Group 13

Author: 任怡静

Validation

Elevator

Table of Contents

[System Architecture 2](#_Toc44624683)

[T1: Unit Test 2](#_Toc44624684)

[T1.1: FloorSensor Unit Test 2](#_Toc44624685)

[T1.2: Elevator Unit Test 3](#_Toc44624686)

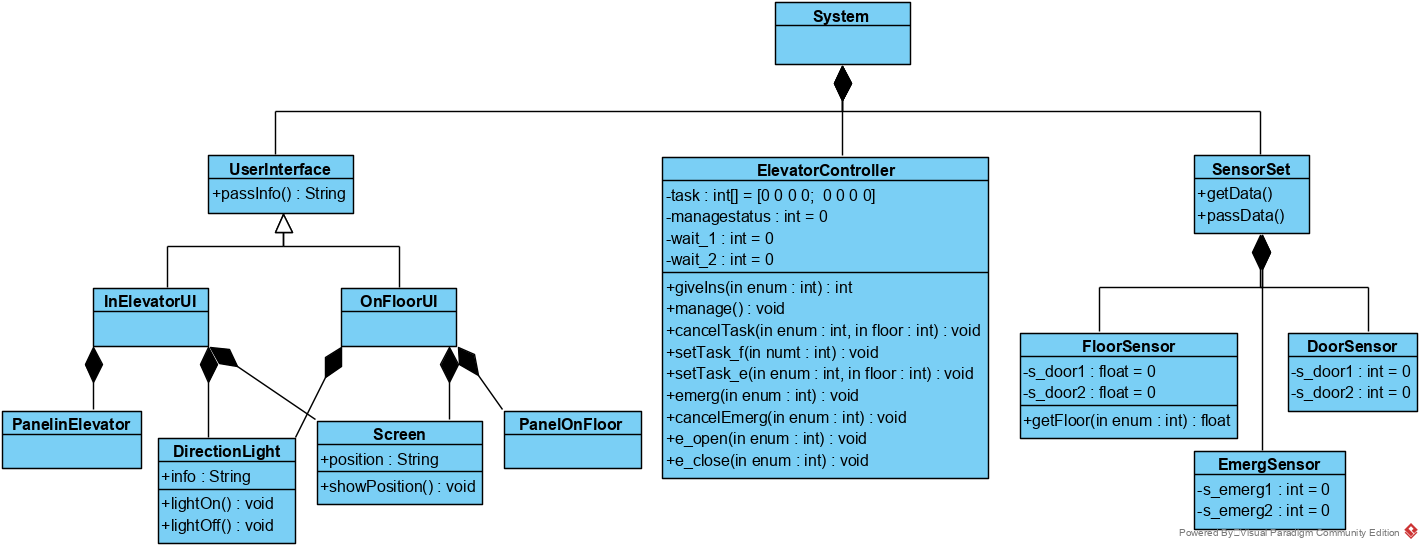
[T1.3: ElevatorController Unit Test 5](#_Toc44624687)

[T2: Integration Test 24](#_Toc44624688)

[T3: Functional Test 25](#_Toc44624689)

## System Architecture

The system architecture is shown below:



## T1: Unit Test

### T1.1: FloorSensor Unit Test

T1.1.1: Test getFloor()

function ret=getFloor(fSensor,enum)

switch enum

case 1

ret=fSensor.f\_e1; Tcover1.1.1.1

case 2

ret=fSensor.f\_e2; Tcover1.1.1.2

end

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.1.1 |
| Coverage Item | Tcover1.1.1.1 |
| Input | enum= 1 |
| State | fSensor = FloorSensor; |
| Expected Output | fSensor.f\_e1 == 1 |
|  | Test Case T1.1.1.2 |
| Coverage Item | Tcover1.1.1.2 |
| Input | enum= 2 |
| State | fSensor = FloorSensor; |
| Expected Output | fSensor.f\_e1 == 1 |

* Test coverage: 2/2=100%
* Test result: 1 passed

### T1.2: Elevator Unit Test

T1.2.1: Test updateFloor()

function updateFloor(elevator,enum)

switch enum

case 1

fl1=elevator.fSensor.f\_e1; T1.2.1.1

elevator.e1.ElevatorGauge.Value=fl1; T1.2.1.1

elevator.f1.Elevator1Gauge.Value=fl1; T1.2.1.1

elevator.f2.Elevator1Gauge.Value=fl1; T1.2.1.1

elevator.f3.Elevator1Gauge.Value=fl1; T1.2.1.1

case 2

fl2=elevator.fSensor.f\_e2; T1.2.1.2

elevator.e2.ElevatorGauge.Value=fl2; T1.2.1.2

elevator.f1.Elevator2Gauge.Value=fl2; T1.2.1.2

elevator.f2.Elevator2Gauge.Value=fl2; T1.2.1.2

elevator.f3.Elevator2Gauge.Value=fl2; T1.2.1.2

end

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.1.1 |
| Coverage Item | Tcover1.2.1.1 |
| Input | enum= 1 |
| State | elevator = Elevator;  elevator.fSensor = FloorSensor;  elevator.f1 = ButtonPanel\_f1;  elevator.f2 = ButtonPanel\_f2;  elevator.f3 = ButtonPanel\_f3;  elevator.e1 = ButtonPanel\_e;  elevator.e2 = ButtonPanel\_e; |
| Expected Output | elevator.e1.ElevatorGauge.Value == fl1  elevator.f1.Elevator1Gauge.Value == fl1  elevator.f2.Elevator1Gauge.Value == fl1  elevator.f3.Elevator1Gauge.Value == fl1 |
|  | Test Case T1.2.1.2 |
| Coverage Item | Tcover1.2.1.2 |
| Input | enum= 2 |
| State | elevator = Elevator;  elevator.fSensor = FloorSensor;  elevator.f1 = ButtonPanel\_f1;  elevator.f2 = ButtonPanel\_f2;  elevator.f3 = ButtonPanel\_f3;  elevator.e1 = ButtonPanel\_e;  elevator.e2 = ButtonPanel\_e; |
| Expected Output | elevator.e1.ElevatorGauge.Value == fl2  elevator.f1.Elevator1Gauge.Value == fl2  elevator.f2.Elevator1Gauge.Value == fl2  elevator.f3.Elevator1Gauge.Value == fl2 |

T1.2.2: Test move()

function move(elevator,enum,inst)

switch enum

case 1

switch inst

case 1

elevator.fSensor.f\_e1=elevator.fSensor.f\_e1+0.1; T1.2.2.1

case 2

elevator.fSensor.f\_e1=elevator.fSensor.f\_e1-0.1; T1.2.2.2

end

case 2

switch inst

case 1

elevator.fSensor.f\_e2=elevator.fSensor.f\_e2+0.1; T1.2.2.3

case 2

elevator.fSensor.f\_e2=elevator.fSensor.f\_e2-0.1; T1.2.2.4

end

end

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.2.1 |
| Coverage Item | Tcover1.2.2.1 |
| Input | enum= 1  inst = 1 |
| State | elevator = Elevator;  elevator.fSensor = FloorSensor;  elevator.f1 = ButtonPanel\_f1;  elevator.f2 = ButtonPanel\_f2;  elevator.f3 = ButtonPanel\_f3;  elevator.e1 = ButtonPanel\_e;  elevator.e2 = ButtonPanel\_e; |
| Expected Output | elevator.fSensor.f\_e1 = elevator.fSensor.f\_e1+0.1 |
|  | Test Case T1.2.2.2 |
| Coverage Item | Tcover1.2.2.2 |
| Input | enum= 1  inst = 2 |
| State | elevator = Elevator;  elevator.fSensor = FloorSensor;  elevator.f1 = ButtonPanel\_f1;  elevator.f2 = ButtonPanel\_f2;  elevator.f3 = ButtonPanel\_f3;  elevator.e1 = ButtonPanel\_e;  elevator.e2 = ButtonPanel\_e; |
| Expected Output | elevator.fSensor.f\_e1 = elevator.fSensor.f\_e1-0.1 |
|  | Test Case T1.2.2.3 |
| Coverage Item | Tcover1.2.2.3 |
| Input | enum= 2  inst = 1 |
| State | elevator = Elevator;  elevator.fSensor = FloorSensor;  elevator.f1 = ButtonPanel\_f1;  elevator.f2 = ButtonPanel\_f2;  elevator.f3 = ButtonPanel\_f3;  elevator.e1 = ButtonPanel\_e;  elevator.e2 = ButtonPanel\_e; |
| Expected Output | elevator.fSensor.f\_e2 = elevator.fSensor.f\_e2+0.1 |
|  | Test Case T1.2.1.4 |
| Coverage Item | Tcover1.2.1.4 |
| Input | enum= 2  inst = 2 |
| State | elevator = Elevator;  elevator.fSensor = FloorSensor;  elevator.f1 = ButtonPanel\_f1;  elevator.f2 = ButtonPanel\_f2;  elevator.f3 = ButtonPanel\_f3;  elevator.e1 = ButtonPanel\_e;  elevator.e2 = ButtonPanel\_e; |
| Expected Output | elevator.fSensor.f\_e2 = elevator.fSensor.f\_e2-0.1 |

* Test coverage: 6/6=100%
* Test result: 2 passed

### T1.3: ElevatorController Unit Test

T1.3.1: Test giveIns()

function ret=giveIns(control,enum)

switch enum

case 1

if control.fSensor.f\_e1<1.01

if control.task(1,1)==1

ret=0;

else

if control.task(1,2)==1||control.task(1,3)==1||control.task(1,4)==1

ret=1;

else

ret=0;

end

end

control.e1.status=ret;

return;

elseif control.fSensor.f\_e1>1.99 && control.fSensor.f\_e1<2.01

if (control.task(1,2)==1 && (control.e1.status==1||control.e1.status==0))

||(control.task(1,3)==1 && (control.e1.status==2||control.e1.status==0))

ret=0;

else

if control.task(1,1)==1

ret=2;

elseif control.task(1,4)==1

ret=1;

else

ret=0;

end

end

control.e1.status=ret;

return;

elseif control.fSensor.f\_e1>2.99

if control.task(1,4)==1

ret=0;

else

if control.task(1,1)==1||control.task(1,2)==1||control.task(1,3)==1

ret=2;

else

ret=0;

end

end

control.e1.status=ret;

return;

else

ret=control.e1.status;

return;

end

case 2

if control.fSensor.f\_e2<1.01

if control.task(2,1)==1

ret=0;

else

if control.task(2,2)==1||control.task(2,3)==1||control.task(2,4)==1

ret=1;

else

ret=0;

end

end

control.e2.status=ret;

return;

elseif control.fSensor.f\_e2>1.99 && control.fSensor.f\_e2<2.01

if (control.task(2,2)==1 && (control.e2.status==1||control.e2.status==0))

||(control.task(2,3)==1 && (control.e2.status==2||control.e2.status==0))

ret=0;

else

if control.task(2,1)==1

ret=2;

elseif control.task(2,4)==1

ret=1;

else

ret=0;

end

end

control.e2.status=ret;

return;

elseif control.fSensor.f\_e2>2.99

if control.task(2,4)==1

ret=0;

else

if control.task(2,3)==1||control.task(2,2)==1||control.task(2,1)==1

ret=2;

else

ret=0;

end

end

control.e2.status=ret;

return;

else

ret=control.e2.status;

return;

end

end

end

|  |  |
| --- | --- |
|  | Test Case T1.3.1.1 |
| Coverage Item | Tcover 1.3.1.1 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 1;  ele\_controller.task = [1 0 0 0;0 0 0 0]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.2 |
| Coverage Item | Tcover1.3.1.2 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 1;  ele\_controller.task = [0 1 1 1;0 0 0 0]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 1; |
|  | Test Case T1.3.1.3 |
| Coverage Item | Tcover1.3.1.3 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 1;  ele\_controller.task = [0 0 0 0;1 0 0 0]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0 |
|  | Test Case T1.3.1.4 |
| Coverage Item | Tcover1.3.1.4 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 2;  ele\_controller.task = [0 1 0 0;0 0 0 0];  ele\_controller.e1.status = 2; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.5 |
| Coverage Item | Tcover1.3.1.5 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 1;  ele\_controller.task = [0 0 1 0;0 0 0 0];  ele\_controller.e1.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.6 |
| Coverage Item | Tcover1.3.1.6 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 2;  ele\_controller.task =[1 0 0 0;0 0 0 0];  ele\_controller.e1.status = 2; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 2; |
|  | Test Case T1.3.1.7 |
| Coverage Item | Tcover1.3.1.7 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 2;  ele\_controller.task = [0 0 0 1;0 0 0 0];  ele\_controller.e1.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 1; |
|  | Test Case T1.3.1.8 |
| Coverage Item | Tcover1.3.1.8 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 2;  ele\_controller.task = [0 0 0 0;1 0 0 0];  ele\_controller.e1.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.9 |
| Coverage Item | Tcover1.3.1.9 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 3;  ele\_controller.task = [0 0 0 1;0 0 0 0];  ele\_controller.e1.status = 2; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.10 |
| Coverage Item | Tcover1.3.1.10 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 3;  ele\_controller.task = [1 0 0 0;0 0 0 0];  ele\_controller.e1.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 2; |
|  | Test Case T1.3.1.11 |
| Coverage Item | Tcover1.3.1.11 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 3;  ele\_controller.task =[0 1 1 0;0 0 0 0];  ele\_controller.e1.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 2; |
|  | Test Case T1.3.1.12 |
| Coverage Item | Tcover1.3.1.12 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 3;  ele\_controller.task =[0 0 0 0;1 0 0 0];  ele\_controller.e1.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |

|  |  |
| --- | --- |
|  | Test Case T1.3.1.13 |
| Coverage Item | Tcover 1.3.1.13 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 1;  ele\_controller.task = [0 0 0 0;1 0 0 0]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.14 |
| Coverage Item | Tcover1.3.1.14 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 1;  ele\_controller.task = [0 0 0 0;0 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 1; |
|  | Test Case T1.3.1.15 |
| Coverage Item | Tcover1.3.1.15 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 1;  ele\_controller.task = [1 0 0 0;0 0 0 0]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0 |
|  | Test Case T1.3.1.16 |
| Coverage Item | Tcover1.3.1.16 |
| Input | enum=2  ele\_controller.fSensor.f\_e2 = 2;  ele\_controller.task = [0 0 0 0;0 1 0 0];  ele\_controller.e2.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.17 |
| Coverage Item | Tcover1.3.1.17 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 2;  ele\_controller.task = [0 0 0 0;0 0 1 0];  ele\_controller.e1.status = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.18 |
| Coverage Item | Tcover1.3.1.18 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 2;  ele\_controller.task =[0 0 0 0;1 0 0 0];  ele\_controller.e1.status = 2; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 2; |
|  | Test Case T1.3.1.19 |
| Coverage Item | Tcover1.3.1. 19 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 2;  ele\_controller.task = [0 0 0 0;0 0 0 1];  ele\_controller.e1.status = 2; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 1; |
|  | Test Case T1.3.1.20 |
| Coverage Item | Tcover1.3.1.20 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 2;  ele\_controller.task = [1 0 0 0;0 0 0 0];  ele\_controller.e1.status = 2; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.21 |
| Coverage Item | Tcover1.3.1.21 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 3;  ele\_controller.task = [0 0 0 0;0 0 0 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |
|  | Test Case T1.3.1.22 |
| Coverage Item | Tcover1.3.1.22 |
| Input | enum= 2  ele\_controller.fSensor.f\_e2 = 3;  ele\_controller.task = [0 0 0 0;1 1 1 0]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 2; |
|  | Test Case T1.3.1.23 |
| Coverage Item | Tcover1.3.1.23 |
| Input | enum= 1  ele\_controller.fSensor.f\_e1 = 3;  ele\_controller.task =[0 1 1 0;0 0 0 0]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | Ret = 0; |

T1.3.2: Test cancelTask()

function cancelTask(control,enum,floor)

switch enum

case 1

switch floor

case 1

control.task(1,1)=0;

case 2

switch control.e1.status

case 1

control.task(1,2)=0;

case 2

control.task(1,3)=0;

case 0

if control.task(1,3)==1

control.task(1,3)=0;

else

control.task(1,2)=0;

end

end

case 3

control.task(1,4)=0;

end

case 2

switch floor

case 1

control.task(2,1)=0;

case 2

switch control.e2.status

case 1

control.task(2,2)=0;

case 2

control.task(2,3)=0;

case 0

if control.task(2,3)==1

control.task(2,3)=0;

else

control.task(2,2)=0;

end

end

case 3

control.task(2,4)=0;

end

end

end

|  |  |
| --- | --- |
|  | Test Case T1.3.2.1 |
| Coverage Item | Tcover1.3.2.1 |
| Input | enum= 1  floor = 1  ele\_controller.task = [1 1 1 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 1 1 1;1 1 1 1] |
|  | Test Case T1.3.2.2 |
| Coverage Item | Tcover1.3.2.2 |
| Input | enum = 1  floor = 2  ele\_controller.e1.status = 1  ele\_controller.task = [0 1 1 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 0 1 1;1 1 1 1] |
|  | Test Case T1.3.2.3 |
| Coverage Item | Tcover1.3.2.3 |
| Input | enum= 1  floor = 2  ele\_controller.e1.status = 2  ele\_controller.task = [0 0 1 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 0 0 1;1 1 1 1] |
|  | Test Case T1.3.2.4 |
| Coverage Item | Tcover1.3.2.4 |
| Input | enum= 1  floor = 2  ele\_controller.e1.status = 0  ele\_controller.task = [1 1 1 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 1 0 1;1 1 1 1] |
|  | Test Case T1.3.2.5 |
| Coverage Item | Tcover1.3.2.5 |
| Input | enum= 1  floor = 2  ele\_controller.e1.status = 0  ele\_controller.task = [1 1 0 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 0 0 1;1 1 1 1] |
|  | Test Case T1.3.2.6 |
| Coverage Item | Tcover1.3.2.6 |
| Input | enum= 1  floor = 3  ele\_controller.e1.status = 0  ele\_controller.task = [1 0 0 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 0 0 0;1 1 1 1] |
|  | Test Case T1.3.2.7 |
| Coverage Item | Tcover1.3.2.7 |
| Input | enum= 2  floor = 1  ele\_controller.task = [1 1 1 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 1 1 1;0 1 1 1] |
|  | Test Case T1.3.2.8 |
| Coverage Item | Tcover1.3.2.8 |
| Input | enum= 2  floor = 2  ele\_controller.e2.status = 1  ele\_controller.task = [1 1 1 1;0 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 1 1 1;0 0 1 1] |
|  | Test Case T1.3.2.9 |
| Coverage Item | Tcover1.3.2.9 |
| Input | enum= 2  floor = 2  ele\_controller.e2.status = 2  ele\_controller.task = [1 1 1 1;0 0 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 1 1 1;0 0 0 1] |
|  | Test Case T1.3.2.10 |
| Coverage Item | Tcover1.3.2.10 |
| Input | enum= 2  floor = 2  ele\_controller.e2.status = 0  ele\_controller.task = [1 1 1 1;1 1 1 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 1 1 1;1 1 0 1] |
|  | Test Case T1.3.2.11 |
| Coverage Item | Tcover1.3.2.11 |
| Input | enum= 2  floor = 2  ele\_controller.e2.status = 0  ele\_controller.task = [1 1 1 1;1 1 0 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 1 1 1;1 0 0 1] |
|  | Test Case T1.3.2.12 |
| Coverage Item | Tcover1.3.2.12 |
| Input | enum= 2  floor = 3  ele\_controller.e2.status = 0  ele\_controller.task = [1 1 1 1;1 0 0 1]; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 1 1 1;1 0 0 0] |

T1.3.3: Test setTask\_e()

function setTask\_e(control,enum,floor)

if floor==1

control.task(enum,floor)=1;

elseif floor==3

control.task(enum,floor+1)=1;

else

switch enum

case 1

if control.fSensor.f\_e1<=2

control.task(enum,2)=1;

else

control.task(enum,3)=1;

end

case 2

if control.fSensor.f\_e2<=2

control.task(enum,2)=1;

else

control.task(enum,3)=1;

end

end

end

end

|  |  |
| --- | --- |
|  | Test Case T1.3.3.1 |
| Coverage Item | Tcover1.3.3.1 |
| Input | enum= 1  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [1 0 0 0;0 0 0 0] |
|  | Test Case T1.3.3.2 |
| Coverage Item | Tcover1.3.3.2 |
| Input | enum= 1  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 3; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 0 0 1;0 0 0 0] |
|  | Test Case T1.3.3.3 |
| Coverage Item | Tcover1.3.3.3 |
| Input | enum= 1  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 2;  ele\_controller.fSensor.f\_e1 = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 1 0 0;0 0 0 0] |
|  | Test Case T1.3.3.4 |
| Coverage Item | Tcover1.3.3.4 |
| Input | enum= 1  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 2;  ele\_controller.fSensor.f\_e1 = 3; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 1 1 0;0 0 0 0] |
|  | Test Case T1.3.3.5 |
| Coverage Item | Tcover1.3.3.5 |
| Input | enum= 2  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 0 0 0;1 0 0 0] |
|  | Test Case T1.3.3.6 |
| Coverage Item | Tcover1.3.3.6 |
| Input | enum= 2  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 3; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 0 0 0;0 0 0 1] |
|  | Test Case T1.3.3.7 |
| Coverage Item | Tcover1.3.3.7 |
| Input | enum= 2  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 2;  ele\_controller.fSensor.f\_e2 = 1; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 0 0 0;0 1 0 0] |
|  | Test Case T1.3.3.8 |
| Coverage Item | Tcover1.3.3.8 |
| Input | enum= 2  ele\_controller.task = [0 0 0 0; 0 0 0 0];  floor = 2;  ele\_controller.fSensor.f\_e2 = 3; |
| State | ele\_controller = ElevatorController;  ele\_controller.f1 = ButtonPanel\_f1;  ele\_controller.f2 = ButtonPanel\_f2;  ele\_controller.f3 = ButtonPanel\_f3;  ele\_controller.e1 = ButtonPanel\_e;  ele\_controller.e2 = ButtonPanel\_e;  ele\_controller.elevator = Elevator;  ele\_controller.fSensor = FloorSensor;  ele\_controller.dSensor = DoorSensor;  ele\_controller.emSensor = EmergSensor; |
| Expected Output | ele\_controller.task == [0 0 0 0;0 1 1 0] |

* Test coverage: 43/43=100%
* Test result: 3 passed

## T2: Integration Test

T2.1: Elevator+ElevatorController+ButtonPanel\_f1+ButtonPanel\_f2+ButtonPanel\_f3+ButtonPanel\_e1+ButtonPanel\_e2 Integration

|  |  |
| --- | --- |
|  | Test Case T2.1.1 |
| Coverage Item | Tcover2.1.1 |
| Input | pause(1);  testCase.press(testCase.e1app.Button\_1);  pause(3);  testCase.press(testCase.e1app.Button\_2);  testCase.press(testCase.e1app.Button\_3);  testCase.press(testCase.e1app.Button\_2);  testCase.press(testCase.e1app.Button\_1);  testCase.press(testCase.e2app.Button\_2);  pause(3);  testCase.press(testCase.f3app.downButton);  testCase.press(testCase.f2app.upButton);  testCase.press(testCase.f1app.upButton);  pause(1);  testCase.press(testCase.e1app.openButton); testCase.press(testCase.e2app.EmergencyButton); |
| State | Elevator 1 and Elevator2 is at floor 1 |
| Expected Output | Elevator waits at 1st floor, then goes to 2nd floor first, then 3rd floor, and go back to 2nd floor, finally go to 1st floor.Then assign near elevators to the requested floors, then open the elevator door, finally alarms it |

T2.2:

Elevator+ElevatorController+ButtonPanel\_f1+ButtonPanel\_f2+ButtonPanel\_f3+ButtonPanel\_e1+ButtonPanel\_e2+ButtonSignals Integration

|  |  |
| --- | --- |
|  | Test Case T2.2.1 |
| Coverage Item | Tcover2.2.1 |
| Input | pause(1);  testCase.press(testCase.e1app.Button\_1);  pause(3);  testCase.press(testCase.e1app.Button\_2);  testCase.press(testCase.e1app.Button\_3);  testCase.press(testCase.e1app.Button\_2);  testCase.press(testCase.e1app.Button\_1);  testCase.press(testCase.e2app.Button\_2);  pause(3);  testCase.press(testCase.f3app.downButton);  testCase.press(testCase.f2app.upButton);  testCase.press(testCase.f1app.upButton);  pause(1);  testCase.press(testCase.e1app.openButton); testCase.press(testCase.e2app.EmergencyButton); |
| State | Elevator 1 and Elevator2 is at floor 1 |
| Expected Output | Elevator waits at 1st floor, then goes to 2nd floor first, then 3rd floor, and go back to 2nd floor, finally go to 1st floor. During which the light in the elevator will show whether the elevator is up or down. The pointer on the floors will show which floor the elevators are on. Then assign near elevators to the requested floors, then open the elevator door, finally alarms it. The alarm light is red and the alarm has a sounds |

## T3: Functional Test

T3.1: Use Case “Move up” and “Move down”

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.1.1 |
| Coverage Item | Tcover3.1.1 |
| Input | testCase.press(testCase.e1app.Button\_1);  pause(3);  testCase.press(testCase.e1app.Button\_2);  testCase.press(testCase.e1app.Button\_3);  testCase.press(testCase.e1app.Button\_2);  testCase.press(testCase.e1app.Button\_1); |
| State | Elevator 1 is at floor 1 |
| Expected Output | Elevator waits at 1st floor, then goes to 2nd floor first, then 3rd floor, and go back to 2nd floor, finally go to 1st floor. |

T3.2: Use Case “Assign Elevator”

|  |  |
| --- | --- |
|  | Test Case T3.2.1 |
| Coverage Item | Tcover3.2.1 |
| Input | testCase.press(testCase.f3app.downButton);  testCase.press(testCase.f2app.upButton);  testCase.press(testCase.f1app.upButton);  pause(1); |
| State | testCase.press(testCase.e2app.Button\_2);  Elevator 1 is at floor 2 |
| Expected Output | Elevator waits at 2nd floor, elevator waits at 1st floor, when there is a request on the third floor, the 2nd floor will move there first. Followed a 2nd floor request, the same distance will move Elevator1 for this always. Followed by up request on 1st floor, the Elevator1 is nearer so go there. |

T3.3: Use Case “OpenDoor”

|  |  |
| --- | --- |
|  | Test Case T3.3.1 |
| Coverage Item | Tcover3.3.1 |
| Input | testCase.press(testCase.e1app.openButton); |
| State | Elevator1 at 1st floor |
| Expected Output | Elevator1 will keep open the door if <|> button is pushed |

T3.4: Use Case “Move independently”

|  |  |
| --- | --- |
|  | Test Case T3.4.1 |
| Coverage Item | Tcover3.4.1 |
| Input | Press 3 in the Elevator1 panel and Press 3 in Elevator2 panel while Elevator1 is going up to the 3rd floor. |
| State | Elevator1 and Elevator2 at 1st floor |
| Expected Output | Elevator2 will not be moved by the press operation in Elevator1 and Elevator1 will not be affected by the press operation in Elevator2 |

T3.5: Use Case “Keep open the door”

|  |  |
| --- | --- |
|  | Test Case T3.5.1 |
| Coverage Item | Tcover3.5.1 |
| Input | Keep on pressing the <|> button multiple times |
| State | Elevator1 at 3rd floor |
| Expected Output | Elevator1 will keep on the state of opening the door |

T3.6: Use Case “Open the door when closed”

|  |  |
| --- | --- |
|  | Test Case T3.6.1 |
| Coverage Item | Tcover3.6.1 |
| Input | Press <|> button when the door is closed and elevator not moving |
| State | Elevator1 at 1st floor |
| Expected Output | Elevator1 will open the door |

T3.7: Use Case “Immediate Close the door”

|  |  |
| --- | --- |
|  | Test Case T3.7.1 |
| Coverage Item | Tcover3.7.1 |
| Input | Press >|< button when the door is opened and elevator not moving |
| State | Elevator1 at 1st floor |
| Expected Output | Elevator1 will close the door immediately |

T3.8: Use Case “Ring the Alarm”

|  |  |
| --- | --- |
|  | Test Case T3.8.1 |
| Coverage Item | Tcover3.8.1 |
| Input | testCase.press(testCase.e2app.EmergencyButton); |
| State | Elevator2 at 3rd floor |
| Expected Output | Elevator2 will stop and ring the alarm, Elevator1 will keep on working. |