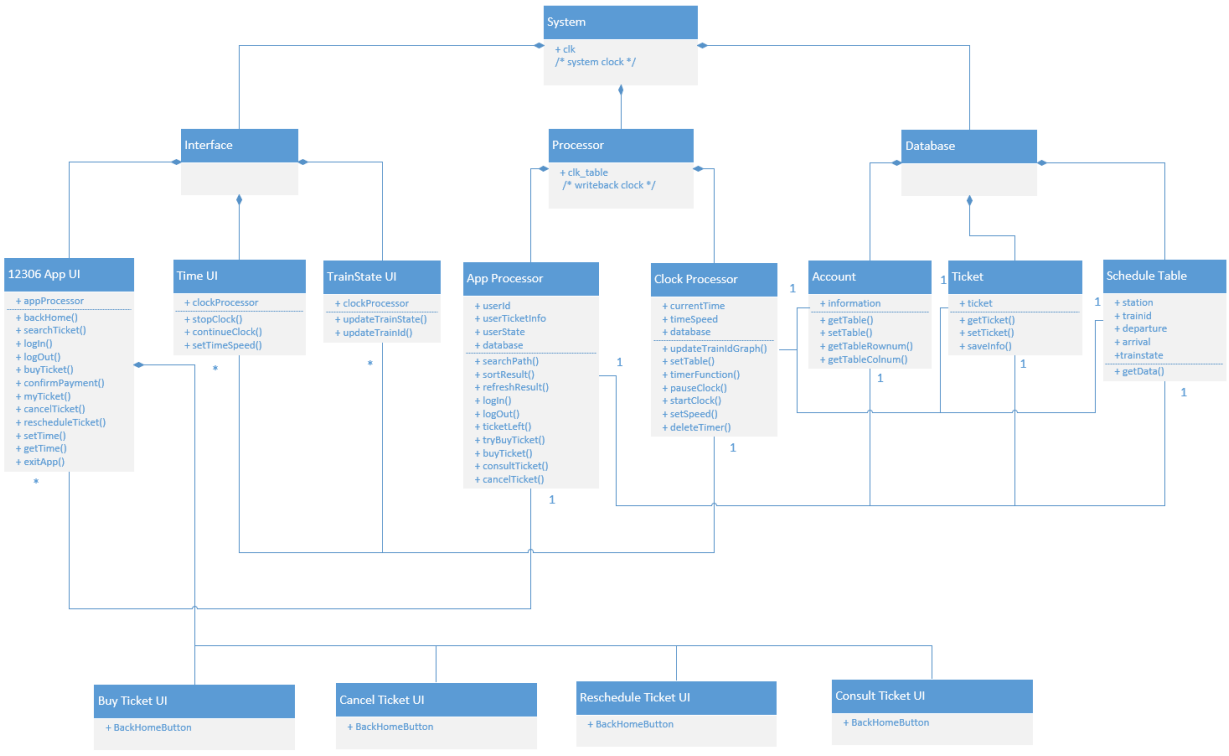
Validation

Contents

System Architecture



File Architecture

```
account.csv
Account.m
Clock.m
Consult.m
ConsultResult.m
DataBase.m
main.m
railway.png
savetime.csv
schedule.xlsx
ticket.csv
Ticket.m
ticket_save.csv
view.mlapp
```

T1 Unit Test

Since Clock.m is nearly impossible to test in unit tests, so we move it to functional tests.

T1.1 DataBase.m Test

T1.1.1 Test getData()

```
function outputArg = getData(obj, row, coloumn)
    if coloumn == 1 %Tcover 1.1.1.1
        outputArg = obj.station(row);
    elseif coloumn == 2 %Tcover 1.1.1.2
        outputArg = obj.trainid(row);
    elseif coloumn == 3 %Tcover 1.1.1.3
        outputArg = obj.arrival(row);
    elseif coloumn == 4 %Tcover 1.1.1.4
        outputArg = obj.departure(row);
    elseif coloumn == 5 %Tcover 1.1.1.5
        outputArg = obj.trainstate(row);
    end
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.1.1.1	tc T1.1.1.2	tc T1.1.1.3	tc T1.1.1.4	tc T1.1.1.5
coverage item	Tcover1.1.1.1	Tcover1.1.1.2	Tcover1.1.1.2	Tcover1.1.1.4	Tcover1.1.1.5
input	row = 1, coloumn = 1	row = 1, coloumn = 2	row = 1, coloumn = 3	row = 1, coloumn = 4	row = 1, coloumn = 5
state	/	/	/	/	/
expected output	{'苏州北'}	{'D11'}	-1	603	0

- Test Coverage = 5/5 = 100%
- Test Passed = 5

T1.2 Ticket.m Test

T1.2.1 setTableInfo()

```
%aux is to control whether to initialize a new line in table.
function setTableInfo(obj, row, col, val, aux)
    if aux == 1 %Tcover 1.2.1.2
        initRow = table([0],[0],[0],[0],[0],{'无'},{'无'},{'C1'},[0],{'无'},[0],
[0],[0]);
        obj.information = [obj.information; initRow];
    end

    obj.information.(col)(row) = val;
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.2.1.1	tc T1.2.1.2
coverage item	Tcover1.2.1	Tcover1.2.1.2
input	row = 1, col = 1, val = 1211, aux = 0	row = 1, col = 1, val = 1211, aux = 1
state	/	/
expected state	obj.information.(1)(1) = 1211	obj.information.(1)(1) = 1211

- Test Coverage = 1/1 = 100%
- Test Passed = 2

T1.2.2 getTableInfo()

```
function outputArg = getTableInfo(obj, row, col)
    outputArg = obj.information.(col)(row);
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.2.2.1
coverage item	Tcover1.2.2
input	row = 1, col = 2
state	obj.information.(2)(1) = 0
expected output	0

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.2.3 getTicket()

```
function outputArg = getTicket(obj, row, col)
    outputArg = obj.ticket.(col)(row);
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.2.3.1
coverage item	Tcover1.2.3
input	row = 1, col = 1
state	obj.ticket.(1)(1) = 2
expected output	2

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.2.4 setTicket()

```
function setTicket(obj, row, col, val)
    obj.ticket.(col)(row) = val;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.2.4.1
coverage item	Tcover1.2.4
input	row = 1, col = 1, val = 5
state	/
expected state	obj.ticket.(1)(1) = 5

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.2.5 getTableRownum()

```
function outputArg = getTableRownum(obj)
    outputArg = size(obj.information, 1);
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.2.5.1
coverage item	Tcover1.2.5
input	/
state	information = table
expected output	0

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.2.6 getTableColnum()

```
function outputArg = getTableColnum(obj)
    outputArg = size(obj.information, 2);
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.2.6.1
coverage item	Tcover1.2.6
input	/
state	information = table
expected output	0

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.2.7 saveInfo()

```
function saveInfo(obj)
    writetable(obj.information, 'account.csv');
    writetable(obj.ticket, 'ticket.csv');
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.2.7.1
coverage item	Tcover1.2.7
input	/
state	information.(1)(1) = 5, ticket.(1)(1) = 5
expected state	csv.information.(1)(1) = 5, csv.ticket.(1)(1) = 5

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.3 Consult.m Test

T1.3.1 setIndex()

```
function obj = setIndex(obj, input)
    obj.index = input;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.3.1.1
coverage item	Tcover1.3.1
input	5
state	/
expected state	consult.index = 5

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.3.2 sorting()

```
function outputArg = sorting(obj, way_sort, arr)
            count = 0;
            for m = 1:30
                if arr(m).departure_time == 0 && arr(m).arrival_time == 0
%Tcover 1.3.2.1
                    count = m - 1;
                    break;
                end
            end
            if strcmp(way_sort, '最少耗时') %Tcover 1.3.2.2
                n = count;
                sortedArray = arr;
                for i = 1:n-1
                    for j = 1:n-1
                        if sortedArray(j).interval_time >
sortedArray(j+1).interval_time                          %Tcover 1.3.2.3
                            temp = sortedArray(j);
                            sortedArray(j) = sortedArray(j+1);
                            sortedArray(j+1) = temp;
                        end
                    end
                end
            end
            if strcmp(way_sort, '最早到时') %Tcover 1.3.2.4
                n = count;
                sortedArray = arr;
                for i = 1:n-1
                    for j = 1:n-1
                        if sortedArray(j).arrival_time >
sortedArray(j+1).arrival_time
                            %Tcover 1.3.2.5
                            temp = sortedArray(j);
                            sortedArray(j) = sortedArray(j+1);
                            sortedArray(j+1) = temp;
                        end
                    end
                end
            end
            if strcmp(way_sort, '最小价格') %Tcover 1.3.2.6
                n = count;
                sortedArray = arr;
                for i = 1:n-1
                    for j = 1:n-1
                        if sortedArray(j).price > sortedArray(j+1).price
                            %Tcover 1.3.2.7
                            temp = sortedArray(j);
                            sortedArray(j) = sortedArray(j+1);
                            sortedArray(j+1) = temp;
                        end
                    end
                end
            end
            obj.array = sortedArray;
            obj.lastArray = sortedArray;
            outputArg = sortedArray;
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.3.2.1	tc T1.3.2.2	tc T1.3.2.3
coverage item	Tcover1.3.2.1, Tcover1.3.2.2, Tcover1.3.2.3	Tcover1.3.2.1, Tcover1.3.2.4, Tcover1.3.2.5	Tcover1.3.2.1, Tcover1.3.2.6, Tcover1.3.2.7
input	way_sort = '最少耗时', arr = testArray	way_sort = '最早到时', arr = testArray	way_sort = '最小价格', arr = testArray
state	testArray(1).interval_time = 60, testArray(1).departure_time = 30, testArray(1).arrival_time = 90, testArray(2).interval_time = 30, testArray(2).departure_time = 30, testArray(2).arrival_time = 60	testArray(1).departure_time = 30, testArray(1).arrival_time = 90, testArray(2).departure_time = 30, testArray(2).arrival_time = 60	testArray(1).price = 6, testArray(1).departure_time = 30, testArray(1).arrival_time = 90, testArray(2).price = 3, testArray(2).departure_time = 30, testArray(2).arrival_time = 60
expected output	outputArg(1).interval_time = 30, outputArg(2).interval_time = 60	outputArg(1).arrival_time = 60, outputArg(2).arrival_time = 90	outputArg(1).price= 3, outputArg(2).price= 6

- Test Coverage = 7/7 = 100%
- Test Passed = 3

T1.3.3 refreshArray()

```
function refreshArray(obj)
            myobj = ConsultResult();
            obj.array = repmat(myobj,[1,30]);
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.3.3.1
coverage item	Tcover1.3.3
input	/
state	testArray(1).price = 6, array = testArray
expected state	array(1).price = 0

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.3.4 search()

```
function outputArg = search(obj, now_time, dep_station,arrival_station, date)
    obj.refreshArray();
    result_cnt = 0;
    for i = 1:90
        cal_time = now_time;
        price = 0;
        if strcmp(obj.database.getData(i, 1), dep_station) && (obj.database.getData(i, 4) > cal_time) %Tcover 1.3.4.1
                    iteri = i + 1;

                    while iteri <= 90
                        break_tag = 0;

                        if obj.database.getData(i, 5) %Tcover 1.3.4.2
                            price = price + 4;
                        else %Tcover 1.3.4.3
                            price = price + 1;
                        end
                        if strcmp(obj.database.getData(iteri, 1), arrival_station) %Tcover 1.3.4.4

                            result_cnt  = result_cnt + 1;
                            obj.array(result_cnt).departure_time = obj.database.getData(i, 4);
                            obj.array(result_cnt).arrival_time = obj.database.getData(iteri, 3);
                            obj.array(result_cnt).departure_station = dep_station;
                            obj.array(result_cnt).arrival_station = arrival_station;
                            obj.array(result_cnt).interval_time =  obj.array(result_cnt).arrival_time - obj.array(result_cnt).departure_time;
                            obj.array(result_cnt).train_id1 = obj.database.getData(i, 2);
                            obj.array(result_cnt).price = price;
                            obj.array(result_cnt).date = date;
                            break;
                        end

                        if obj.database.getData(iteri, 4) == -1 %Tcover 1.3.4.5
                            cal_time = obj.database.getData(iteri, 3);
                            save_price = price;
                            for j = 1:90
                                if strcmp(obj.database.getData(j, 1), obj.database.getData(iteri, 1)) && (obj.database.getData(j, 4) > cal_time) %Tcover 1.3.4.6
                                    transfer_time = obj.database.getData(j, 4);
                                    iterj = j + 1;

                                    if obj.database.getData(iterj, 5) %Tcover 1.3.4.7

                                        if strcmp(obj.database.getData(iterj, 1), dep_station) %Tcover 1.3.4.8
                                            continue;
                                        end

                                        if strcmp(obj.database.getData(iterj, 1), arrival_station) %Tcover 1.3.4.9
                                            result_cnt  = result_cnt + 1;
                                            obj.array(result_cnt).departure_time = obj.database.getData(i, 4);
                                            obj.array(result_cnt).arrival_time = obj.database.getData(iterj, 3);
                                            obj.array(result_cnt).departure_station = dep_station;
                                            obj.array(result_cnt).arrival_station = arrival_station;
                                            obj.array(result_cnt).interval_time =  obj.array(result_cnt).arrival_time - obj.array(result_cnt).departure_time;
                                            obj.array(result_cnt).train_id1 = obj.database.getData(i, 2);
                                            obj.array(result_cnt).train_id2 = obj.database.getData(j, 2);
                                            obj.array(result_cnt).whether_transfer = true;
                                            obj.array(result_cnt).transfer_station = obj.database.getData(iteri, 1);
                                            obj.array(result_cnt).price = price + 4;
                                            price = save_price;
                                            obj.array(result_cnt).date = date;
                                            obj.array(result_cnt).transfer_time = transfer_time;
                                            continue;
                                        end
                                    end

                                    while obj.database.getData(iterj, 4) ~= -1

                                        if obj.database.getData(j, 5) %Tcover 1.3.4.10
                                            price = price + 4;
                                        else %Tcover 1.3.4.11
                                            price = price + 1;
                                        end

                                        if strcmp(obj.database.getData(iterj, 1), dep_station) %Tcover 1.3.4.12
                                            break;
                                        end

                                        if strcmp(obj.database.getData(iterj, 1), arrival_station) %Tcover 1.3.4.13
                                            result_cnt  = result_cnt + 1;
                                            obj.array(result_cnt).departure_time = obj.database.getData(i, 4);
                                            obj.array(result_cnt).arrival_time = obj.database.getData(iterj, 3);
                                            obj.array(result_cnt).departure_station = dep_station;
                                            obj.array(result_cnt).arrival_station = arrival_station;
                                            obj.array(result_cnt).interval_time =  obj.array(result_cnt).arrival_time - obj.array(result_cnt).departure_time;
                                            obj.array(result_cnt).train_id1 = obj.database.getData(i, 2);
                                            obj.array(result_cnt).train_id2 = obj.database.getData(j, 2);
                                            obj.array(result_cnt).whether_transfer = true;
                                            obj.array(result_cnt).transfer_station = obj.database.getData(iteri, 1);
                                            obj.array(result_cnt).price = price;
                                            price = save_price;
                                            obj.array(result_cnt).date = date;
                                            obj.array(result_cnt).transfer_time = transfer_time;
                                            break;
                                        end

                                        iterj = iterj + 1;
                                    end
                                    price = save_price;
                                end
                                if j == 90 %Tcover 1.3.4.14
                                    break_tag = 1;
                                end
```

```
                    end
                end

                if break_tag == 1 %Tcover 1.3.4.15
                    break;
                end
                iteri = iteri + 1;
            end
        end
    end
    obj.lastArray = obj.array;
    outputArg = obj.array;
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.3.4.1	tc T1.3.4.2	tc T1.3.4.3
coverage item	Tcover1.3.4.1, Tcover1.3.4.2, Tcover1.3.4.3, Tcover1.3.4.4	Tcover1.3.4.1, Tcover1.3.4.2, Tcover1.3.4.5, Tcover1.3.4.6, Tcover1.3.4.7, Tcover1.3.4.8, Tcover1.3.4.11, Tcover1.3.4.12, Tcover1.3.4.13, Tcover1.3.4.14, Tcover1.3.4.15	Tcover1.3.4.1, Tcover1.3.4.3, Tcover1.3.4.5, Tcover1.3.4.6, Tcover1.3.4.7, Tcover1.3.4.8, Tcover1.3.4.9, Tcover1.3.4.10, Tcover1.3.4.11, Tcover1.3.4.12, Tcover1.3.4.13, Tcover1.3.4.14, Tcover1.3.4.15
input	now_time = 1, dep_station = '上海虹桥', arrival_station = '南京南', date = datetime('today')	now_time = 1, dep_station = '上海虹桥', arrival_station = '溧阳', date = datetime('today')	now_time = 1, dep_station = '溧阳', arrival_station = '上海虹桥', date = datetime('today')
state	/	/	/
expected output	ans(1).train_id1{1}(1) = 'G' ans(1).train_id1{1}(1) = 'D'	ans(1).train_id1{1}(1) = 'G' , ans(1).train_id1{1}(1) = 'D' , ans(1).whether_transfer = true	E<> ans(i).train_id2{1}(1) = 'G'

- In this expected output, the first line is in least time, the second line is in least price)
- Test Coverage = 15/15 = 100%
- Test Passed = 3

T1.4 Clock.m Test

T1.4.1 pauseClock()

```
function pauseClock(obj)
    if obj.running == 1 %Tcover 1.4.1.1
        stop(obj.clk);
        obj.running = 0;
    end
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.4.1.1	tc T.4.1.2
coverage item	Tcover1.4.1	Tcover1.4.1.1
input	/	/
state	start(clk), running = 1	running = 1
expected state	running = 0	running = 0

- Test Coverage = 1/1 = 100%
- Test Passed = 2

T1.4.2 startClock()

```
function startClock(obj)
    if obj.running == 0 %Tcover 1.4.2.1
        start(obj.clk);
        obj.running = 1;
    end
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.4.2.1	tc T1.4.2.2
coverage item	Tcover1.4.1	Tcover1.4.2.1
input	/	/
state	stop(clk), running = 0	running = 1
expected state	running = 1	running = 1

- Test Coverage = 1/1 = 100%
- Test Passed = 2

T1.5 Account.m Test

T1.5.1 setClock

```
function setClock(obj, clock)
    obj.clock = clock;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.5.1.1
coverage item	Tcover1.5.1
input	clock
state	clock = timer
expected state	account.clock = clock

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.5.2 logout()

```
function obj = logout(obj)
    obj.logInAccountId = -1;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T1.5.2.1
coverage item	Tcover1.5.2
input	/
state	/
expected state	logInAccountId = -1

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.5.3 login()

```
function obj = login(obj, accountId)
            obj.logInAccountId = accountId;
            tableRownum = obj.ticket.getTableRownum();
            findSuccess = 0;
            for i = 1:tableRownum
                if accountId == obj.ticket.getTableInfo(i, 1) %Tcover 1.5.3.1
                    findSuccess = 1;
                    break;
                end
            end

            if findSuccess == 0 %Tcover 1.5.3.2
                obj.ticket.setTableInfo(tableRownum + 1, 1, accountId, 1);
            end
```

```
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.5.3.1	tc T1.5.3.2
coverage item	Tcover1.5.3.1	Tcover1.5.3.2
input	accountId = 201	accountId = 201, 202, 203
state	/	/
expected state	logInAccountId = 201	logInAccountId = 201

- Test Coverage = 2/2 = 100%
- Test Passed = 2

T1.5.4 ticketLeft()

```
function outputArg = ticketLeft(obj, structIn, seat_quality)
            if structIn.price == 0 %Tcover 1.5.4.1
                outputArg = 0;
                return;
            end

            if structIn.departure_time == 0 %Tcover 1.5.4.2
                outputArg = 0;
                return;
            end

            if structIn.whether_transfer == 0 %Tcover 1.5.4.3
                for i = 1:90
                    if (strcmp(obj.database.getData(i, 1),
structIn.departure_station)
                    && obj.database.getData(i, 4) == structIn.departure_time)
                        outputArg = obj.ticket.getTicket(i, seat_quality);
                        iteri = i + 1; %Tcover 1.5.4.4

                        while iteri <= 90
                            if strcmp(obj.database.getData(iteri, 1), \
                            structIn.arrival_station) %Tcover 1.5.4.5
                                break;
                            end


                            if obj.ticket.getTicket(iteri, seat_quality) <
outputArg
                                outputArg = obj.ticket.getTicket(iteri,
seat_quality);
                            end %Tcover 1.5.4.6

                            iteri = iteri + 1;
                        end

                        return;
                    end
                end
```

```
            elseif structIn.whether_transfer == 1 %Tcover 1.5.4.7
                for i = 1:90
                    if (strcmp(obj.database.getData(i, 1),
structIn.departure_station)
                    && obj.database.getData(i, 4) == structIn.departure_time)
                        outputArg = obj.ticket.getTicket(i, seat_quality);
                        iteri = i + 1; %Tcover 1.5.4.8

                        while iteri <= 90
                            if strcmp(obj.database.getData(iteri, 1), \ %Tcover
1.5.4.9
                            structIn.transfer_station)
                                break;
                            end

                            if obj.ticket.getTicket(iteri, seat_quality) <
outputArg
                                outputArg = obj.ticket.getTicket(iteri,
seat_quality);
                            end %Tcover 1.5.4.10

                            iteri = iteri + 1;
                        end

                        break;
                    end
                end

                for j = 1:90
                    if (strcmp(obj.database.getData(j, 1),
structIn.transfer_station)
                    && obj.database.getData(j, 4) == structIn.transfer_time
                    && strcmp(obj.database.getData(j, 2), structIn.train_id2))
                    %Tcover 1.5.4.11
                        if obj.ticket.getTicket(j, seat_quality) < outputArg
                            outputArg = obj.ticket.getTicket(j, seat_quality);
                        end %Tcover 1.5.4.12

                        iterj = j + 1;

                        while iterj <= 90
                            if strcmp(obj.database.getData(iterj, 1), \ %Tcover
1.5.4.13
                            structIn.arrival_station)
                                break;
                            end

                            if obj.ticket.getTicket(iterj, seat_quality) <
outputArg
                                outputArg = obj.ticket.getTicket(iterj,
seat_quality);
                            end %Tcover 1.5.4.14

                            iterj = iterj + 1;

                        end

                        return;
                    end
                end
            end

        end
```

- Coverage Criteria: Branch Coverage

- Test case

	tc T1.5.4.1	tc T1.5.4.2	tc T1.5.4.3	tc T1.5.4.4	tc T1.5.4.5
coverage item	Tcover1.5.4.1	Tcover1.5.4.2	Tcover1.5.4.3, Tcover1.5.4.4. Tcover1.5.4.5, Tcover1.5.4.6	Tcover1.5.4.3, Tcover1.5.4.4. Tcover1.5.4.5, Tcover1.5.4.6	Tcover1.5.4.7, Tcover1.5.4.8. Tcover1.5.4.9, Tcover1.5.4.10, Tcover1.5.4.11, Tcover1.5.4.12, Tcover1.5.4.13, Tcover1.5.4.14
input	Array.price = 0, Array.dep_time = 2	Array.price = 1, Array.dep_time = 0	Array = search('苏州北', '常州北')	Array = search('苏州北', '南京南')	Array = search('苏州北', '湖州')
state	/	/	/	/	/
expected state	outputArg = 0	outputArg = 0	outputArg = ticket.getTicket(1,1)	outputArg = ticket.getTicket(1,1)	outputArg = ticket.getTicket(1,1)

- Test Coverage = 14/14 = 100%
- Test Passed = 5

T1.5.5 tryBuyTicket()

```
function outputArg = tryBuyTicket(obj, structIn, seat_quality)
            if structIn.whether_transfer == 0 % Tcover 1.5.5.1
                for i = 1:90
                    if (strcmp(obj.database.getData(i, 1),
structIn.departure_station)
                    && obj.database.getData(i, 4) == structIn.departure_time)
                    % Tcover 1.5.5.2
                        obj.ticket.setTicket(i, seat_quality, \
                        obj.ticket.getTicket(i, seat_quality) - 1);
                        iteri = i + 1;

                        while iteri <= 90
                            if strcmp(obj.database.getData(iteri, 1), \
                            structIn.arrival_station)
                            % Tcover 1.5.5.3
                                break;
                            end

                            obj.ticket.setTicket(iteri, seat_quality, \
                            obj.ticket.getTicket(iteri, seat_quality) - 1);

                            iteri = iteri + 1;
                        end
                        outputArg = 1;
                        return;
                    end
                end

            elseif structIn.whether_transfer == 1 % Tcover 1.5.5.4
                for i = 1:90
                    if (strcmp(obj.database.getData(i, 1),
structIn.departure_station)
                    && obj.database.getData(i, 4) == structIn.departure_time)
                    % Tcover 1.5.5.5
                        obj.ticket.setTicket(i, seat_quality,\
                        obj.ticket.getTicket(i, seat_quality) - 1);
                        iteri = i + 1;

                        while iteri <= 90
```

```
                            if strcmp(obj.database.getData(iteri, 1),\
                            structIn.transfer_station)
                            % Tcover 1.5.5.6
                                break;
                            end

                            obj.ticket.setTicket(iteri, seat_quality, \
                            obj.ticket.getTicket(iteri, seat_quality) - 1);

                            iteri = iteri + 1;
                        end

                        break;
                    end
                end

                for j = 1:90
                    if (strcmp(obj.database.getData(j, 1),
structIn.transfer_station)
                    && obj.database.getData(j, 4) == structIn.transfer_time
                    && strcmp(obj.database.getData(j, 2), structIn.train_id2))
                    % Tcover 1.5.5.7
                        obj.ticket.setTicket(j, seat_quality,\
                        obj.ticket.getTicket(j, seat_quality) - 1);

                        iterj = j + 1;

                        while iterj <= 90
                            if strcmp(obj.database.getData(iterj, 1), \
                            % Tcover 1.5.5.8
                            structIn.arrival_station)
                                break;
                            end

                            obj.ticket.setTicket(iterj, seat_quality,\
                            obj.ticket.getTicket(iterj, seat_quality) - 1);

                            iterj = iterj + 1;

                        end
                        outputArg = 1;
                        return;
                   end
                end

            end

 end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.5.5.1	tc T1.5.5.2	tc T1.5.5.3
coverage item	Tcover1.5.5.1, Tcover1.5.5.2, Tcover1.5.5.3	Tcover1.5.5.1, Tcover1.5.5.2, Tcover1.5.5.3	Tcover1.5.5.4, Tcover1.5.5.5, Tcover1.5.5.6, Tcover1.5.5.7, Tcover1.5.5.8
input	Array = search('苏州北', '常州北')	Array = search('苏州北', '南京南')	Array = search('苏州北', '湖州')
state	login(200)	login(200)	login(200)
expected state	outputArg = 1	outputArg = 1	outputArg = 1

- Test Coverage = 8/8 = 100%
- Test Passed = 3

T1.5.6 buyTicket()

```
function success = buyTicket(obj, structIn, seat_quality)
            if obj.logInAccountId == -1 %Tcover 1.5.6.1
                success = 0;
                return;
            end

            tableRownum = obj.ticket.getTableRownum();
            for i = 1:tableRownum
                if obj.logInAccountId == obj.ticket.getTableInfo(i, 1)
                    %Tcover 1.5.6.2
                    if obj.ticketLeft(structIn, seat_quality) < 1 %Tcover
1.5.6.3

                        success = 0;
                        msgbox('Error! No Enough Tickets!', 'error');
                        return;
                    end

                    if obj.ticket.getTableInfo(i, 2) == 1 %Tcover 1.5.6.4
                        success = 0;
                        msgbox('Error! Cannot Buy Two Tickets!', 'error');
                        return;
                    end
                    obj.tryBuyTicket(structIn, seat_quality);
                    obj.ticket.setTableInfo(i, 2, 1, 0);
                    obj.ticket.setTableInfo(i, 3, structIn.departure_time, 0);
                    obj.ticket.setTableInfo(i, 4, structIn.arrival_time, 0);
                    obj.ticket.setTableInfo(i, 5, structIn.interval_time, 0);
                    obj.ticket.setTableInfo(i, 6, {structIn.departure_station},
0);
                    obj.ticket.setTableInfo(i, 7, {structIn.arrival_station},
0);
                    obj.ticket.setTableInfo(i, 8, {structIn.train_id1}, 0);
                    obj.ticket.setTableInfo(i, 9, structIn.whether_transfer, 0);
                    obj.ticket.setTableInfo(i, 10, {structIn.transfer_station},
0);
                    obj.ticket.setTableInfo(i, 11, structIn.price, 0);
                    obj.ticket.setTableInfo(i, 12, seat_quality, 0);
                    obj.ticket.setTableInfo(i, 13, structIn.transfer_time, 0);
                    obj.ticket.setTableInfo(i, 14, {structIn.train_id2}, 0);
```

```
                    success = 1;
                    return;
                end
            end
            success = 0;
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.5.6.1	tc T1.5.6.2	tc T1.5.6.3
coverage item	Tcover1.5.6.1	Tcover1.5.6.2, Tcover1.5.6.4	Tcover1.5.6.2, Tcover1.5.6.3
input	Array = search('苏州北', '湖州')	Array = search('苏州北', '湖州')	Array = search('苏州北', '湖州')
state	logout()	login(200)	login(200)
expected state	outputArg = 0	outputArg = 0	outputArg = 0

- Test Coverage = 8/8 = 100%
- Test Passed = 3

T1.5.7 consultTicket()

```
function outputArg = consultTicket(obj, input)
            tableRownum = obj.ticket.getTableRownum();
             for i = 1:tableRownum
                if obj.logInAccountId == obj.ticket.getTableInfo(i, 1) %T1.5.7.1
                    outputArg = obj.ticket.getTableInfo(i, input);
                end
             end
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.5.7.1
coverage item	Tcover1.5.7.1
input	1
state	login(200)
expected state	outputArg = 200

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T1.5.8 tryCancelTicket()

```
function success = tryCancelTicket(obj)
            if obj.logInAccountId == -1 %Tcover 1.5.8.1
                success = 0;
                return;
            end

            tableRownum = obj.ticket.getTableRownum();
            for i = 1:tableRownum
                if obj.logInAccountId == obj.ticket.getTableInfo(i, 1) %Tcover
1.5.8.2
                    if obj.ticket.getTableInfo(i, 2) == 0 %Tcover 1.5.8.3
                        success = 0;
                        msgbox('No Ticket To Cancel!');
                        return;
                    end
                    whether_transfer = obj.ticket.getTableInfo(i, 9);
                    departure_station = obj.ticket.getTableInfo(i, 6);
                    departure_time = obj.ticket.getTableInfo(i, 3);
                    transfer_station = obj.ticket.getTableInfo(i, 10);
                    transfer_time = obj.ticket.getTableInfo(i, 13);
                    arrival_station = obj.ticket.getTableInfo(i, 7);
                    seat_quality = obj.ticket.getTableInfo(i, 12);
                    train_id2 = obj.ticket.getTableInfo(i, 14);
                end
            end

            if whether_transfer == 0 && obj.clock.save_time < departure_time
            %Tcover 1.5.8.4
                for i = 1:90
                    if (strcmp(obj.database.getData(i, 1), departure_station)
                    && obj.database.getData(i, 4) == departure_time)
                    %Tcover 1.5.8.5
                        obj.ticket.setTicket(i, seat_quality, \
                        obj.ticket.getTicket(i, seat_quality) + 1);
                        iteri = i + 1;

                        while iteri <= 90
                            if strcmp(obj.database.getData(iteri, 1),
arrival_station{1})
                                break;
                            end %Tcover 1.5.8.6

                            obj.ticket.setTicket(iteri, seat_quality, \
                            obj.ticket.getTicket(iteri, seat_quality) + 1);

                            iteri = iteri + 1;
                        end
                        success = 1;
                        break;
                    end
                end

            elseif whether_transfer == 1  %Tcover 1.5.8.7
            && obj.clock.save_time < departure_time
                for i = 1:90
                    if (strcmp(obj.database.getData(i, 1), departure_station)
                    && obj.database.getData(i, 4) == departure_time)
                    %Tcover 1.5.8.8
                        obj.ticket.setTicket(i, seat_quality, \
                        obj.ticket.getTicket(i, seat_quality) + 1);
                        iteri = i + 1;
```

```
                        while iteri <= 90
                            if strcmp(obj.database.getData(iteri, 1), \
                            transfer_station{1})
                            %Tcover 1.5.8.9
                                break;
                            end

                            obj.ticket.setTicket(iteri, seat_quality, \
                            obj.ticket.getTicket(iteri, seat_quality) + 1);

                            iteri = iteri + 1;
                        end

                        break;
                    end
                end

                for j = 1:90
                    if (strcmp(obj.database.getData(j, 1), transfer_station{1})
                    && obj.database.getData(j, 4) == transfer_time
                    && strcmp(obj.database.getData(j, 2), train_id2{1}))
                    %Tcover 1.5.8.10
                        obj.ticket.setTicket(j, seat_quality, \
                        obj.ticket.getTicket(j, seat_quality) + 1);

                        iterj = j + 1;

                        while iterj <= 90
                            if strcmp(obj.database.getData(iterj, 1),
arrival_station{1})
                                break;
                            end %Tcover 1.5.8.11

                            obj.ticket.setTicket(iterj, seat_quality, \
                            obj.ticket.getTicket(iterj, seat_quality) + 1);

                            iterj = iterj + 1;

                        end

                        break;
                    end
                end
            end


            for i = 1:tableRownum
                if obj.logInAccountId == obj.ticket.getTableInfo(i, 1) %Tcover
1.5.8.12
                    obj.ticket.setTableInfo(i, 1, 0, 0);
                    obj.ticket.setTableInfo(i, 2, 0, 0);
                    TableRownum = obj.ticket.getTableRownum();
                    obj.ticket.setTableInfo(TableRownum + 1, 1,
obj.logInAccountId, 1);
                    success = 1;
                end
            end

        end


    end
end
```

- Coverage Criteria: Branch Coverage

- Test case

	tc T1.5.4.1	tc T1.5.4.2	tc T1.5.4.3	tc T1.5.4.4	tc T1.5.4.5
coverage item	Tcover1.5.8.1	Tcover1.5.8.2, Tcover1.5.8.3	Tcover1.5.8.2， Tcover1.5.8.7, Tcover1.5.8.8. Tcover1.5.8.9, Tcover1.5.8.10, Tcover1.5.8.11, Tcover1.5.8.12,	Tcover1.5.8.2, Tcover1.5.8.4. Tcover1.5.8.5, Tcover1.5.8.6，Tcover1.5.8.12	Tcover1.5.8.2， Tcover1.5.8.7, Tcover1.5.8.8. Tcover1.5.8.9, Tcover1.5.8.10, Tcover1.5.8.11, Tcover1.5.8.12,
input	/	/	/	/	/
state	logout()	no_ticket	ticket = ('苏州北', '湖州')	ticket = ('苏州北', '常州北')	ticket = ('苏州北', '南京南')
expected state	outputArg = 0	outputArg = 0	outputArg = 1	outputArg = 1	outputArg = 1

- Test Coverage = 12/12 = 100%
- Test Passed = 5

T1.6 UI Test

T1.6.1 convertSeatQuality()

```
function outputArg = convertSeatQuality(~, input)
            if strcmp(input, '一等座') %Tcover 1.6.1.1
                outputArg = 1;
            elseif strcmp(input, '二等座') %Tcover 1.6.1.2
                outputArg = 2;
            end
end
```

- Coverage Criteria: Branch Coverage
- Test case

	tc T1.6.1.1	tc T1.6.1.2
coverage item	Tcover1.6.1.1	Tcover1.6.1.2
input	一等座	二等座
state	/	/
expected output	outputArg = 1	outputArg = 2

- Test Coverage = 2/2 = 100%
- Test Passed = 2

T2 Functional Test

T2.1 Cancel Ticket

T2.1.1 Cancel Ticket Test

```
function cancelTest(tc)

            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.login);
            tc.press(tc.ui.MyTicket);
            tc.press(tc.ui.Button_12);
            tc.press(tc.ui.Button);
            tc.press(tc.ui.Buy_Button_1);
            tc.press(tc.ui.buyAssure);
            tc.press(tc.ui.successButton);
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 0);
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.MyTicket);
            tc.press(tc.ui.tryCancelTicket);
            tc.press(tc.ui.cancelTicket);
            tc.press(tc.ui.successButton);

            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 2
                    success = 1;
                end
            end
            tc.verifyEqual(success, 1);
            tc.clock.delete_timer();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.1.1
coverage item	Tcover2.1.1
input	press login -> press frontpage -> press search -> press buyticket1 -> press confirm -> cancel ticket
state	login
expected output	buy ticket success = 1, cancel ticket success (browse database)

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T2.2 Reschedule Ticket

T2.2.1 Reschedule Ticket Test

```
function rescheduleTest(tc)
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.login);
            tc.press(tc.ui.MyTicket);
            tc.press(tc.ui.Button_12);
            tc.press(tc.ui.Button);
            tc.press(tc.ui.Buy_Button_1);
            tc.press(tc.ui.buyAssure);
            tc.press(tc.ui.successButton);
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 0);
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.MyTicket);
            tc.press(tc.ui.RescheduleButton);
            tc.press(tc.ui.Button_re_search);
            tc.press(tc.ui.Buy_Button_6);
            tc.press(tc.ui.buyAssure_2);
            tc.press(tc.ui.successButton);
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 1;
                    break;
                end
            end
            tc.verifyEqual(success, 1);
            tc.clock.delete_timer();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.2.1
coverage item	Tcover2.2.1
input	press login -> press frontpage -> press search -> press buyticket1 -> press confirm -> reschedule ticket
state	login
expected output	buy ticket success = 1, reschedule ticket success (browse database)

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T2.3 Clock

T2.3.1 Clock Test

```
function clockTest(tc)
            tc.press(tc.ui.timeStop);
            tc.verifyEqual(tc.clock.running, 0);
            tc.press(tc.ui.timeContinue);
            tc.verifyEqual(tc.clock.running, 1);
            tc.clock.delete_timer();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.3.1
coverage item	Tcover2.3.1
input	stop clock -> start clock
state	/
expected output	clock.running = 0 -> clock.running = 1

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T2.3.2 Train State Test

```
function trainStateTest(tc)
            tc.ui.EditField_3.Value = 100;
            tc.press(tc.ui.setTimeSpeed);
            pause(20);
            tc.clock.delete_timer();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.3.2
coverage item	Tcover2.3.2
input	/
state	Timespeed = 100
expected output	train ID emerge from departure station and goes to arrival station as the schedule and stays at station for 3 min and stay on way for 27 min each step. Difference are existed in each direction on the same railway.

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T2.3.3 Pass Day Test

```
function passDayTest(tc)
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.login);
            tc.press(tc.ui.MyTicket);
            tc.press(tc.ui.Button_12);
            tc.press(tc.ui.Button);
            tc.press(tc.ui.Buy_Button_1);
            tc.press(tc.ui.buyAssure);
            tc.press(tc.ui.successButton);
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 0);
            tc.ui.EditField_3.Value = 100;
            tc.press(tc.ui.setTimeSpeed);
            pause(30);
            success = 1;
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 1);
            tc.clock.delete_timer();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.3.3
coverage item	Tcover2.3.3
input	press login -> press frontpage -> press search -> press buyticket1 -> press confirm -> wait 30s for next day
state	/
expected state	buy ticket success = 1, account has no ticket (browse database)

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T2.3.4 Board Test

```
function boardTest(tc)
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.login);
            tc.press(tc.ui.MyTicket);
            tc.press(tc.ui.Button_12);
            tc.press(tc.ui.Button);
            tc.press(tc.ui.Buy_Button_1);
            tc.press(tc.ui.buyAssure);
            tc.press(tc.ui.successButton);
            success = 0;
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 0);
            tc.ui.EditField_3.Value = 100;
            tc.press(tc.ui.setTimeSpeed);
            pause(10);
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 0);
            pause(20);
            tc.clock.delete_timer();
            tc.ticket.saveInfo();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.3.4
coverage item	Tcover2.3.4
input	press login -> press frontpage -> press search -> press buyticket1 -> press confirm -> wait 10s for expire
state	/
expected state	buy ticket success = 1, no ticket, seat not released (browse database)

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T2.3.5 Unboard Test

```
function unboardTest(tc)
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.login);
            tc.press(tc.ui.MyTicket);
            tc.press(tc.ui.Button_12);
            tc.press(tc.ui.Button);
            tc.press(tc.ui.Buy_Button_1);
            tc.press(tc.ui.buyAssure);
            tc.press(tc.ui.successButton);
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 0);
            tc.ui.boardButton.Value = 0;
            tc.ui.EditField_3.Value = 100;
            tc.press(tc.ui.setTimeSpeed);
            pause(20);
            success = 1;
            for i = 1:90
                if tc.ticket.getTicket(i, 2) == 1
                    success = 0;
                    break;
                end
            end
            tc.verifyEqual(success, 1);
            pause(10);
            tc.clock.delete_timer();
            tc.ticket.saveInfo();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.3.5
coverage item	Tcover2.3.5
input	press login -> press frontpage -> press search -> press buyticket1 -> press confirm -> wait 20s for expire
state	/
expected state	buy ticket success = 1, account has ticket, seat released (browse database)

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T2.3.6 Concurrency Test

```
function multiTest(tc)

            pause(7);
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.login);
            tc.press(tc.ui.Button_12);
            tc.press(tc.ui.Button);
            tc.press(tc.ui.Buy_Button_1);
            tc.press(tc.ui.buyAssure);
            tc.press(tc.ui.successButton);

            pause(5);
            tc.press(tc.ui1.frontPageMine);
            tc.ui1.accountId.Value = 114515;
            tc.press(tc.ui1.login);
            tc.press(tc.ui1.Button_12);
            tc.press(tc.ui1.Button);
            pause(5);
            tc.press(tc.ui.frontPageMine);
            tc.press(tc.ui.logout);
            tc.ui.accountId.Value = 114516;
            tc.press(tc.ui.login);
            tc.press(tc.ui.Button_12);
            tc.press(tc.ui.Button);
            tc.press(tc.ui.Buy_Button_1);
            tc.press(tc.ui.buyAssure);
            tc.press(tc.ui.successButton);

            pause(5);
            tc.press(tc.ui1.Buy_Button_1);
            tc.press(tc.ui1.buyAssure);
            pause(5);

            tc.clock.time = 1430;
            tc.ui.EditField_3.Value = 100;
            pause(1);
            tc.press(tc.ui.setTimeSpeed);
            pause(1);

            tc.clock1.time = 1430;
```

```
            tc.ui1.EditField_3.Value = 100;
            pause(1);
            tc.press(tc.ui1.setTimeSpeed);
            pause(1);

            tc.clock.delete_timer();
            tc.clock1.delete_timer();
            tc.ticket.saveInfo();
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T2.3.6
coverage item	Tcover2.3.6
input	buy 2 tickets -> buy 1 ticket
state	/
expected state	no ticket

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T3 Intergration Test

T3.1 Test

T3.1.1 Single UI Test

```
function intergration(tc)
            pause(7);
            tc.press(tc.ui.frontPageMine);
            pause(3);
            tc.press(tc.ui.login);
            pause(3);
            tc.press(tc.ui.MyTicket);
            pause(2);
            tc.press(tc.ui.Button_13);
            pause(2);
            tc.press(tc.ui.Button_12);
            pause(3);
            tc.press(tc.ui.Button);
            pause(7);
            tc.press(tc.ui.Buy_Button_1);
            pause(19);
            tc.press(tc.ui.Button_15);
            pause(2);
            tc.press(tc.ui.Button);
            pause(7);
            tc.press(tc.ui.Buy_Button_1);
            pause(5);
            tc.press(tc.ui.buyAssure);
            pause(3);
```

```
tc.press(tc.ui.successButton);

pause(15);
tc.press(tc.ui.frontPageMine);
pause(2);
tc.press(tc.ui.MyTicket);
pause(6);
tc.press(tc.ui.tryCancelTicket);
pause(2);
tc.press(tc.ui.cancelTicket);
pause(2);
tc.press(tc.ui.successButton);
pause(1);
tc.press(tc.ui.Button);
pause(7);
tc.press(tc.ui.Buy_Button_1);
pause(5);
tc.press(tc.ui.buyAssure);
pause(2);
tc.press(tc.ui.successButton);


pause(2);
tc.press(tc.ui.frontPageMine);
pause(2);
tc.press(tc.ui.MyTicket);
pause(2);
tc.press(tc.ui.RescheduleButton);
pause(3);
tc.press(tc.ui.Button_re_search);
pause(3);
tc.press(tc.ui.Buy_Button_6);
pause(2);
tc.press(tc.ui.buyAssure_2);
pause(2);
tc.press(tc.ui.successButton);
pause(2);
tc.press(tc.ui.frontPageMine);
pause(2);
tc.press(tc.ui.MyTicket);
pause(2);
tc.press(tc.ui.ticketBackToMine);

pause(2);
tc.ui.EditField_3.Value = 100;
pause(1);
tc.press(tc.ui.setTimeSpeed);
pause(3);
tc.ui.EditField_3.Value = 10;
pause(1);
tc.press(tc.ui.setTimeSpeed);
pause(30);
tc.press(tc.ui.MyTicket);
pause(2);
tc.ui.EditField_3.Value = 100;
pause(1);
tc.press(tc.ui.setTimeSpeed);
pause(8);
tc.ui.EditField_3.Value = 1;
pause(1);
tc.press(tc.ui.setTimeSpeed);
pause(1);
tc.press(tc.ui.Button_13);
pause(3);

tc.press(tc.ui.Button_12);
```

```
            pause(2);
            tc.press(tc.ui.Button);
            pause(2);
            tc.press(tc.ui.Buy_Button_1);
            pause(2);
            tc.press(tc.ui.buyAssure);
            pause(2);
            tc.press(tc.ui.successButton);
            pause(2);
            tc.press(tc.ui.frontPageMine);
            pause(2);
            tc.press(tc.ui.MyTicket);
            pause(2);
            tc.ui.boardButton.Value = 0;
            pause(3);
            tc.press(tc.ui.ticketBackToMine);
            pause(2);
            tc.ui.EditField_3.Value = 100;
            pause(1);
            tc.press(tc.ui.setTimeSpeed);
            pause(8);
            tc.press(tc.ui.timeStop);
            pause(10);
            tc.press(tc.ui.MyTicket);
            pause(1);
            tc.press(tc.ui.tryCancelTicket);
            pause(2);
            tc.press(tc.ui.cancelTicket);
            pause(2);
            tc.press(tc.ui.successButton);
            pause(1);

            stop(tc.clock.clk);
            stop(tc.clock.clk_table);
            delete(tc.clock.clk);
            delete(tc.clock.clk_table);
            close all force;
end
```

- Coverage Criteria: State Coverage
- Test case

	tc T3.1.1
coverage item	Tcover3.1.1
input	press Mine ->press login -> press myTicket -> press confirm -> press FrontPage -> press search -> press buyTicket1 -> wait 15 seconds -> press backToFrontPage -> press search -> press buyTicket1 -> press confirm -> press backToFrontPage -> press Mine -> press Mine Ticket -> press cancelTicket -> press confirm -> press back -> press search -> press buyTicket1 -> press confirm -> press backToFrontPage -> press Mine -> press myTicket -> press reschedule -> press search -> press buyTicket2 -> press confirm -> press backToFrontPage -> press Mine -> press MyTicket -> press back -> set EditField_timespeed to 100 -> press set -> press myTicket -> set EditField_timespeed to 10 -> press set -> set EditField_timespeed to 100 -> press set -> press myTicket -> set EditField_timespeed to 11 -> press set -> press confirm -> press frontPage -> press search -> press buyTicket1 -> press confirm -> press backToFrontPage -> press Mine -> press myTicket -> unpress board -> press back -> set EditField_timespeed to 100 -> press set -> press timeStop -> press myTicket -> press cancel -> press confirm
state	No Account, No Ticket
expected output	Successfully trigger timeout -> trigger buy ticket -> trigger cancel ticket -> trigger reschedule ticket -> press or unpress board button ticket -> train graph and table -> timer passes a day

- Test Coverage = 1/1 = 100%
- Test Passed = 1

T4 Risk Management

M1 User buy a multi-station ticket

Risk Analysis

- Hazard Situation: 12306 doesn't lock all seats along one's routine
- Possible Cause: User buy a multi-station ticket

Risk Evaluation

- For a system, it is catastrophic and may cause no enough seats.

Risk Control

- In our system, we will check and lock the seats along the way in our implementation.

M2 User not log in

Risk Analysis

- Hazard Situation: 12306 can't buy ticket.
- Possible Cause: User not log in

Risk Evaluation

- For a system, it may sometimes encounter this situation.

Risk Control

- In our system, we set msgbox for this condition and it is designed to search without login.

M3 No Enough Tickets

Risk Analysis

- Hazard Situation: 12306 doesn't have enough tickets for two requests
- Possible Cause: User buy too much tickets

Risk Evalution

- For a ticket system, this is likely to happen

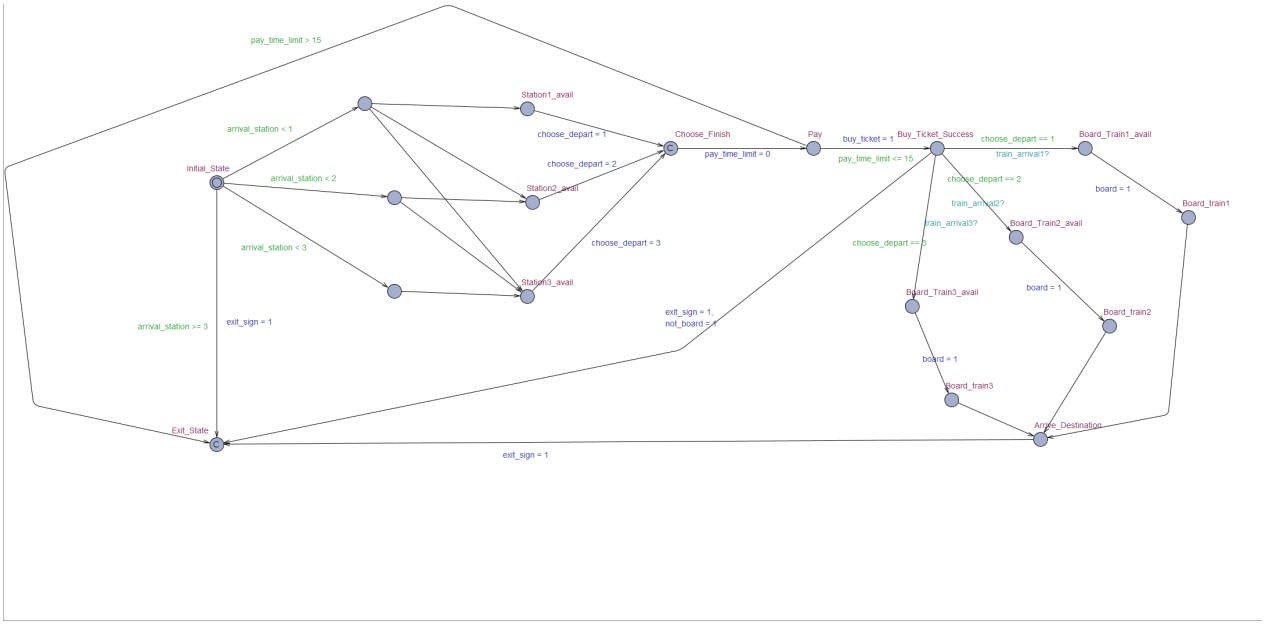
Risk Control

- Since matlab only use one core and is't actually parallel, so the early one will get te ticket and the rest will only receive a not enough ticket msgbox.

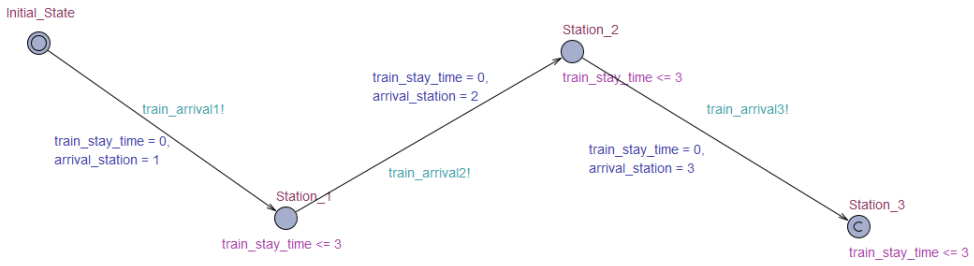
T5 UPPAAL

T5.1 Model

T5.1.1 App Model



T5.1.2 Train Model



T5.2 Model Properties Test

- One can buy a ticket.
- One can cancel a ticket.
- One can arrrive Station_1.
- One can arrrive Station_2.
- One can arrrive Station_3.
- One can pay for a ticket.
- All trains won't stay for longer than 3 minutes at a station.
- Train can arrive Station_1.
- Train can arrive Station_2.
- Train can arrive Station_3.
- One can board a train.
- One can not board a train.
- At all circumstance there won't be a ticket to Station_2 arrives at Station_3.
- At all circumstance there won't be a ticket to Station_1 arrives at Station_3.
- There exists a ticket to Station_3 that arrives at station 3.
- There exists a ticket to Station_3 that arrives at station 2.
- There exists a ticket to Station_3 that arrives at station 1.
- There exists a ticket to Station_2 that arrives at station 2.

- There exists a ticket to Station_2 that arrives at station 1.
- There exists a ticket to Station_1 that arrives at station 1.
- There exists a path to exit status.