Group 13

Author: Liu Chengqi

Software Specifications

Elevator

Table of Contents

[System Architecture 2](#_Toc10413072)

[Software Specifications 2](#_Toc10413073)

[S1: ButtonPanel\_e implementation 2](#_Toc10413074)

[S1.1: Choose one floor to arrive 2](#_Toc10413075)

[S1.2: Open the door 5](#_Toc10413075)

[S1.3: Close the door 5](#_Toc10413075)

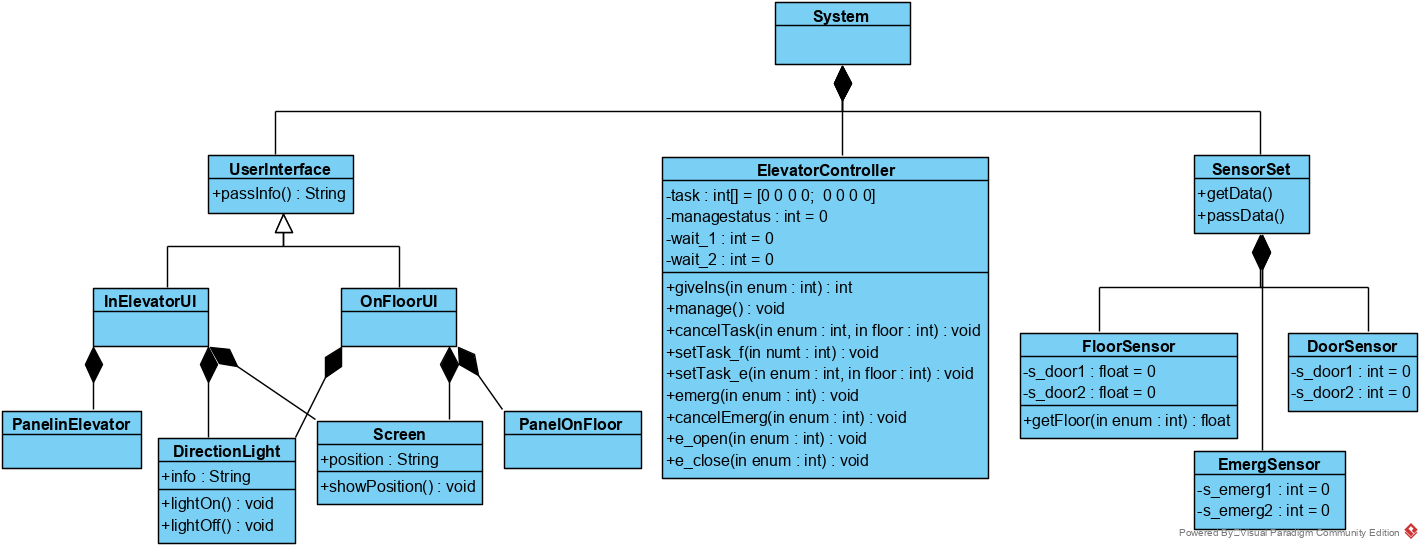
[S1.4: Ask for help in emergency 5](#_Toc10413075)

[S2: ButtonPanel\_f Implementation 5](#_Toc10413075)

[S2.1: Choose one direction to ask for the elevator 6](#_Toc10413075)

## System Architecture

The system architecture is shown below:



(InElevator UI means the ButtonPanel\_e.m. OnFloorUI means the ButtonPanel\_f1.m, ButtonPanel\_f2.m and ButtonPanel\_f3.m.)

## Software Specifications

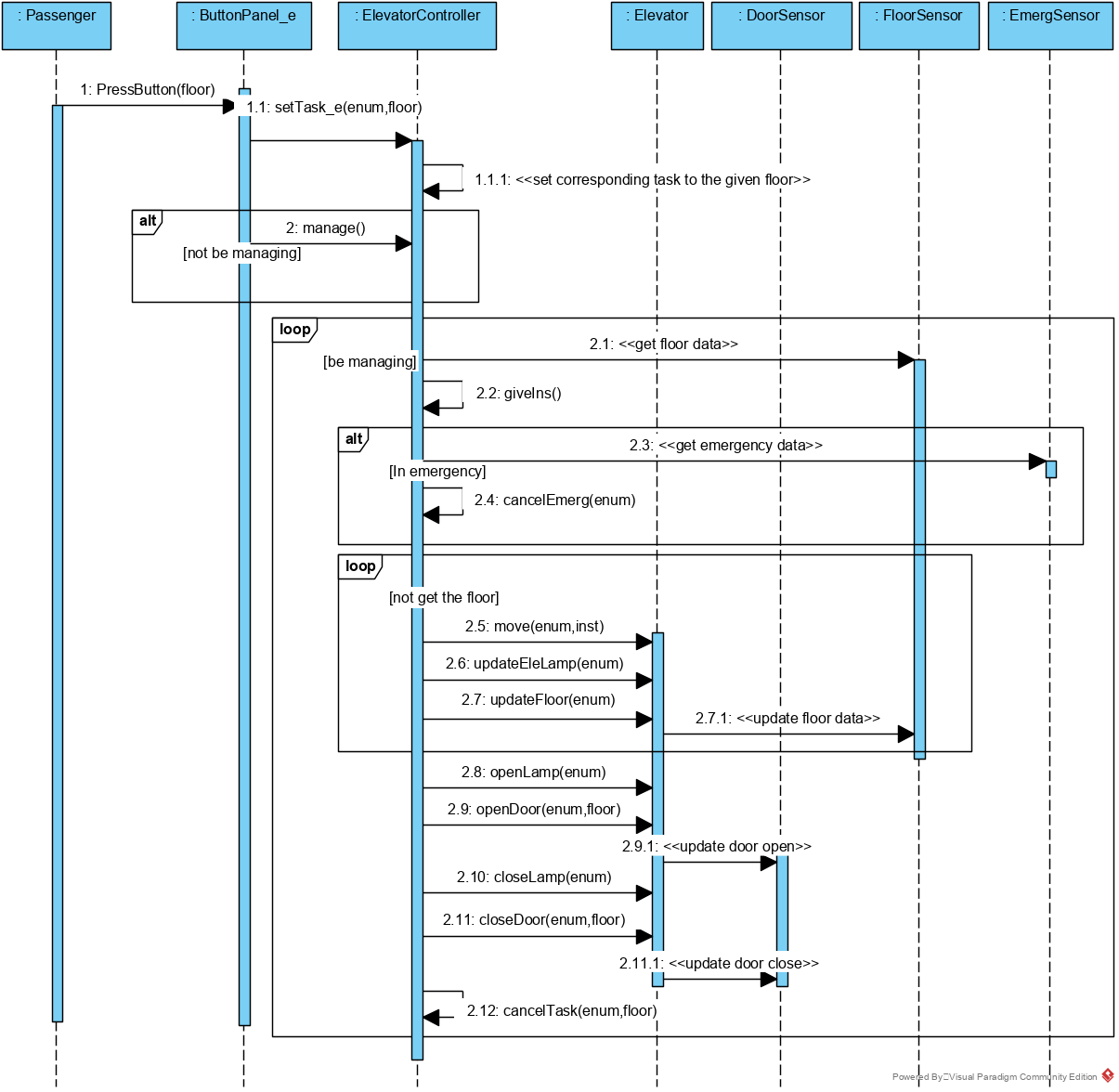
### S1: ButtonPanel\_e implementation

(Let's take elevator1 as an example. Because the implementation of elevator2 and elevator1 are exactly the same.)



#### S1.1: Choose one floor to arrive

(In the figure below, ‘enum’ means the elevator’s number. It is 1 or 2. The same is below in the after figure.)



* S1.1.1: Choose one floor to arrive

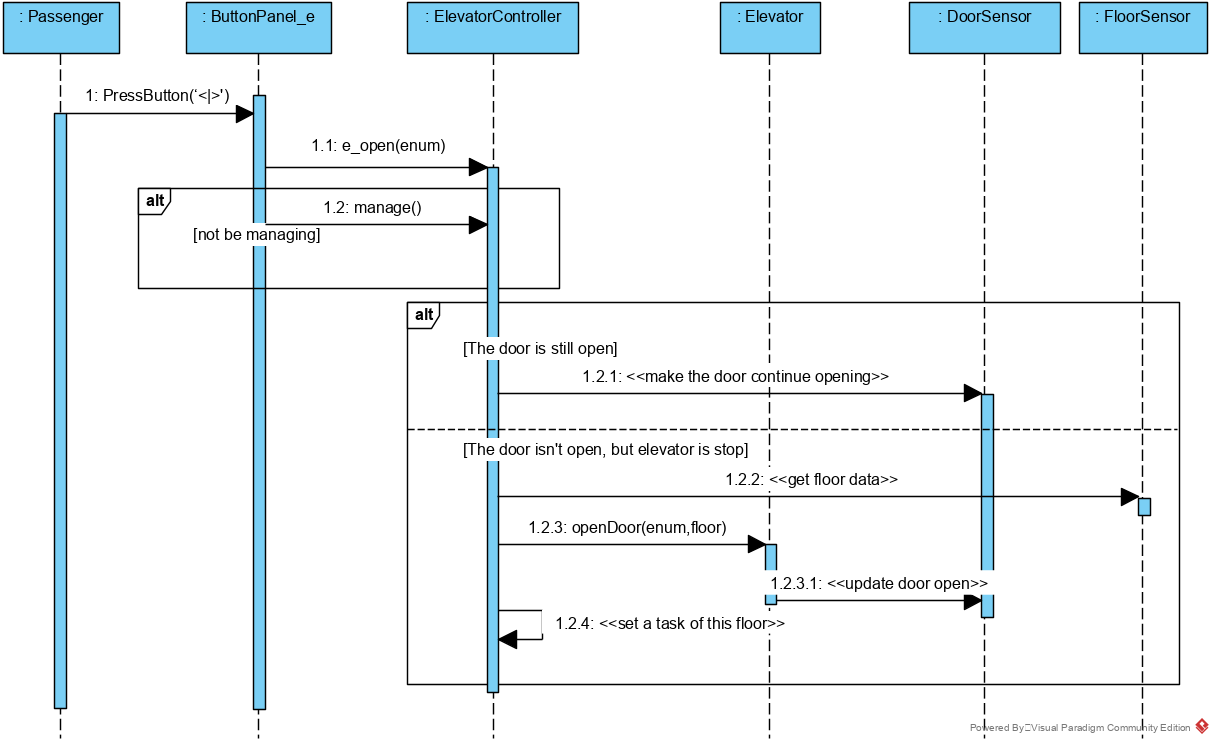
1. Press the button of the floor number
2. Make the button unable to be press again
3. If the elevator is still on the floor,

Make this operation invalid and return.

1. Ask the ElevatorController to set the task for elevator
2. If the ElevatorController is not managing

Start it managing

#### S1.2: Open the door



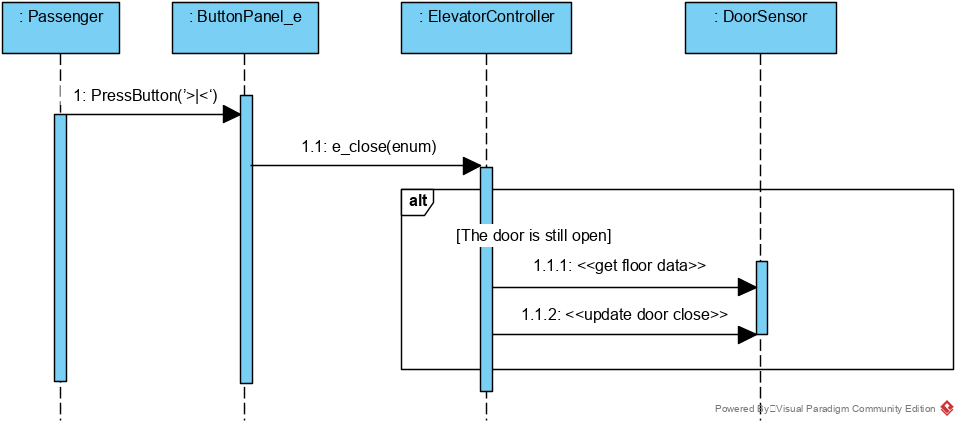
* S1.2.1: Open the door

1. Press the open-door button <|>
2. Make the button unable to be press again
3. Ask ElevatorController to open the door
4. If the ElevatorController is not managing

Start it managing

1. Let the button pop back

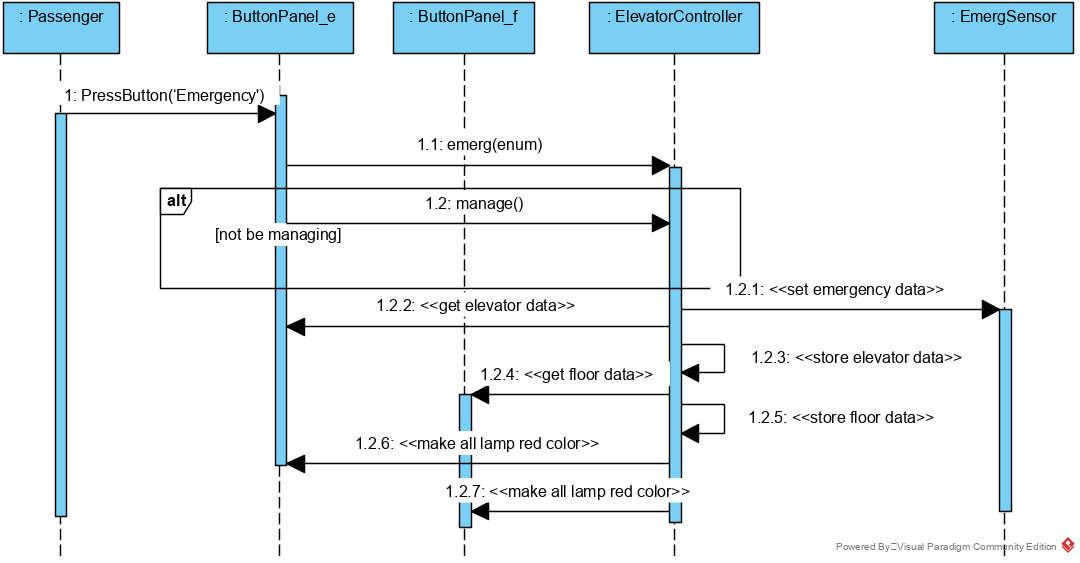
#### S1.3: Close the door



* S1.3.1: Close the door

1. Press the close-door button >|<
2. Make the button unable to be press again
3. Ask ElevatorController to close the door
4. Let the button pop back

#### S1.4: Ask for help in emergency



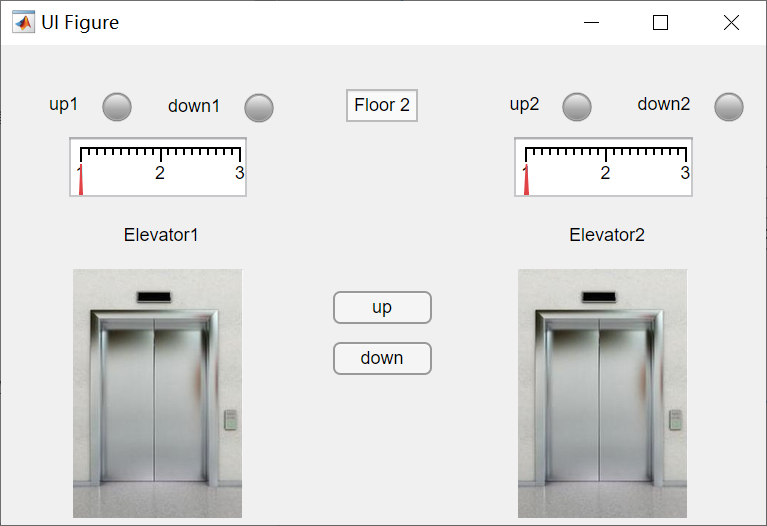
* S1.4.1: Ask for help in emergency

1. Press the emergency button
2. Make the button unable to be press again
3. Ask ElevatorController to deal with the emergency
4. If the ElevatorController is not managing

Start it managing

### S2: ButtonPanel\_f Implementation

(There are three UI implemented just in the same way: ButtonPanel\_f1, ButtonPanel\_f2 and ButtonPanel\_f3. For simplicity, take ButtonPanel\_f2 as an example. The other two UI just act the same.)



#### S2.1: Choose one direction to ask for the elevator

(In the figure below, ‘numt’ means the number of task. It is 1, 2, 3 or 4. It encodes different tasks into numbers. In detail, it is implemented in this way:

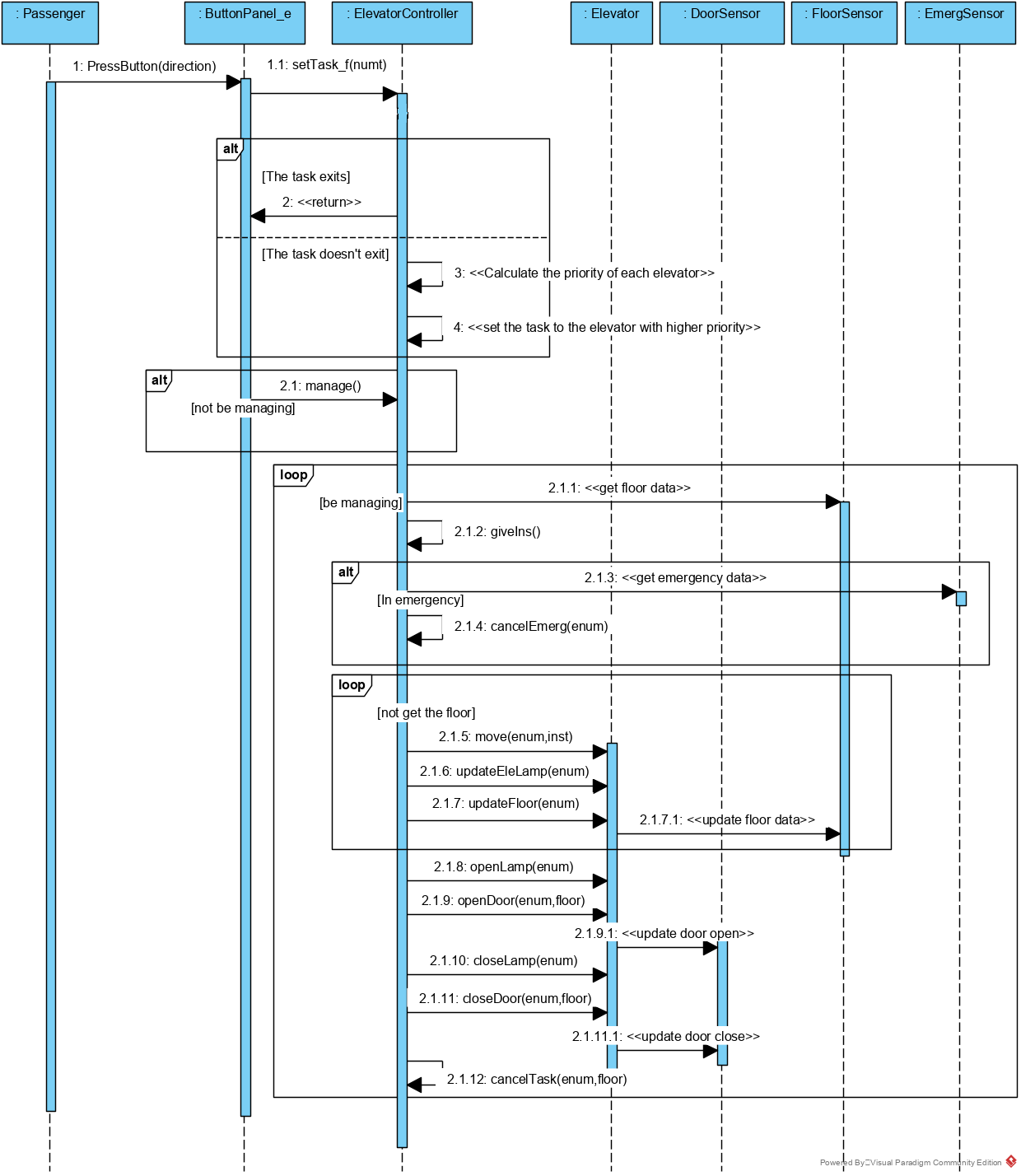
If the passenger press ‘up’ at floor 1, numt is 1.

If the passenger press ‘up’ at floor 2, numt is 2.

If the passenger press ‘down’ at floor 2, numt is 3.

If the passenger press ‘down’ at floor 3, numt is 4.

)



* S1.1.1: Choose one floor to arrive

1. Press the button of one chosen direction
2. Make the button unable to be press again
3. Ask the ElevatorController to set the task for floor
4. If the ElevatorController is not managing

Start it managing