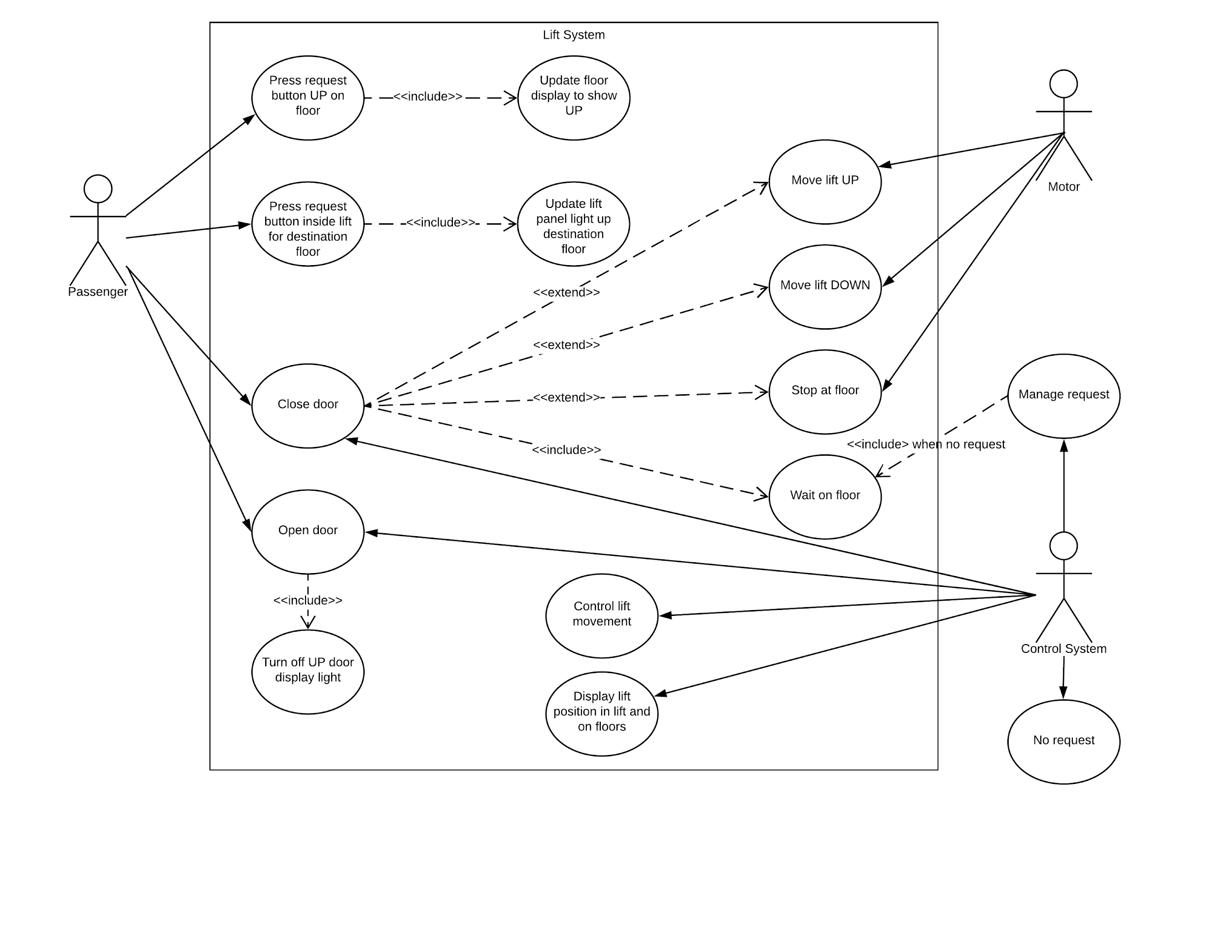
## Analysis

In analysing the description of the problem some assumptions were made:

* The system only needs to manage one lift
* The system does not need to queue requests
* Ground floor would be floor 1 and top floor would be floor 4
* If a passenger presses the UP button on a floor they are requesting to go up
* If a passenger presses the DOWN button on a floor they are requesting to go down
* Regardless of pressing UP or DOWN on a floor the description does not specify that this sets the direction of the lift, it merely calls it to the passengers floor
* After the lift arrives for the passenger they are free to change their mind about going up or down by selecting the exact destination floor from inside the lift

## Use Case Diagram



## Use Case

**Actor** = Passenger

**Summary** = Passenger travels to destination floor of their choice

**Main success scenario:**

1. Passenger calls the lift from the floor that they are currently on
2. Lift moves to the floor the passenger called from
3. Passenger enters the lift
4. Passenger chooses a floor to go to
5. Lift moves the passenger to the floor they chose
6. Passenger exits the lift

**Extensions:**

2a) Lift is already waiting at the floor where passenger is

Lift doors open

5a) Floor selection is made for a floor that doesn’t exist

Selection is rejected and new selection can be made

5b) Floor selection is made for current floor

System shows lift waiting at current floor

Doors will open, if not open already

## User Stories

**User story**

As a passenger I want to use the lift so that I can travel to the floor of my choice

**Acceptance criteria**

1. User should be able to press a lift call button on their current floor to call the lift
2. User should be able to enter the lift
3. User should be able to choose a destination floor
4. User should be transported in the lift to destination floor
5. User should be able to exit the lift

**User story**

As a passenger I want the lift doors to open immediately if I request the lift and it is on the same floor as my current floor

**Acceptance criteria**

1. User should be able to press a lift call button on their current floor to call the lift
2. User should be able to get into the lift

**User story**

As a passenger I want to be able to see where the lift is and its direction of movement before I call it

**Acceptance criteria**

1. User should be able to see a display that shows the current floor floor of the lift
2. User should be able to see a display that shows which direction the lift is moving, up or down

**User story**

As a passenger I want to be able to see which floor I have selected when I press the button in the lift and a confirmation that I am at the destination floor when I reach it

**Acceptance criteria**

1. User should be able to press a button that allow them to select the destination floor
2. When the user presses a button it should visually represent that it has been selected
3. User should see visual confirmation of the selection while the lift is moving
4. User should be able to see visual confirmation that the lift has arrived when it does

**User story**

As a passenger I want the lift to stop moving before it opens the doors for me to enter/exit

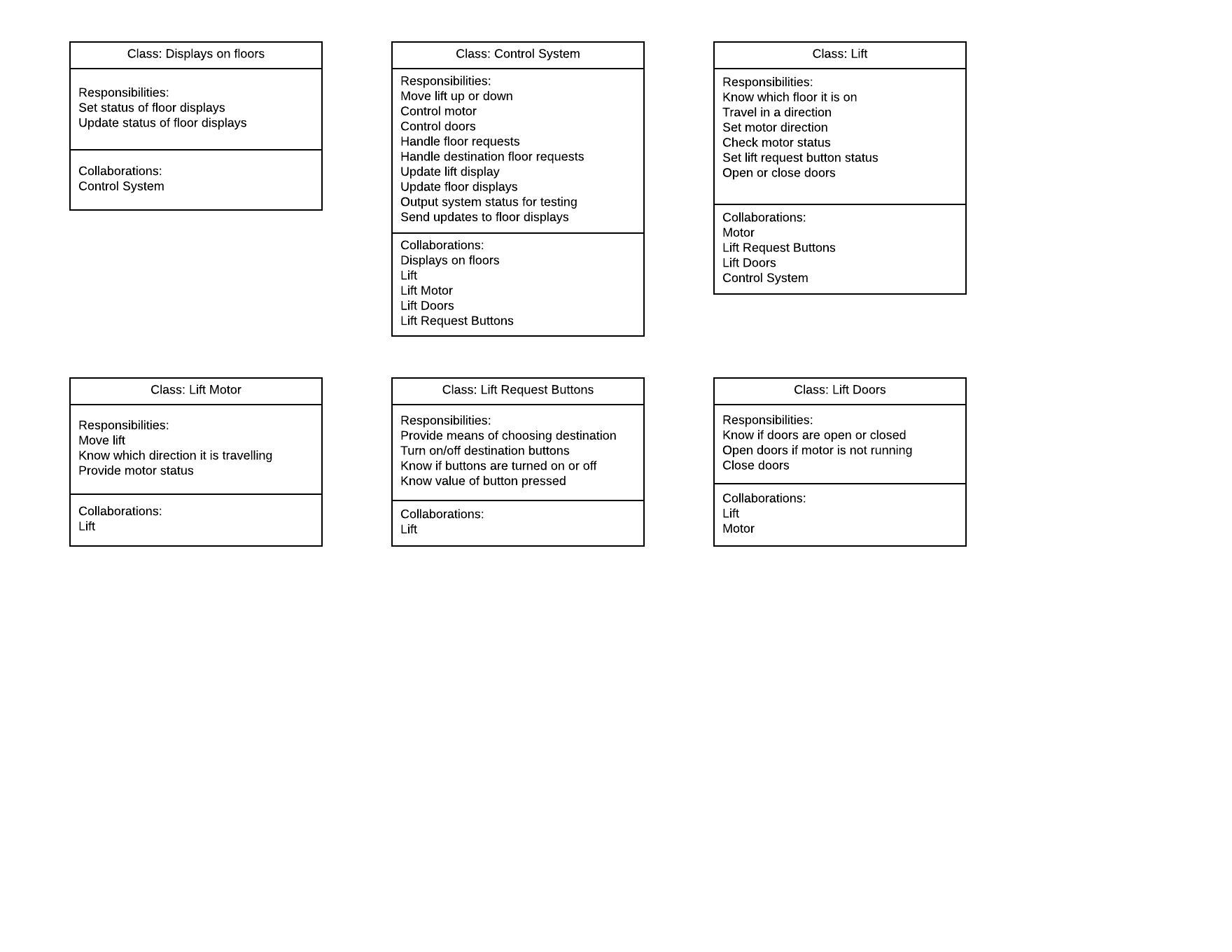
**Acceptance criteria**

1. Lift should not be able to open the doors unless the motor has been turned off

## 

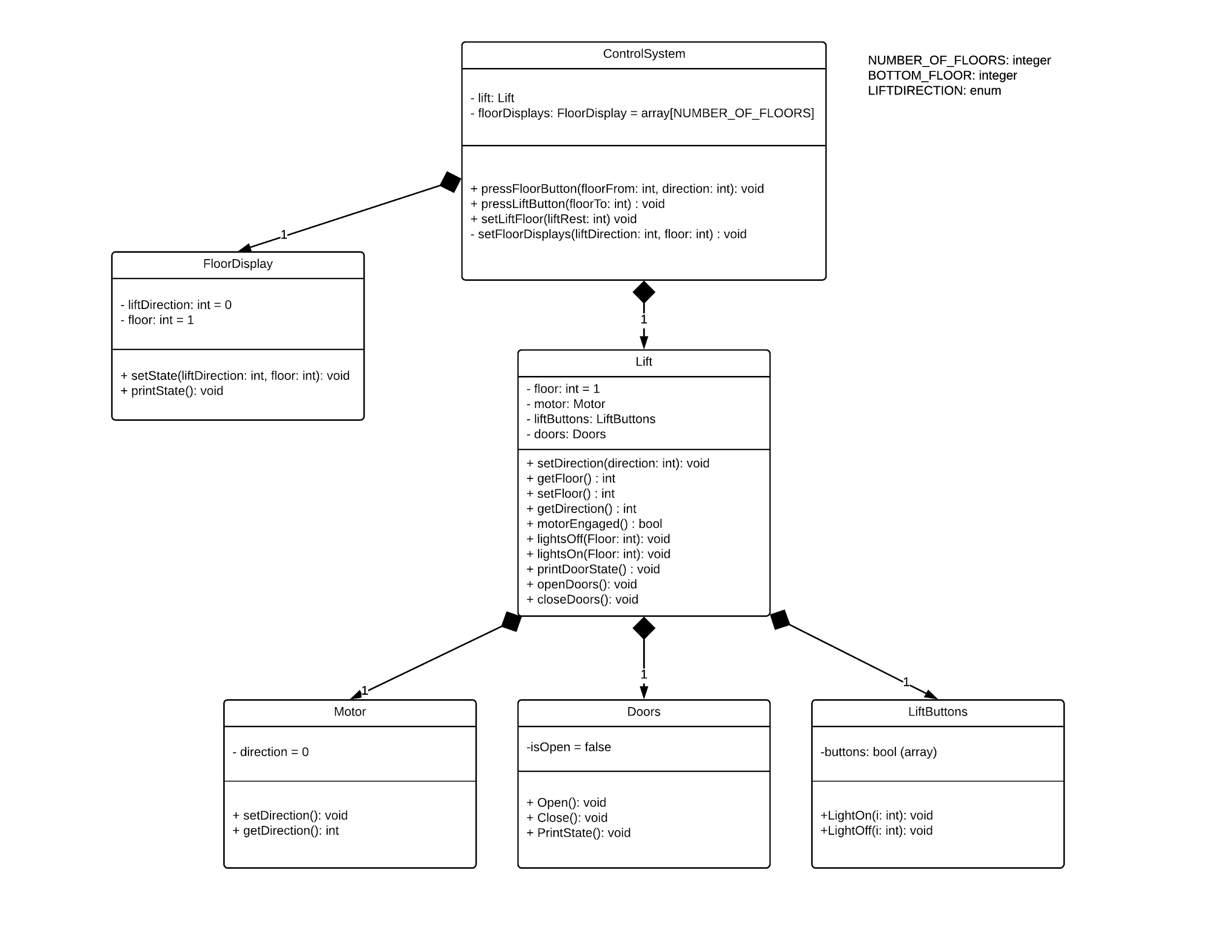
## Design

### CRC cards



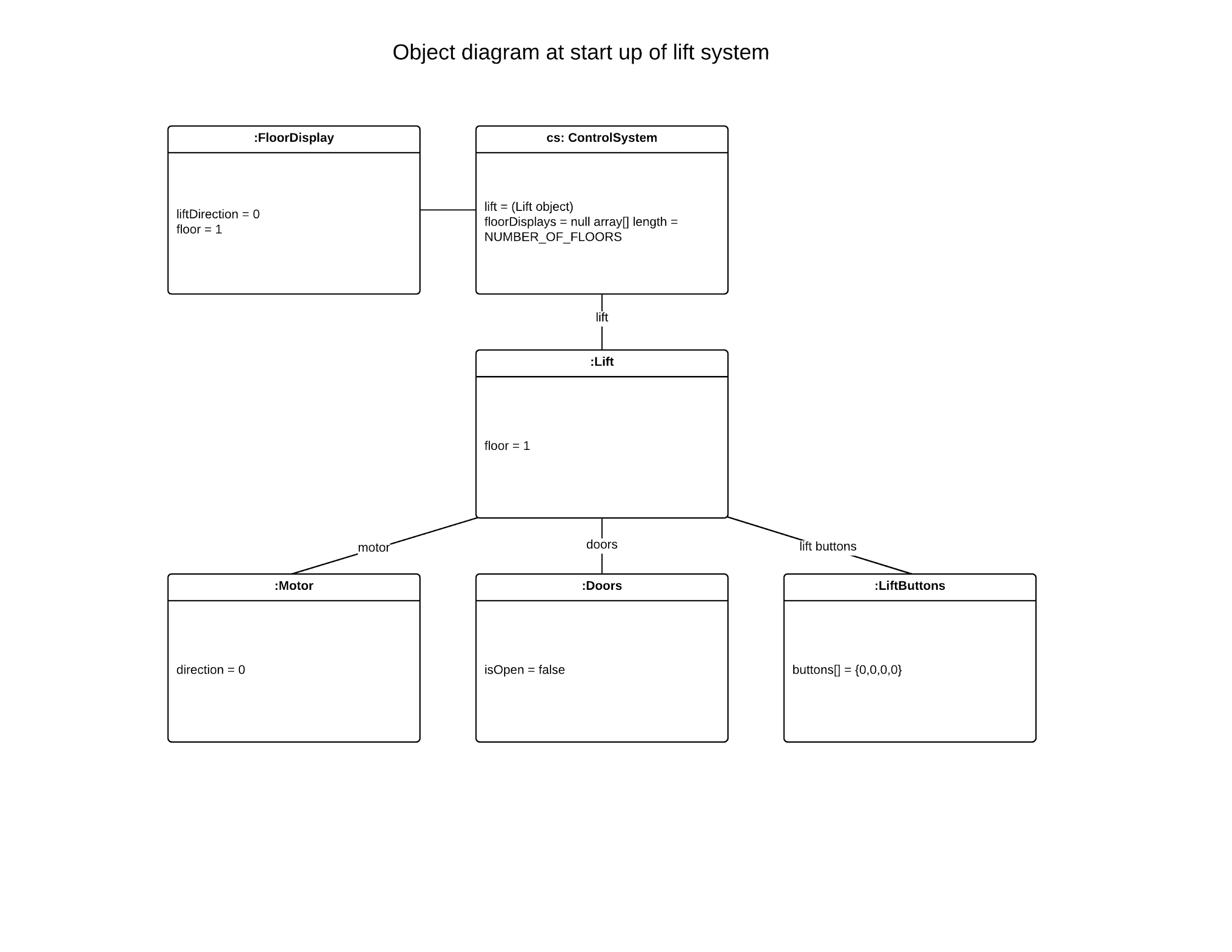
### 

### Class diagram



### 

### Object Diagrams



### 

### 

### 

### 

### 

### 

### 

### Pseudocode for main program

Set topfloor value as 4

Set bottomfloor value as 1

Create new control system which contains a lift, a lift motor and lift buttons

Create structure representing display for all floors with up/down info and number of floor lift is on

Receive input of up/down button on one of four floors calling the lift

Send input to the control system

IF The lift is below the called floor

Close doors

Update floor displays to show lift going up, with current lift floor

Engage motor and set it to move the lift up

ELSE IF The lift is above the called floor

Close doors

Update floor displays to show lift going up, with current lift floor

Engage motor and set it to move the lift down

ELSE Set floor displays to show lift is waiting at lift floor

WHILE The lift floor is not equal to the called floor

IF The lift is below the called floor

Update floor panels with lift direction up and current floor

Move the lift up

ELSE IF The lift is below the called floor

Update floor panels with lift direction down and current floor

Move the lift down

END WHILE

Turn motor off

Update lift floor displays to show lift as waiting at current lift floor

IF Motor is off then open doors

ELSE Throw an exception, we cannot open doors if motor is not off

Receive button input of internal lift panel selecting destination floor

Send input to control system

IF Destination floor button selected is higher than the top floor value OR less than it

Report that an invalid choice has been made

Go back to receiving button input of internal lift panel

Illuminate the destination floor button

IF The lift is below the called destination floor

Update floor panels with lift direction up and current floor

Move the lift up

ELSE IF The lift is below the called floor

Update floor panels with lift direction down and current floor

Move the lift down

ELSE Don’t move the lift, show it as waiting on display

WHILE The lift floor is not equal to the destination floor

IF The lift is below the destination floor

Update floor panels with lift direction up and current floor

Move the lift up

ELSE IF The lift is below the destination floor

Update floor panels with lift direction down and current floor

Move the lift down

END WHILE

Turn motor off

Update lift floor displays to show lift as waiting at current lift floor

Turn off destination floor light inside lift

IF Motor is off then open doors

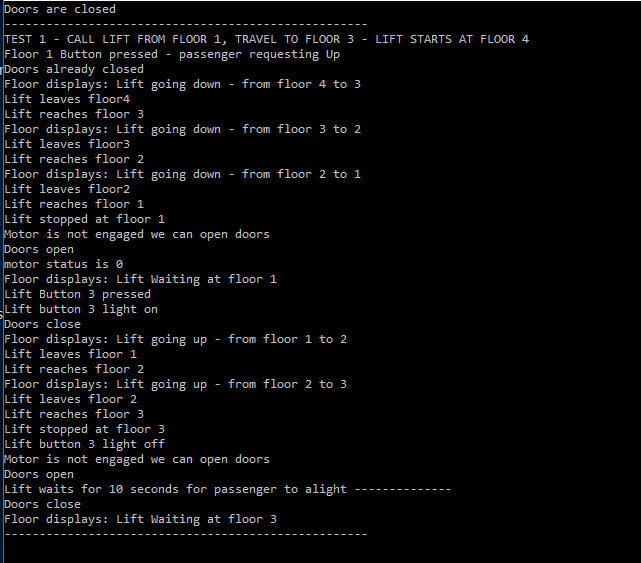
ELSE Throw an exception, we cannot open doors if motor is not off

## 

## Testing

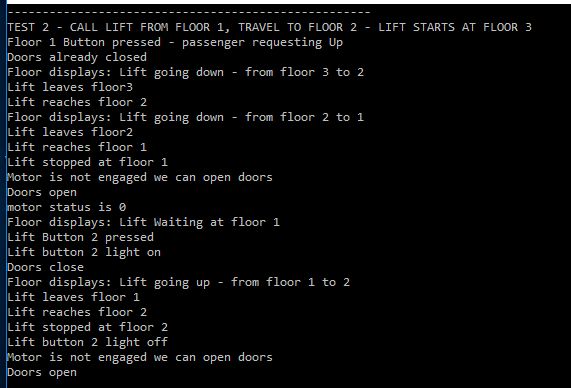
Test 1 - Test lift can go down to request and up after destination selection with normal boundary

Call lift from 1st floor while it is at the top floor, then travel to floor 3



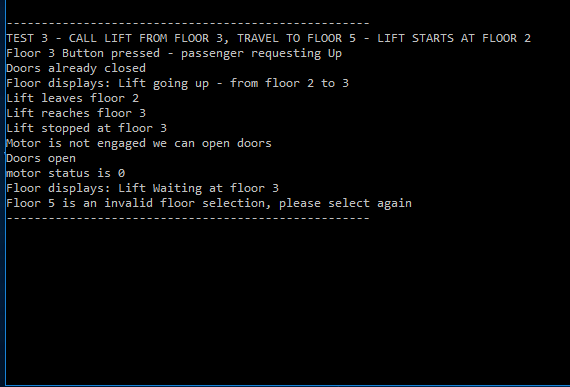
Test 2 - Test lift can travel down to request and travel up to destination floor within normal boundary

Call lift from floor 1 while it is at floor 3, travel to floor 2



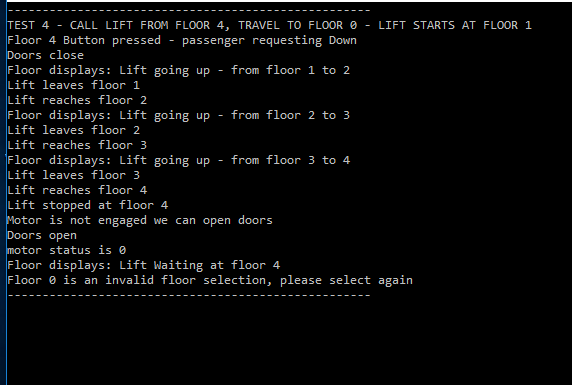
Test 3 - Test lift can go up to a floor request and travel up again to destination of top floor

Call lift from floor 3 while it is at floor 2, travel to floor 5

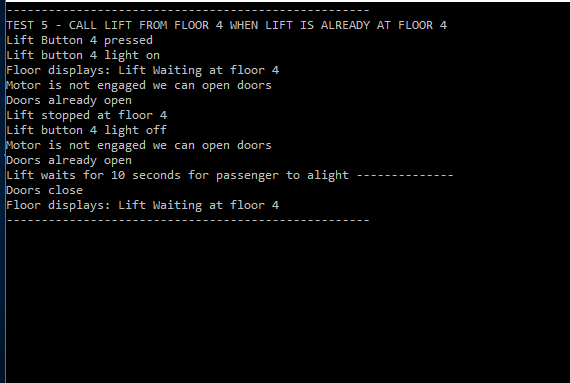


Test 4 - Test lift can travel up to a floor request and handle an invalid floor selection by reporting invalid destination selection

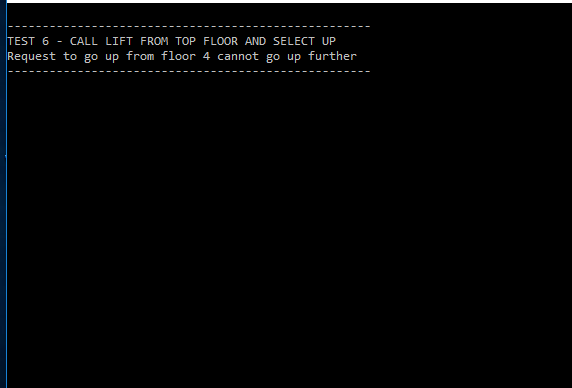
Call lift from floor 4 while it is at floor 1, travel to floor 0 (which is an invalid floor)



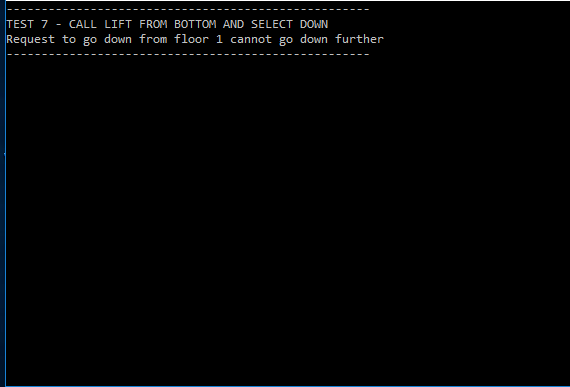
Test 5 - Test lift will not move when called from the floor it is already on



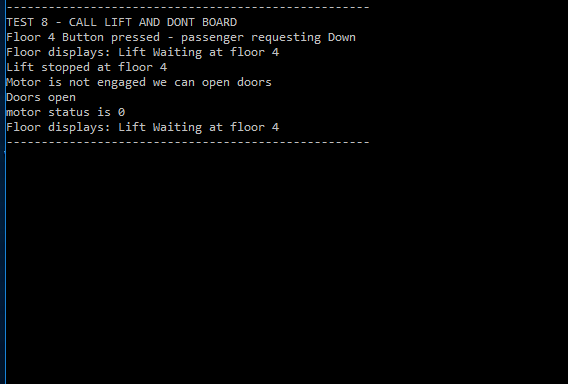
Test 6 - Test lift will not go above top floor even if UP request is received from that floor, test that an error is displayed



Test 7 - Test lift will not go below bottom floor even if DOWN request is received from the floor, test than an error is displayed



Test 8 - Call the lift and do not select a floor destination, lift should wait



## Conclusions

Significant improvements could be made to the solution, moving the logic that moves the lift up/down should perhaps be moved to the lift class, this would help with enabling the system to manage multiple lifts in the building by allowing different lift objects to control their own movement, the building may have multiple lifts with different numbers of floors for example.

A class could be used to represent the floors themselves, storing the floor number and the existence of an UP or DOWN button, or both. The floor objects could be passed to the lift as it moved, this could send the lift the current floor position and the status of the lift being between floors.

A queue structure (perhaps using the queue container adapter in C++ for a FIFO approach or similar) could be implemented that stored multiple floor requests, a structure could also be used to handle multiple destination floor buttons being pressed from within the lift.

Defining the number of floors and the bottom floor variables outside of classes could be improved by making them static const variables for the control system or the lift class, this would need to be done if the system were to handle multiple lifts in a building with different bottom and top floor values.

An abstract button base class could be created with two child classes derived from it such as FloorButton and LiftButton.