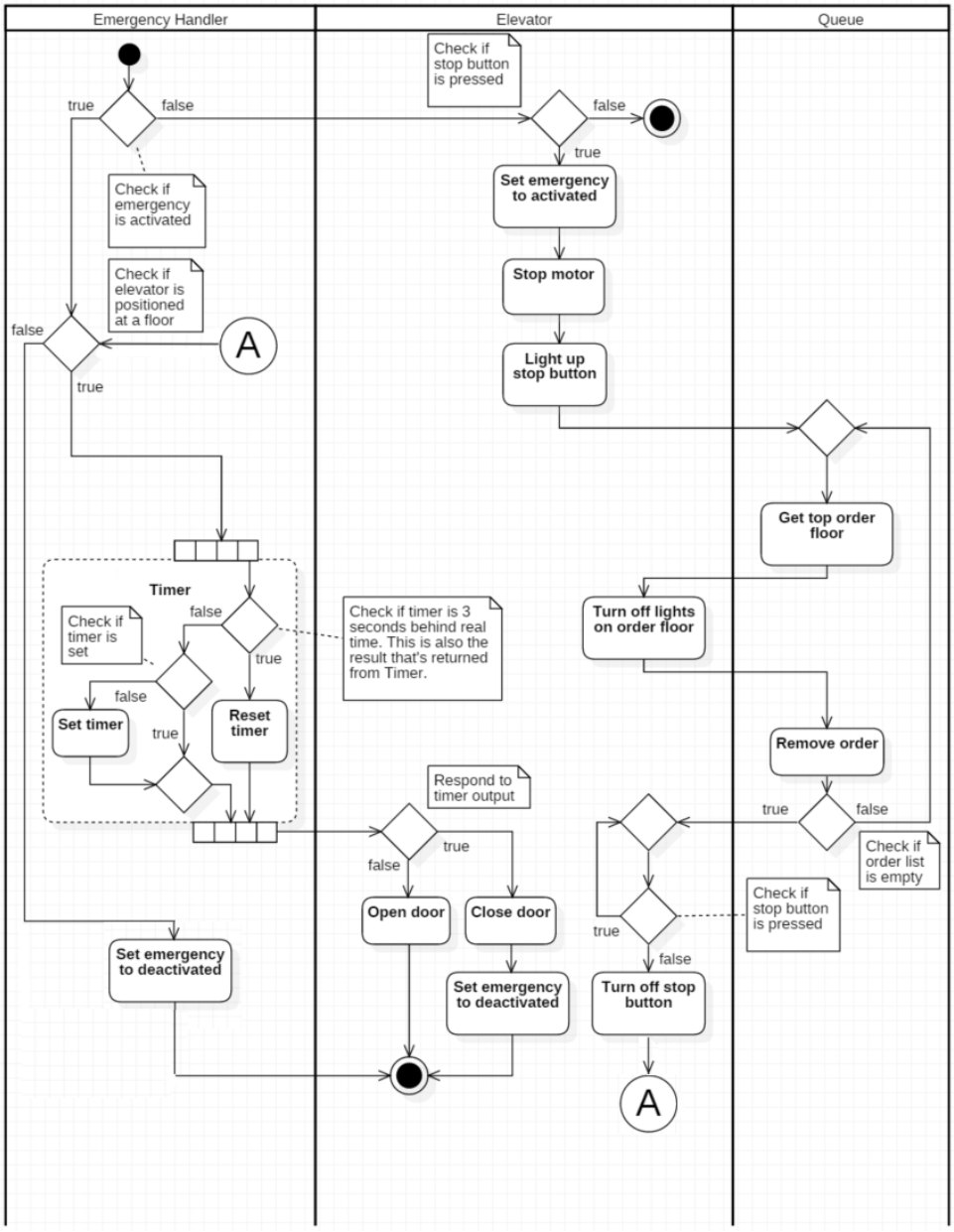

Figure 4.1: Activity diagram of 'Elevator passenger pressing stop button'

As we decided only to let orders with solvable requirements to be added, this meaning there is no placeholder objectives filling up unused space, and because this was a feature we didn't initially plan to implement, we've included an activity diagram to the process of adding an order. The diagram is an illustration of the `queue_addOrder()` in the 'Queue' module.

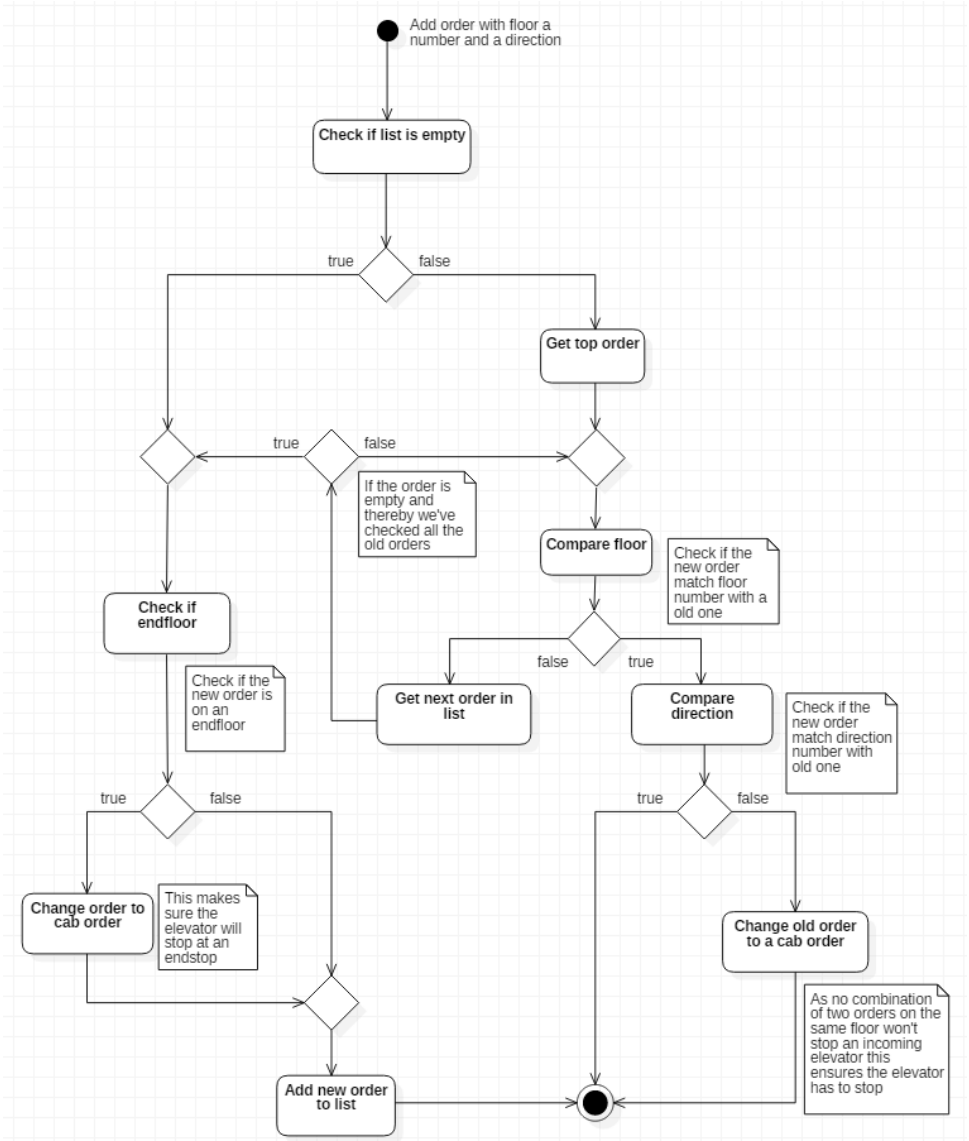

Figure 4.2: Activity diagram of the `queue_addOrder()` function.

This function allowed for simplifications many places in the system, as for example illustrated in Figure 4.1 making it easy to remove orders, check for stops and if check if there are no orders.