

# OOAD Project Presentation

## SPEED CHANGE & SKIP ELEVATOR

200714170 모진종

200714171 박현준

200714173 심상문

## Team Practice #2 – 2030 amendment part

- Controller addition to Sequence Diagram
- There is Controller on inside of elevator
- Controller manages elevator on the whole

# Activity 2041. Design Real Use Cases

Name	1. Elevator call	
Actor	User	
Description	<ul style="list-style-type: none"> <li>- This Use Case can call located elevator of another floors to floor which user is.</li> <li>- If presses the button, light to the button.</li> <li>- Comes at the layer where the elevator has become calling.</li> </ul>	
Type	Primary	
Cross-Reference	R1	
Use Case	Elevator call	
Precondition	There must be to a another floor the elevator.	
Typical Course of Event	Actor	System
	1. presses the button.	2. Light to the button. 3. Show transport process that denote LED's number of layers elevator moves.
Alternative Course		
Exception Course	Line 1. If other users pressed first button trouble need be. Line 2. May not press if there is already light to button. Line 3. Stop on course which is called if was called during transfer.	

# Activity 2041. Design Real Use Cases

Name	2. Open door	
Actor	User	
Description	<ul style="list-style-type: none"> <li>- Presses button when rides elevator and opens the door.</li> <li>- After elevator arrives, the door opens automatically.</li> <li>- Inside the elevator opens the door there is a possibility with the button.</li> </ul>	
Type	Primary	
Cross-Reference	R2.1	
Use Case	Open door, Close door	
Precondition	The elevator door should be closed.	
Typical Course of Event	[1] That is position of elevator and user's position operation in case of do	
	Actor	System
	1. presses the button.	2. Light to the button. 3. The door of elevator opens.
	[2] In case of position of elevator and user's position differ	
	Actor	System
	1. presses the button.	2. Light to the button. 3. Elevator comes to user's position. 4. The door of elevator opens.
	[3] In case of there is user inside elevator	
	Actor	System
	1. presses the button.	2. The door of elevator opens.
Alternative Course		
Exception Course	<p>[1,2,3] Line 1. If other users pressed first button trouble need be.          [1,2] Line 2. May not press if there is already light to button.</p>	



# Activity 2041. Design Real Use Cases

<b>Name</b>	3. Close door	
<b>Actor</b>	User	
<b>Description</b>	<ul style="list-style-type: none"> <li>- This Use Case is user boards and the door is closed automatically.</li> <li>- Presses button and closes the door.</li> </ul>	
<b>Type</b>	Primary	
<b>Cross-Reference</b>	R2.2	
<b>Use Case</b>	Open door, Close door	
<b>Precondition</b>	The elevator door should be opened.	
<b>Typical Course of Event</b>	<b>[1] Board on elevator and in case of given time passed</b>	
	<b>Actor</b>	System
	<b>2. Board on elevator.</b>	<ul style="list-style-type: none"> <li>1. The elevator door is being opened.</li> <li>3. If given time passes, the elevator door is closed.</li> </ul>
	<b>[2] Close the elevator door compulsorily</b>	
	<b>2. Board on elevator.</b> <b>3. presses the button.</b>	<ul style="list-style-type: none"> <li>1. The elevator door is being opened.</li> <li>4. The elevator door is closed.</li> </ul>
<b>Alternative Course</b>		
<b>Exception Course</b>	<ul style="list-style-type: none"> <li>[1] Line 3. If weight is exceeded, the door is not closed.</li> <li>[1] Line 3. If is pressing that is pressing button or opens inside button, the door is not closed.</li> <li>[2] Line 4. If weight is exceeded, the door is not closed.</li> </ul>	

# Activity 2041. Design Real Use Cases

<b>Name</b>	4. Movement by wanting layer	
<b>Actor</b>	User	
<b>Description</b>	- The door is closed if boards on elevator and presses button of wanted course and moves.	
<b>Type</b>	Primary	
<b>Cross-Reference</b>	R3	
<b>Use Case</b>		
<b>Precondition</b>	User should boards on elevator.	
<b>Typical Course of Event</b>	<b>Actor</b>	<b>System</b>
	1. Board on elevator. 2. Press button of wanting floor.	3. Light enters to button. 4. The elevator door is closed. 5. Movement by wanting layer
<b>Alternative Course</b>		
<b>Exception Course</b>	Line 3. Light does not enter to button in case of it is floor which wanted course is in current position. Line 4. If weight is exceeded, the door is not closed. Line 5. Elevator stops in course which is called in case of was called while move.	

# Activity 2041. Design Real Use Cases

<b>Name</b>	5. Location mark	
<b>Actor</b>	None	
<b>Description</b>	<ul style="list-style-type: none"> <li>- This Use Case marks current position of elevator.</li> <li>- Mark floor of moving elevator.</li> </ul>	
<b>Type</b>	Primary	
<b>Cross-Reference</b>	R4	
<b>Use Case</b>	Location mark	
<b>Precondition</b>	Should agree correctly with floor which position of elevator is marking.	
<b>Typical Course of Event</b>	<b>Actor</b>	<b>System</b>
		<ol style="list-style-type: none"> <li>1. Current position of elevator that is stopping indication</li> <li>2. Moving present indication by real-time</li> </ol>
<b>Alternative Course</b>		
<b>Exception Course</b>		

# Activity 2041. Design Real Use Cases

Name	6. Full mark	
Actor	None	
Description	-These Use Case when the total weight of the passengers gets near to the weight limit the elevator will be able to move, mark full.	
Type	Primary	
Cross-Reference	R5	
Use Case	Full mark	
Precondition	Total weight of passengers should be calculated correctly.	
Typical Course of Event	Actor	System
		1. Full mark to user
Alternative Course		
Exception Course	Line 1. Do not mark if do not cross weight limit.	



# Activity 2041. Design Real Use Cases

<b>Name</b>	7. User urgency call	
<b>Actor</b>	User	
<b>Description</b>	<p>- This Use Case requests help using emergency currency if crisis situation happens in elevator.</p> <p>- Presses the emergency currency button from the inside and is connected with the administrator.</p>	
<b>Type</b>	Primary	
<b>Cross-Reference</b>	R6.1	
<b>Use Case</b>	User urgency call, Breakdown urgency call	
<b>Precondition</b>	It must be crisis situation.	
<b>Typical Course of Event</b>	<b>Actor (User)</b>	<b>System</b>
	1. Press emergency currency button. 3. The administrator and talked over the phone.	2. Connect to administrator.
	<b>Actor (관리자)</b>	<b>System</b>
	2. The user and talked over the phone. 3. Solve crisis situation.	1. Emergency currency bell rings.
<b>Alternative Course</b>		
<b>Exception Course</b>		

# Activity 2041. Design Real Use Cases

<b>Name</b>	8. Breakdown urgency call	
<b>Actor</b>	None	
<b>Description</b>	- This Use Case is linked automatically to administrator if elevator broke downs.	
<b>Type</b>	Primary	
<b>Cross-Reference</b>	R6.2	
<b>Use Case</b>	User urgency call, Breakdown urgency call	
<b>Precondition</b>	It must be internal breakdown of elevator.	
<b>Typical Course of Event</b>	<b>Actor</b>	<b>System</b>
		<ol style="list-style-type: none"> <li>1. Elevator breakdown</li> <li>2. Connection to administrator</li> </ol>
<b>Alternative Course</b>		
<b>Exception Course</b>		

# Activity 2041. Design Real Use Cases

Name	9. No person, speed increases.	
Actor	None	
Description	- If a person moves by state that is not on elevator, operating speed is increased.	
Type	Secondary	
Cross-Reference	R7.1	
Use Case	No person, speed increases, Board, speed return	
Precondition	<ul style="list-style-type: none"> <li>- There must not be user on elevator.</li> <li>- The operation speed which increases sets the administrator.</li> </ul>	
Typical Course of Event	Actor	System
		<ul style="list-style-type: none"> <li>1.Sensor confirms that elevator internal user is absent.</li> <li>2.Moves with the speed which increases.</li> </ul>
Alternative Course		
Exception Course	Line 2. User is run at safety operating speed if there is user.	

# Activity 2041. Design Real Use Cases

<b>Name</b>	10. Board, speed return	
<b>Actor</b>	None	
<b>Description</b>	- When the elevator does an unmanned movement and the user boards, returns with a safe operation speed	
<b>Type</b>	Secondary	
<b>Cross-Reference</b>	R7.2	
<b>Use Case</b>	No person, speed increases, Board, speed return	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>- Must be user on elevator.</li> <li>- administrator decides safety operating speed.</li> </ul>	
<b>Typical Course of Event</b>	<b>Actor</b>	<b>System</b>
		<ul style="list-style-type: none"> <li>1.Sensor confirms that user boards on elevator.</li> <li>2. Moves with a safety operation speed.</li> </ul>
<b>Alternative Course</b>		
<b>Exception Course</b>	Line 2. User is run at augmented rate if there is no user.	



# Activity 2041. Design Real Use Cases

<b>Name</b>	11. Operation speed value set	
<b>Actor</b>	administrator	
<b>Description</b>	- This Use Case sets high speed operating speed and safety operating speed and stores administrator.	
<b>Type</b>	Secondary	
<b>Cross-Reference</b>	R8	
<b>Use Case</b>	Operation speed value set	
<b>Precondition</b>	It must be state that suspend running of elevator.	
<b>Typical Course of Event</b>	<b>Actor</b>	<b>System</b>
	<ol style="list-style-type: none"> <li>1. Enters in a administrator mode.</li> <li>3. Set safety operating speed and high speed operating speed.</li> </ol>	<ol style="list-style-type: none"> <li>2. There is no passenger, and elevator must be standby.</li> <li>4. Stores input contents and sends end message.</li> </ol>
<b>Alternative Course</b>		
<b>Exception Course</b>	Line 1. If there is passenger or elevator is not standby, input is refused.	

# Activity 2041. Design Real Use Cases

Name	12. Weight limits set.	
Actor	administrator	
Description	- administrator set weight limits to this Use Case.	
Type	Secondary	
Cross-Reference	R9	
Use Case	Weight limits set.	
Precondition	There is no passenger and must be a in the standby.	
Typical Course of Event	Actor	System
	<ol style="list-style-type: none"> <li>1. Enters in a administrator mode.</li> <li>3. Weight limits set.</li> <li>4. Stores an input contents in system.</li> </ol>	<ol style="list-style-type: none"> <li>2. There is no passenger, and elevator must be standby.</li> </ol>
Alternative Course		
Exception Course	Line 1. When there is a user or the elevator is in the process of using, will not be able to input.	

# Activity 2041. Design Real Use Cases

<b>Name</b>	13. boarding wrong message	
<b>Actor</b>	None	
<b>Description</b>	For this Use Case, sensor compares weight and I inform waiting boarding possibility availability.	
<b>Type</b>	Secondary	
<b>Cross-Reference</b>	R10	
<b>Use Case</b>	boarding wrong message	
<b>Precondition</b>	Waiting weight and elevator inside's user's weight should be compared by real-time.	
<b>Typical Course of Event</b>	<b>Actor</b>	<b>System</b>
	1. Waiting calls elevator.	2. Sensor measures waiting weight. 3. The elevator compares accommodating possible weight the inside. 4. Inform boarding prohibition message.
<b>Alternative Course</b>		
<b>Exception Course</b>	Line 4. If the weight which perceives from the atmosphere layer sensor is fewer accommodating possible weight like the call signaling which is general is controlled.	

# Activity 2041. Design Real Use Cases

Name	14. Compare whether or not boarding possibility and skip	
Actor	None	
Description	-This Use Case does course which is called because elevator compares weight with user and waiting skip.	
Type	Secondary	
Cross-Reference	R11.1	
Use Case	<ul style="list-style-type: none"> <li>- Compare whether or not boarding possibility and skip</li> <li>- Compare whether or not boarding possibility and stop</li> </ul>	
Precondition	Waiting weight and elevator inside's user's weight should be compared by real-time.	
Typical Course of Event	Actor	System
	<ol style="list-style-type: none"> <li>1. Call elevator.</li> </ol>	<ol style="list-style-type: none"> <li>2. Sensor measures waiting weight.</li> <li>3. The elevator compares accommodating possible weight the inside.</li> <li>4. If approximate in limit weight, called course does skip.</li> </ol>
Alternative Course		
Exception Course		



# Activity 2041. Design Real Use Cases

Name	15. Compare whether or not boarding possibility and stop	
Actor		
Description	-This Use Case does course which is called because elevator compares weight with user and waiting stop.	
Type	Secondary	
Cross-Reference	R11.2	
Use Case	<ul style="list-style-type: none"> <li>- Compare whether or not boarding possibility and skip</li> <li>- Compare whether or not boarding possibility and stop</li> </ul>	
Precondition	Waiting weight and elevator inside's user's weight should be compared by real-time.	
Typical Course of Event	Actor	System
	<ol style="list-style-type: none"> <li>1. Call elevator.</li> </ol>	<ol style="list-style-type: none"> <li>2. Sensor measures waiting weight.</li> <li>3. The elevator compares accommodating possible weight the inside.</li> <li>4. Suspend on course which is called if do not approximate in limit weight.</li> </ol>
Alternative Course		
Exception Course		

# Activity 2042. Define Reports, UI, and Storyboard

1. Administer Mode
  - (1) Weight Limits Set

Administrator Mode

---

1	2	3
4	5	6
7	8	9
M/s	0	<b>1</b> Kg
Speed	save	Weight Limit



Administrator Mode

---

<b>2</b> 1	2	3
4	5	6
7	8	9
M/s	<b>3</b> 0	Kg
Speed	save	Weight Limit

**4**

# Activity 2042. Define Reports, UI, and Storyboard

- (2) Operation Speed Value Set
- ① Safe Operation Speed

Administrator Mode

1	2	3
4	5	6
7	8	9
① M/s	0	Kg
Speed	save	Weight Limit



Administrator Mode

1	② 2	3
4	5	6
7	8	9
M/s	0	Kg
Speed	③ save	Weight Limit

# Activity 2042. Define Reports, UI, and Storyboard

## ② No Person Speed

Administrator Mode

---

1	2	3
4	5	6
7	8	9
① M/s	0	Kg
Speed	save	Weight Limit



Administrator Mode

---

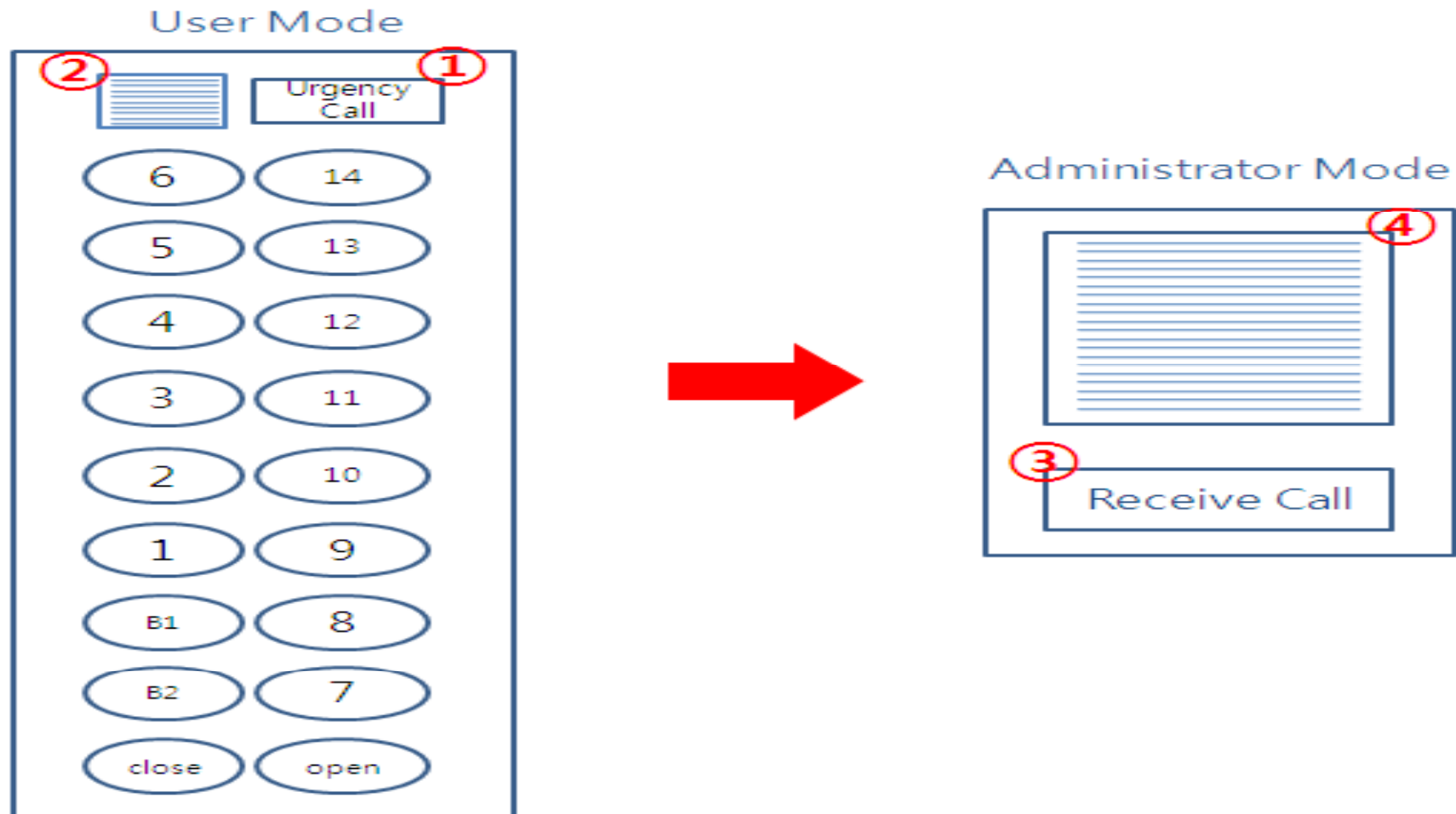
1	2	3
② 4	5	6
7	8	9
M/s	0	Kg
Speed	③ save	Weight Limit



# Activity 2042. Define Reports, UI, and Storyboard

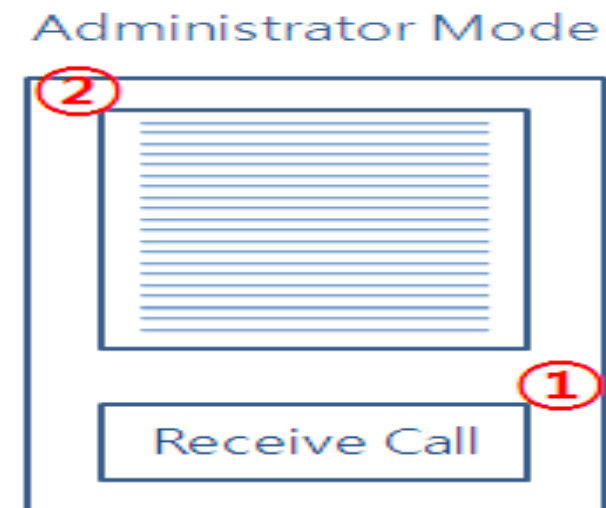
## 2. User Mode, Administrator Mode

### (1) Urgency Call (User → Administrator)

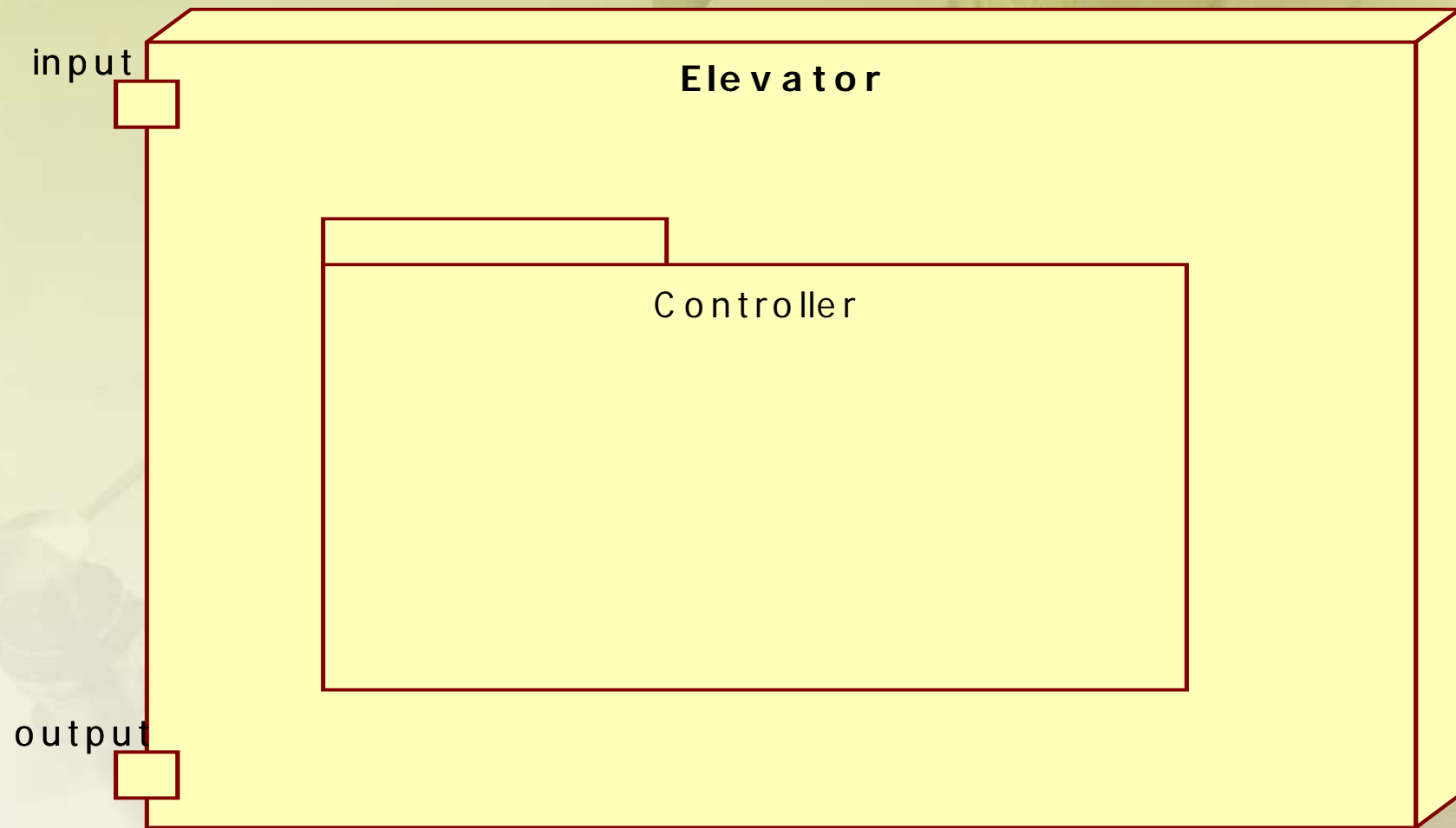


# Activity 2042. Define Reports, UI, and Storyboard

(2) Urgency Call (Administrator → User)

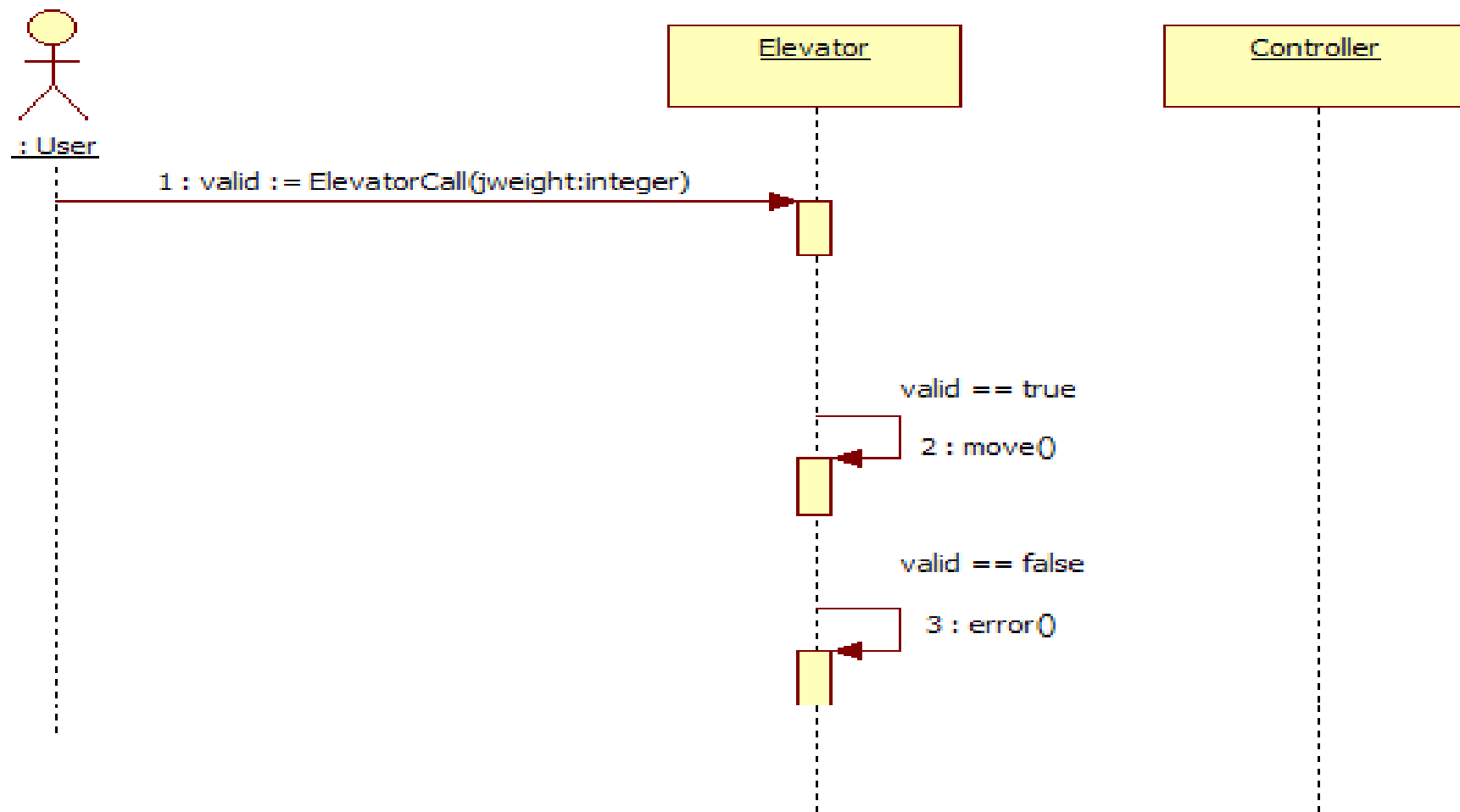


# Activity 2043. Refine System Architecture



# Activity 2044. Define Interaction Diagrams

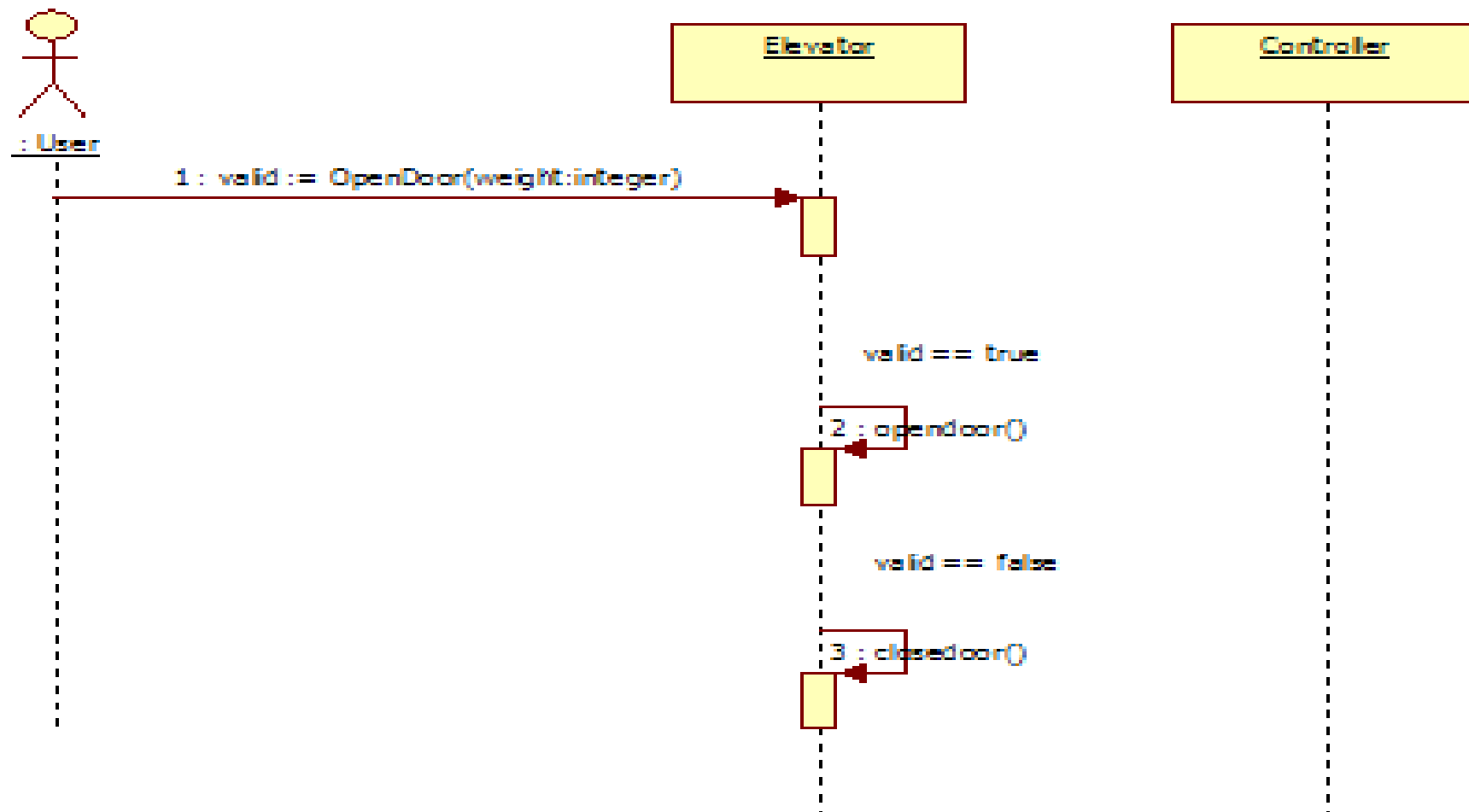
## 1. ElevatorCall()





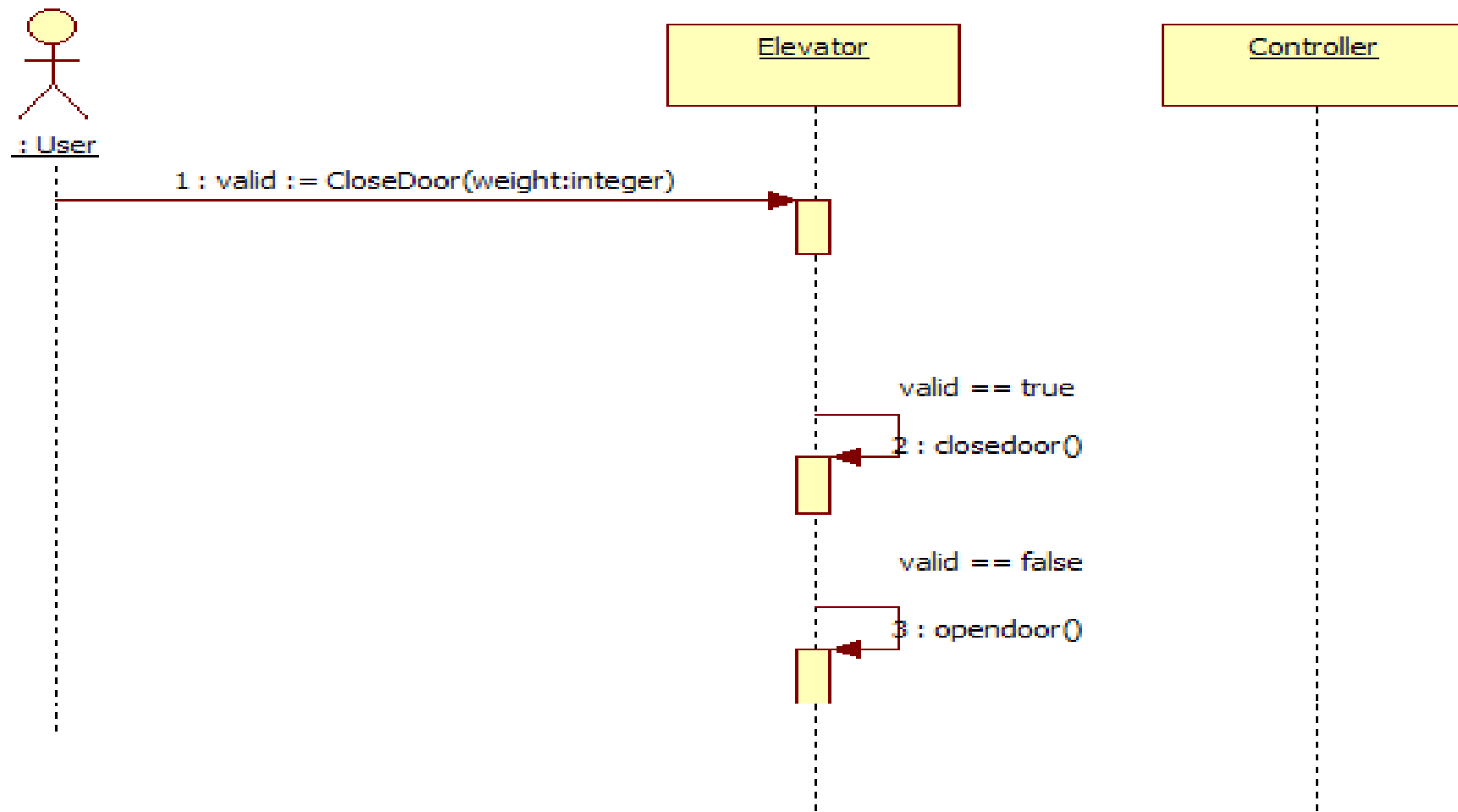
# Activity 2044. Define Interaction Diagrams

## 2. OpenDoor()



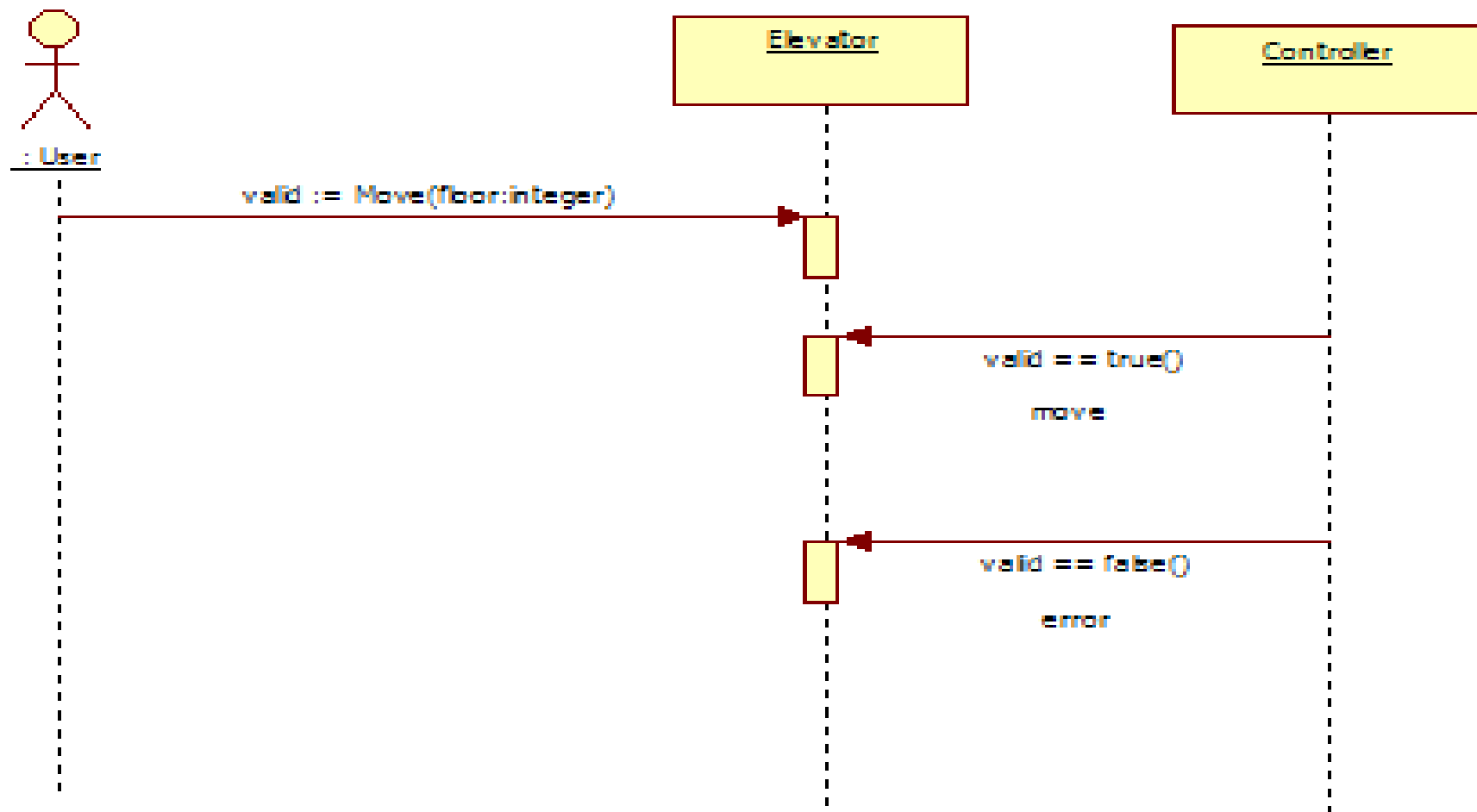
# Activity 2044. Define Interaction Diagrams

## 3. CloseDoor()



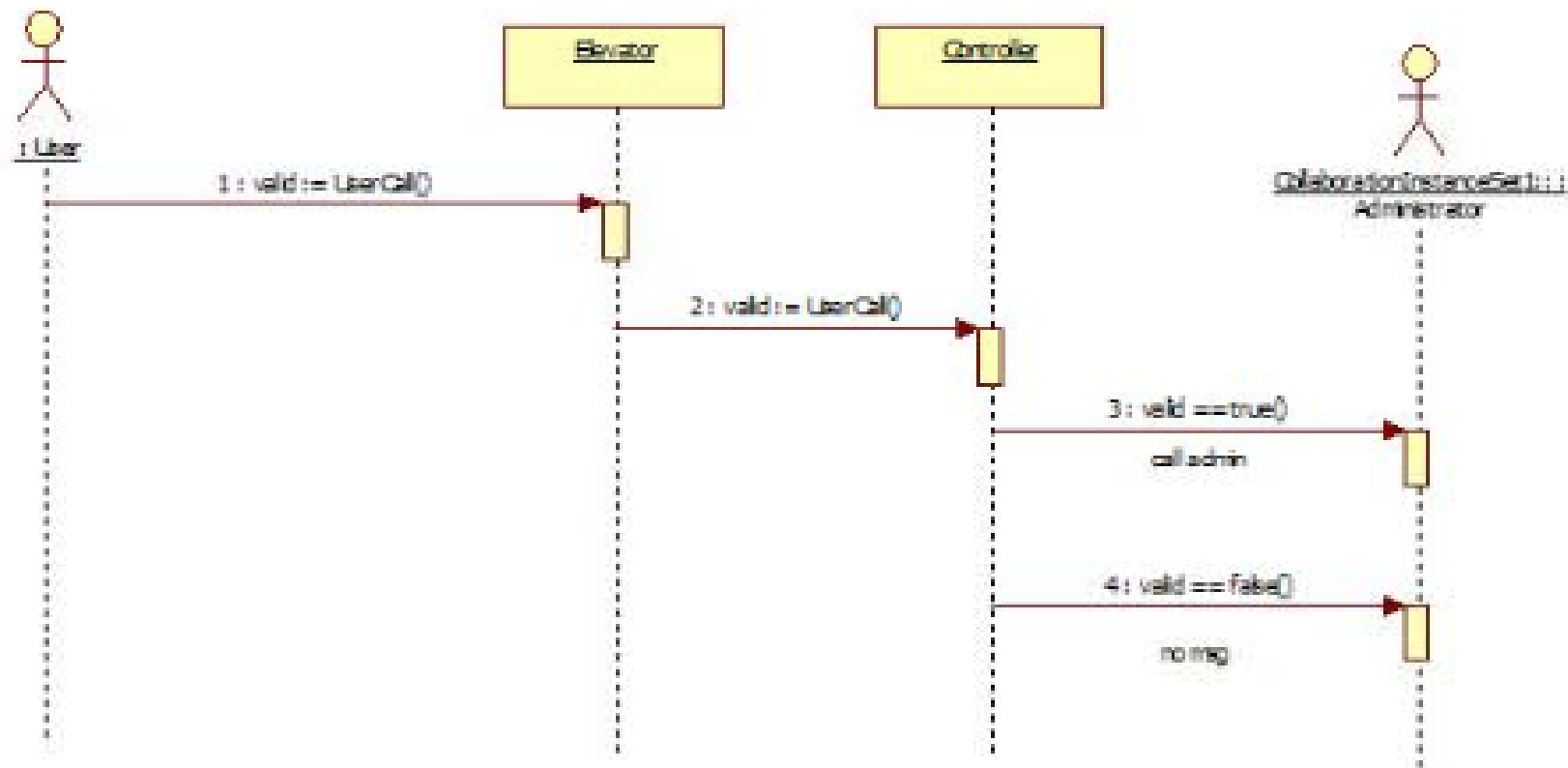
# Activity 2044. Define Interaction Diagrams

## 4. Move()



# Activity 2044. Define Interaction Diagrams

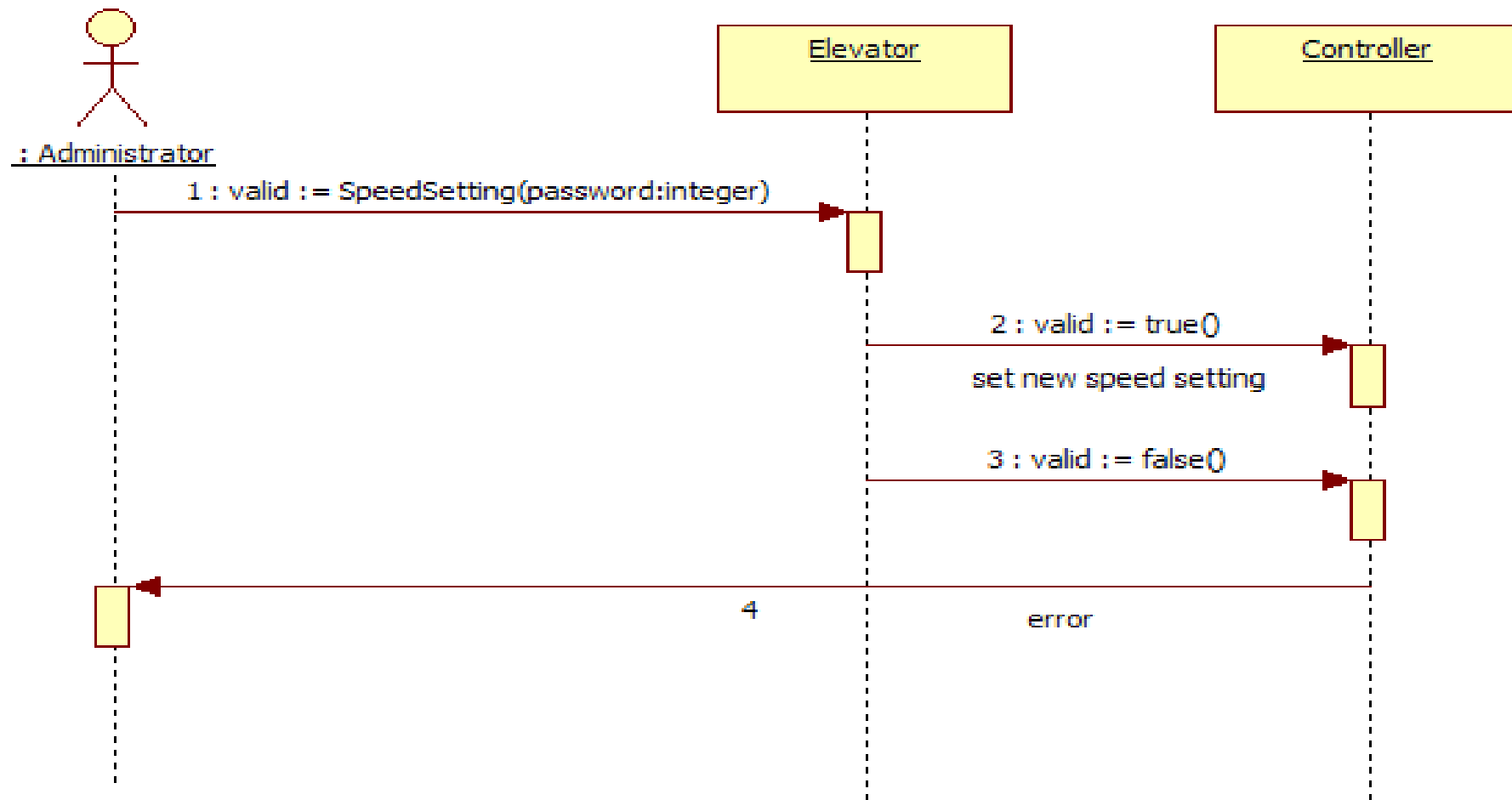
## 5. UserCall()





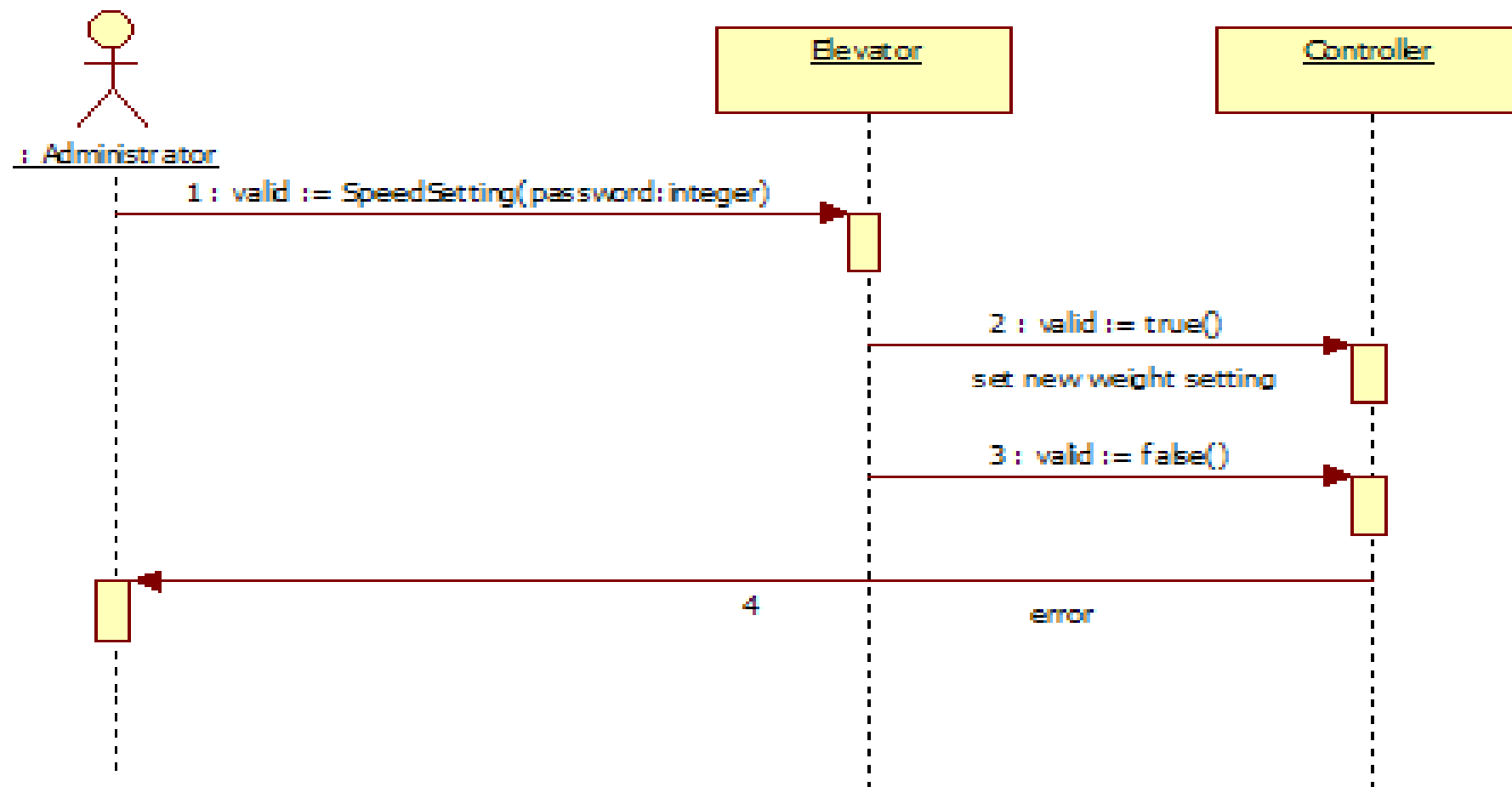
# Activity 2044. Define Interaction Diagrams

## 6. SpeedSetting()



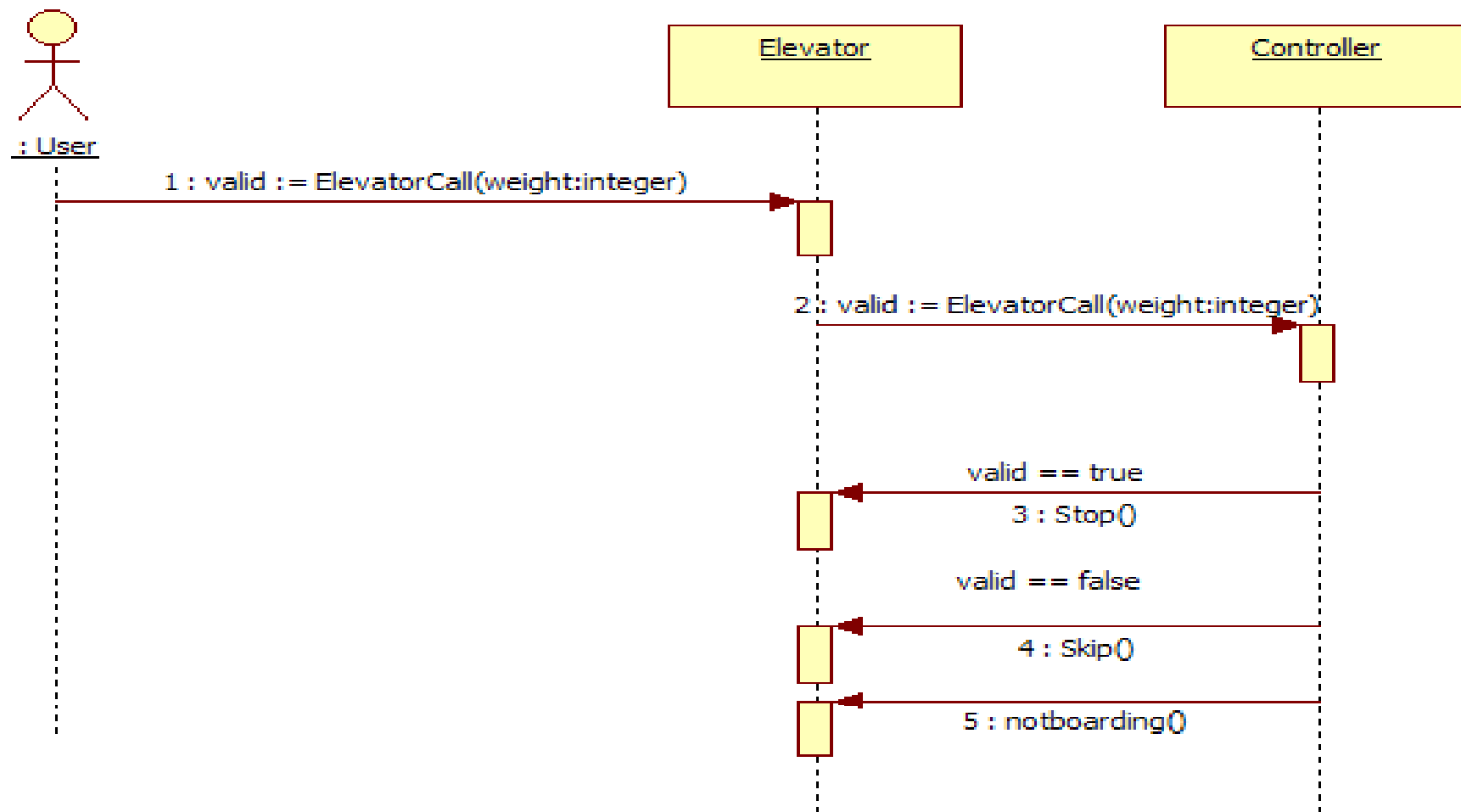
# Activity 2044. Define Interaction Diagrams

## 7. WeightLimitSet()



# Activity 2044. Define Interaction Diagrams

## 8. NotBoarding()



# Activity 2045. Define Design Class Diagram

