Group 13

Author: Liu Chengqi

Software VALIDATION

Painkiller Injection System

Table of Contents

[System Architecture 2](#_Toc44617158)

[T1: Unit Test 2](#_Toc44617159)

[T1.1: Pump Unit Test 2](#_Toc44617160)

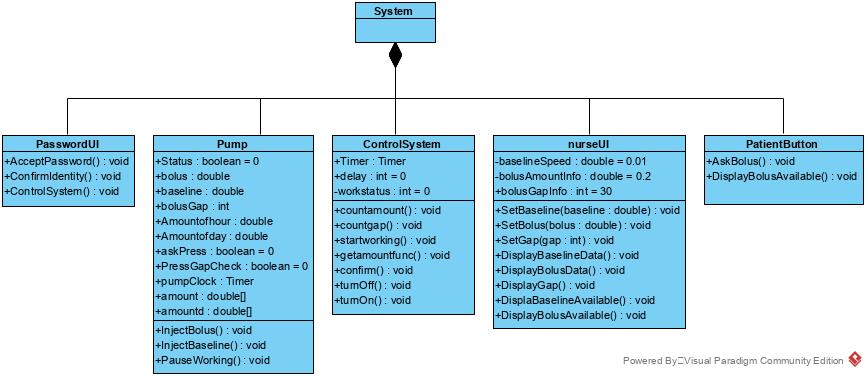
[T1.2: ControlSystem Unit Test 3](#_Toc44617161)

[T2: Integration Test 21](#_Toc44617162)

[T3: Functional Test 23](#_Toc44617163)

## System Architecture

The system architecture is shown below:



## T1: Unit Test

### T1.1: Pump Unit Test

T1.1.1: Test injectBolus ()

function injectBolus(pump)

pump.amount(pump.index)=pump.bolus; Tcover1.1.1.1

pump.amountd(pump.indexd)=pump.bolus;

pump.askPress = 0;

pump.PressGapcheck = 0;

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.1.1 |
| Coverage Item | Tcover1.1.1.1 |
| Input | pump.bolus=1;  pump.index=2;  pump.indexd=3;  pump.amount(pump.index)==pump.bolus;  pump.amountd(pump.indexd==pump.bolus;  pump.askPress == 0;  pump.PressGapcheck == 0; |
| State | pump.amount(pump.index)=pump.bolus;  pump.amountd(pump.indexd)=pump.bolus;  pump.askPress = 0;  pump.PressGapcheck = 0; |
| Expected Output | pump.amount(pump.index)==pump.bolus;  pump.amountd(pump.indexd==pump.bolus;  pump.askPress == 0;  pump.PressGapcheck == 0; |

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

T1.1.2: Test injectbaseline ()

function injectbaseline(pump)

pump.amount(pump.index)=pump.baseline; Tcover1.1.2.1

pump.amountd(pump.indexd)=pump.baseline;

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.2.1 |
| Coverage Item | Tcover1.1.2.1 |
| Input | pump.baseline=1;  pump.index=2;  pump.indexd=3; |
| State | pump.injectbaseline(); |
| Expected Output | pump.amount(pump.index)==pump.baseline;  pump.amountd(pump.indexd)==pump.baseline; |

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

T1.1.3: Test pauseinjection ()

function pauseinjection(pump)

pump.amount(pump.index)=0; Tcover1.1.3.1

pump.amountd(pump.indexd)=0;

pump.PressAvailable = 0;

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.3.1 |
| Coverage Item | Tcover1.1.3.1 |
| Input | pump.indexd=1;  pump.index=2; |
| State | pump.pauseinjection(); |
| Expected Output | pump.amount(pump.index)==0;  pump.amountd(pump.indexd)==0;  pump.PressAvailable==0; |

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

### T1.2: ControlSystem Unit Test

T1.2.1: Test countamount ()

function countamount(controlsys)

controlsys.pump.amount\_of\_hour = sum(controlsys.pump.amount,'all'); Tcover1.2.1.1

controlsys.pump.amount\_of\_day = sum(controlsys.pump.amountd,'all');

controlsys.pump.patient.Amountofhour = controlsys.pump.amount\_of\_hour;

controlsys.pump.patient.Amountofday = controlsys.pump.amount\_of\_day;

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.1.1 |
| Coverage Item | Tcover1.2.1.1 |
| Input | controlsys.pump.amount=ones(1,60);  controlsys.pump.amountd=ones(1,24\*60); |
| State | controlsys.countamount(); |
| Expected Output | controlsys.pump.amount\_of\_hour==sum(controlsys.pump.amount,'all');  controlsys.pump.amount\_of\_day==sum(controlsys.pump.amountd,'all');  controlsys.pump.patient.Amountofhour==controlsys.pump.amount\_of\_hour;  controlsys.pump.patient.Amountofday==controlsys.pump.amount\_of\_day; |

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

T1.2.2: Test countgap ()

function countgap(controlsys)

controlsys.pump.askPress = 0; Tcover1.2.2.1 and Tcover1.2.2.2

if controlsys.pump.gap < controlsys.pump.bolusGap

controlsys.pump.gap = controlsys.pump.gap + 1;

else

controlsys.pump.PressGapcheck = 1;

controlsys.pump.gap = 0;

end

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.2.1 |
| Coverage Item | Tcover1.2.2.1 |
| Input | controlsys.pump.gap=1;  controlsys.pump.bolusGap=2; |
| State | controlsys.countgap(); |
| Expected Output | controlsys.pump.askPress==0;  controlsys.pump.gap==2; |

|  |  |
| --- | --- |
|  | Test Case T1.2.2.2 |
| Coverage Item | Tcover1.2.2.2 |
| Input | controlsys.pump.gap=2;  controlsys.pump.bolusGap=1; |
| State | controlsys.countgap(); |
| Expected Output | controlsys.pump.PressGapcheck==1;  controlsys.pump.gap==0; |

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

T1.2.3: Test startwoking ()

function startwoking(controlsys)

Tcover1.2.3.1

controlsys.nurseUI.Message\_3.Text = ' Not get bolus';

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.Timer = timer(...

'ExecutionMode', 'fixedRate', ... % Run timer repeatedly

'Period', 0.1, ... % Period is 1 second

'BusyMode', 'queue',... % Queue timer callbacks when busy

'TimerFcn', @controlsys.getamountfunc); % Specify callback function

start(controlsys.Timer);

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.3.1 |
| Coverage Item | Tcover1.2.3.1 |
| Input | This function doesn’t need to input. |
| State | controlsys.startwoking(); |
| Expected Output | controlsys.nurseUI.Message\_3.Text=='Notgetbolus';  controlsys.nurseUI.BolusAvailableLamp.Color==[0.00 1.00 0.00];  controlsys.patientUI.BolusAvailableLamp.Color==[0.00 1.00 0.00]; |

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

T1.2.4: Test confirm ()

function confirm(controlsys)

controlsys.nurseUI.PUMPSwitch.Enable = 1; Tcover1.2.4.1

controlsys.pump.baseline = controlsys.nurseUI.baselineSpeed;

controlsys.pump.bolus = controlsys.nurseUI.bolusAmountInfo;

controlsys.pump.bolusGap = controlsys.nurseUI.bolusGapInfo;

%controlsys.pump.status = 1;

controlsys.pump.PressAvailable=1;

controlsys.pump.gap = 0;

controlsys.nurseUI.Message\_4.Text = num2str(controlsys.nurseUI.baselineSpeed);

controlsys.nurseUI.Message\_5.Text = num2str(controlsys.nurseUI.bolusAmountInfo);

controlsys.nurseUI.Message\_6.Text = num2str(controlsys.nurseUI.bolusGapInfo);

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.4.1 |
| Coverage Item | Tcover1.2.4.1 |
| Input | ontrolsys.nurseUI.baselineSpeed=1;  controlsys.nurseUI.bolusAmountInfo=2;  controlsys.nurseUI.bolusGapInfo=3; |
| State | controlsys.confirm(); |
| Expected Output | controlsys.nurseUI.PUMPSwitch.Enable==1;  controlsys.pump.baseline==controlsys.nurseUI.baselineSpeed;  controlsys.pump.bolus==controlsys.nurseUI.bolusAmountInfo;  controlsys.pump.bolusGap==controlsys.nurseUI.bolusGapInfo;  controlsys.pump.PressAvailable==1;  controlsys.pump.gap==0;  controlsys.nurseUI.Message\_4.Text==num2str(controlsys.nurseUI.baselineSpeed);  controlsys.nurseUI.Message\_5.Text==num2str(controlsys.nurseUI.bolusAmountInfo);  controlsys.nurseUI.Message\_6.Text==num2str(controlsys.nurseUI.bolusGapInfo); |

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

T1.2.5: Test turnOff ()

function turnOff(controlsys)

controlsys.nurseUI.PUMPSwitch.Enable = 0; Tcover1.2.5.1

controlsys.workstatus = 0;

controlsys.pump.baseline = 0.01;

controlsys.pump.bolus = 0.2;

controlsys.pump.bolusGap = 30;

controlsys.pump.status = 0;

controlsys.pump.amount=zeros(1,60);

controlsys.pump.amountd=zeros(1,24\*60);

controlsys.pump.amount\_of\_hour = 0;

controlsys.pump.amount\_of\_day = 0;

controlsys.pump.gap = 0;

controlsys.pump.PressAvailable=1;

controlsys.pump.PressGapcheck=1;

controlsys.nurseUI.BaselinemlminSpinner.Value = 0.01;

controlsys.nurseUI.BolusmlshotSpinner.Value = 0.2;

controlsys.nurseUI.BolusGapminSpinner.Value = 30;

delete(controlsys.Timer);

controlsys.nurseUI.Message.Text = '';

controlsys.nurseUI.Message\_2.Text = '';

controlsys.nurseUI.Message\_3.Text = '';

controlsys.nurseUI.BaselineAvailableLamp.Color = [0.65,0.65,0.65];

controlsys.nurseUI.BolusAvailableLamp.Color = [0.65,0.65,0.65];

controlsys.patientUI.BolusAvailableLamp.Color = [0.65,0.65,0.65];

controlsys.delay = 0;

controlsys.nurseUI.Message\_4.Text = '';

controlsys.nurseUI.Message\_5.Text = '';

controlsys.nurseUI.Message\_6.Text = '';

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.5.1 |
| Coverage Item | Tcover1.2.5.1 |
| Input | This function doesn’t need to input. |
| State | controlsys.turnOff(); |
| Expected Output | controlsys.nurseUI.PUMPSwitch.Enable==0;  controlsys.workstatus==0;  controlsys.pump.baseline==0.01;  controlsys.pump.bolus==0.2;  controlsys.pump.bolusGap==30;  controlsys.pump.status==0;  controlsys.pump.amount==zeros(1,60);  controlsys.pump.amountd==zeros(1,24\*60);  controlsys.pump.amount\_of\_hour==0;  controlsys.pump.amount\_of\_day==0;  controlsys.pump.gap==0;  controlsys.pump.PressAvailable==1;  controlsys.pump.PressGapcheck==1;  controlsys.nurseUI.BaselinemlminSpinner.Value==0.01;  controlsys.nurseUI.BolusmlshotSpinner.Value==0.2;  controlsys.nurseUI.BolusGapminSpinner.Value==30;  controlsys.nurseUI.Message.Text=='';  controlsys.nurseUI.Message\_2.Text=='';  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.BaselineAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.nurseUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.patientUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.delay==0;  controlsys.nurseUI.Message\_4.Text=='';  controlsys.nurseUI.Message\_5.Text=='';  controlsys.nurseUI.Message\_6.Text==''; |

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

T1.2.6: Test turnOn ()

function test\_turnOn(testCase)

testCase.controlsys.turnOn(); Tcover1.2.6.1

testCase.verifyEqual(testCase.controlsys.nurseUI.Message.Text , '0');

testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_2.Text , '0');

testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_3.Text , ' Not get bolus');

testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color , [0.65,0.65,0.65]);

testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color , [0 1 0]);

testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color , [0 1 0]);

testCase.verifyEqual(testCase.controlsys.nurseUI.controlsys.delay , 0);

testCase.verifyEqual(testCase.controlsys.workstatus , 1);

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.6.1 |
| Coverage Item | Tcover1.2.6.1 |
| Input | This function doesn’t need to input. |
| State | controlsys.turnOn(); |
| Expected Output | controlsys.nurseUI.Message.Text=='0';  controlsys.nurseUI.Message\_2.Text=='0';  controlsys.nurseUI.Message\_3.Text=='Notgetbolus';  controlsys.nurseUI.BaselineAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.nurseUI.BolusAvailableLamp.Color==[0 1 0];  controlsys.patientUI.BolusAvailableLamp.Color==[0 1 0];  controlsys.nurseUI.controlsys.delay==0;  controlsys.workstatus==1; |

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

T1.2.7: Test getamountfunc ()

controlsys.nurseUI.Message.Text = num2str(controlsys.pump.amount\_of\_hour); From Tcover1.2.7.1 to Tcover1.2.7.9

controlsys.nurseUI.Message\_2.Text = num2str(controlsys.pump.amount\_of\_day);

%Set the index of day(amountd[indexd])

if controlsys.pump.indexd == 24\*60

controlsys.pump.indexd = 1;

else

controlsys.pump.indexd = controlsys.pump.indexd + 1;

end

%Set the index of hour(amount[index])

if controlsys.pump.index == 60

controlsys.pump.index = 1;

else

controlsys.pump.index = controlsys.pump.index + 1;

end

%count gap

if controlsys.pump.PressGapcheck == 0

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.countgap;

end

%Define whether the button is available to be presses by

%hourlimit

if (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) > (controlsys.pump.hourlimit + 0.001)

controlsys.pump.PressAvailable = 0;

controlsys.pump.askPress = 0;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

else

if (controlsys.pump.amount\_of\_day + controlsys.pump.bolus) > (controlsys.pump.daylimit + 0.001)

controlsys.pump.PressAvailable = 0;

controlsys.pump.askPress = 0;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

else

controlsys.pump.PressAvailable = 1;

end

end

%Define the working status by the amount(day) of injection

if controlsys.delay == 0

if (controlsys.pump.amount\_of\_day + controlsys.pump.baseline) <= (controlsys.pump.daylimit + 0.0001) && (controlsys.pump.amount\_of\_hour + controlsys.pump.baseline) <= (controlsys.pump.hourlimit + 0.0001)

controlsys.pump.status=1;

controlsys.nurseUI.BaselineAvailableLamp.Color = [0.00,1.00,0.00];

if controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

end

else

if (controlsys.pump.amount\_of\_day + controlsys.pump.baseline) > (controlsys.pump.daylimit + 0.001)

controlsys.delay = 1;

end

controlsys.pump.status=0;

controlsys.nurseUI.BaselineAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.pump.pauseinjection;

controlsys.pump.askPress = 0;

end

else

if (controlsys.pump.amount\_of\_day + controlsys.nurseUI.pump.bolus) > (controlsys.pump.daylimit + 0.0001) || (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) > (controlsys.pump.hourlimit + 0.0001)

controlsys.pump.status=0;

controlsys.nurseUI.BaselineAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.pump.pauseinjection;

controlsys.pump.askPress = 0;

%controlsys.nurseUI.Message\_13.Text = '1';

elseif (controlsys.pump.amount\_of\_day + controlsys.pump.bolus) >= controlsys.pump.daylimit && (controlsys.pump.amount\_of\_day + controlsys.pump.bolus) < (controlsys.pump.daylimit + 0.0001)

if (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) < (controlsys.pump.hourlimit + 0.0001)

controlsys.pump.status=0;

controlsys.nurseUI.BaselineAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.pump.pauseinjection;

controlsys.pump.PressAvailable = 1;

if controlsys.pump.PressGapcheck==1

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

end

if controlsys.pump.askPress == 1 && controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.Message\_3.Text = datestr(now,0);

controlsys.pump.injectBolus;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.npatientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

end

end

%controlsys.nurseUI.Message\_13.Text = '2';

elseif (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) >= controlsys.pump.hourlimit && (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) < (controlsys.pump.hourlimit + 0.0001)

if (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) < (controlsys.pump.daylimit + 0.0001)

controlsys.pump.status=0;

controlsys.nurseUI.BaselineAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.pump.pauseinjection;

controlsys.pump.PressAvailable = 1;

if controlsys.pump.PressGapcheck==1

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

end

if controlsys.pump.askPress == 1 && controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.Message\_3.Text = datestr(now,0);

controlsys.pump.injectBolus;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

end

end

%controlsys.nurseUI.Message\_13.Text = '3';

elseif (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus + controlsys.pump.baseline) > (controlsys.pump.hourlimit + 0.0001) && (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) < controlsys.pump.hourlimit

if (controlsys.pump.amount\_of\_day + controlsys.pump.bolus) < (controlsys.pump.daylimit + 0.0001)

controlsys.pump.status=0;

controlsys.nurseUI.BaselineAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.pump.pauseinjection;

controlsys.pump.PressAvailable = 1;

if controlsys.pump.PressGapcheck==1

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

end

if controlsys.pump.askPress == 1 && controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.Message\_3.Text = datestr(now,0);

controlsys.pump.injectBolus;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

end

end

%controlsys.nurseUI.Message\_13.Text = '4';

elseif (controlsys.pump.amount\_of\_day + controlsys.pump.bolus + controlsys.pump.baseline) > (controlsys.pump.daylimit + 0.0001) && (controlsys.pump.amount\_of\_day + controlsys.pump.bolus) < controlsys.pump.daylimit

if (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) < (controlsys.pump.hourlimit + 0.0001)

controlsys.pump.status=0;

controlsys.nurseUI.BaselineAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.pump.pauseinjection;

controlsys.pump.PressAvailable = 1;

if controlsys.pump.PressGapcheck==1

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

end

if controlsys.pump.askPress == 1 && controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.Message\_3.Text = datestr(now,0);

controlsys.pump.injectBolus;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

end

end

%controlsys.nurseUI.Message\_13.Text = '5';

elseif (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus + controlsys.pump.baseline) > (controlsys.pump.hourlimit + 0.0001) && (controlsys.pump.amount\_of\_hour + controlsys.pump.bolus) <= controlsys.pump.hourlimit

if (controlsys.pump.amount\_of\_day + controlsys.pump.bolus) < (controlsys.pump.daylimit + 0.0001)

controlsys.pump.status=0;

controlsys.nurseUI.BaselineAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.pump.pauseinjection;

controlsys.pump.PressAvailable = 1;

if pp.pump.PressGapcheck==1

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

end

if controlsys.pump.askPress == 1 && controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.Message\_3.Text = datestr(now,0);

controlsys.pump.injectBolus;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

end

end

%controlsys.nurseUI.Message\_13.Text = '6';

else

controlsys.pump.status=1;

if controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [0.00,1.00,0.00];

end

controlsys.nurseUI.BaselineAvailableLamp.Color = [0.00,1.00,0.00];

%controlsys.nurseUI.Message\_13.Text = '7';

end

end

%The inject func when pump working status = 1

if controlsys.nurseUI.pump.status==1

%inject bolus

%controlsys.nurseUI.Message\_13.Text = '8';

if controlsys.pump.askPress == 1 && controlsys.pump.PressAvailable==1 && controlsys.pump.PressGapcheck==1

controlsys.nurseUI.Message\_3.Text = datestr(now,0);

controlsys.pump.injectBolus;

controlsys.nurseUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

controlsys.patientUI.BolusAvailableLamp.Color = [1.00,0.00,0.00];

%inject baseline

else

controlsys.pump.injectbaseline;

end

end

%count the amount

controlsys.countamount;

end

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.7.1 |
| Coverage Item | Tcover1.2.7.1 |
| Input | type(passwordUI.StaffIDEditField=='9999';  type(passwordUI.PasswordEditField=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=24\*60;  controlsys.pump.index=60;  controlsys.pump.PressGapcheck=0;  controlsys.pump.amount\_of\_hour=1;  controlsys.pump.bolus=1;  controlsys.pump.hourlimit=0;  controlsys.delay=0;  controlsys.pump.amount\_of\_day=5;  controlsys.pump.baseline=1;  controlsys.pump.daylimit=0;  controlsys.nurseUI.pump.status=1; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==1;  controlsys.nurseUI.BolusAvailableLamp.Color==[1.00 0.00 0.00];  controlsys.patientUI.BolusAvailableLamp.Color==[1.00 0.00 0.00];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==1;  controlsys.pump.index==1;  controlsys.pump.PressGapcheck==1;  controlsys.pump.amount\_of\_hour==0;  controlsys.pump.bolus==1;  controlsys.pump.hourlimit==0;  controlsys.delay==1;  controlsys.pump.amount\_of\_day==0;  controlsys.pump.baseline==1;  controlsys.pump.daylimit==0;  controlsys.nurseUI.pump.status==0; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.2 |
| Coverage Item | Tcover1.2.7.2 |
| Input | type(passwordUI.StaffIDEditField)=='9999';  type(passwordUI.PasswordEditField)=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=0;  controlsys.pump.index=0;  controlsys.pump.PressGapcheck=0;  controlsys.pump.amount\_of\_hour=0;  controlsys.pump.bolus=1;  controlsys.pump.hourlimit=0;  controlsys.delay=2;  controlsys.pump.amount\_of\_day=0;  controlsys.pump.baseline=1;  controlsys.pump.daylimit=0;  controlsys.nurseUI.pump.status=1; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==1;  controlsys.nurseUI.BolusAvailableLamp.Color==[1.00 0.00 0.00];  controlsys.patientUI.BolusAvailableLamp.Color==[1.00 0.00 0.00];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==1;  controlsys.pump.index==1;  controlsys.pump.PressGapcheck==1;  controlsys.pump.amount\_of\_hour==0;  controlsys.pump.bolus==1;  controlsys.pump.hourlimit==0;  controlsys.delay==2;  controlsys.pump.amount\_of\_day==0;  controlsys.pump.baseline==1;  controlsys.pump.daylimit==0;  controlsys.nurseUI.pump.status==0; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.3 |
| Coverage Item | Tcover1.2.7.3 |
| Input | type(passwordUI.StaffIDEditField)=='9999';  type(passwordUI.PasswordEditField)=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=7;  controlsys.pump.index=5;  controlsys.pump.PressGapcheck=6;  controlsys.pump.amount\_of\_hour=3;  controlsys.pump.bolus=1;  controlsys.pump.hourlimit=99;  controlsys.delay=2;  controlsys.pump.amount\_of\_day=0;  controlsys.pump.baseline=1;  controlsys.pump.daylimit=99;  controlsys.nurseUI.pump.status=1; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==8;  controlsys.nurseUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.patientUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==8;  controlsys.pump.index==6;  controlsys.pump.PressGapcheck==6;  controlsys.pump.amount\_of\_hour==1;  controlsys.pump.bolus==1;  controlsys.pump.hourlimit==99;  controlsys.delay==2;  controlsys.pump.amount\_of\_day==1;  controlsys.pump.baseline==1;  controlsys.pump.daylimit==99;  controlsys.nurseUI.pump.status==1; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.4 |
| Coverage Item | Tcover1.2.7.4 |
| Input | type(passwordUI.StaffIDEditField=='9999';  type(passwordUI.PasswordEditField=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=2;  controlsys.pump.index=2;  controlsys.pump.PressGapcheck=2;  controlsys.pump.amount\_of\_hour=2;  controlsys.pump.bolus=9;  controlsys.pump.hourlimit=20;  controlsys.delay=2;  controlsys.pump.amount\_of\_day=3;  controlsys.pump.baseline=1;  controlsys.pump.daylimit=99;  controlsys.nurseUI.pump.status=1; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==3;  controlsys.nurseUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.patientUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==3;  controlsys.pump.index==3;  controlsys.pump.PressGapcheck==2;  controlsys.pump.amount\_of\_hour==1;  controlsys.pump.bolus==9;  controlsys.pump.hourlimit==20;  controlsys.delay==2;  controlsys.pump.amount\_of\_day==1;  controlsys.pump.baseline==1;  controlsys.pump.daylimit==99;  controlsys.nurseUI.pump.status==1; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.5 |
| Coverage Item | Tcover1.2.7.5 |
| Input | type(passwordUI.StaffIDEditField)=='9999';  type(passwordUI.PasswordEditField)=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=0;  controlsys.pump.index=0;  controlsys.pump.PressGapcheck=0;  controlsys.pump.amount\_of\_hour=0;  controlsys.pump.bolus=0;  controlsys.pump.hourlimit=0;  controlsys.delay=0;  controlsys.pump.amount\_of\_day=0;  controlsys.pump.baseline=0;  controlsys.pump.daylimit=0;  controlsys.nurseUI.pump.status=0; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==1;  controlsys.nurseUI.BolusAvailableLamp.Color==[0 1 0];  controlsys.patientUI.BolusAvailableLamp.Color==[0 1 0];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==1;  controlsys.pump.index==1;  controlsys.pump.PressGapcheck==1;  controlsys.pump.amount\_of\_hour==0;  controlsys.pump.bolus==0;  controlsys.pump.hourlimit==0;  controlsys.delay==0;  controlsys.pump.amount\_of\_day==0;  controlsys.pump.baseline==0;  controlsys.pump.daylimit==0;  controlsys.nurseUI.pump.status==1; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.6 |
| Coverage Item | Tcover1.2.7.6 |
| Input | type(passwordUI.StaffIDEditField)=='9999';  type(passwordUI.PasswordEditField)=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=3;  controlsys.pump.index=5;  controlsys.pump.PressGapcheck=9;  controlsys.pump.amount\_of\_hour=33;  controlsys.pump.bolus=2;  controlsys.pump.hourlimit=2;  controlsys.delay=1;  controlsys.pump.amount\_of\_day=4;  controlsys.pump.baseline=4;  controlsys.pump.daylimit=5;  controlsys.nurseUI.pump.status=8; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.nurseUI.BolusAvailableLamp.Color==[1 0 0];  controlsys.patientUI.BolusAvailableLamp.Color==[1 0 0];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==4;  controlsys.pump.index==6;  controlsys.pump.PressGapcheck==9;  controlsys.pump.amount\_of\_hour==0;  controlsys.pump.bolus==2;  controlsys.pump.hourlimit==2;  controlsys.delay==1;  controlsys.pump.amount\_of\_day==0;  controlsys.pump.baseline==4;  controlsys.pump.daylimit==5;  controlsys.nurseUI.pump.status==0; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.7 |
| Coverage Item | Tcover1.2.7.7 |
| Input | type(passwordUI.StaffIDEditField)=='9999';  type(passwordUI.PasswordEditField)=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=0;  controlsys.pump.index=30;  controlsys.pump.PressGapcheck=30;  controlsys.pump.amount\_of\_hour=2;  controlsys.pump.bolus=5;  controlsys.pump.hourlimit=8;  controlsys.delay=3;  controlsys.pump.amount\_of\_day=0;  controlsys.pump.baseline=3;  controlsys.pump.daylimit=3;  controlsys.nurseUI.pump.status=0; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==1;  controlsys.nurseUI.BolusAvailableLamp.Color==[1 0 0];  controlsys.patientUI.BolusAvailableLamp.Color==[1 0 0];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==1;  controlsys.pump.index==31;  controlsys.pump.PressGapcheck==30;  controlsys.pump.amount\_of\_hour==0;  controlsys.pump.bolus==5;  controlsys.pump.hourlimit==8;  controlsys.delay==3;  controlsys.pump.amount\_of\_day==0;  controlsys.pump.baseline==3;  controlsys.pump.daylimit==3;  controlsys.nurseUI.pump.status==0; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.8 |
| Coverage Item | Tcover1.2.7.8 |
| Input | type(passwordUI.StaffIDEditField)=='9999';  type(passwordUI.PasswordEditField)=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=1;  controlsys.pump.index=1;  controlsys.pump.PressGapcheck=-1;  controlsys.pump.amount\_of\_hour=-1;  controlsys.pump.bolus=-1;  controlsys.pump.hourlimit=0;  controlsys.delay=-1;  controlsys.pump.amount\_of\_day=-1;  controlsys.pump.baseline=-1;  controlsys.pump.daylimit=9;  controlsys.nurseUI.pump.status=0; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==2;  controlsys.nurseUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.patientUI.BolusAvailableLamp.Color==[0.65 0.65 0.65];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==2;  controlsys.pump.index==2;  controlsys.pump.PressGapcheck==-1;  controlsys.pump.amount\_of\_hour==-1;  controlsys.pump.bolus==-1;  controlsys.pump.hourlimit==0;  controlsys.delay==-1;  controlsys.pump.amount\_of\_day==-1;  controlsys.pump.baseline==-1;  controlsys.pump.daylimit==9;  controlsys.nurseUI.pump.status==1; |

|  |  |
| --- | --- |
|  | Test Case T1.2.7.9 |
| Coverage Item | Tcover1.2.7.9 |
| Input | type(passwordUI.StaffIDEditField)=='9999';  type(passwordUI.PasswordEditField)=='9999';  press(passwordUI.ConfirmButton);  controlsys.pump.indexd=1;  controlsys.pump.index=1;  controlsys.pump.PressGapcheck=3;  controlsys.pump.amount\_of\_hour=1;  controlsys.pump.bolus=2;  controlsys.pump.hourlimit=0;  controlsys.delay=1;  controlsys.pump.amount\_of\_day=5;  controlsys.pump.baseline=7;  controlsys.pump.daylimit=9;  controlsys.nurseUI.pump.status=3; |
| State | controlsys.getamountfunc(); |
| Expected Output | controlsys.pump.indexd==2;  controlsys.nurseUI.BolusAvailableLamp.Color==[1 0 0];  controlsys.patientUI.BolusAvailableLamp.Color==[1 0 0];  controlsys.nurseUI.Message\_3.Text=='';  controlsys.nurseUI.Message\_13.Text=='';  controlsys.pump.indexd==2;  controlsys.pump.index==2;  controlsys.pump.PressGapcheck==3;  controlsys.pump.amount\_of\_hour==0;  controlsys.pump.bolus==2;  controlsys.pump.hourlimit==0;  controlsys.delay==1;  controlsys.pump.amount\_of\_day==0;  controlsys.pump.baseline==7;  controlsys.pump.daylimit==9;  controlsys.nurseUI.pump.status==0; |

* Test coverage: 9/9=100%
* Test result: 9 passed

## T2: Integration Test

T2.1: Pump+ CotrolSystem+ nurseUI+ PasswordUI+ PatientButton Integration

This part is the integration test for the whole project. It contains the two unit tests before (Pump and ControlSystem, as well as all UI. It integrates all the functions of this project.

Considering that the same program may run at different speeds on different devices, this will affect the correctness of the test. Therefore, the following parts will be displayed with video.

The codes can be seen in the “test\_Integration.m”, and the video can be seen in the “test\_Integration.mp4”. Some Chinese comments in the code can help understand the meaning of the test content.

|  |  |
| --- | --- |
|  | Test Case T2.1.1 |
| Coverage Item | Tcover2.1.1 |
| Input | No additional setting input is required. |
| State | testCase.type(testCase.passwordUI.StaffIDEditField,'abcd');  pause(0.5);  testCase.type(testCase.passwordUI.PasswordEditField,'1234');  pause(0.5);  testCase.press(testCase.passwordUI.ConfirmButton);  pause(0.5);  testCase.type(testCase.passwordUI.StaffIDEditField,'9999');  pause(0.5);  testCase.type(testCase.passwordUI.PasswordEditField,'9999');  pause(0.5);  testCase.press(testCase.passwordUI.ConfirmButton);  pause(1);  testCase.press(testCase.nur.nUIapp.BaselinemlminSpinner,'up');  pause(0.5);  testCase.press(testCase.nur.nUIapp.BolusmlshotSpinner,'up');  pause(0.5);  testCase.press(testCase.nur.nUIapp.BolusGapminSpinner,'up');  pause(0.5);  testCase.press(testCase.nur.nUIapp.ConfirmButton);  pause(1);  testCase.press(testCase.nur.nUIapp.PUMPSwitch);  pause(2);  testCase.press(testCase.paitentUI.GetBolusButton);  pause(1);  testCase.press(testCase.paitentUI.GetBolusButton);  pause(0.5);  testCase.press(testCase.paitentUI.GetBolusButton);  testCase.press(testCase.nur.nUIapp.BaselinemlminSpinner,'down');  pause(4);  testCase.press(testCase.nur.nUIapp.ConfirmButton);  pause(6);  pause(140);  testCase.press(testCase.paitentUI.GetBolusButton);  pause(4.5);  testCase.press(testCase.nur.nUIapp.PUMPSwitch);  pause(3); |
| Expected Output | testCase.verifyEqual(testCase.passwordUI.Label.Text,'Wrongidorpassword');  testCase.verifyEqual(testCase.passwordUI.nurse.ID,testCase.passwordUI.id);  testCase.verifyEqual(testCase.passwordUI.nurse.password,testCase.passwordUI.password);  testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!');  testCase.verifyEqual(testCase.controlsys.nurseUI.PUMPSwitch.Enable,'on');  testCase.verifyEqual(testCase.controlsys.pump.baseline,testCase.controlsys.nurseUI.baselineSpeed);  testCase.verifyEqual(testCase.controlsys.pump.bolus,testCase.controlsys.nurseUI.bolusAmountInfo);  testCase.verifyEqual(testCase.controlsys.pump.bolusGap,testCase.controlsys.nurseUI.bolusGapInfo);  testCase.verifyEqual(testCase.controlsys.pump.PressAvailable,1);  testCase.verifyEqual(testCase.controlsys.pump.gap,0);  testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_4.Text,num2str(testCase.controlsys.nurseUI.baselineSpeed));  testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_5.Text,num2str(testCase.controlsys.nurseUI.bolusAmountInfo));  testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_6.Text,num2str(testCase.controlsys.nurseUI.bolusGapInfo));  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[1.00,0.00,0.00]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[1.00,0.00,0.00]);  pause(4.5);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[010]);  testCase.press(testCase.paitentUI.GetBolusButton);  pause(15);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[1.00,0.00,0.00]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[1.00,0.00,0.00]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[1.00,0.00,0.00]);  testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!');  testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!');  testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!'); |

(The code in the source file “test\_Integration.m” contains more detailed segmentation and comments, which is easier to understand.)

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

## T3: Functional Test

As in the previous section, this part of the test also contains both video and code files.

T3.1: Use Case “Input Password”

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.1.1 |
| Coverage Item | Tcover3.1.1 |
| Input | No additional setting input is required. |
| State | estCase.type(testCase.passwordUI.StaffIDEditField,'abcd');  testCase.type(testCase.passwordUI.PasswordEditField,'1234');  testCase.press(testCase.passwordUI.ConfirmButton);  testCase.type(testCase.passwordUI.StaffIDEditField,'9999');  testCase.type(testCase.passwordUI.PasswordEditField,'9999');  testCase.press(testCase.passwordUI.ConfirmButton); |
| Expected Output | testCase.verifyEqual(testCase.passwordUI.Label.Text,'Wrongidorpassword');  testCase.verifyEqual(testCase.passwordUI.nurse.ID,testCase.passwordUI.id);  testCase.verifyEqual(testCase.passwordUI.nurse.password,testCase.passwordUI.password); |

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

T3.2: Use Case “Set Initial Value”

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.2.1 |
| Coverage Item | Tcover3.2.1 |
| Input | testCase.type(testCase.passwordUI.StaffIDEditField,'9999');  testCase.type(testCase.passwordUI.PasswordEditField,'9999');  testCase.press(testCase.passwordUI.ConfirmButton); |
| State | testCase.press(testCase.nur.nUIapp.BaselinemlminSpinner,'up');  testCase.press(testCase.nur.nUIapp.BolusmlshotSpinner,'up');  testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!');  testCase.press(testCase.nur.nUIapp.BolusGapminSpinner,'up');  testCase.press(testCase.nur.nUIapp.ConfirmButton); |
| Expected Output | testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!');  testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!');  testCase.verifyEqual(testCase.controlsys.nurseUI.PUMPSwitch.Enable,'on');  testCase.verifyEqual(testCase.controlsys.pump.baseline,testCase.controlsys.nurseUI.baselineSpeed);  testCase.verifyEqual(testCase.controlsys.pump.bolus,testCase.controlsys.nurseUI.bolusAmountInfo);  testCase.verifyEqual(testCase.controlsys.pump.bolusGap,testCase.controlsys.nurseUI.bolusGapInfo);  testCase.verifyEqual(testCase.controlsys.pump.PressAvailable,1);  testCase.verifyEqual(testCase.controlsys.pump.gap,0);  testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_4.Text,num2str(testCase.controlsys.nurseUI.baselineSpeed));  testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_5.Text,num2str(testCase.controlsys.nurseUI.bolusAmountInfo));  testCase.verifyEqual(testCase.controlsys.nurseUI.Message\_6.Text,num2str(testCase.controlsys.nurseUI.bolusGapInfo)); |

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

T3.3: Use Case “Inject the Painkiller”

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.3.1 |
| Coverage Item | Tcover3.3.1 |
| Input | testCase.type(testCase.passwordUI.StaffIDEditField,'9999');  testCase.type(testCase.passwordUI.PasswordEditField,'9999');  testCase.press(testCase.passwordUI.ConfirmButton);  testCase.press(testCase.nur.nUIapp.ConfirmButton);  testCase.press(testCase.nur.nUIapp.PUMPSwitch);  pause(2); |
| State | %bolus  testCase.press(testCase.paitentUI.GetBolusButton);  pause(1);  testCase.press(testCase.paitentUI.GetBolusButton);  pause(0.5);  testCase.press(testCase.paitentUI.GetBolusButton); |
| Expected Output | testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[1.00,0.00,0.00]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[1.00,0.00,0.00]);  pause(4.5);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[010]);  testCase.press(testCase.paitentUI.GetBolusButton);  pause(0.5);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[100]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[1.00,0.00,0.00]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[010]);  pause(5);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[010]); |

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

T3.4: Use Case “Change the value while injecting”

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.4.1 |
| Coverage Item | Tcover3.4.1 |
| Input | testCase.type(testCase.passwordUI.StaffIDEditField,'9999');  testCase.type(testCase.passwordUI.PasswordEditField,'9999');  testCase.press(testCase.passwordUI.ConfirmButton);  testCase.press(testCase.nur.nUIapp.ConfirmButton);  testCase.press(testCase.nur.nUIapp.PUMPSwitch); |
| State | pause(1);  %invalid  testCase.press(testCase.nur.nUIapp.BolusmlshotSpinner,'up');  pause(4);  %confirm,valid  testCase.press(testCase.nur.nUIapp.ConfirmButton);  pause(2); |
| Expected Output | testCase.verifyEqual(testCase.nur.nUIapp.Message\_13.Text,'NotConfirmed!'); |

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

T3.5: Use Case “Inject to the next day”

* Coverage Criteria: Statement coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.5.1 |
| Coverage Item | Tcover3.5.1 |
| Input | testCase.type(testCase.passwordUI.StaffIDEditField,'9999');  testCase.type(testCase.passwordUI.PasswordEditField,'9999');  testCase.press(testCase.passwordUI.ConfirmButton);  testCase.press(testCase.nur.nUIapp.ConfirmButton);  testCase.press(testCase.nur.nUIapp.PUMPSwitch); |
| State | %nextday,waitforlongtime：  %%%%%%%  pause(130); |
| Expected Output | testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[100]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[100]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[100]);  pause(30);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[010]);  testCase.press(testCase.paitentUI.GetBolusButton);  pause(0.5);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[100]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[100]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[100]);  pause(4.5);  %%%%%%% |

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

T3.6: Use Case “Stop Injection”

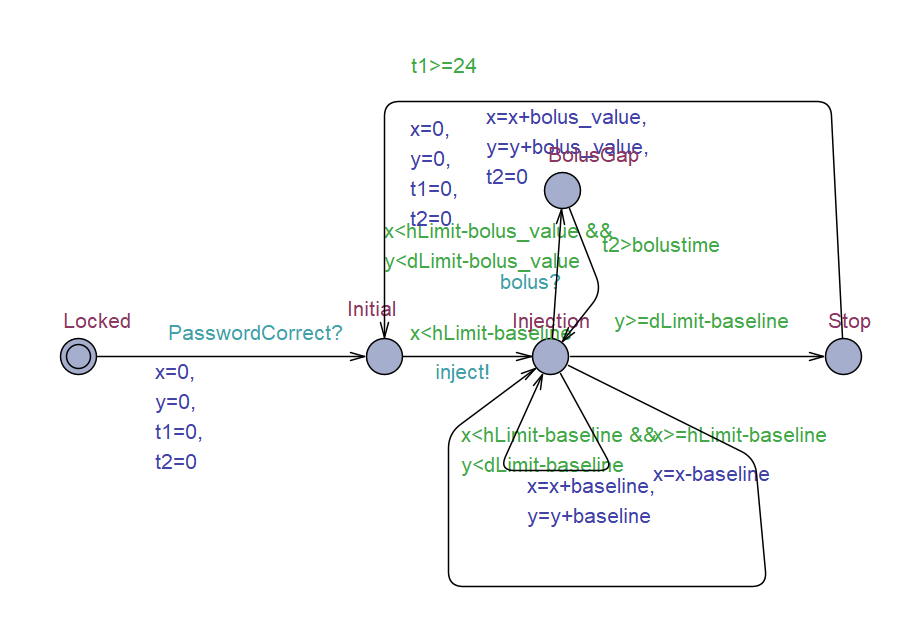
* Coverage Criteria: Statement coverage
* Test case

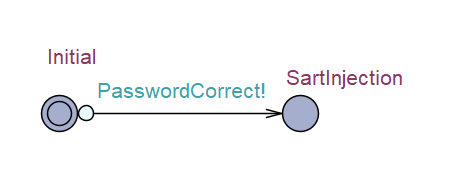
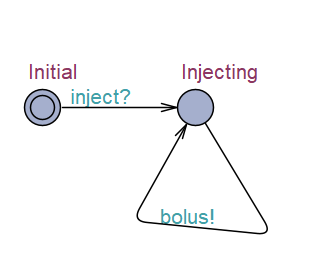
|  |  |
| --- | --- |
|  | Test Case T3.6.1 |
| Coverage Item | Tcover3.6.1 |
| Input | testCase.type(testCase.passwordUI.StaffIDEditField,'9999');  testCase.type(testCase.passwordUI.PasswordEditField,'9999');  testCase.press(testCase.passwordUI.ConfirmButton);  testCase.press(testCase.nur.nUIapp.ConfirmButton);  testCase.press(testCase.nur.nUIapp.PUMPSwitch);  pause(2); |
| State | %stopinjection  testCase.press(testCase.nur.nUIapp.PUMPSwitch);  pause(0.5); |
| Expected Output | testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[010]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[010]);  pause(3);  testCase.verifyEqual(testCase.controlsys.nurseUI.BolusAvailableLamp.Color,[0.650.650.65]);  testCase.verifyEqual(testCase.controlsys.patientUI.BolusAvailableLamp.Color,[0.650.650.65]);  testCase.verifyEqual(testCase.controlsys.nurseUI.BaselineAvailableLamp.Color,[0.650.650.65]); |

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

## Model Checking

These are the system, nurse and patient.



It has these two properties:

A[]x<dLimit and x<hLimit

A[]y<dLimit