Software Requirements

Elevator System

Group 13

Author: 陈怡珺

Table of Contents

[System Objective 2](#_Toc10411767)

[Domain Analysis 2](#_Toc10411768)

[System Architecture 4](#_Toc10411769)

[Use Cases 5](#_Toc10411770)

[Software Requirements 5](#_Toc10411771)

[R1: InElevatorUI 6](#_Toc10411772)

[R2: OnFloorUI 6](#_Toc10411773)

[R3: ElevatorController 6](#_Toc10411774)

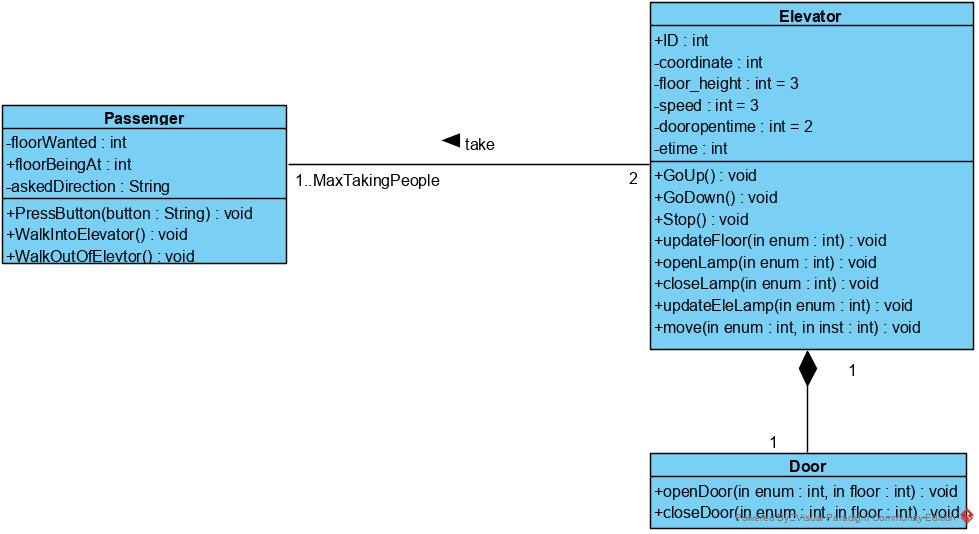
[R4: SensorSet 6](#_Toc10411775)

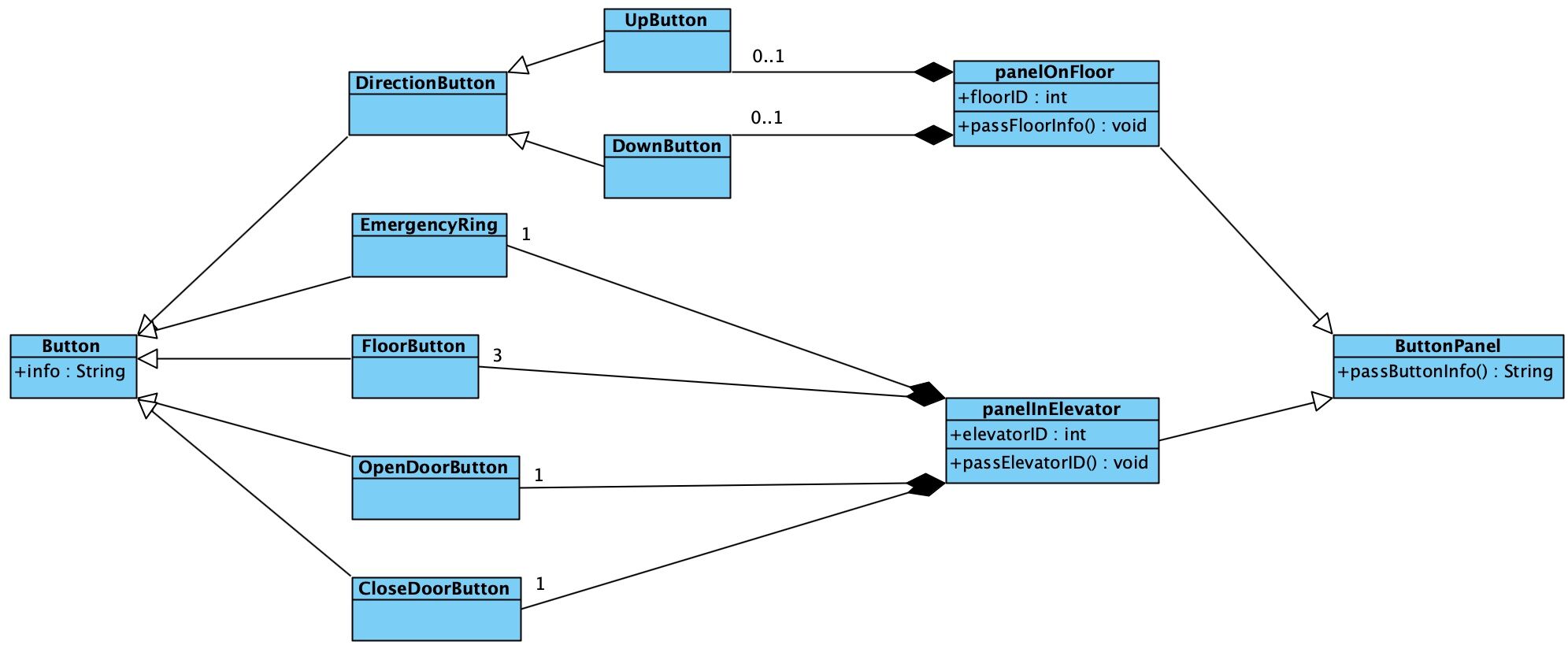
## System Objective

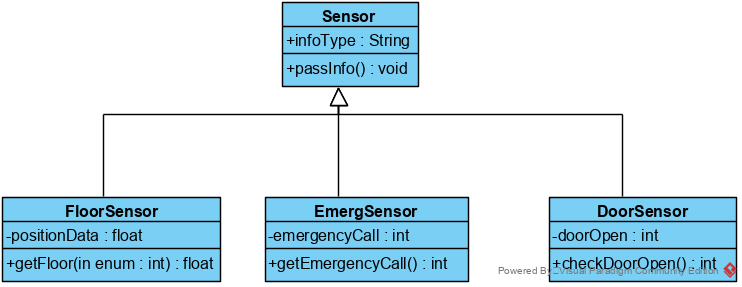
In this project, we are developing a software that can manage the running of two elevators in a building with three floors and improve the efficiency of this process. By providing user interfaces on each floor and in each elevator which contains button panel and screen to the people need to take elevator in the building, the system can reduce delays and the number of errors during taking elevator, which can improve passengers’ satisfaction.

## Domain Analysis

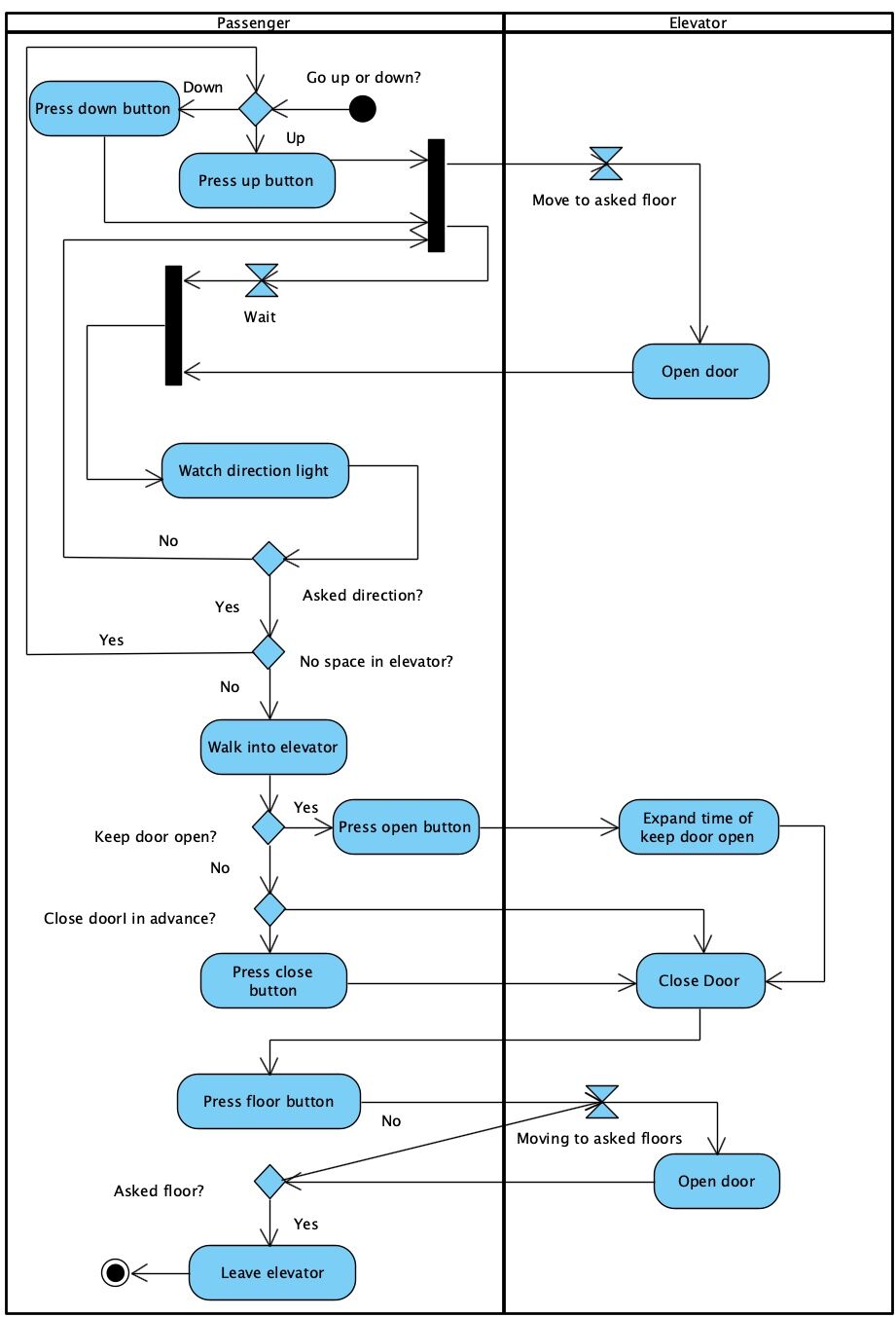
The participants of activities in a restaurant can be categorized into Passenger and Elevator. The relationships among different participants are shown as follows:



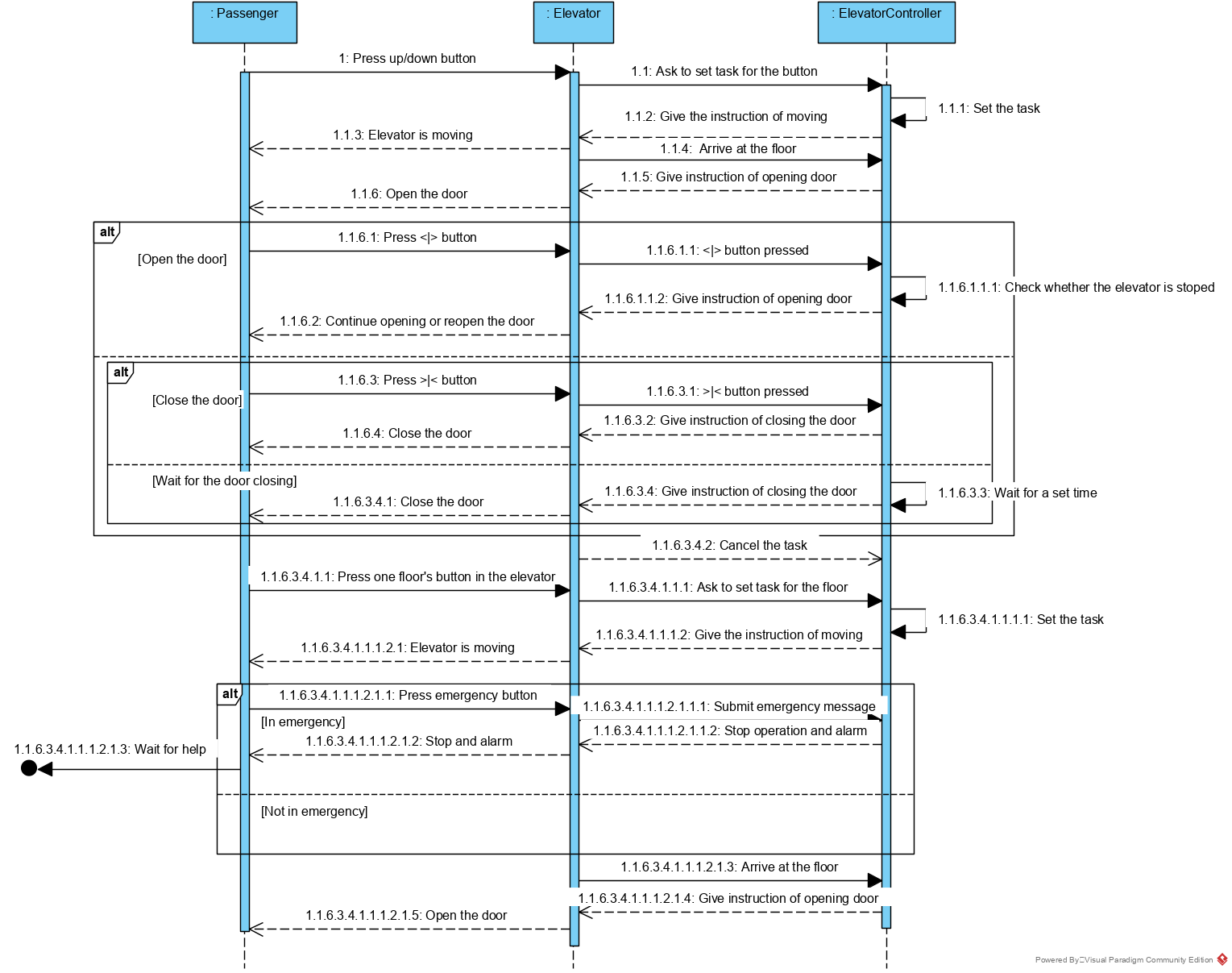




Here is the sequence of events for taking elevator:

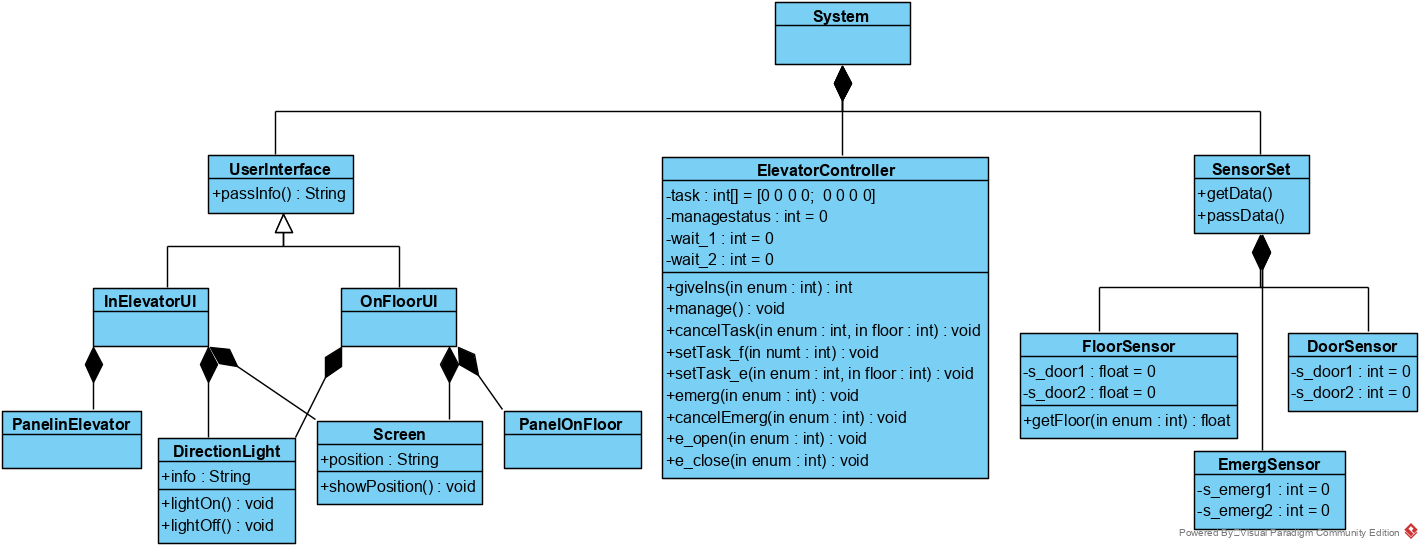


Here is the sequence of data passing when passengers taking elevator:



## System Architecture

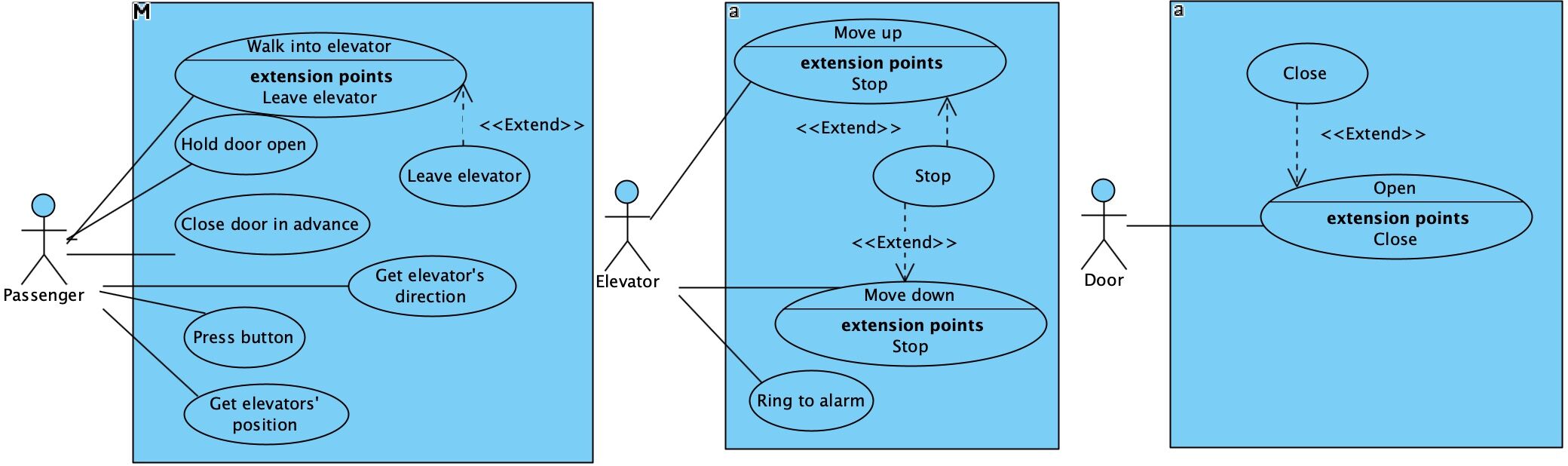
From the information above, we will design a software system that allows the passenger to take elevator to move to the floor they want. Passenger can pass there ask of floor, control door with limit and know the elevators’ moving direction and position, so that it is more efficient and convenient for passenger to take elevator. 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.)

## Use Cases

The system can achieve the following use cases from the server’s and the chef’s perspectives:



## Software Requirements

### R1: InElevatorUI (ButtonPanel\_e)

* R1.1: The passenger should be able to get information of elevator.
  + R1.1.1: The passenger should be able to know elevator’s position.
  + R1.1.2: The passenger should be able to know elevator’s moving direction .
* R1.2: The passenger should be able to control door while the elevator stops moving.
  + R1.2.1: The passenger should be able to ask door to keep open when the door is open.
  + R1.2.2: The passenger should be able to ask door to close in advance.
* R1.3: The passenger should be able to call staff when emergency occurs.
* R1.4: The passenger should be able to select the floor where he/she wants to moving to.
* R1.5: The controller should be able to get the information of the button pressed by passenger.

### R2: OnFloorUI（ButtonPanel\_f1, ButtonPanel\_f2 and ButtonPanel\_f3）

* R2.1: The passenger should be able to get information about two elevators.
  + R1.1.1: The passenger should be able to know elevator’s position.
  + R1.1.2: The passenger should be able to know elevator’s next moving direction.
* R2.2: The passenger should be able to know the floor he/she is on.
* R2.3: The passenger should be able to ask for taking elevator
* R2.4: The passenger should be able to select the direction he/she wants to move.
* R2.5: The controller should be able to get the information of the button pressed by passenger.

### R3: ElevatorController

* R3.1: The controller should be able to assign elevator to handle the ask of taking elevator on each floor.
  + R3.1.1: The controller should not assign both elevators moving to the same floor to take passenger at the same time.
  + R3.1.2: The controller should assign one elevator to moving to the floor once there exists an ask.
  + R3.1.3: The controller should assign the closer elevator to moving to the floor if the elevator is moving at the same direction as the direction to take passenger or it isn’t moving.
  + R3.1.4: The controller should assign the No.1 elevator to moving to the floor when the two elevators are of the same situation.
* R3.2: The controller should react when emergency occur.
* R3.3: The controller should be able to set tasks to both elevators.
  + R3.3.1: The controller should set the order of moving downward/upward.
  + R3.3.2: The controller should set the order of close/open door.
  + R3.3.2: The controller should set the order of moving/stop moving.
* R3.4: The controller should be able to delete task when the task is finish.
* R3.5: The controller should be able to process data to know the information of elevators’ working.
  + R3.5.1: The controller should know elevators’ moving situation.
  + R3.5.2: The controller should know elevator doors’ situation.
  + R3.5.3: The controller should know elevators’ position.
* R3.6: The controller should be able to control door of each elevator.
  + R3.6.1: The controller should ask door to close after opening for a fixed time.
  + R3.6.2: The controller should keep door closed while elevator moving.
  + R3.6.3: The controller should open door after the elevators stop moving.
  + R3.6.4: The controller should hold door when passenger askes before it is time to close door.
  + R3.6.5: The controller should close door in advance when passenger askes.
* R3.7 The controller should be able to assign elevator to handle the ask pass by UI in elevator.
  + R3.7.1: The controller should assign elevator moving to and stopping on each floor asked by passenger.
  + R3.7.2: The controller should assign elevator moving to and stopping on each floor asked in order of the LOOK algorithm.
  + R3.7.3: The controller should not assign elevator moving to floor isn’t asked.

### R4: SensorSet

* R4.1: The controller should get data of elevators all the time.
  + R4.1.1: The controller should be able to get the data that whether the door is open.
  + R4.1.2: The controller should be able to get the position of two elevators.
* R4.2: The controller should get information when emergency occurs.