

Unit Test Report for Public Transportation System

- Test Cases Specification
- Test Summary Report

Project Team
Team 3

Latest update on:
2014-11-21

Team Information
201311269 김제현
201311275 박상희
201311276 박형민
201311287 엄현식

Table of Contents

1 Introduction3

1.1 Objectives.....3

1.2 References.....3

2 Unit test case specification	3
2.1 Test case specification identifier	3
2.1.1 Public Transportation System	3
2.1.2 Recharger System.....	9
2.1.3 Fee Calculation System	10
2.2 Test items.....	15
2.2.1 Public Transportation System	15
2.2.2 Recharger System.....	20
2.2.3 Fee Calculation System	22
2.3 Input specifications.....	26
2.3.1 Public Transportation System	26
2.3.2 Recharger System.....	26
2.3.3 Fee Calculation System	26
2.4 Output specifications.....	26
2.4.1 Public Transportation System	26
2.4.2 Recharger System.....	26
2.4.3 Fee Calculation System	26
3 Environmental needs	27
4 Unit test summary report	27
4.1 Test summary report identifier.....	27
4.1.1 Public Transportation System	27
4.1.2 Recharger System.....	28

4.1.3 Fee Calculation System	31
4.2 Evaluation	32

1 Introduction

1.1 Objectives

본 문서는 Public Transportation System, Recharger System, Fee calculation System 을 수행한 결과에 대한 Report 문서이다. Test 요소들에 대한 Test Case 와 Test 수행 결과에 대한 내용을 담고 있다.

1.2 References

DS-2014SE-PTS-SRS-1.0

T3-2014-PTS-SRA-1.3

T3-2014-PTS-SDS-2.0

T3-2014-PTS-UTP-1.0

2 Unit test case specification

2.1 Test case specification identifier

2.1.1 Public Transportation System

<Table 1.1 Test Case Identification>

Test Case Identifier	Input Specification	Output Specification
PTS.UTC_1200_000	File!=NULL/ card_Info->CID==temp_card_info->CID	Card_info 를 card_Reader_1.txt 에 쓴다.
PTS.UTC_1200_001	File!=NULL/ card_Info>CID!=temp_card_info->CID	stat = InvalidInput; card_info 의 값을 0 으로 초기화시킨다.
PTS.UTC_1200_002	File==Null	printf("파일이 없습니다. 파일 열기 실패\n")

PTS.UTC_2111_000	stat==Normal	Error 검사를 실행하고 가격을 책정한다.
PTS.UTC_2111_001	stat!=Normal	
PTS.UTC_2112_000	interval_sec <=0	interval_sec += 60;
PTS.UTC_2112_001	state==1/interval_sec<15/getout==0/CRID==card_info	interval_sec += 60; stat = HopInProcessing;

	->CRID	
PTS.UTC_2112_002	state==1/interval_sec>=15/getout==0/CRID==card_info->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_003	state==1/interval_sec<15/getout!=0/CRID==card_info->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_004	state==1/interval_sec<15/getout==0/CRID!=card_info->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_005	state==1/interval_sec<15/getout!=0/CRID!=card_info->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_006	state==1/interval_sec>=15/getout!=0/CRID==card_info->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_007	state==1/interval_sec>=15/getout==0/CRID!=card_info->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_008	state==1/interval_sec>=15/getout!=0/CRID!=card_info->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_009	state==0/interval_sec<15/getout==0/CRID==card_info->CRID	stat = GetOffProcessing;

PTS.UTC_2112_010	state==0/interval_sec>=15 /getout==0/CRID==card_in fo->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_011	state==0/interval_sec<15/g etout!=0/CRID==card_info- >CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_012	state==0/interval_sec<15/g etout==0/CRID!=card_info- >CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_013	state==0/interval_sec<15/g etout!=0/CRID!=card_info- >CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_014	state==0/interval_sec>=15	Stat 의 값을 변경시키지 않는다.

	/getout!=0/CRID==card_inf o->CRID	
PTS.UTC_2112_015	state==0/interval_sec>=15 /getout==0/CRID!=card_inf o->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2112_016	state==0/interval_sec>=15 /getout!=0/CRID!=card_inf o->CRID	Stat 의 값을 변경시키지 않는다.
PTS.UTC_2113_000	interval_sec <= 0	interval_sec += 60 * 60
PTS.UTC_2113_001	CRID < 10	tp = 0;
PTS.UTC_2113_002	CRID >= 10	tp = 1;
PTS.UTC_2113_003	(CRID%10) == 1 / state == 1 / transfer == 1 / tp == 1	price = 1650
PTS.UTC_2113_004	(CRID%10) == 1 / state == 1 / transfer == 1 / tp == 0	price = 1550
PTS.UTC_2113_005	(CRID%10) == 1 / state == 1 / transfer == 1 / tp != 1 / tp != 0	Error 출력

PTS.UTC_2113_006	(CRID%10) == 1 / state == 1 / transfer == 0 / tp == 1	price = 1250
PTS.UTC_2113_007	(CRID%10) == 1 / state == 1 / transfer == 0 / tp == 0	price = 1050
PTS.UTC_2113_008	(CRID%10) == 1 / state == 1 / transfer == 0 / tp != 1 / tp != 0	Error 출력
PTS.UTC_2113_009	(CRID%10) == 1 / state == 1 / transfer != 0 / transfer != 1	Error 출력
PTS.UTC_2113_010	(CRID%10) == 1 / state == 0 / transfer == 0 / interval_sec <= 15 / tp == card_info->tp	price = 1050
PTS.UTC_2113_011	(CRID%10) == 1 / state == 0 / transfer == 0 / interval_sec <= 15 / tp == 1 / card_info->tp == 0	price = 600
PTS.UTC_2113_012	(CRID%10) == 1 / state == 0 / transfer == 0 /	price = 500

	interval_sec <= 15 && tp == 0 && card_info->tp == 1	
PTS.UTC_2113_013	(CRID%10) == 1 / state == 0 / transfer == 0 / interval_sec <= 15 / tp != 0 / tp != 1 / card_info->tp != 0 / card_info->tp != 1	Error 출력
PTS.UTC_2113_014	(CRID%10) == 1 / state == 0 / transfer == 0 / interval_sec > 15	price = 1050

PTS.UTC_2113_015	(CRID%10) == 1 / state == 0 / transfer != 0	Error 출력
PTS.UTC_2113_016	(CRID%10) == 0 / state == 1 / transfer == 1 / tp == 1 / interval_station == 1 interval_station == 4	price = 300
PTS.UTC_2113_017	(CRID%10) == 0 / state == 1 / transfer == 1 / tp == 1 / interval_station == 2 interval_station == 3	price = 600
PTS.UTC_2113_018	(CRID%10) == 0 / state == 1 / transfer == 1 / tp == 1 / interval_station < 1 interval_station > 4	price = 0
PTS.UTC_2113_019	(CRID%10) == 0 / state == 1 / transfer == 1 / tp == 0 / (interval_sec/30) > 5	price = 500
PTS.UTC_2113_020	(CRID%10) == 0 / state == 1 / transfer == 1 / tp == 0 / (interval_sec/30) <= 5	price = interval_sec / 30 * 100
PTS.UTC_2113_021	(CRID%10) == 0 / state == 1 / transfer == 1 / tp != 0 / tp != 1	Error 출력
PTS.UTC_2113_022	(CRID%10) == 0 / state == 1 / transfer == 0 / tp == 1 / interval_station == 1	price = 0

	interval_station == 4	
PTS.UTC_2113_023	(CRID%10) == 0 / state == 1 / transfer == 0 / tp == 1 / interval_station == 2 interval_station == 3	price = 200

PTS.UTC_2113_024	(CRID%10) == 0 / state == 1 / transfer == 0 / tp == 1 / interval_station < 1 interval_station > 4	price = 0
PTS.UTC_2113_025	(CRID%10) == 0 / state == 1 / transfer == 0 / tp == 0	price = 0
PTS.UTC_2113_026	(CRID%10) == 0 / state == 1 / transfer == 0 / tp != 1 / tp != 0	Error 출력
PTS.UTC_2113_027	(CRID%10) == 0 / state != 1	Error 출력
PTS.UTC_2113_028	(CRID%10)!=0/ (CRID%10) != 1	Error 출력
PTS.UTC_2121_000	stat == Normal	Short()호출
PTS.UTC_2121_001	Short() / stat == Normal	Calculation()호출
PTS.UTC_2122_000	(card_info->cash-price) < 0	stat = EShort
PTS.UTC_2122_000	CRID < 10	tp = 0
PTS.UTC_2123_001	CRID >= 10	tp = 1
PTS.UTC_2123_002	card_info->transfer == 0	card_info->cash -= price
PTS.UTC_2123_003	(CRID%10) == 0	card_info->state = 0
PTS.UTC_2123_004	(CRID%10) != 0	card_info->state = 1
PTS.UTC_2130_000	stat == Normal	CardUpdate(), CardReaderRecord() 호출
PTS.UTC_2140_000	stat == Normal / transfer == 1	0, cash 출력
PTS.UTC_2140_001	stat == Normal / transfer != 1	price, cash 출력
PTS.UTC_2140_002	stat == HopnInProcessing	Error 메시지 출력
PTS.UTC_2140_003	stat == GetoffProcessing	Error 메시지 출력
PTS.UTC_2140_004	stat == EShort	Error 메시지 출력

PTS.UTC_2140_005	stat == NotAddjust	Error 메시지 출력
PTS.UTC_2140_006	stat == InvalidInput	Error 메시지 출력
PTS.UTC_2150_000	file != NULL / newFile !=	newFile 에 update 된 card_info 를
	NULL / feof(file) == false / CID == fileCID	기록한다.
PTS.UTC_2150_001	file != NULL / newFile != NULL / feof(file) == false / CID == fileCID	기존의 카드값을 유지한다.
PTS.UTC_2160_000	file != NULL	card_info 를 card_Reader_1.txt 에 쓴다.

2.1.2 Recharger System

<Table 2.1 Test Case Identification>

Test Case Identifier	Input specification	Output specification
PTS.UTC_120_000	CID=1000 / if(file != NULL)	card_info={CID=1000, CRID=11, tagTime={2014, 11, 20, 19, 52, 18}, tp=1, state=1, cash=30900, transfer=0, getout=1}
PTS.UTC_120_001	CID=100 / if(file != NULL)	card_info={CID=0, CRID=0, tagTime={?, ?, ?, ?, ?, ?}, tp=0, state=0, cash=0, transfer=0, getout=0}
PTS.UTC_120_002	if(file == NULL)	"파일이 없습니다. 파일 열기 실패" exit(1);exit(1);
PTS.UTC_211_000	CID=1000 / money=20000	card_info.cash+=money / trigger "Update"/ trigger "Display"
PTS.UTC_211_001	CID=100 / money=20000	card_info.cash+=money / trigger "Update"/ trigger "Display"

PTS.UTC_212_000	card_info={CID=1000, CRID=11, tagTime={2014, 11, 20, 19, 52, 18}, tp=1, state=1, cash=50900, transfer=0, getout=1}	fprintf(newFile, "...", card_info->CID, card_info->getout);이 한번 수행
PTS.UTC_212_001	card_info={CID=0, CRID=0, tagTime={?, ?, ?, ?, ?, ?}, tp=0, state=0, cash=0, transfer=0, getout=0}	fprintf(newFile, "...", card_info->CID, card_info->getout);이 한번도 수행되지 못함.
PTS.UTC_212_002	if(file == NULL)	"파일이 없습니다. 파일 열기 실패" exit(1);
PTS.UTC_213_000	card_info={CID=1000, CRID=11, tagTime={2014, 11, 20, 19, 52, 18}, tp=1, state=1, cash=50900, transfer=0, getout=1} / money=20000	"충전 한 시각 : 2014 년 11 월 21 일 03 시 16 분 44 초" "충전 전의 금액 : 30900 원" "충전 한 금액 : 20000 원" "충전 후의 금액 : 50900 원"
PTS.UTC_213_001	card_info={CID=0, CRID=0, tagTime={?, ?, ?, ?, ?, ?}, tp=0, state=0, cash=0, transfer=0, getout=0} / money=20000	"충전 한 시각 : 2014 년 11 월 21 일 03 시 16 분 44 초" "충전 전의 금액 : 0 원" "충전 한 금액 : 20000 원" "충전 후의 금액 : 20000 원"

2.1.3 Fee Calculation System

<Table 3.1 Test Case Identification>

Test Case Identifier	Input Specification	Output Specification
FCS.UTC.010.000	Tick()==0/ *.txt(단말기)!=NULL	fscanf(, &CardReader_info[i].CID,.....)

FCS.UTC.010.001	Tick()==0/ *.txt(단말기)==NULL	
FCS.UTC.010.002	Tick()==1/ *.txt(단말기)!=NULL	
FCS.UTC.010.003	Tick()==1/ *.txt(단말기)==NULL	
FCS.UTC.010.004	CardReader_info[i].CID==1000	fprint(file_CID_1000 , ,CardReader_info[i].CID,,,,,,)
FCS.UTC.010.005	CardReader_info[i].CID==1001	fprint(file_CID_1001, ,CardReader_info[i].CID,,,,,,)
FCS.UTC.010.006	CardReader_info[i].CID==1002	fprint(file_CID_1002, ,CardReader_info[i].CID,,,,,,)
FCS.UTC.010.007	CardReader_info[i].CID==1003	fprint(file_CID_1003, ,CardReader_info[i].CID,,,,,,)
FCS.UTC.010.008	CardReader_info[i].CID==1004	fprint(file_CID_1004, ,CardReader_info[i].CID,,,,,,)
FCS.UTC.010.009	CardReader_info[i].CID==1005	fprint(file_CID_1005, ,CardReader_info[i].CID,,,,,,)
FCS.UTC.010.010	CardReader_info[i].CID==1006	fprint(file_CID_1006, ,CardReader_info[i].CID,,,,,,)
FCS.UTC.010.011	CardReader_info[i].CID>1006 CardReader_info[i].CID<1000	
FCS.UTC.021.000	AdjustStart()	c_well=1
FCS.UTC.021.001	!AdjustStart()	c_well=1

FCS.UTC.021.002	CardReader_info[i].state==1	CardReader_info[i].getout=1
FCS.UTC.021.003	CardReader_info[i].state==1	CardReader_info[i].getout=0
FCS.UTC.021.004	CardReader_info[i].state==0	CardReader_info[i].getout=1
FCS.UTC.021.005	CID_Sort()/CID_*.txt !=NULL	fscanf(CID_*.txt, ,temp_CardReader_info[i].CID,,,,)
FCS.UTC.021.006	CID_Sort()/CID_*.txt ==NULL	
FCS.UTC.021.007	CID_Sort()/temp_CardReader_info[i]!=NULL &temp_CardReader_info[i].tagTime 작은 순 으로	real_CardReader_info[i]=temp_CardReader_info[j]
FCS.UTC.021.008	CID_Sort()/temp_CardReader_info[i]==NULL &temp_CardReader_info[i].tagTime 작은 순 으로	
FCS.UTC.021.009	real_CarfReader_info!=NULL, line!=NULL, &bus_fee!=NULL, &metro_fee!=NULL	Adjust();

FCS.UTC.021.010	<code>real_CarfReader_info==NULL line==NULL &bus_fee==NULL &metro_fee==NULL</code>	
FCS.UTC.021.011	<code>i>0, real_CardReader_info[i].price>1050</code>	<code>real_Card_info[i].price += real_Card_info[i].price-1050, real_Card_info[i].price-=1050.</code>
FCS.UTC.021.012	<code>i<=0 real_CardReader_info[i].price<=1050</code>	<code>real_Card_info[i].price += real_Card_info[i].price-1050, real_Card_info[i].price-=1050.</code>
FCS.UTC.021.013	<code>Index=size-1; for(i=size-1;i>=0;i--) if((real_card_info[i].state==1)&& (real_card_info[i].transfer==0))</code>	<code>index=i-1;</code>
FCS.UTC.021.014	<code>Index=size-1; for(i=size-1;i>=0;i--) if((real_card_info[i].state!=1) (real_card_info[i].transfer!=0))</code>	<code>.</code>
FCS.UTC.021.016	<code>Index=size-1; for(i=size-1;i>=0;i--) { if((real_card_info[i].state==1)&& (real_card_info[i].transfer==0)) { for(j=index;j>=i;j--)}</code>	<code>real_fee+=real_card_info[j].price</code>
FCS.UTC.021.017	<code>Index==NULL</code>	<code>real_fee+= real_CardReader_info[i].price</code>
FCS.UTC.021.018	<code>real_CardReader_info[i-1].tp !=</code>	<code>total_fee+=real_fee</code>

	<code>real_CardReader_info[i].tp</code>	
FCS.UTC.021.019	<code>real_CardReader_info[i-1].tp == real_CardReader_info[i].tp</code>	<code>total_fee+=real_fee</code>

FCS.UTC.021.020	Index=size-1; for(i=size-1;i>=0;i--) { if((real_card_info[i].state==1)&& (real_card_info[i].transfer==0)) { for(j=index;j>=i;j--)}	temp_fee+=real_CardReader_info[i].price
FCS.UTC.021.021	index=NULL	temp_fee+=real_CardReader_info[i].price
FCS.UTC.021.022	if(j=i (real_card_info[j-1].tp !=real_card_info[j].tp)){ if(real_card_info[j].tp==0)	*bus_fee+=temp_fee/total_fee*real_fee
FCS.UTC.021.023	if(j!=i (real_card_info[j-1].tp ==real_card_info[j].tp)){ if(real_card_info[j].tp==0)	*bus_fee+=temp_fee/total_fee*real_fee
FCS.UTC.021.024	if(j=i (real_card_info[j-1].tp !=real_card_info[j].tp)){ if(real_card_info[j].tp==1)	Adjust 상태 일 때, ((real_CardReader_info[i-1].tp !=real_CardReader_info[i].tp) i==0), real_CardReader_info[i].tp==1 이면 *metro_fee+=temp_fee/total_fee*real_fee 를 수행 한다.
FCS.UTC.021.025	if(j!=i (real_card_info[j-1].tp ==real_card_info[j].tp)){ if(real_card_info[j].tp==1)	*metro_fee+=temp_fee/total_fee*real_fee
FCS.UTC.021.026	Index=size-1; for(i=size-1;i>=0;i--) { if((real_card_info[i].state==1)&& (real_card_info[i].transfer==0)) { for(j=index; j>=i; j--){} }	index=i-1, total_fee=0, real_fee=0, temp_fee=0
FCS.UTC.021.027	If 문 끝난 다음이 아닐때	index=i-1, total_fee=0, real_fee=0, temp_fee=0
FCS.UTC.021.028	bus_fee!=NULL, metro_fee!=NULL	c_well=0
FCS.UTC.021.029	bus_fee==NULL, metro_fee==NULL	c_well=0

FCS.UTC.021.029	c_well==0	enable Display.
FCS.UTC.021.030	c_well!=0	enable Display.
FCS.UTC.021.031	c_well==0	Trigger Send
FCS.UTC.021.032	c_well!=0	Trigger Send
FCS.UTC.021.033	c_well==0	Trigger Reset
FCS.UTC.021.034	c_well!=0	Trigger Reset
FCS.UTC.022.000	Enable/c_well==0	printf("버스 정산 금액: %d\n",bus_fee); printf("지하철 정산 금액: %d\n",metro_fee);
FCS.UTC.022.001	Enable/c_well==0	printf("버스 정산 금액: %d\n",bus_fee); printf("지하철 정산 금액: %d\n",metro_fee);
FCS.UTC.022.002	c_well!=0	
FCS.UTC.023.000	Trigger/c_well==0	file_bus=fopen("send_bus.txt","w"); fprintf(file_bus,"%d\n",c_well); fprintf(file_bus,"정산금액: %d\n",bus_fee); file_metro=fopen("send_metro.txt","w"); fprintf(file_metro,"%d\n",c_well); fprintf(file_metro,"총 정산금액: %d",metro_fee);
FCS.UTC.023.001	Trigger/c_well==0	file_bus=fopen("send_bus.txt","w"); fprintf(file_bus,"%d\n",c_well); fprintf(file_bus,"정산금액: %d\n",bus_fee); file_metro=fopen("send_metro.txt","w"); fprintf(file_metro,"%d\n",c_well); fprintf(file_metro,"총 정산금액: %d",metro_fee);
FCS.UTC.023.002	c_well!=0	
FCS.UTC.024.000	Trigger/c_well==0	remove(fname); rename("temp.txt",fname);
FCS.UTC.024.001	Trigger/c_well==0	remove(fname); rename("temp.txt",fname);
FCS.UTC.024.002	c_well!=0	

2.2 Test items

2.2.1 Public Transportation System

<Table 1.2 Test Design Identification>

Identifier	Feature	Valid/ Invalid Value
PTS_UTC_1200_000	1.2 Card Info Loader	File!=NULL 상태에서 유효한 card_info 의 입력이 들어온다.
PTS_UTC_1200_001	1.2 Card Info Loader	File!=NULL 유효하지 않은 card_info 의 입력이 들어온다.
PTS_UTC_1200_002	1.2 Card Info Loader	File==NULL 인 입력이 들어온다.
PTS_UTC_2111_000	2.1.1.1 Catch Error Controller	card_info 의 값과 정수인 price 의 값과 stat==Normal 인 입력이 들어온다
PTS_UTC_2111_001	2.1.1.1 Catch Error Controller	card_info 의 값과 정수인 price 의 값과 stat!=Normal 인 입력이 들어온다
PTS_UTC_2112_000	2.1.1.2 Error	interval_sec <=0 인 입력이 들어온다.
PTS_UTC_2112_001	2.1.1.2 Error	state==1,interval_sec<15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어온다.
PTS_UTC_2112_002	2.1.1.2 Error	state==1,interval_sec>=15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어온다.
PTS_UTC_2112_003	2.1.1.2 Error	state==1,interval_sec<15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어온다.
PTS_UTC_2112_004	2.1.1.2 Error	state==1,interval_sec<15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어온다.

PTS.UTC_2112_005	2.1.1.2 Error	state==1,interval_sec<15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어 온다.
PTS.UTC_2112_006	2.1.1.2 Error	state==1,interval_sec>=15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어 온다.
PTS.UTC_2112_007	2.1.1.2 Error	state==1,interval_sec>=15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어 온다.
PTS.UTC_2112_008	2.1.1.2 Error	state==1,interval_sec>=15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어 온다.
PTS.UTC_2112_009	2.1.1.2 Error	state==0,interval_sec<15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어 온다.

PTS.UTC_2112_010	2.1.1.2 Error	state==0,interval_sec>=15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어 온다.
PTS.UTC_2112_011	2.1.1.2 Error	state==0,interval_sec<15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어 온다.
PTS.UTC_2112_012	2.1.1.2 Error	state==0,interval_sec<15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어 온다.
PTS.UTC_2112_013	2.1.1.2 Error	state==0,interval_sec<15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어 온다.
PTS.UTC_2112_014	2.1.1.2 Error	state==0,interval_sec>=15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 같은 입력이 들어 온다.

PTS.UTC_2112_015	2.1.1.2 Error	state==0,interval_sec>=15,getout==0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어 온다.
PTS.UTC_2112_016	2.1.1.2 Error	state==0,interval_sec>=15,getout!=0 이고 CRID 의 값이 card_info 의 CRID 의 값과 다른 입력이 들어 온다.
PTS.UTC_2113_000	2.1.1.3 FixPrice	interval_sec <= 0 인 입력을 들어온다.
PTS.UTC_2113_001	2.1.1.3 FixPrice	CRID < 10 인 입력이 들어온다.
PTS.UTC_2113_002	2.1.1.3 FixPrice	CRID >= 10 인 입력이 들어온다.
PTS.UTC_2113_003	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 1 && transfer == 1 && tp == 1 이다.
PTS.UTC_2113_004	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 1 && transfer == 1 && tp == 0 이다.
PTS.UTC_2113_005	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 1 && transfer == 1 && tp != 1 && tp != 0 이다.
PTS.UTC_2113_006	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 1 && transfer == 0 && tp == 1 이다.
PTS.UTC_2113_007	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 1 && transfer == 0 && tp == 0 이다.
PTS.UTC_2113_008	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 1 && transfer == 0 && tp != 1 && tp != 0 이다.
PTS.UTC_2113_009	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 1 && transfer != 0 && transfer != 1 이다.
PTS.UTC_2113_010	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 0 && transfer == 0 && interval_sec <= 15 && tp == card_info->tp 이다.
PTS.UTC_2113_011	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 0 && transfer == 0 && interval_sec <= 15 && tp == 1 && card_info->tp == 0 이다.

PTS.UTC_2113_012	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 0 && transfer == 0 && interval_sec <= 15 && tp == 0 && card_info->tp == 1 이다.
PTS.UTC_2113_013	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 0 && transfer == 0 && interval_sec <= 15 && tp != 0 && tp != 1 && card_info->tp != 0 && card_info->tp != 1 이다.
PTS.UTC_2113_014	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 0 && transfer == 0 && interval_sec > 15
PTS.UTC_2113_015	2.1.1.3 FixPrice	(CRID%10) == 1 상태에서 state == 0 && transfer != 0
PTS.UTC_2113_016	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 1 && tp == 1 이면서 interval_station 가 1 또는 4 이다.
PTS.UTC_2113_017	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 1 && tp == 1 이면서 interval_station 가 2 또는 3 이다.
PTS.UTC_2113_018	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 1 && tp == 1 이면서 interval_station 가 1~4 가 아니다.
PTS.UTC_2113_019	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 1 && tp == 0 이면서 (interval_sec/30) > 5 이다.
PTS.UTC_2113_020	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 1 && tp == 0 이면서 (interval_sec/30) <= 5 이다.
PTS.UTC_2113_021	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 1 && tp != 0 && tp != 1 이다.
PTS.UTC_2113_022	2.1.1.3 FixPrice	PTS.UTC_2113_021 (CRID%10) == 0 상태에서 state == 1 && transfer == 0 && tp == 1 이면서 interval_station 가 1 또는 4 이다.
PTS.UTC_2113_023	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 &&

		transfer == 0 && tp == 1 이면서 interval_station 가 2 또는 3 이다.
PTS.UTC_2113_024	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 0 && tp == 1 이면서 interval_station 가 1~4 가 아니다.
PTS.UTC_2113_025	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 0 && tp == 0 이다.
PTS.UTC_2113_026	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state == 1 && transfer == 0 && tp != 1 && tp != 0 이다.
PTS.UTC_2113_027	2.1.1.3 FixPrice	(CRID%10) == 0 상태에서 state != 1 이다.
PTS.UTC_2113_028	2.1.1.3 FixPrice	(CRID%10) != 0 && (CRID%10) != 1 이다.
PTS.UTC_2121_000	2.1.2.1 Money Check Controller	stat == Normal 이다.
PTS.UTC_2121_001	2.1.2.1 Money Check Controller	Short 호출 후 stat == Normal 이다.
PTS.UTC_2122_000	2.1.2.2 Short	(cash-price) < 0 이다.
PTS.UTC_2122_000	2.1.2.3 Calculation	CRID < 10 값이 들어온다.
PTS.UTC_2123_001	2.1.2.3 Calculation	CRID >= 10 값이 들어온다.
PTS.UTC_2123_002	2.1.2.3 Calculation	카드에 저장된 transfer 값이 0 이다.
PTS.UTC_2123_003	2.1.2.3 Calculation	(CRID%10)==0 이다.
PTS.UTC_2123_004	2.1.2.3 Calculation	(CRID%10)!=0 이다.
PTS.UTC_2130_000	2.1.3 Payment Controller	stat == Normal 값이 들어온다.
PTS.UTC_2140_000	2.1.4 Result	stat == Normal && transfer == 1 인 값이 들어온다.
PTS.UTC_2140_001	2.1.4 Result	Normal && transfer != 1 인 값이 들어온다.
PTS.UTC_2140_002	2.1.4 Result	stat == HopInProcessing 값이 들어온다.
PTS.UTC_2140_003	2.1.4 Result	stat == GetoffProcessing 값이 들어온다.
PTS.UTC_2140_004	2.1.4 Result	stat == EShort 값이 들어온다.

PTS_UTC_2140_005	2.1.4 Result	stat == NotAddjust 값이 들어온다.
PTS_UTC_2140_006	2.1.4 Result	stat == InvalidInput 값이 들어온다.
PTS_UTC_2150_000	2.1.5 Card update	file != NULL && newFile != NULL && feof(file) == false 상태에서 CID == fileCID 값이 들어온다.
PTS_UTC_2150_001	2.1.5 Card update	file != NULL && newFile != NULL && feof(file) == false 상태에서 CID != fileCID 값이 들어온다.
PTS_UTC_2160_000	2.1.6 Card Reader Record	file != NULL 이다.

2.2.2 Recharger System

<Table 2.2 Test Design Identification>

Identifier	Feature	Valid/ Invalid Value
PTS_UTC_120_000	1.2 Card Info Loader	Card.txt 파일을 열어 입력받은 CID 값이 있는 줄을 찾고, 찾으면 카드의 정보를 덮어씌운다.
PTS_UTC_120_001	1.2 Card Info Loader	Card.txt 파일을 열어 입력받은 CID 값이 있는 줄을 찾고, 찾지 못하면 카드의 정보를 초기화해 준다.
PTS_UTC_120_002	1.2 Card Info Loader	Card.txt 파일이 지정한 상대경로에 존재하지 않는다면, 파일열기 실패를 출력하면서 프로그램이 종료된다.
PTS_UTC_211_000	2.1.1 Recharger Controller	적합한 CID 값을 입력받아서, card_info 값을 갱신한 후에는, money 값을 입력 받고, 잔액을 충전한 후에, 충전된 정보로 교통카드를 갱신하고, 충전시각과 함께 교통카드에 충전된 정보를 Monitor 에 보여준다.
PTS_UTC_211_001	2.1.1 Recharger Controller	적합한 CID 값을 입력받지 못해서, card_info 값을 0 으로 모두 초기화한 후일지라도, money 값을 입력 받고, 잔액을 충전한다. 충전된 정보로 교통카드를 갱신하려 하지만, Card.txt 파일에서 일치하는 CID 정보가 없으므로 실질적으로 갱신은 이루어지지 않는다. 이후, 충전시각과 함께 교

		통카드에 충전된 정보를 Monitor 에 보여준다.
PTS_UTC_212_000	2.1.2. Update	입력받은 충전할 금액과 카드의 잔액을 더한 카드의 정보를 입력 받아서, Card.txt 파일과 newCard.txt 파일을 열은 후, Card.txt 파일에서, 입력받은 CID 값이 있는 줄이면 충전한 카드정보를 newCard.txt 에 기록하고, 다른 CID 값이 있는 줄이면 그대로 newCard.txt 에 기록한다. 이후, Card.txt 파일은 제거하고, newCard.txt 의 이름을 Card.txt 로 변경한다.
PTS_UTC_212_001	2.1.2. Update	적합한 CID 값을 입력받지 못해서, card_info 값을 0 으로 모두 초기화한 카드의 정보를 입력받을 경우, Card.txt 파일에서, CID 값이 0 인 줄이 없으므로, Card.txt 파일의 값이 그대로 newCard.txt 에
		기록되면서 실질적으로 바뀌는 부분은 없다. 이후, Card.txt 파일은 제거하고, newCard.txt 의 이름을 Card.txt 로 변경한다.
PTS_UTC_212_002	2.1.2. Update	Card.txt 파일이 지정한 상대경로에 존재하지 않는다면, 파일열기 실패를 출력하면서 프로그램이 종료된다.
PTS_UTC_213_000	2.1.3. Display	적합한 CID 값을 입력받아서, card_info 값을 갱신한 후에는, 충전된 카드의 정보와 충전한 금액을 입력 받았을 때, 충전한 시각을 구한 후, 충전한 시각, 충전 전의 금액, 충전 한 금액, 충전 후의 금액을 보여준다.

PTS.UTC.213.001	2.1.3. Display	적합한 CID 값을 입력받지 못해서, card_info 값을 0 으로 모두 초기화한 카드가 충전된 정보와 충전한 금액을 입력 받았을 때, 충전한 시각을 구한 후, 충전 한 시각, 충전 전의 금액(=0 원), 충전 한 금액, 충전 후의 금액(=충전 한 금액)을 보여준다.
-----------------	----------------	--

2.2.3 Fee Calculation System

<Table 1.2 Test Design Identification>

		CID_1003.txt 에 저장한다.
FCS.UTC.010.008	1 Card Reader Info Loader	CardReader_info[i].CID==1004 일 때, CardReader_info[i]를 CID_1004.txt 에 저장한다.
FCS.UTC.010.009	1 Card Reader Info Loader	CardReader_info[i].CID==1005 일 때, CardReader_info[i]를 CID_1005.txt 에 저장한다.
FCS.UTC.010.010	1 Card Reader Info Loader	CardReader_info[i].CID==1006 일 때, CardReader_info[i]를 CID_1006.txt 에 저장한다.

Identifier	Feature	Valid/Invalid value
FCS.UTC.010.000	Card Reader Info Loader	Tick()==0 일 때, 각 단말기 파일 != NULL 이면 모든 단말기 파일에 대하여 CardReader_info!=NULL 입력이 들어온다.
FCS.UTC.010.001	Card Reader Info Loader	Tick()==0 일 때, 각 단말기 파일 == NULL 이면 모든 단말기 파일에 대하여 CardReader_info!=NULL 입력이 들어온다.
FCS.UTC.010.002	Card Reader Info Loader	Tick()==1 일 때, 각 단말기 파일 != NULL 이면 모든 단말기 파일에 대하여 CardReader_info!=NULL 입력이 들어온다.
FCS.UTC.010.003	Card Reader Info Loader	Tick()==1 일 때, 각 단말기 파일 == NULL 이면 모든 단말기 파일에 대하여 CardReader_info==NULL 입력이 들어온다.
FCS.UTC.010.004	Card Reader Info Loader	CardReader_info[i].CID==1000 일 때, CardReader_info[i]를 CID_1000.txt 에 저장한다.
FCS.UTC.010.005	Card Reader Info Loader	CardReader_info[i].CID==1001 일 때, CardReader_info[i]를 CID_1001.txt 에 저장한다.
FCS.UTC.010.006	Card Reader Info Loader	CardReader_info[i].CID==1002 일 때, CardReader_info[i]를 CID_1002.txt 에 저장한다.
FCS.UTC.010.007	Card Reader Info Loader	CardReader_info[i].CID==1003 일 때, CardReader_info[i]를

FCS.UTC.010.011	1 Card Reader Info Loader	CardReader_info[i].CID==NULL 일 때, CardReader_info[i]를 CID_*.txt 에 저장한다.
FCS.UTC.021.000	2.1 Fee Calculation Controller	AdjustStart 상태 일 때, c_well==1 을 입력한다.
FCS.UTC.021.001	2.1 Fee Calculation Controller	AdjustStart 상태 일 때, c_well==0 을 입력한다.
FCS.UTC.021.002	2.1 Fee Calculation Controller	GuestOut 상태 일 때, state==1 이면 getout==1 을 입력한다.
FCS.UTC.021.003	2.1 Fee Calculation Controller	GuestOut 상태 일 때, state==1 이면 getout==0 을 입력한다.
FCS.UTC.021.004	2.1 Fee Calculation Controller	GuestOut 상태 일 때, state==0 이면 getout==1 을 입력한다.
FCS.UTC.021.005	2.1 Fee Calculation Controller	CID_Sort 상태 일 때, CID_*.txt != NULL 이면 CID_*.txt 데이터를 CardReader_info 배열에 저장한다.
FCS.UTC.021.006	2.1 Fee Calculation Controller	CID_Sort 상태 일 때, CID_*.txt == NULL 이면 CID_*.txt 데이터를 CardReader_info 배열에 저장한다.
FCS.UTC.021.007	2.1 Fee Calculation Controller	CID_Sort 상태 일 때, CardReader_info 배열이 !=NULL 이면 tagTime 이 작은 순으로 real_CardReader_info 배열에 저장한다.
FCS.UTC.021.008	2.1 Fee Calculation Controller	CID_Sort 상태 일 때, CardReader_info 배열이 ==NULL 이면 tagTime 이 작은 순으로 real_CardReader_info 배열에 저장한다.
FCS.UTC.021.009	2.1 Fee Calculation Controller	UnAdjust 상태 일 때, real_CarfReader_info!=NULL, line!=NULL, &bus_fee!=NULL, &metro_fee!=NULL 이면 Adjust 를 실행한다.
FCS.UTC.021.010	2.1 Fee Calculation Controller	UnAdjust 상태 일 때, real_CarfReader_info==NULL line==NULL &bus_fee==NULL &metro_fee==NULL 이 면 Adjust 를 실행한다.
FCS.UTC.021.011	2.1 Fee Calculation Controller	Adjust 상태 일 때, i>0, real_CardReader_info[i].price>1050 면 real_Card_info[i].price += real_Card_info[i].price-1050, real_Card_info[i].price-=1050 을 실행한다.
FCS.UTC.021.012	2.1 Fee Calculation Controller	Adjust 상태 일 때, i>0, real_CardReader_info[i].price>1050 면 real_Card_info[i].price += real_Card_info[i].price-1050, real_Card_info[i].price-=1050 을 실행하지 않는다.

FCS.UTC.021.013	2.1 Fee Calculation Controller	Adjust 상태 일 때, (배열의 처음) (배열의 마지막) (state==1, transfer==0)이면 다음 state==1, transfer==0 인
		전까지를 하거나 state==1, transfer==0 인 곳이 없으면 배열의 처음이나 마지막이면 처음이나 마지막을 정산 계산을 할 구간을 정해서 index 에 배열 인덱스를 넣는다.
FCS.UTC.021.014	2.1 Fee Calculation Controller	Adjust 상태 일 때, (배열의 처음) (배열의 마지막) (state==1, transfer==0)이면 다음 state!=1, transfer!=0 인 전까지 정산 계산을 할 구간을 정해서 index 에 배열 인덱스를 넣는다.
FCS.UTC.021.015	2.1 Fee Calculation Controller	Adjust 상태 일 때, index!=NULL 이면 real_fee+= real_CardReader_info[i].price 를 수행한다.
FCS.UTC.021.016	2.1 Fee Calculation Controller	Adjust 상태 일 때, index==NULL 이면 real_fee+= real_CardReader_info[i].price 를 수행한다.
FCS.UTC.021.017	2.1 Fee Calculation Controller	Adjust 상태 일 때, real_CardReader_info[i-1].tp != real_CardReader_info[i].tp 이면 total_fee+=real_fee 를 수행한다.
FCS.UTC.021.018	2.1 Fee Calculation Controller	Adjust 상태 일 때, real_CardReader_info[i-1].tp == real_CardReader_info[i].tp 이면 total_fee+=real_fee 를 수행한다.
FCS.UTC.021.019	2.1 Fee Calculation Controller	Adjust 상태 일 때, index!=NULL 이면 temp_fee+= real_CardReader_info[i].price 를 수행한다.
FCS.UTC.021.020	2.1 Fee Calculation Controller	Adjust 상태 일 때, index==NULL 이면 temp_fee+= real_CardReader_info[i].price 를 수행한다.
FCS.UTC.021.021	2.1 Fee Calculation Controller	Adjust 상태 일 때, ((real_CardReader_info[j-1].tp != real_CardReader_info[j].tp) i==j), real_CardReader_info[j].tp==0 이면 *bus_fee+=temp_fee/total_fee*real_fee 를 수행한다.
FCS.UTC.021.022	2.1 Fee Calculation Controller	Adjust 상태 일 때, ((real_CardReader_info[j-1].tp == real_CardReader_info[j].tp) i!=j), real_CardReader_info[j].tp==0 이면 *bus_fee+=temp_fee/total_fee*real_fee 를 수행한다.

FCS.UTC.021.023	2.1 Fee Calculation Controller	Adjust 상태 일 때, ((real_CardReader_info[j-1].tp != real_CardReader_info[j].tp) i==j), real_CardReader_info[j].tp==1 이면 *metro_fee+=temp_fee/total_fee*real_fee 를 수행한다.
FCS.UTC.021.024	2.1 Fee Calculation Controller	Adjust 상태 일 때, ((real_CardReader_info[j-1].tp == real_CardReader_info[j].tp) i!=j), real_CardReader_info[j].tp==1 이면 *metro_fee+=temp_fee/total_fee*real_fee 를 수행한다.
FCS.UTC.021.025	2.1 Fee Calculation Controller	Index 구간이 끝나면, index=i-1, total_fee=0, real_fee=0, temp_fee=0 를 수행한다.
FCS.UTC.021.026	2.1 Fee Calculation Controller	Index 구간이 끝나지 않아도, index=i-1, total_fee=0, real_fee=0, temp_fee=0 를 수행한다.
FCS.UTC.021.027	2.1 Fee Calculation Controller	Adjus 상태가 끝나고, bus_fee!=NULL, metro_fee!=NULL 이면 c_well=0 을 수행한다.
FCS.UTC.021.028	2.1 Fee Calculation Controller	Adjus 상태가 끝나고, bus_fee==NULL, metro_fee==NULL 이 면 c_well=0 을 수행한다.
FCS.UTC.021.029	2.1 Fee Calculation Controller	c_well==0 면 enable Display.
FCS.UTC.021.030	2.1 Fee Calculation Controller	c_well!=0 면 enable Display.
FCS.UTC.021.031	2.1 Fee Calculation Controller	c_well==0 이면 Trigger Send
FCS.UTC.021.032	2.1 Fee Calculation Controller	c_well!=0 이면 Trigger Send
FCS.UTC.021.033	2.1 Fee Calculation Controller	c_well==0 이면 Trigger Reset
FCS.UTC.021.034	2.1 Fee Calculation Controller	c_well!=0 이면 Trigger Reset
FCS.UTC.022.000	2.2 Display	Enable 이 들어오고 c_well==0 이면 bus_fee, metro_fee, t_now 를 print 한다
FCS.UTC.022.001	2.2 Display	Enable 이 들어오고 c_well!=0 이면 bus_fee, metro_fee, t_now 를 print 한다
FCS.UTC.022.002	2.2 Display	Enable 이 들어오고 c_well!=0 이면 bus_fee, metro_fee, t_now 를 print 안한다
FCS.UTC.023.000	2.3 Send	Trigger 가 들어오고 c_well==0 이면 c_well, bus_fee, metro_fee 를 지하철 회사, 버스 회사에 보낸다.

FCS.UTC.023.001	2.3 Send	Trigger 가 들어오고 c_well==0 이면 c_well, bus_fee, metro_fee 를 지하철 회사, 버스 회사에 보낸다.
FCS.UTC.023.002	2.3 Send	Trigger 가 들어오고 c_well!=0 이면 c_well, bus_fee, metro_fee 를 지하철 회사, 버스 회사에 안보낸다.
FCS.UTC.024.000	2.4 Reset	Trigger 가 들어오고 c_well == 0 이면 모든 단말기파일을 초기화한다.
FCS.UTC.024.001	2.4 Reset	Trigger 가 들어오고 c_well != 0 이면 모든 단말기파일을 초기화한다.
FCS.UTC.024.002	2.4 Reset	Trigger 가 들어오고 c_well != 0 이면 모든 단말기파일을 초기화하지 않는다.

2.3 Input specifications

2.3.1 Public Transportation System

<Table 1.1 Test Case Identification> 참조

2.3.2 Recharger System

<Table 2.1 Test Case Identification> 참조

2.3.3 Fee Calculation System

<Table 3.1 Test Case Identification> 참조

2.4 Output specifications

2.4.1 Public Transportation System

<Table 1.1 Test Case Identification> 참조

2.4.2 Recharger System

<Table 2.1 Test Case Identification> 참조

2.4.3 Fee Calculation System

<Table 3.1 Test Case Identification> 참조

3 Environmental needs

T3-2014-PTS-UTP-1.0 Environmental needs 항목 참조

4 Unit test summary report

4.1 Test summary report identifier

4.1.1 Public Transportation System

PTS - Public Transportation System 버스 승차단말기(Card Leader) - CRID : 01 태그하실 카드의 ID를 입력해 주세요. : 1010 승차 단말기에 태그하였습니다. 카드 상태가 하차상태입니다. 카드 상태가 환승이 아닙니다. 하차태그 후 15초 이후에 태그하였습니다. 1050 태그 한 시각 : 2014년 11월 21일 06시 53분 40초 태그시 부과금액: 1050 태그 후의 잔액 : 21450	PTS - Public Transportation System 버스 하차단말기(Card Leader) - CRID : 00 태그하실 카드의 ID를 입력해 주세요. : 1010 하차 단말기에 태그하였습니다. 카드 상태가 승차상태입니다. 카드 상태가 환승이 아닙니다. 버스 단말기를 태그하였습니다. 태그 한 시각 : 2014년 11월 21일 06시 53분 46초 태그시 부과금액: 0 태그 후의 잔액 : 21450
PTS - Public Transportation System 건대입구역 승차단말기(Card Leader) - CRID : 11 태그하실 카드의 ID를 입력해 주세요. : 1010 승차 단말기에 태그하였습니다. 카드 상태가 하차상태입니다. 카드 상태가 환승이 아닙니다. 하차태그 후 15초 이내에 태그하였습니다. 버스 -> 지하철 태그 600 환승입니다. 태그 한 시각 : 2014년 11월 21일 06시 53분 51초 태그시 부과금액: 0 태그 후의 잔액 : 21450	PTS - Public Transportation System 신림역 하차단말기(Card Leader) - CRID : 30 태그하실 카드의 ID를 입력해 주세요. : 1010 하차 단말기에 태그하였습니다. 카드 상태가 승차상태입니다. 카드 상태가 환승상태입니다. 지하철 단말기에 태그하였습니다. 2개역을 이동하였습니다. 태그 한 시각 : 2014년 11월 21일 06시 53분 59초 태그시 부과금액: 600 태그 후의 잔액 : 20850
PTS - Public Transportation System 신림역 승차단말기(Card Leader) - CRID : 31 태그하실 카드의 ID를 입력해 주세요. : 1010 승차 단말기에 태그하였습니다. 카드 상태가 하차상태입니다. 카드 상태가 환승이 아닙니다. 하차태그 후 15초 이내에 태그하였습니다. 지하철 -> 지하철, 버스 -> 버스 태그 1050 태그 한 시각 : 2014년 11월 21일 06시 54분 03초 태그시 부과금액: 1050 태그 후의 잔액 : 19800	PTS - Public Transportation System 건대입구역 하차단말기(Card Leader) - CRID : 10 태그하실 카드의 ID를 입력해 주세요. : 1010 하차 단말기에 태그하였습니다. 카드 상태가 승차상태입니다. 카드 상태가 환승이 아닙니다. 지하철 단말기를 태그했습니다. 2개역을 이동하였습니다. 태그 한 시각 : 2014년 11월 21일 06시 54분 06초 태그시 부과금액: 200 태그 후의 잔액 : 19600

4.1.2 Recharger System

CID	CRID	tagTime	tp	state	cash	transfer	getout
1000	11	2014 11 21 6 23 53	1	1	50900	0	0
1001	11	2014 11 16 3 13 26	1	1	6250	0	0
1002	10	2014 11 3 8 26 30	0	0	1000	0	0
1003	11	2014 11 4 23 30 44	1	1	8450	0	1
1004	11	2014 11 4 23 36 52	1	1	28750	0	0
1005	11	2014 11 16 3 13 37	1	1	17200	0	1
1006	30	2014 11 3 8 27 30	1	0	20000	0	0
1007	31	2014 11 3 8 27 45	1	1	20000	1	0
1008	40	2014 11 3 8 28 0	1	0	20000	0	1
1009	41	2014 11 3 8 28 15	1	1	50000	1	0
1010	11	2014 11 16 2 19 12	1	1	101250	0	1

```

C:\windows\system32\cmd.exe
SoftwareEngineering Team Project
PTS - Recharger System
충전하실 카드의 ID를 입력해 주세요. : 39.384sbfh25nga0
CID : 39384250
카드에 충전하실 금액을 입력해 주세요. : nj3q.3jw.3j000
충전 한 시각 : 2014년 11월 21일 06시 31분 52초
충전 전의 금액 : 0 원
충전 한 금액 : 333000 원
충전 후의 금액 : 333000 원
계속 충전하시겠습니까? Y<Yes>, N<No> :

```

CID	CRID	tagTime	tp	state	cash	transfer	getout
1000	11	2014 11 21 6 23 53	1	1	50900	0 0	
1001	11	2014 11 16 3 13 26	1	1	6250	0 0	
1002	10	2014 11 3 8 26 30	0	0	1000	0 0	
1003	11	2014 11 4 23 30 44	1	1	8450	0 1	
1004	11	2014 11 4 23 36 52	1	1	28750	0 0	
1005	11	2014 11 16 3 13 37	1	1	17200	0 1	
1006	30	2014 11 3 8 27 30	1	0	20000	0 0	
1007	31	2014 11 3 8 27 45	1	1	20000	1 0	
1008	40	2014 11 3 8 28 0	1	0	20000	0 1	
1009	41	2014 11 3 8 28 15	1	1	50000	1 0	
1010	11	2014 11 16 2 19 12	1	1	101250	0 1	

```

C:\windows\system32\cmd.exe
SoftwareEngineering Team Project
PTS - Recharger System
충전하실 카드의 ID를 입력해 주세요. : 100

CID : 100
카드에 충전하실 금액을 입력해 주세요. : 20000
충전 한 시각 : 2014년 11월 21일 06시 30분 49초
충전 전의 금액 : 0 원
충전 한 금액 : 20000 원
충전 후의 금액 : 20000 원

계속 충전하시겠습니까? Y<Yes>, N<No> :

```

CID	CRID	tagTime	tp	state	cash	transfer	getout
1000	11	2014 11 21 6 23 53	1	1	50900	0 0	
1001	11	2014 11 16 3 13 26	1	1	6250	0 0	
1002	10	2014 11 3 8 26 30	0	0	1000	0 0	
1003	11	2014 11 4 23 30 44	1	1	8450	0 1	
1004	11	2014 11 4 23 36 52	1	1	28750	0 0	
1005	11	2014 11 16 3 13 37	1	1	17200	0 1	
1006	30	2014 11 3 8 27 30	1	0	20000	0 0	
1007	31	2014 11 3 8 27 45	1	1	20000	1 0	
1008	40	2014 11 3 8 28 0	1	0	20000	0 1	
1009	41	2014 11 3 8 28 15	1	1	50000	1 0	
1010	11	2014 11 16 2 19 12	1	1	71250	0 1	

```

C:\windows\system32\cmd.exe
SoftwareEngineering Team Project
PTS - Recharger System
충전하실 카드의 ID를 입력해 주세요. : 1010

CID : 1010
카드에 충전하실 금액을 입력해 주세요. : 30000
충전 한 시각 : 2014년 11월 21일 06시 30분 17초
충전 전의 금액 : 71250 원
충전 한 금액 : 30000 원
충전 후의 금액 : 101250 원

계속 충전하시겠습니까? Y<Yes>, N<No> :

```

CID	CRID	tagTime	tp	state	cash	transfer	getout
1000	11	2014 11 21 6 23 53	1	1	50900	0	0
1001	11	2014 11 16 3 13 26	1	1	6250	0	0
1002	10	2014 11 3 8 26 30	0	0	1000	0	0
1003	11	2014 11 4 23 30 44	1	1	8450	0	1
1004	11	2014 11 4 23 36 52	1	1	28750	0	0
1005	11	2014 11 16 3 13 37	1	1	7200	0	1
1006	30	2014 11 3 8 27 30	1	0	20000	0	0
1007	31	2014 11 3 8 27 45	1	1	20000	1	0
1008	40	2014 11 3 8 28 0	1	0	20000	0	1
1009	41	2014 11 3 8 28 15	1	1	50000	1	0
1010	11	2014 11 16 2 19 12	1	1	71250	0	1


```

C:\windows\system32\cmd.exe
SoftwareEngineering Team Project
PTS - Recharger System
충전하실 카드의 ID를 입력해 주세요. : 1005

CID : 1005
카드에 충전하실 금액을 입력해 주세요. : 10000
충전 한 시각 : 2014년 11월 21일 06시 29분 30초
충전 전의 금액 : 7200 원
충전 한 금액 : 10000 원
충전 후의 금액 : 17200 원

계속 충전하시겠습니까? Y<Yes>, N<No> :

```


CID	CRID	tagTime	tp	state	cash	transfer	getout
1000	11	2014 11 21 6 23 53	1	1	30900	0	0
1001	11	2014 11 16 3 13 26	1	1	6250	0	0
1002	10	2014 11 3 8 26 30	0	0	1000	0	0
1003	11	2014 11 4 23 30 44	1	1	8450	0	1
1004	11	2014 11 4 23 36 52	1	1	28750	0	0
1005	11	2014 11 16 3 13 37	1	1	7200	0	1
1006	30	2014 11 3 8 27 30	1	0	20000	0	0
1007	31	2014 11 3 8 27 45	1	1	20000	1	0
1008	40	2014 11 3 8 28 0	1	0	20000	0	1
1009	41	2014 11 3 8 28 15	1	1	50000	1	0
1010	11	2014 11 16 2 19 12	1	1	71250	0	1


```

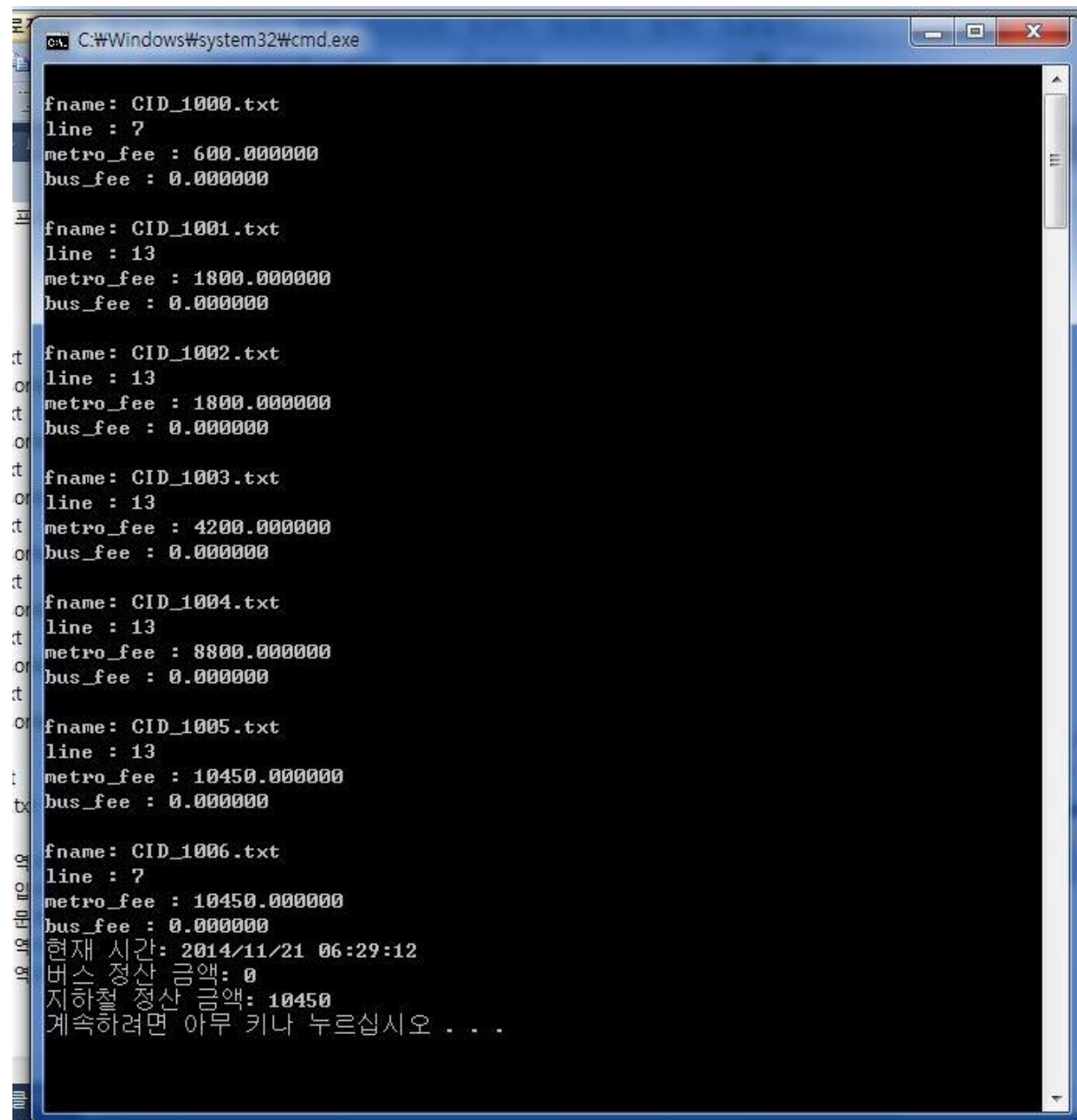
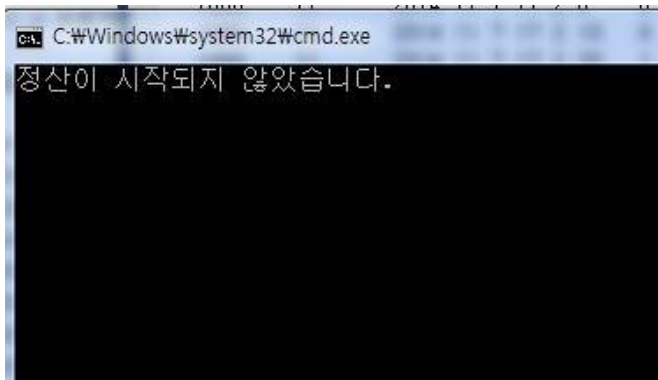
C:\windows\system32\cmd.exe
SoftwareEngineering Team Project
PTS - Recharger System
충전하실 카드의 ID를 입력해 주세요. : 1000

CID : 1000
카드에 충전하실 금액을 입력해 주세요. : 20000
충전 한 시각 : 2014년 11월 21일 06시 27분 54초
충전 전의 금액 : 30900 원
충전 한 금액 : 20000 원
충전 후의 금액 : 50900 원

계속 충전하시겠습니까? Y<Yes>, N<No> :

```

4.1.3 Fee Calculation System



CID	CRID	tagTime	tp	state	price	transfer	getout
1000	11	2014 11 7 17 2 0	0	1	100 0		0
1000	11	2014 11 7 17 2 10	0	0	100 0		0
1000	11	2014 11 7 17 2 20	1	1	100 1		0
1000	11	2014 11 7 17 2 30	1	0	100 0		0
1000	11	2014 11 7 17 2 40	0	1	100 0		0
1000	11	2014 11 7 17 2 57	0	0	100 0		0

```

C:\Windows\system32\cmd.exe

fname: CID_1000.txt
line : 7
metro_fee : 133.333328
bus_fee : 466.666656

fname: CID_1001.txt
line : 13
metro_fee : 1333.333252
bus_fee : 466.666656

fname: CID_1002.txt
line : 13
metro_fee : 1333.333252
bus_fee : 466.666656

fname: CID_1003.txt
line : 13
metro_fee : 3733.333252
bus_fee : 466.666656

fname: CID_1004.txt
line : 13
metro_fee : 8333.333008
bus_fee : 466.666656

fname: CID_1005.txt
line : 13
metro_fee : 9983.333008
bus_fee : 466.666656

fname: CID_1006.txt
line : 7
metro_fee : 9983.333008
bus_fee : 466.666656
현재 시간: 2014/11/21 06:35:03
버스 정산 금액: 466
지하철 정산 금액: 9983
계속하려면 아무 키나 누르십시오 . . .

```

4.2 Evaluation

Test Tools 다를 줄 몰라서 CMD 창에서 작동하는 모습으로 평가했다. 그래서 Test Tools

를 사용해 테스트 평가를 못한 것이 아쉽다.