



# SOFTWARE SPECIFICATIONS

Elevator System

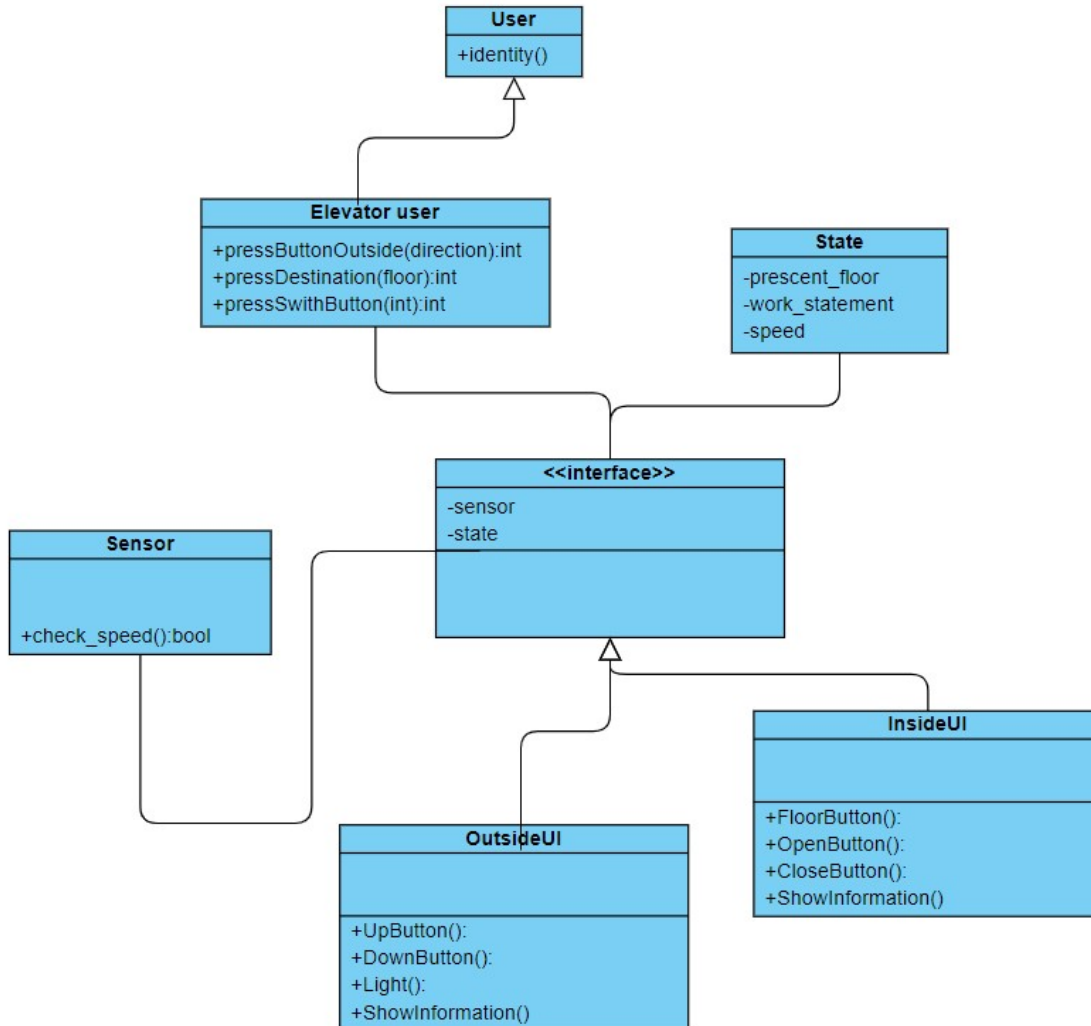
Group 25  
Chuyi Zhao

# Table of Contents

System Architecture.....	2
Software Specifications.....	3
S1: InsideUI implementation.....	3
S1.1: Interface UI design.....	3
S1.2: Push Floor Button.....	4
S1.3: Push Open/ Close Button.....	5
S2: OutsideUIimplementation.....	5
S2.1: Interface UI design.....	6
S2.2: Push Up/ Down Button.....	7
S2.3: Open Door if reach destination.....	7
S3: Controller Implementation.....	8
S3.1: Set Floorheight.....	8
S3.3: Set Open Period.....	8
S3.4: Scheduleing Design.....	8

## System Architecture

The system architecture is shown below:

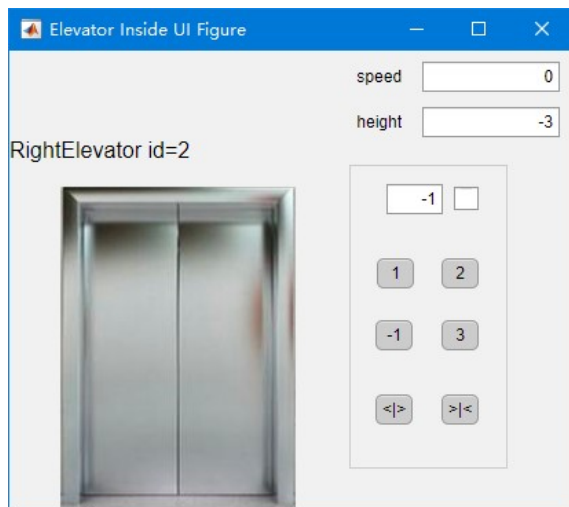
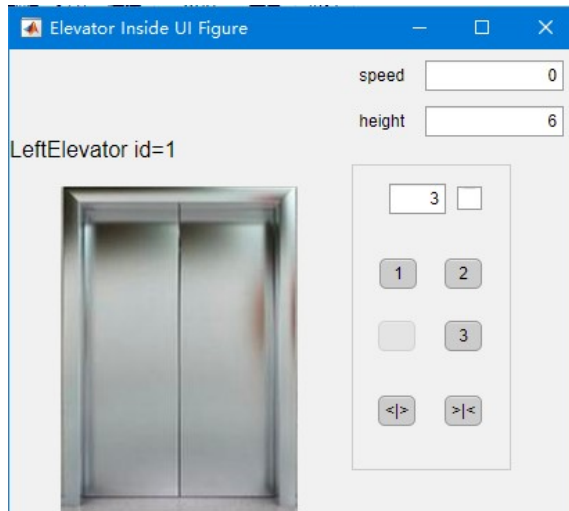


## Software Specifications

### S1: insideUI implementation

#### S1.1: Interface UI Design

This part shows the interior part of the elevator. There are two elevators with index id 1 and 2 respectively. Bellow mainly take elevator 1 as an example.



#### S1.1.1: Speed

Each elevator has a max speed 3m/s, with an acceleration of  $3 \text{ m/s}^2$ , showing in the right up part of the wall.

#### S1.1.2: Height

The current height of the elevator lowest point in meter unit is shown on the right up part of the wall, just bellow the speed box.

#### S1.1.3: Position and Direction

The current position of the elevator lowest point in floor unit is shown on the upmost part of

the elevator panel. For elevator 1, it is in range [1, 2, 3] Floor. For elevator 2, it is in range [-1, 1, 2, 3] Floor.

The current direction of the elevator is shown on the right upmost part of the elevator panel, in range ["↑", "↓", ""], indicating the direction of the elevator speed. Here blank "" means the elevator is idle with no task.

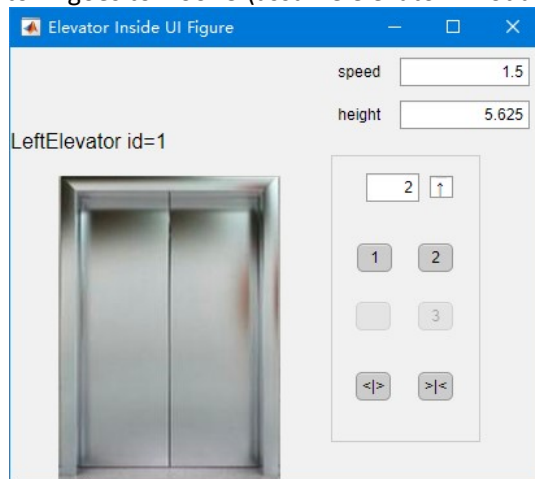
#### S1.1.4: Interact buttons

Different buttons are shown in the panel right to the elevator door, including floor button and open/ close button. Exact usage are discribed more detailedly in the following parts.

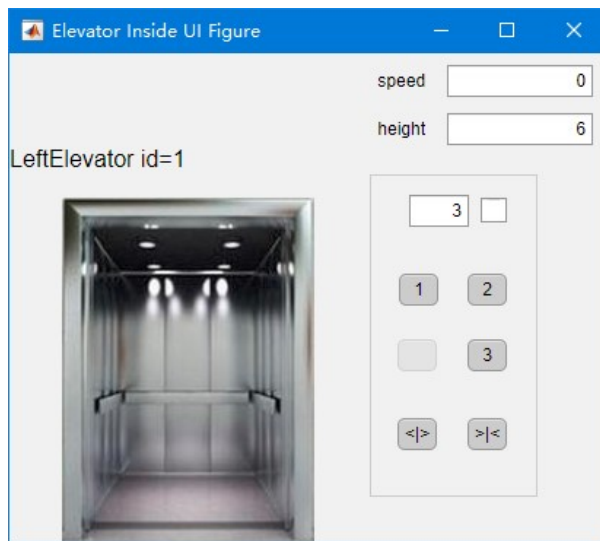
#### S1.2: Push Floor Button

For each elevator, there are three/ four floor buttons on the panel right to door. Bellow take Floor Button 3 for an example.

1. Push Button 3.
2. Button 1 is set unale, meaning the elevatord recieved the instruction signal.
3. Ask the elevaterController to move elevator 1 to floor 3.
4. Elevator 1 goes to Floor 3.(assume elevator 1 not at 3F now)



- (1) Speed acceleratd from 0.
- (2) Direction turns up.
- (3) Speed slows down to 0 from max speed when needed.
5. Once Reach Floor 3 and speed equals 0, door opens.(figure bellow)
6. Button 3 is set enabled.



### S1.3: Push Open/Close Button

#### S1.3.1 Close Button

1. Push `>|<` Close Button
2. If the door is open, it is instantly closed.
3. If the door is close, nothing happens

#### S1.3.1 Open Button

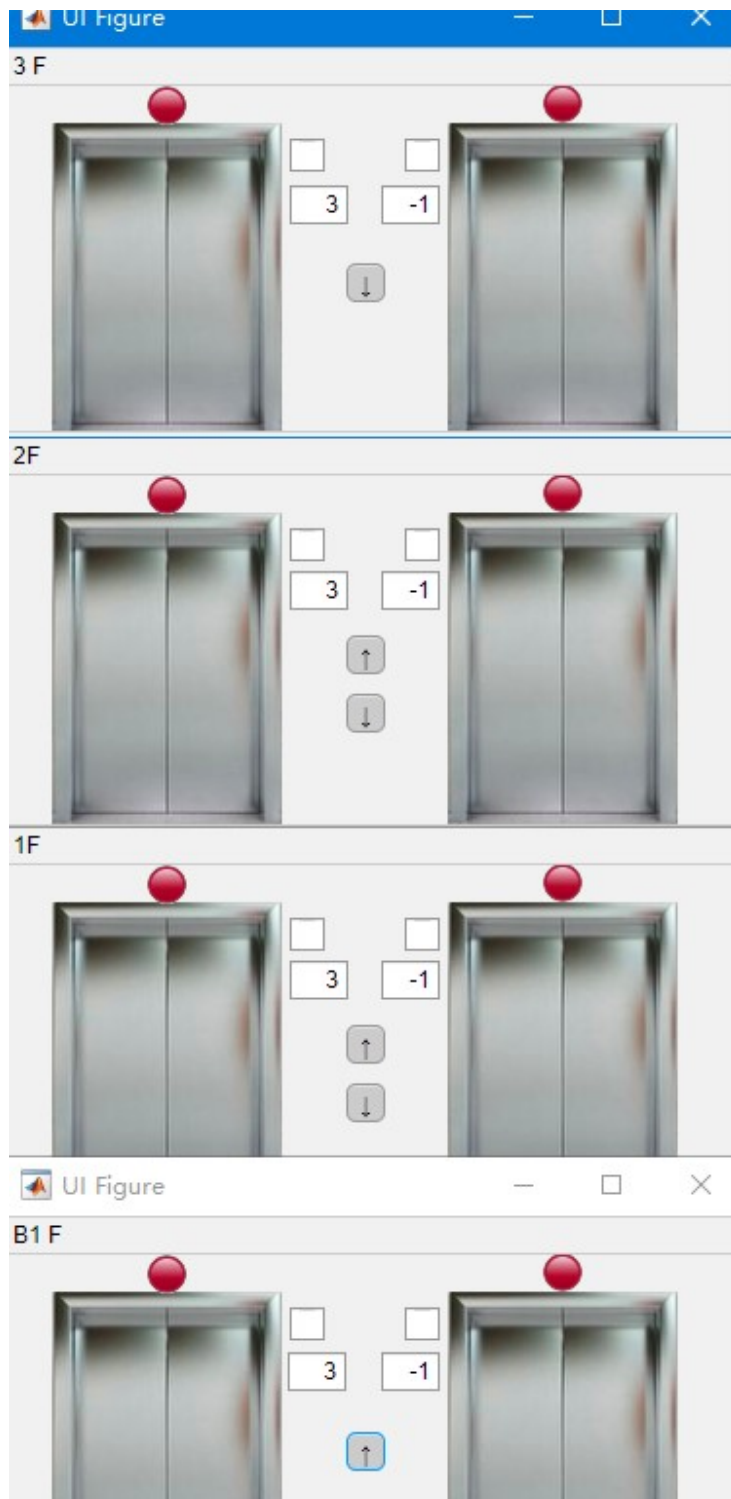
1. Push `<|>` Open Button
2. If the door is close and the elevator is stable<sup>1</sup>:
  - (1) If the speed equals 0, it is instantly open.
  - (2) If the speed  $\neq 0$ , the speed slows down to 0 then close the door.
3. If the door is close and the elevator is not stable<sup>2</sup>, the elevator will stop at the nearest floor and open the door.
4. The door opening last for a `Open Period`. (described later in the last part)
5. If the door is open, its open time last with a delay of `Open Period`.

### S2: OutsideUI Implementation

This part shows the outer part of the elevator. There are two elevators with index id 1 and 2 and 4 floors indexing B1, 1, 2, 3F respectively. Therefore there are 4 outside UIs. Each elevator has a label on its inside UI and each floor has a label indicating the floor index on its outer UI. Bellow mainly take Floor 2 as an example.

<sup>1</sup> The elevator satisfies the condition:  $\text{Height} == \text{Floor Index} * \text{Floor Height}$

<sup>2</sup> The elevator satisfies the condition:  $\text{Height} == \text{Floor Index} * \text{Floor Height}$



*S2.1: Interface UI design*

#### S2.1.1: Positions and Directions

The current position of the two elevators are shown on the middle part of each floor panel. For elevator 1(the left one), it is in range [1, 2, 3] Floor. For elevator 2, it is in range [-1, 1, 2, 3] Floor.

The current direction of the two elevators are shown on the middle part of each floor panel, just beside the position box, in range ["↑", "↓", ""], indicating the direction of the elevator speed. Here blank "" means the elevator is idle with no task.

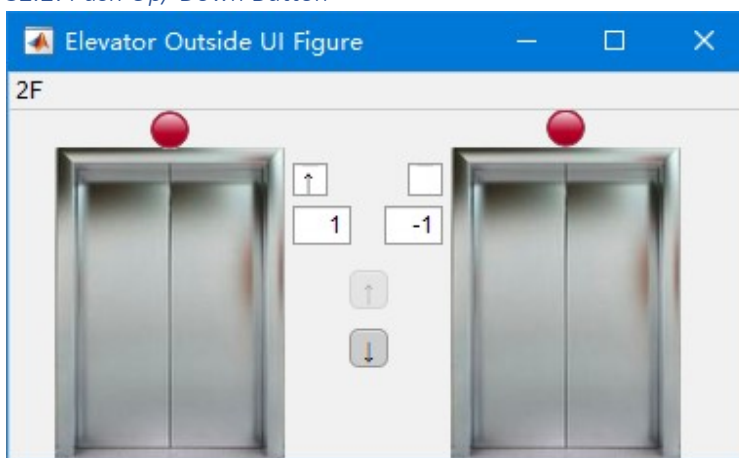
#### S2.1.2: Light

There is one light over every door. It turns green only when the door is open and turns red otherwise. The purpose is to give user a safety signal to enter the elevator.

#### S2.1.3: Interact buttons

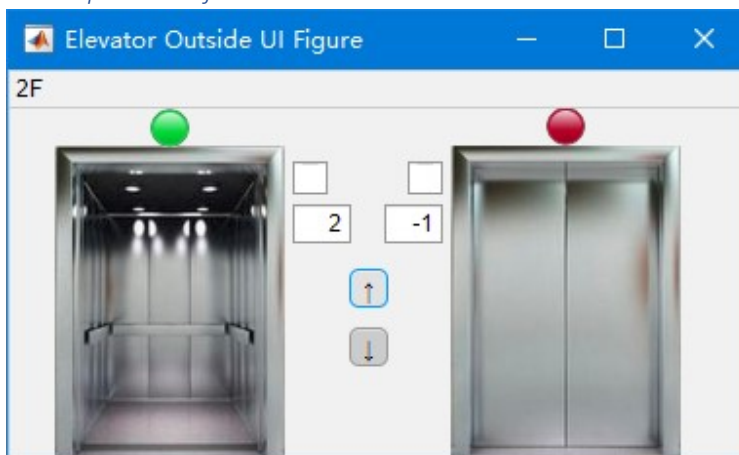
One or Two are shown in the panel right to the elevator door, including floor button and open/ close button. Exact usage are described more detailedly in the following parts.

#### S2.2: Push Up/ Down Button



1. Push the up/ down button at F2.
2. If certain elevator is at F2:
  - (1) The pushed button is set inactive.
  - (2) Ask elevatorController to move any elevator to floor F2.
  - (3) Once any elevator reaches F2, open the door.
  - (4) Button is enabled.
3. Else:
  - (1) Open the door
  - (2) If the button at F2 is pushed again, open time delays

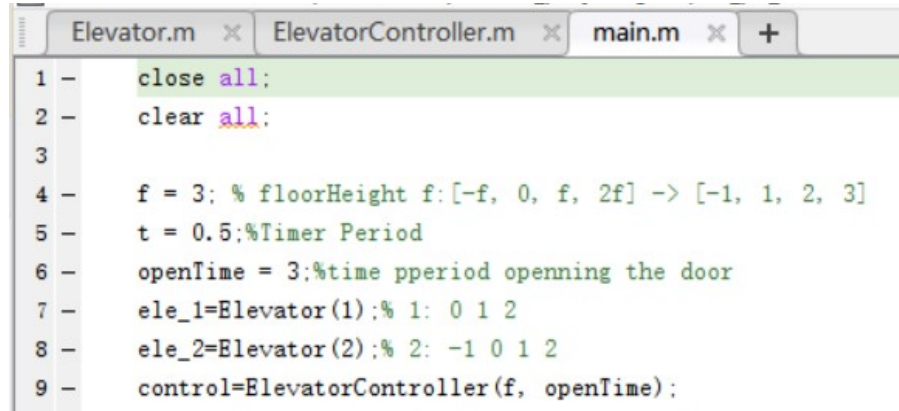
#### S2.3: Open Door if reach destination





As mentioned above, all door UI have such function and the door state at all places are synchronized as the same.

### S3: Controller Implementation



```

1 - close all;
2 - clear all;
3
4 - f = 3; % floorHeight f: [-f, 0, f, 2f] -> [-1, 1, 2, 3]
5 - t = 0.5; %Timer Period
6 - openTime = 3; %time pperiod openning the door
7 - ele_1=Elevator(1); % 1: 0 1 2
8 - ele_2=Elevator(2); % 2: -1 0 1 2
9 - control=ElevatorController(f, openTime);

```

#### S3.1: Set Floorheight

All parameters are set in 'main.m'. Just set variable 'f' to the needed value in main file.

#### S3.2: Set Timer Clock

Set variable 't' to the needed value in main file.

#### S3.3: Set Open Period

Set variable 'openTime' to the needed value in main file.

#### S3.4: Scheduling Design

1. The overall design is composed of 3 '.m' files and 4 '.mlapp' files.
  - (1) .m: main, Elevator, ElevatorController
  - (2) .mlapp: ElevatorInside, ElevatorOutside\_B1, ElevatorOutside\_F1\_F2, ElevatorOutside\_F3
2. For any instruction to certain floor (assumed y here), give ElevatorController instruction for elevator x to go to floor y.
  - (1) Instruction from InsideUI: x=ElevatorInside.id
  - (2) Instruction from OutsideUI:
    - ① If y=-1: x=2
    - ② Else: x=1, 2
3. If x is idle, set y as its destination and go to y.
4. Else: add y to its task list.
5. Once reach y, mark y as finished.
6. Once finish the destination, pop out a destination from task list.
7. Timer Function 'TimerFcn' in ElevatorController.m controls the synchronization of UI sections, e.g. light, door, position, speed, height, direction, etc. And it manages destination and movement (including accelerating, slowing down, opening and closing doors, etc) control. This way elevators can work independently<sup>3</sup> and finish all tasks.

<sup>3</sup> Different elevator state does not influence each other, i.e. one is moving while another one is opening the door with no delay.