

# Unit Test Report for PTS System

- Test Cases Specification
- Test Summary Report

**Project Team**

**Team 4**

Latest update on:

**2014-11-20**

---

**노은방 200811428**

**김상민 200910044**

**박수민 201111353**

**한별 201214217**

## Table of Contents

1	Introduction .....	3
1.1	Objectives.....	3
1.2	References.....	3
2	Unit test case specification.....	3
2.1	Test case specification identifier.....	3
2.2	Test items .....	3
2.3	Input specifications.....	3
2.4	Output specifications.....	4
3	Environmental needs.....	4
4	Unit test summary report .....	4
4.1	Test summary report identifier.....	4
4.2	Evaluation.....	14

## 1 Introduction

### 1.1 Objectives

본 문서는 2014년 2학기 소프트웨어 공학 개론 수업의 T4가 개발한 Public Transportation System (PTS)을 Unit Testing 결과 문서이다.

### 1.2 References

T4\_UTP\_ver1

## 2 Unit test case specification

### 2.1 Test case specification identifier

< Table 5 Test Case Identification-버스 >

<Table 6 Test Case Identification-지하철 >

<Table 7 Test Case Identification-정산 >

#### 참조

### 2.2 Test items

< Table 5 Test Case Identification-버스 >

<Table 6 Test Case Identification-지하철 >

<Table 7 Test Case Identification-정산 >

#### 참조

### 2.3 Input specifications

< Table 5 Test Case Identification-버스 >

<Table 6 Test Case Identification-지하철 >

<Table 7 Test Case Identification-정산 >

#### 참조

2.4 Output specifications

< Table 5 Test Case Identification-버스 >

< Table 6 Test Case Identification-지하철 >

< Table 7 Test Case Identification-정산 >

참조

3 Environmental needs

4 Unit test summary report

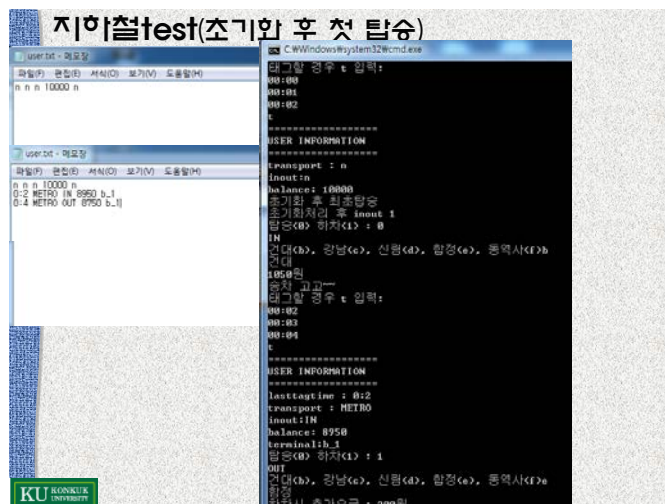
4.1 Test summary report identifier

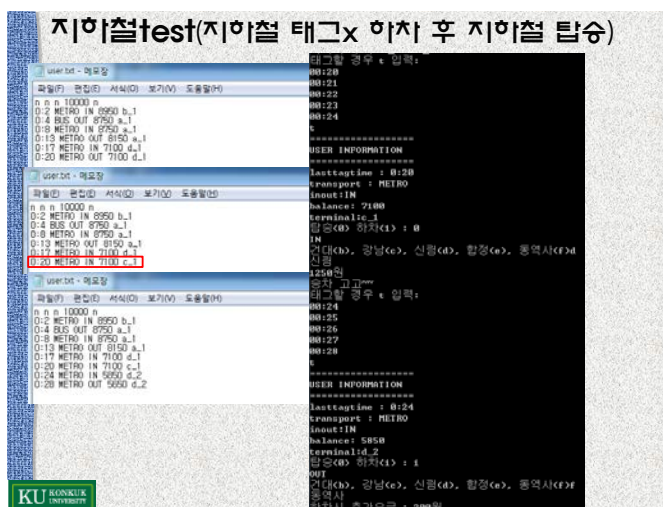
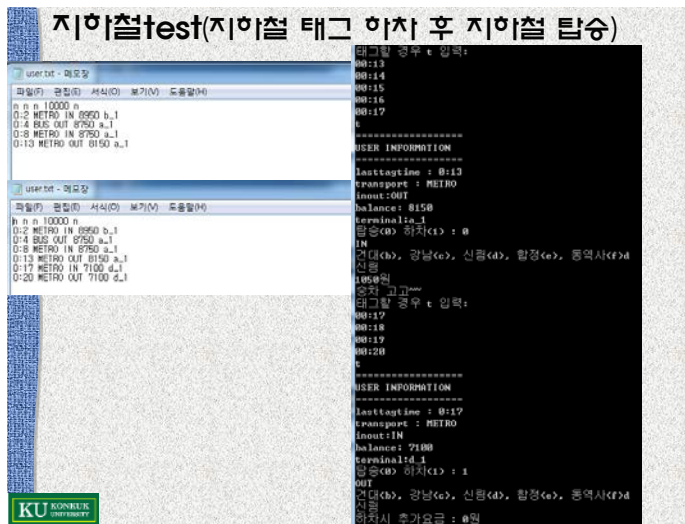
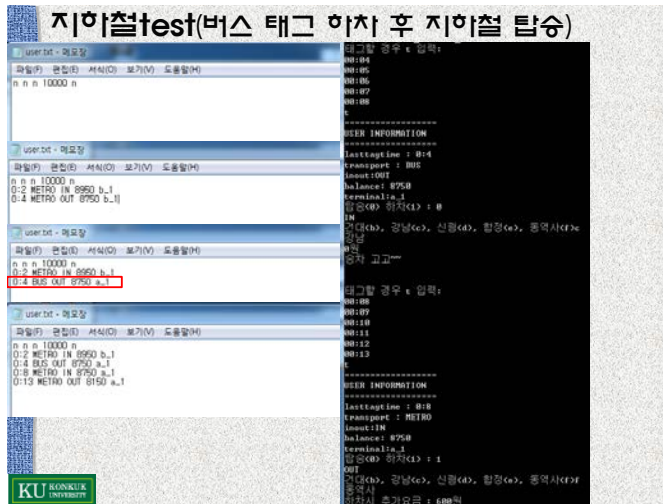
		initialize function?	cleanup function?	Test Count	Active?
	지하철 승차				
Suit	1650원 버스-지하철 환승후 내릴 때 태그 안하고 승차	yes	yes	2	yes
	1750원 지하철-버스 환승후 내릴 때 태그 안하고 승차				
suit	1250원 지하철 승차 지하철 승차	yes	yes	1	yes
suit	1050원 최초 승차	yes	yes	4	yes
	버스 승차 지하철 승차				
	버스 하차 후 환승 시간경과				
	지하철 하차 후 지하철 승차				

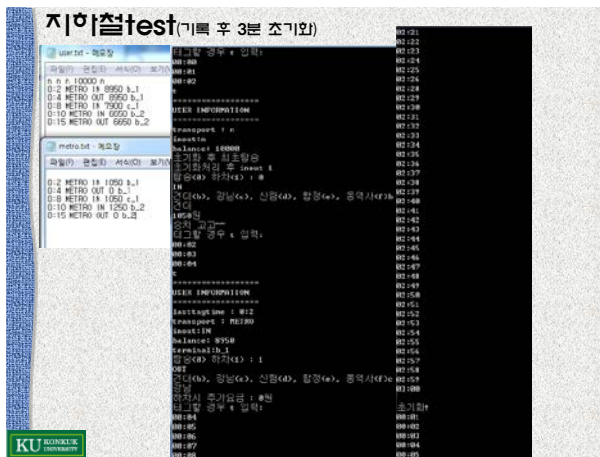
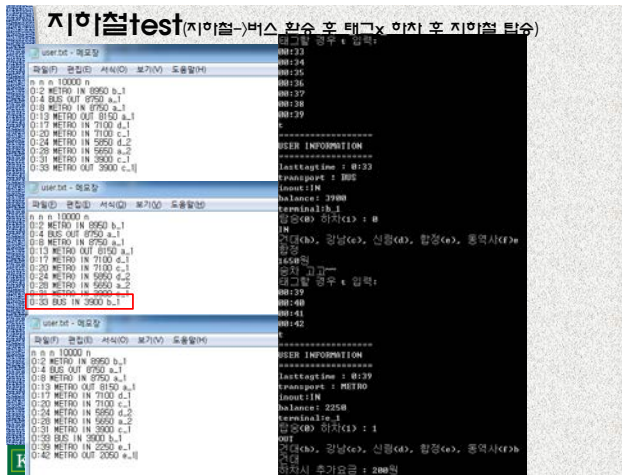
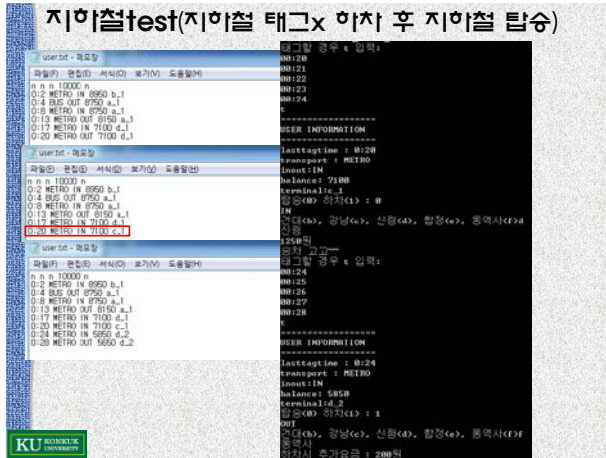
suit	0원 환승	yes	yes	1	yes
	<b>지하철 하차</b>				
suit	0원 환승 아닐 때 0개 정거장	yes	yes	3	yes
	1개 정거장 이동				
	환승일 때 0개 정거장 이동				
suit	200원 환승 아닐 때 : 2개 정거장 이동	yes	yes	1	yes
suit	300원 환승일 때 : 1개 정거장 이동	yes	yes	1	yes
suit	600원 환승일 때 : 2개 정거장 이동	yes	yes	1	yes
	<b>버스 승차</b>				
suit	1650원 버스-지하철 환승후 내릴 때 태그 안하고 탔을때	yes	yes	3	yes
	1750원 지하철-버스 환승후 내릴 때 태그 안하고 탔을때				
	1250원 지하철 승차 버스 승차				
suit	1050원 최초 승차,	yes	yes	1	yes
	환승시간 경과,				

	버스승차 버스승차				
	버스 하차 버스 승차				
suit	0원 환승일 때	yes	yes	1	yes
	<b>버스 하차</b>				
suit	0원 환승인데 시간이 초과 안됐거나 환승 아닐 때	yes	yes	6	yes
	100원				
	200원				
	300원				
	400원				
	500원				

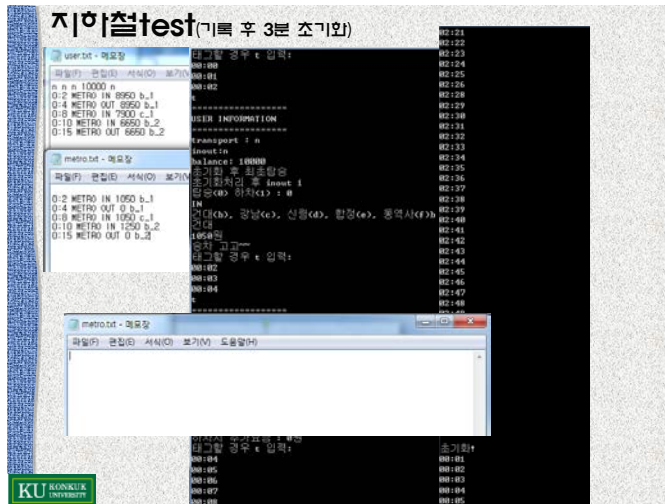
## 지하철



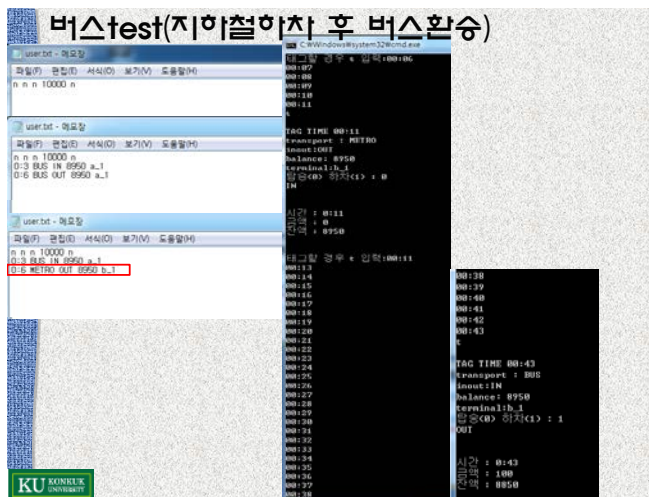
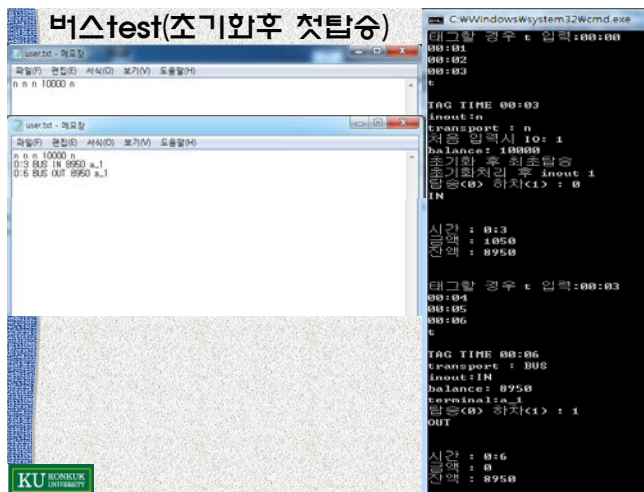


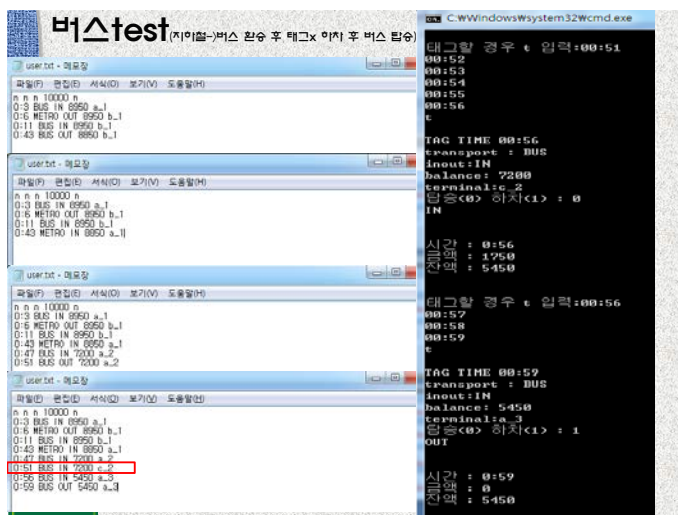
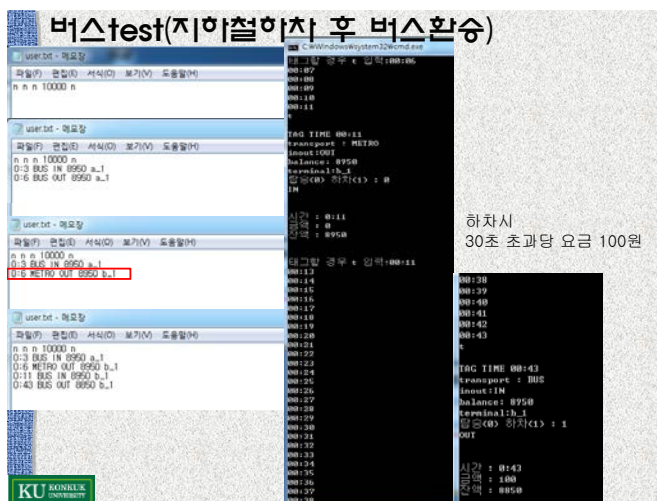
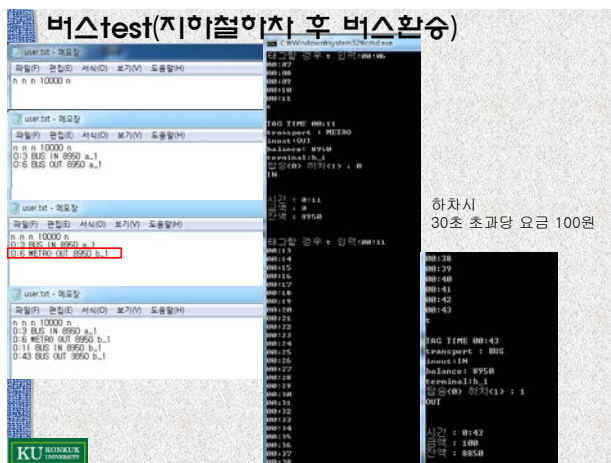


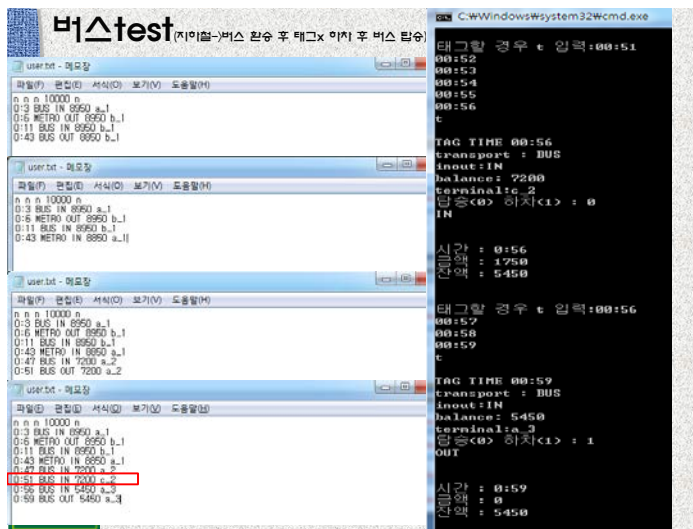
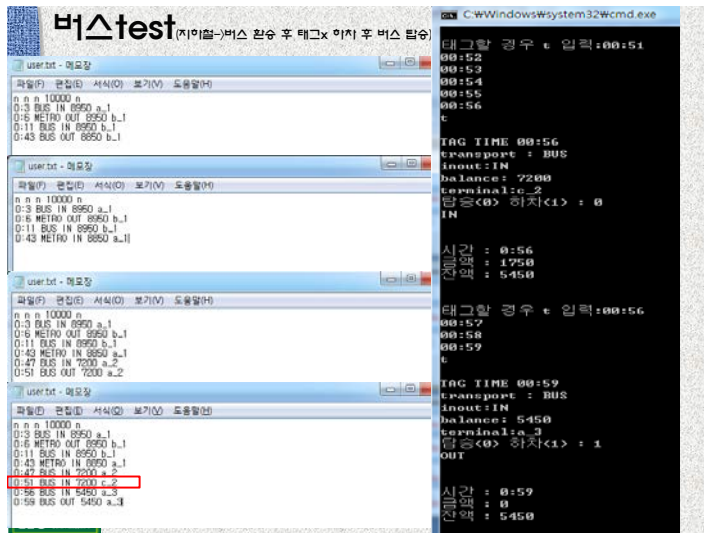
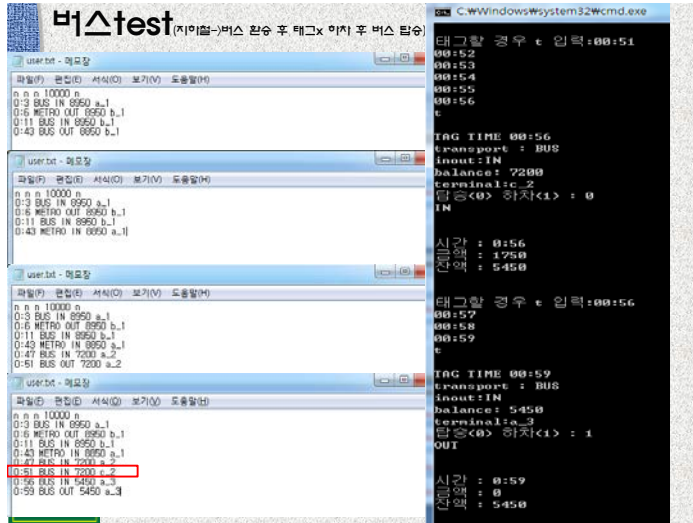




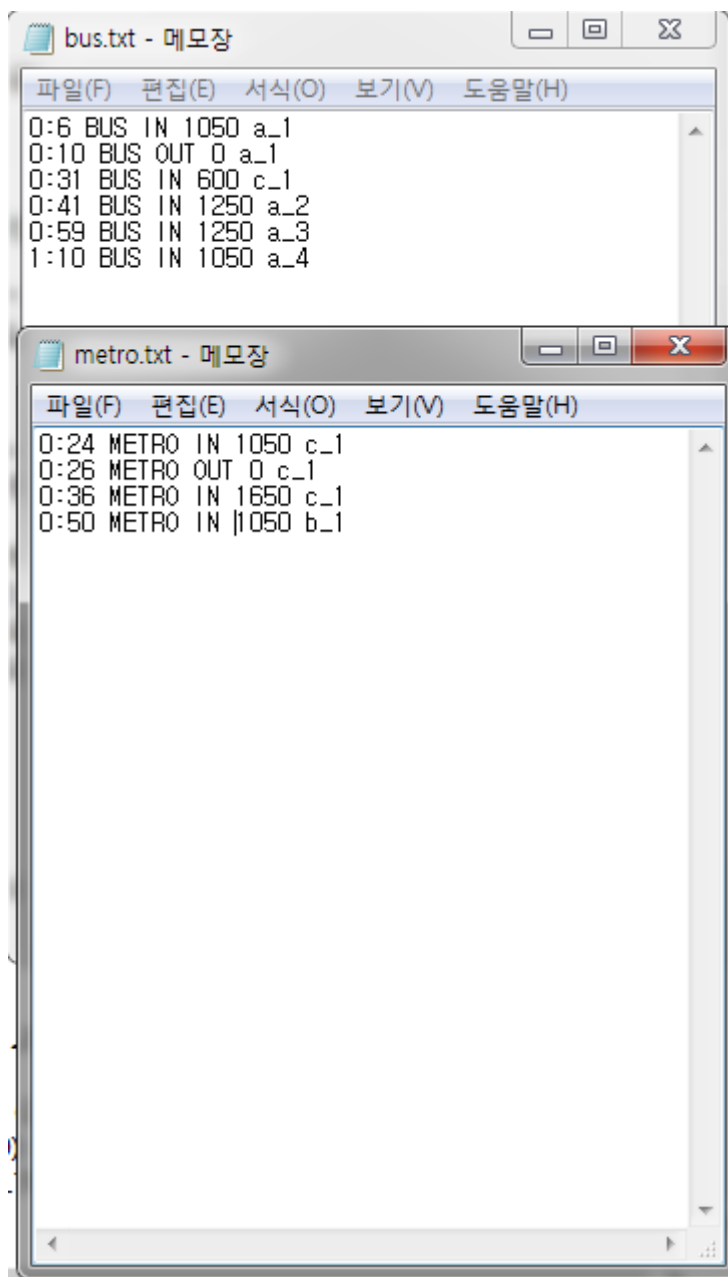
## 버스

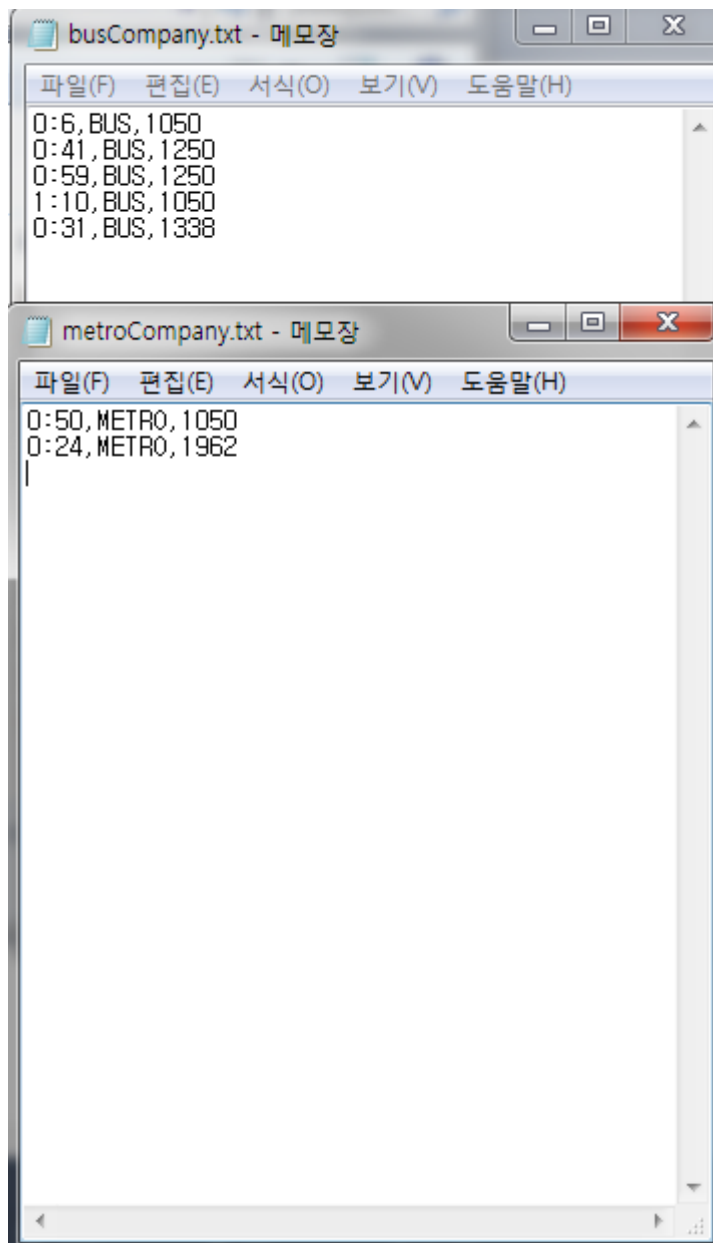


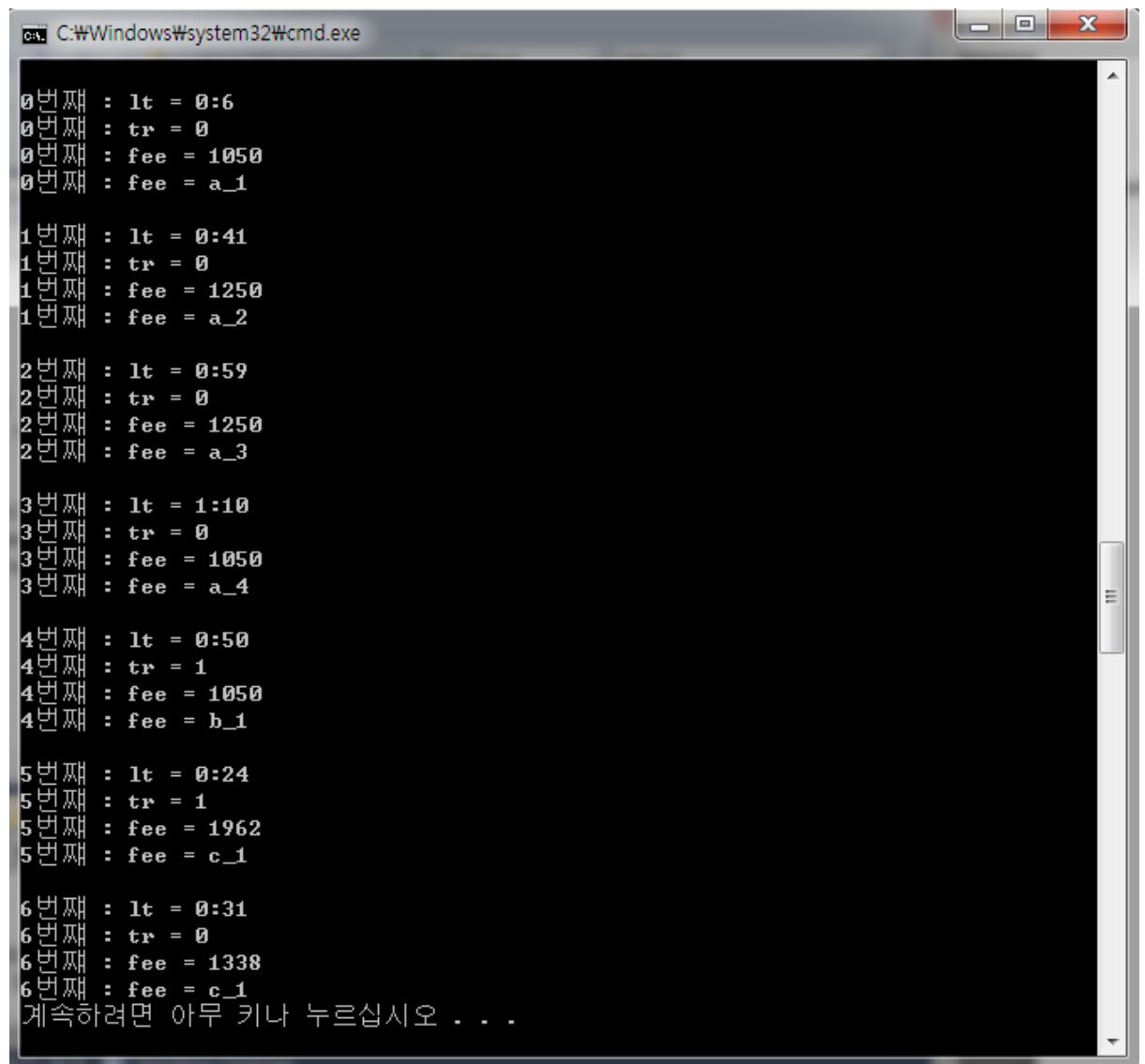




## 정산







```
C:\Windows\system32\cmd.exe

0번째 : lt = 0:6
0번째 : tr = 0
0번째 : fee = 1050
0번째 : fee = a_1

1번째 : lt = 0:41
1번째 : tr = 0
1번째 : fee = 1250
1번째 : fee = a_2

2번째 : lt = 0:59
2번째 : tr = 0
2번째 : fee = 1250
2번째 : fee = a_3

3번째 : lt = 1:10
3번째 : tr = 0
3번째 : fee = 1050
3번째 : fee = a_4

4번째 : lt = 0:50
4번째 : tr = 1
4번째 : fee = 1050
4번째 : fee = b_1

5번째 : lt = 0:24
5번째 : tr = 1
5번째 : fee = 1962
5번째 : fee = c_1

6번째 : lt = 0:31
6번째 : tr = 0
6번째 : fee = 1338
6번째 : fee = c_1
계속하려면 아무 키나 누르십시오 . . .
```

## 4.2 Evaluation



Table 5 Test case Identification-버스

Identifier	Input	Pass/Fail Criteria
PTS.UTC.1000	유저가 태그 하였을 시 유저 카드에 저장된 정보를 입력 받아 저장해주는 부분	
PTS.UTC.1000.00	규모가 큰 Unit.	
PTS.UTC.1100	받은 사용자 카드에서 마지막으로 태그 된 시간 정보를 전달한다.	
PTS.UTC.1100.00	Interfaces는 Testing 제외	
PTS.UTC.1200	받은 사용자 카드에 저장된 마지막으로 태그 된 교통수단 정보를 전달한다.	
PTS.UTC.1200.00	Interfaces는 Testing 제외	
PTS.UTC.1300	받은 사용자 카드에 저장된 마지막으로 태그 된 승.하차 정보를 전달한다.	
PTS.UTC.1300.00	Interfaces는 Testing 제외	
PTS.UTC.1400	받은 사용자 카드에 저장된 잔액 정보를 전달한다.	
PTS.UTC.1400.00	Interfaces는 Testing 제외	
PTS.UTC.1500	받은 사용자 카드에 저장된 마지막으로 태그 된 단말기 정보를 전달한다.	
PTS.UTC.1500.00	Interfaces는 Testing 제외	
PTS.UTC.1600	유저가 태그를 했을 시 유저의 입력을 받은 부분을 모아서 컨트롤 부에서 이용할 수 있는 자료구조로 따로 저장해준다. 또한 태그를 한 시간을 따로 저장해준다.	
PTS.UTC.1600.00	LastTagTime  transportation  in_out  balance	userInfo에 저장, currentInfo에 저장.



	terminalinfo	
PTS.UTC.2000	Tag 정보에서 입력을 받아 정보처리를 한 후 각각 화면에 출력하거나 저장을 한다.	
PTS.UTC.2000.00	규모가 큰 Unit. 단순한 Spec으로 인해 Testing 불가	
PTS.UTC.2100	승객이 승차를 하는지 하차를 하는지에 따라서 case로 나눈다.	
PTS.UTC.2100.00	규모가 큰 Unit. 단순한 Spec으로 인해 Testing 불가	
PTS.UTC.2200	승객이 승차를 했을 때 지불해야 할 요금계산 후 요금의 충분 부족에 따라 각 저장/출력 명령을 출력한다.	
PTS.UTC.2200.00	State == "case 승차"	저장 command ==fputa 출력 command==printf
PTS.UTC.2300	승객이 하차를 했을 때 지불해야 할 요금계산 후 저장/출력 명령을 출력한다.	
PTS.UTC.2300.00	State == "Case 하차"	저장 command ==fputa 출력 command ==printf
PTS.UTC.2400	저장에 대한 명령어를 받아서 각각 지하철과 버스 단말기 정보 저장소 파일에 요금정보를 포함해서 저장을 하고, 승객의 카드정보에도 잔액 정보를 포함해서 저장을 한다.	
PTS.UTC.2400.00	Interface는 Testing 제외	
PTS.UTC.2500	출력에 대한 명령어를 받아서 요금과 현재 시간에 대한 정보를 단말기(여기선 터미널)에 출력한다.	
PTS.UTC.2500.00	Interface는 Testing 제외	
PTS.UTC.1610	유저가 태그를 했을 시 유저의 입력을 받은 부분을 모아서 컨트롤 부에서 이용할 수 있는 자료구조로 따로 저장해준다. 또한 태그 한 시간을 따로 저장해 준다. 그리고 3분마다 초기화 판단 및 시행을 해준다.	

PTS.UTC.1610.00	LastTagTime == ???  Transportation == ???  in_out == ???  balance == ???  terminalinfo == ??? c.time != 03:00	struct userInfo  struct currentInfo  secTime
PTS.UTC.1610.01	LastTagTime == ???  Transportation == ???  in_out == ???  balance == ???  terminalinfo == ??? c.time == 03:00	Trigger"초기화"
PTS.UTC.1620	3분주기마다 시행이 되며 유저카드 이외의 정보를 초기화하는 command를 출력한다.	
PTS.UTC.1620.00	Trigger	"초기화!" 출력
PTS.UTC.2110	승객의 승차, 하차를 구분 짓는 컨트롤로 사용자 정보와 현재시간을 입력 받아 승차인지 하차인지 결정 짓는다.	
PTS.UTC.2110.00	Currentinfo. In_out == 0	Trigger"승차Control"
PTS.UTC.2110.01	Currentinfo. In_out == 1	Trigger"하차Control"

PTS.UTC.2210 실질적으로 요금의 경우를 나누어 주는 주 컨트롤러로, 사용자의 정보를 입력 받고, 환승 여부, 미정산 여부, 최초 탑승여부의 조합에 따라 경우를 나누어 각 계산해야 되는 요금식을 판단해주고 다음 프로세스에 트리거를 전송해준다.		
PTS.UTC.2210.00	Trigger "승차Control" Currentinfo. In_out == 0 Userinfo.In_Out == 1 Userinfo.transportation ==1 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime - userinfo.lasttagtime < 15	Trigger"정산완료&환승"
PTS.UTC.2210.01	Trigger "승차Control" Currentinfo. In_out == 0 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] =='a'	Trigger"미정산&환승"
PTS.UTC.2210.02	Trigger "승차Control" Currentinfo. In_out == 0 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] != 'a'	Trigger"미정산&환승"
PTS.UTC.2210.03	Trigger "승차Control" Currentinfo. In_out == 0 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] != 'a'	Trigger"미정산&환승"

PTS.UTC.2210.04	Trigger "승차Control" Currentinfo.In_out == 0 Userinfo.In_Out == 1 Userinfo.transportation ==0	Enable"기본 요금"
PTS.UTC.2210.05	Trigger "승차Control" Currentinfo.In_out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] == 'a'	Enable"기본 요금"
PTS.UTC.2210.06	Trigger "승차Control" Currentinfo.In_out == 0 Userinfo.In_Out == 1 Userinfo.transportation ==1 Userinfo.terminalinfo[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime > 15	Enable"기본 요금"
PTS.UTC.2220	사용자가 이전에 정산을 하고 환승을 하였을 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.	
PTS.UTC.2220.00	Trigger"정산완료&환승" Currentinfo.In_out == 0 Userinfo.In_Out == 1 Userinfo.transportation ==1 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime - userinfo.lasttagtime < 15	Currentinfo.balance == 0
PTS.UTC.2230	사용자가 최초탑승이거나 이전에 버스를 탔거나 환승시간이 넘은 경우, 기본 금액을 다음 컨트롤러로 전달해 준다.	
PTS.UTC.2230.01	Enable"기본 요금"	Currentinfo.balance == 1050

	<pre>Currentinfo.In_out == 0 Userinfo.In_Out == 1 Userinfo.transportation ==0</pre>	
PTS.UTC.2230.02	<pre>Enable "기본 요금" Currentinfo.In_out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] == 'a'</pre>	Currentinfo.balance == 1050
PTS.UTC.2230.03	<pre>Enable"기본 요금" Currentinfo.In_out == 0 Userinfo.In_Out == 1 Userinfo.transportation ==1 Userinfo.terminalinfo[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime &gt; 15</pre>	Currentinfo.balance == 1050
PTS.UTC.2240	<p>사용자가 미정산을 하고 환승을 안한 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.</p>	
PTS.UTC.2240.00	<pre>Trigger"미정산&amp;환승" Currentinfo.In_out == 0 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] =='a'</pre>	Currentinfo.balance == 1650
PTS.UTC.2240.01	<pre>Trigger"미정산&amp;환승" Currentinfo.In_out == 0 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] != 'a'</pre>	Currentinfo.balance == 1650

PTS.UTC.2240.02	Trigger"미정산&환승" Currentinfo.In_out == 0 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] != 'a'	Currentinfo.balance == 1250
PTS.UTC.2250	전달받은 금액을 유저의 잔액과 비교해서 요금부족 여부를 판단하여 각각 트리거를 전달 해준다.	
PTS.UTC.2250.00	Currentinfo.balance == 0 Userinfo.balance == ???	Userinfo.balance >= 500 Trigger"충분"
PTS.UTC.2250.01	Currentinfo.balance == 0 Userinfo.balance == ???	Userinfo.balance < 500 Enable"부족"
PTS.UTC.2250.02	Currentinfo.balance == 1050 Userinfo.balance == ???	Userinfo.balance >= 1050 Trigger"충분"
PTS.UTC.2250.03	Currentinfo.balance == 1050 Userinfo.balance == ???	Userinfo.balance < 1050 Enable"부족"
PTS.UTC.2250.04	Currentinfo.balance == 1250 Userinfo.balance == ???	Userinfo.balance >= 1250 Trigger"충분"
PTS.UTC.2250.05	Currentinfo.balance == 1250 Userinfo.balance == ???	Userinfo.balance < 1250 Enable"부족"
PTS.UTC.2250.06	Currentinfo.balance == 1650 Userinfo.balance == ???	Userinfo.balance >= 1650 Trigger"충분"
PTS.UTC.2250.07	Currentinfo.balance == 1650 Userinfo.balance == ???	Userinfo.balance < 1650 Enable"부족"
PTS.UTC.2250.08	Currentinfo.balance == 1750 Userinfo.balance == ???	Userinfo.balance >= 1750 Trigger"충분"

PTS.UTC.2250.09	Currentinfo.balance == 1750 Userinfo.balance == ???	Userinfo.balance < 1750 Enable "부족"
PTS.UTC.2260	요금정보와 그에 해당하는 정보를 사용자 카드와 단말기에 각각 저장시키고, 금액과 시간을 출력하는 명령을 전송한다.	
PTS.UTC.2260.00	Trigger "충분"	저장 Command == fputa 출력 Command == printf
PTS.UTC.2270	최종처리에 금액이 부족한 경우 활성화 되며, 금액 부족 문구를 출력하는 명령을 전송한다.	
PTS.UTC.2270.00	Enable. "부족"	출력 Command == printf
PTS.UTC.2310	사용자가 하차태그를 하였을 시 환승을 하였는지 안 하였는지의 여부에 대한 트리거를 다음 프로세스에 전송해준다.	
PTS.UTC.2310.00	Trigger "하차 Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation == 0 Userinfo.terminalinfo[0] == 'a'	Trigger "미환승 요금 계산"
PTS.UTC.2310.01	Trigger "하차Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation == 0 Userinfo.terminalinfo[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime < 30	Trigger "환승 요금 계산"
PTS.UTC.2310.02	Trigger "하차Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0	Trigger "환승 요금 계산"

	Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime – userinfo.lasttagtime <60	
PTS.UTC.2310.03	Trigger "하차Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime – userinfo.lasttagtime < 90	Trigger"환승 요금 계산"
PTS.UTC.2310.04	Trigger "하차Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime – userinfo.lasttagtime < 120	Trigger"환승 요금 계산"
PTS.UTC.2310.05	Trigger "하차Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime – userinfo.lasttagtime < 150	Trigger"환승 요금 계산"
PTS.UTC.2310.06	Trigger "승차Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==0	Trigger"환승 요금 계산"



	Userinfo.terminalinfo[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime < 180	
PTS.UTC.2320	하차시 미환승 하였을 경우 지금까지 이동한 시간정보를 입력 받아서 해당하는 금액을 계산하고 해당 금액과 해당하는 정보를 사용자 카드와 단말기에서 각각 저장시키고, 금액과 시간을 출력한다.	
PTS.UTC.2320.00	Trigger, "미환승 요금계산" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation == 0 Userinfo.terminalinfo[0] == 'a' c.time	Currentinfo.balance == 0 저장 Command == fputa 출력 Command == printf
PTS.UTC.2330	25차시환 승 하였을 경우 지금까지 이동한 시간정보를 입력 받아서 해당하는 금액을 계산하고 해당금액과 해당하는 정보를 사용자 카드와 단말기에 각각 저장시키고, 금액과 시간을 출력한다.	
PTS.UTC.2330.00	Trigger "환승 요금 계산" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation == 0 Userinfo.terminalinfo[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime < 30	Currentinfo.balance == 0 저장 Command == fputa 출력 Command == printf
PTS.UTC.2330.01	Trigger "환승 요금 계산" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation == 0 Userinfo.terminalinfo[0] != 'a'	Currentinfo.balance == 100 저장 Command == fputa 출력 Command == printf

	Currentinfo.lasttagtime – userinfo.lasttagtime < 60 c.time	
PTS.UTC.2330.02	Trigger“환승 요금 계산” Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime – userinfo.lasttagtime < 90 c.time	Currentinfo.balance == 200 저장 Command == fputa 출력 Command == printf
PTS.UTC.2330.03	Trigger“환승 요금 계산” Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime – userinfo.lasttagtime < 120 c.time	Currentinfo.balance == 300 저장 Command == fputa 출력 Command == printf
PTS.UTC.2330.04	Trigger“환승 요금 계산” Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime – userinfo.lasttagtime < 150 c.time	Currentinfo.balance == 400 저장 Command == fputa 출력 Command == printf
PTS.UTC.2330.05	Trigger“환승 요금 계산” Currentinfo. In_out == 1	Currentinfo.balance == 500 저장 Command == fputa

	Userinfo.In_Out == 0 Userinfo.transportation ==0 Userinfo.terminalinfo[0] !='a' Currentinfo.lasttagtime - userinfo.lasttagtime < 180 c.time	출력 Command == printf
--	---	----------------------

**Table 6 Test case Identification-지하철**

Identifier	Input	Pass/Fail Criteria
PTS.UTC.1000	유저가 태그 하였을 시 유저 카드에 저장된 정보를 입력 받아 저장해주는 부분	
PTS.UTC.1000.00	규모가 큰 Unit.	
PTS.UTC.1100	받은 사용자 카드에서 마지막으로 태그 된 시간 정보를 전달한다.	
PTS.UTC1100.00	Interfaces는 Testing 제외	
PTS.UTC.1200	받은 사용자 카드에 저장된 마지막으로 태그 된 교통수단 정보를 전달한다.	
PTS.UTC1200.00	Interfaces는 Testing 제외	
PTS.UTC.1300	받은 사용자 카드에 저장된 마지막으로 태그 된 승.하차 정보를 전달한다.	
PTS.UTC1300.00	Interfaces는 Testing 제외	
PTS.UTC.1400	받은 사용자 카드에 저장된 잔액 정보를 전달한다.	
PTS.UTC1400.00	Interfaces는 Testing 제외	
PTS.UTC.1500	받은 사용자 카드에 저장된 마지막으로 태그 된 단말기 정보를 전달한다.	
PTS.UTC1500.00	Interfaces는 Testing 제외	

PTS.UTC.1600	유저가 태그를 했을 시 유저의 입력을 받은 부분을 모아서 컨트롤 부에서 이용할 수 있는 자료구조로 따로 저장해준다. 또한 태그를 한 시간을 따로 저장해준다.	
PTS.UTC1600.00	LastTagTime  transportation  in_out  balance  terminalinfo	userInfo에 저장, currentInfo에 저장.
PTS.UTC.2000	Tag 정보에서 입력을 받아 정보처리를 한 후 각각 화면에 출력하거나 저장을 한다.	
PTS.UTC.2000.00	규모가 큰 Unit. 단순한 Spec으로 인해 Testing 불가	
PTS.UTC.2100	승객이 승차를 하는지 하차를 하는지에 따라서 case로 나눈다.	
PTS.UTC.2100.00	규모가 큰 Unit. 단순한 Spec으로 인해 Testing 불가	
PTS.UTC.2200	승객이 승차를 했을 때 지불해야 할 요금계산 후 요금의 충분 부족에 따라 각 저장/출력 명령을 출력한다.	
PTS.UTC.2200.00	State == "case 승차"	저장 command ==fputa 출력 command==printf
PTS.UTC.2300	승객이 하차를 했을 때 지불해야 할 요금계산 후 저장/출력 명령을 출력한다.	
PTS.UTC.2300.00	State == "Case 하차"	저장 command ==fputa 출력 command ==printf
PTS.UTC.2400	저장에 대한 명령어를 받아서 각각 지하철과 버스 단말기 정보 저장소 파일에 요금정보를 포함해서 저장을 하고, 승객의 카드정보에도 잔액 정보를 포함	

	해서 저장을 한다.	
PTS.UTC.2400.00	Interface는 Testing 제외	
PTS.UTC.2500	출력에 대한 명령어를 받아서 요금과 현재 시간에 대한 정보를 단말기(여기선 터미널)에 출력한다.	
PTS.UTC.2500.00	Interface는 Testing 제외	
PTS.UTC.1610	유저가 태그를 했을 시 유저의 입력을 받은 부분을 모아서 컨트롤 부에서 이용 할 수 있는 자료구조로 따로 저장해준다. 또한 태그 한 시간을 따로 저장해 준다. 그리고 3분마다 초기화 판단 및 시행을 해준다.	
PTS.UTC.1610.00	LastTagTime == ??? Transportation ==??? in_out == ??? balance == ??? terminalinfo == ??? c.time != 03:00	struct userInfo  struct currentInfo  secTime
PTS.UTC.1610.01	LastTagTime == ??? Transportation == ??? in_out == ??? balance == ??? terminalinfo == ???	Trigger" 초기화"

	c.time == 03:00	
PTS.UTC.1620	3분주기마다 시행이 되며 유저카드 이외의 정보를 초기화하는 command를 출력한다.	
PTS.UTC.1620.00	Trigger	"초기화!" 출력
PTS.UTC.2110	승객의 승차, 하차를 구분 짓는 컨트롤로 사용자 정보와 현재시간을 입력 받아 승차인지 하차인지 결정 짓는다.	
PTS.UTC.2110.00	Currentinfo. In_out == 0	Trigger"승차Control"
PTS.UTC.2110.01	Currentinfo. In_out == 1	Trigger"하차Control"
PTS.UTC.2210	실질적으로 요금의 경우를 나누어 주는 주 컨트롤러로, 사용자의 정보를 입력 받고, 환승 여부, 미정산 여부, 최초 탑승여부의 조합에 따라 경우를 나누어 각 계산해야 되는 요금식을 판단해주고 다음 프로세스에 트리거를 전송해준다.	
PTS.UTC.2210.00	Trigger,"승차Control" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation ==1 Currentinfo.terminal[0] != 'a'	Trigger"미정산환승"
PTS.UTC.2210.01	Trigger,"승차Control" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] != 'a' Userinfo.transportation ==0 Currentinfo.terminal[0] != 'a'	Trigger"미정산환승"

PTS.UTC.2210.02	Trigger,"승차Control" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] != 'a' Userinfo.transportation ==1 Currentinfo.terminal[0] != 'a'	Trigger"미정산미환승"
PTS.UTC.2210.03	Trigger,"승차Control" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation ==0 Currentinfo.terminal[0] != 'a'	Trigger"기본요금"
PTS.UTC.2210.04	Trigger,"승차Control" Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] != 'a' Userinfo.transportation == 1 Currentinfo.terminal[0] != 'a'	Trigger"기본요금"
PTS.UTC.2210.05	Trigger,"승차Control" Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation == 1 Currentinfo.terminal[0] != 'a'	Trigger"기본요금"
PTS.UTC.2210.06	Trigger,"승차Control"	Trigger"기본요금"

	Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation == 0 Currentinfo.terminal[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime > 15	
PTS.UTC.2210.07	Trigger,"승차Control" Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation == 0 Currentinfo.terminal[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime <= 15	Trigger"정산환승"
PTS.UTC.2220	사용자가 이전에 정산을 하고 환승을 하였을 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.	
PTS.UTC.2220.00	Trigger,"미정산환승" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] != 'a' Userinfo.transportation ==1 Currentinfo.terminal[0] != 'a'	Currentinfo.balance == 1750
PTS.UTC.2220.01	Trigger,"미정상환승" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] != 'a'	Currentinfo.balance == 1650



	Userinfo.transportation ==0 Currentinfo.terminal[0] != 'a'	
PTS.UTC.2230	사용자가 미성잔을 하고 환승을 안 한 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.	
PTS.UTC.2230.00	Trigger"미정산미환승" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] != 'a' Userinfo.transportation ==1 Currentinfo.terminal[0] != 'a'	Currentinfo.balance == 1250
PTS.UTC.2240	사용자가 정산을 하고 환승을 안한 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.	
PTS.UTC.2240.00	Trigger"정산환승" Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation == 0 Currentinfo.terminal[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime <= 15	Currentinfo.balance == 0
PTS.UTC.2250	사용자가 최초탑승이거나 이전에 버스를 탔거나 환승시간이 넘은 경우, 기본 금액을 다음 컨트롤러로 전달해준다.	
PTS.UTC.2250.00	Enable"기본요금" Userinfo.In_Out == 0 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a'	Currentinfo.balance == 1050

	Userinfo.transportation == 0 Currentinfo.terminal[0] != 'a'	
PTS.UTC.2250.01	Enable "기본요금" Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] != 'a' Userinfo.transportation == 1 Currentinfo.terminal[0] != 'a'	Currentinfo.balance == 1050
PTS.UTC.2250.02	Enable "기본요금" Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation == 1 Currentinfo.terminal[0] != 'a'	Currentinfo.balance == 1050
PTS.UTC.2250.03	Enable "기본요금" Userinfo.In_Out == 1 Currentinfo. In_out == 0 Userinfo.terminalinfo[0] == 'a' Userinfo.transportation == 0 Currentinfo.terminal[0] != 'a' Currentinfo.lasttagtime - userinfo.lasttagtime > 15	Currentinfo.balance == 1050
PTS.UTC.2260	전달받은 금액을 유저의 잔액과 비교해서 요금부족 여부를 판단하여 각각 트리를 전달해준다.	
PTS.UTC.2260.00	Currentinfo.balance == 0 Userinfo.balance == ???	Money = Currentinfo.balance + 600 Userinfo.balance < Money

		Enable,"부족",
PTS.UTC.2260.01	Currentinfo.balance == 0 Userinfo.balance == ???	Money = Currentinfo.balance + 600 Userinfo.balance >= Money Trigger,"충분",
PTS.UTC.2260.02	Currentinfo.balance != 0 Userinfo.balance == ???	Money = Currentinfo.balance + 200 Userinfo.balance >= Money Trigger,"충분",
PTS.UTC.2260.03	Currentinfo.balance == 1050 Userinfo.balance == ???	Money = Currentinfo.balance + 200 Userinfo.balance < Money Enable,"부족",
PTS.UTC.2270	요금정보와 그에 해당하는 정보를 사용자 카드와 단말기에 각각 저장시키고, 금액과 시간을 출력하는 명령을 전송한다.	
PTS.UTC.2270.00	Trigger,"충분"	저장 Command == fputa 출력 Command == printf
PTS.UTC.2280	금액이 부족한 경우 활성화 되며, 금액 부족 문구를 출력하는 명령을 전송한다.	
PTS.UTC.2270.01	Enable. "부족"	출력 Command == printf
PTS.UTC.2310	사용자가 하차태그를 하였을 시 환승을 하였는지 안 하였는지의 여부에 대한 트리거를 다음 프로세스에 전송해준다.	
PTS.UTC.2310.00	Trigger "하차 Control" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] != 'a' Currentinfo.terminalinfo[0] != 'a'	Trigger"미환승 요금계산"

PTS.UTC.2310.01	Trigger "하차 Control" Currentinfo.In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] == 'a' Currentinfo.terminalinfo[0] != 'a'	Enable"환승 요금계산"
PTS.UTC.2320	하차시 미환승 하였을 경우 지금까지 이동한 시간정보를 입력 받아서 해당하는 금액을 계산하고 해당 금액과 해당하는 정보를 사용자 카드와 단말기에서 각각 저장시키고, 금액과 시간을 출력한다.	
PTS.UTC.2320.00	Trigger"미환승 요금계산" Currentinfo.In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] != 'a' Currentinfo.terminalinfo[0] != 'a' c.time	<pre> userinfo.terminalinfo[0] == 'b' &amp;&amp; currentinfo.terminalinfo[0] == 'b')    (userinfo.terminalinfo[0] == 'c' &amp;&amp; currentinfo.terminalinfo[0] == 'c')    (userinfo.terminalinfo[0] == 'd' &amp;&amp; currentinfo.terminalinfo[0] == 'd')    (userinfo.terminalinfo[0] == 'e' &amp;&amp; currentinfo.terminalinfo[0] == 'e')    (userinfo.terminalinfo[0] == 'f' &amp;&amp; currentinfo.terminalinfo[0] == 'f')  Currentinfo.balance == 0 저장 Command == fputa 출력 Command == printf </pre>
PTS.UTC.2320.01	Trigger"미환승 요금계산" Currentinfo.In_out == 1 Userinfo.In_Out == 0	<pre> userinfo.terminalinfo[0] == 'b' &amp;&amp; (currentinfo.terminalinfo[0] == 'f'    currentinfo.terminalinfo[0] == 'c')) </pre>

	<pre>Userinfo.transportation ==1 Userinfo.terminalinfo[0] !='a' Currentinfo.terminalinfo[0] != 'a' c.time</pre>	<pre>   (userinfo.terminalinfo[0] == 'c' &amp;&amp; (currentinfo.terminalinfo[0] == 'b'    currentinfo.terminalinfo[0] == 'd'))    (userinfo.terminalinfo[0] == 'd' &amp;&amp; (currentinfo.terminalinfo[0] == 'c'    currentinfo.terminalinfo[0] == 'e'))    (userinfo.terminalinfo[0] == 'e' &amp;&amp; (currentinfo.terminalinfo[0] == 'd'    currentinfo.terminalinfo[0] == 'f'))    (userinfo.terminalinfo[0] == 'f' &amp;&amp; (currentinfo.terminalinfo[0] == 'e'    currentinfo.terminalinfo[0] == 'b'  Currentinfo.balance == 0 저장 Command == fputa 출력 Command == printf</pre>
<p>PTS.UTC.2320.02</p>	<pre>Trigger“미환승 요금계산” Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] !='a' Currentinfo.terminalinfo[0] != 'a' c.time</pre>	<pre>userinfo.terminalinfo[0] == 'b' &amp;&amp; (currentinfo.terminalinfo[0] == 'd'    currentinfo.terminalinfo[0] == 'e'))    (userinfo.terminalinfo[0] == 'c' &amp;&amp; (currentinfo.terminalinfo[0] == 'e'    currentinfo.terminalinfo[0] == 'f'))    (userinfo.terminalinfo[0] == 'd' &amp;&amp; (currentinfo.terminalinfo[0] == 'f'    currentinfo.terminalinfo[0] == 'b'))</pre>

		<pre>    (userinfo.terminalinfo[0] == 'e' &amp;&amp; (currentinfo.terminalinfo[0] == 'b'    currentinfo.terminalinfo[0] == 'c'))    (userinfo.terminalinfo[0] == 'f' &amp;&amp; (currentinfo.terminalinfo[0] == 'c'    currentinfo.terminalinfo[0] == 'd'))  Currentinfo.balance == 200 저장 Command == fputa 출력 Command == printf </pre>
PTS.UTC.2330	하차시 환승 하였을 경우 지금까지 이동한 시간정보를 입력 받아서 해당하는 금액을 계산하고 해당금액과 해당하는 정보를 사용자 카드와 단말기에 각각 저장시키고, 금액과 시간을 출력한다.	
PTS.UTC.2330.00	<pre> Enable,"환승 요금계산" Currentinfo.In_out == 1 UserInfo.In_Out == 0 UserInfo.transportation ==1 UserInfo.terminalinfo[0] =='a' Currentinfo.terminalinfo[0] != 'a' c.time </pre>	<pre> (c == 'b' &amp;&amp; currentinfo.terminalinfo[0] == 'b')    (c == 'c' &amp;&amp; currentinfo.terminalinfo[0] == 'c')    (c == 'd' &amp;&amp; currentinfo.terminalinfo[0] == 'd')    (c == 'e' &amp;&amp; currentinfo.terminalinfo[0] == 'e')    (c == 'f' &amp;&amp; currentinfo.terminalinfo[0] == 'f')  Currentinfo.balance == 0 저장 Command == fputa 출력 Command == printf </pre>
PTS.UTC.2330.01	<pre> Enable,"환승 요금계산" Currentinfo.In_out == 1 UserInfo.In_Out == 0 </pre>	<pre> (c == 'b' &amp;&amp; (currentinfo.terminalinfo[0] == 'c'    currentinfo.terminalinfo[0] == 'f'))    (c == 'c' &amp;&amp; (currentinfo.terminalinfo[0] == 'd' </pre>

	<pre>Userinfo.transportation ==1 Userinfo.terminalinfo[0] =='a' Currentinfo.terminalinfo[0] != 'a' c.time</pre>	<pre>   currentinfo.terminalinfo[0] == 'b'))    (c == 'd' &amp;&amp; (currentinfo.terminalinfo[0] == 'e'    currentinfo.terminalinfo[0] == 'c'))    (c == 'e' &amp;&amp; (currentinfo.terminalinfo[0] == 'f'    currentinfo.terminalinfo[0] == 'd'))    (c == 'f' &amp;&amp; (currentinfo.terminalinfo[0] == 'b'    currentinfo.terminalinfo[0] == 'e'  Currentinfo.balance == 300 저장 Command == fputa 출력 Command == printf</pre>
<p>PTS.UTC.2330.02</p>	<pre>Enable,"환승 요금계산" Currentinfo. In_out == 1 Userinfo.In_Out == 0 Userinfo.transportation ==1 Userinfo.terminalinfo[0] =='a' Currentinfo.terminalinfo[0] != 'a' c.time</pre>	<pre>c == 'b' &amp;&amp; (currentinfo.terminalinfo[0] == 'd'    currentinfo.terminalinfo[0] == 'e'))    (c == 'c' &amp;&amp; (currentinfo.terminalinfo[0] == 'e'    currentinfo.terminalinfo[0] == 'f'))    (c == 'd' &amp;&amp; (currentinfo.terminalinfo[0] == 'f'    currentinfo.terminalinfo[0] == 'b'))    (c == 'e' &amp;&amp; (currentinfo.terminalinfo[0] == 'b'    currentinfo.terminalinfo[0] == 'c'))    (c == 'f' &amp;&amp; (currentinfo.terminalinfo[0] == 'c'    currentinfo.terminalinfo[0] == 'd'  Currentinfo.balance == 600 저장 Command == fputa 출력 Command == printf</pre>

Table 7 Test case Identification-정산

Identifier	Input	Pass/Fail Criteria	비고
PTS.UTC.1000	각 버스와 지하철 단말기에서부터 정보를 입력받아 저장을 해주는 부분이다.		
PTS.UTC.1000	Subway data Input, Bus data Input	Save(단말기 결제정보)	
PTS.UTC.2000	단말기 결제정보에서 정산에 필요한 정보를 불러와 정산처리를 한 뒤 정산된 금액을 각 회사에 보내고, 화면에 출력한다.		
PTS.UTC.2000.00	Load(단말기 결제정보)	Transmit, Print	
PTS.UTC.1100	지하철의 데이터를 입력 받아서 정보를 전송해 준다.		
PTS.UTC1100.00	Subway data Input	Save(단말기 결제정보)	
PTS.UTC.1200	버스의 데이터를 입력 받아서 정보를 전송해 준다.		
PTS.UTC1200.00	Bus data Input	Save(단말기 결제정보)	
PTS.UTC.2100	승객이 승차를 하는지 하차를 하는지에 따라서 case로 나눈다.		
PTS.UTC2100.00	User 정보, 현재시간	Case 승차, Case 하차	
PTS.UTC.2200	승객이 승차를 했을 때 지불해야 할 요금계산 후 요금의 충분 부족에 따라 각 저장/출력 명령을 출력한다. 그리고 3분마다 초기화 명령을 출력한다.		
PTS.UTC2200.00	Case 승차	저장 Command, 출력 Command, 초기화 Command	
PTS.UTC.2300	승객이 하차를 했을 때 지불해야 할 요금 계산 후 요금의 충분 부족에 따라 각 저장/출력 명령을 출력한다.		
PTS.UTC2300.00	Case 하차	저장 Command, 출력 Command	
PTS.UTC.2110	각 환승 방향별로 정산금액을 계산해주고, 해당 정보		



를 저장하며, 3분마다 전송부를 활성화 시킨다.			
PTS.UTC2110.00	Load tick	Enable/Disable"정산실행"	
PTS.UTC.2120	정산된 금액에 대해서 각각 지하철과 버스회사로 전송 해주고, 정산된 금액을 출력해준다		
PTS.UTC2120.00	Enable, Disable	Save, Print	