

Unit Test Report for PTS System

- Test Cases Specification
- Test Summary Report

Project Team

Team 6

Latest update on:

2014-11-30

Team Information

김창규(201110202)

김민우(201111339)

배장길(201111359)

전진영(201111382)

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	4
2.3	Input specifications.....	10
2.4	Output specifications.....	11
3	Environmental needs.....	11
4	Unit test summary report	11
4.1	Test summary report identifier.....	11
4.2	Evaluation.....	13

1 Introduction

1.1 Objectives

본 문서는 Public Transportation System (PTS)의 unit test를 수행한 결과에 대한 report 문서이다. Test요소들에 대한 Test case와 Test수행 결과에 대한 내용을 담고있다.

1.2 References

2014SE_B_TP2_T6_SRA_Rev_2.3

(http://dslab.konkuk.ac.kr/Class/2014/14SE/Team_Project_B/TP2/2014SE_B_TP2_T6_SRA_Rev_2.3.pdf)

2014SE_B_TP2_T6_SDS_Rev_1.2

(http://dslab.konkuk.ac.kr/Class/2014/14SE/Team_Project_B/TP2/2014SE_B_TP2_T6_SDS_Rev_1.2.pdf)

2014SE_B_TP2_T6_UTP

실습 수업 주제(SRS)

(<http://dslab.konkuk.ac.kr/Class/2014/14SE/PTS%20SRS%20Ver%201.0.pdf>)

2 Unit test case specification

2.1 Test case specification identifier

Table 1 Test Design Identification

<Card Tagger>		
Identifier	Feature(Process ID in DFD)	Valid / Invalid value
PTS.UTC_000	1 Card Reader Interface	카드를 읽어서 단말기ID와 Time Tick로부터 들어온 태그시간은 Data Store에 저장
PTS.UTC_001	2.1 Get on/off Process	카드정보와 단말기 정보를 받아서 승하차 여부 확인
PTS.UTC_002	2.2 Transfer Process	카드정보와 단말기 정보를 받아서 환승 여부 확인

PTS.UTC_003	2.3 Balance Process	카드정보와 단말기 정보를 받아서 미정산 여부 확인
PTS.UTC_004	2.4 T.D Process	카드정보와 단말기 정보를 받아서 대중교통종류 확인
PTS.UTC_005	2.5 F.T Process	카드정보와 단말기 정보를 받아서 부과될 요금과 Writer info를 return
PTS.UTC_006	3.1 B.D Control	Charge를 바탕으로 승차의 승인여부 판단
PTS.UTC_007	3.2 Saving	Enable신호를 받아서 데이터를 단말기에 저장
PTS.UTC_008	3.5 Send Process	Card Interface에 Writer Info, Charge를 전송
PTS.UTC_009	4 LED Interface	들어온 정보를 LED에 표시

<Calculating>		
Identifier	Feature(Process ID in DFD)	Valid / Invalid value
PTS.UTC_010	1.6 Time Check	Tag Stop / Data Receive Process로 Trigger를 보내 동작 확인
PTS.UTC_011	1.7 Data Receive Process	데이터를 받아서 정리한 후 return 확인.
PTS.UTC_012	1.1 Classify Process	정산된 금액을 버스금액과 지하철금액으로 분류 확인
PTS.UTC_013	1.2 Added Charge	분류된 각각의 버스금액과 지하철 금액을 하나의 버스금액과 지하철 금액으로 더함 확인

2.2 Test items

Table 2 Test Case Identification

<Card Tagger>		
Identifier	Input specification	Output specification
PTS.UTC_000_000	//structure중 tag == 0인경우	//Structure(Tag info)를 다음 데이터스토어로 전송하는 함수 실행
PTS.UTC_000_001	//structure중 tag == 1인경우	//Structure(Tag info)를 다음 데이터스토어로 전송하는 함수 실행

PTS.UTC_000_002	//structure중 tag == 2인경우	//Structure(Tag info)를 다음 데이터스토어로 전송하는 함수 실행
PTS.UTC_000_003	//structure중 tag == 3인경우	//Structure(Tag info)를 다음 데이터스토어로 전송하는 함수 실행
PTS.UTC_000_004	//structure중 tag == 4인경우	//Structure(Tag info)를 다음 데이터스토어로 전송하는 함수 실행
PTS.UTC_000_005	//structure중 tag == 5인경우	//Structure(Tag info)를 다음 데이터스토어로 전송하는 함수 실행
PTS.UTC_000_006	//structure중 tag != 0,1,2,3,4,5인경우	//Structure(Tag info)를 다음 데이터스토어로 전송하는 함수 미실행
PTS.UTC_001_000	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?}; </pre>	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, ? , ?}; </pre>
PTS.UTC_001_001	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?}; </pre>	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,OUT, BUS_0, ? , ?}; </pre>
PTS.UTC_001_002	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,?, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?}; </pre>	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,?, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?}; </pre>
PTS.UTC_002_000	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 22,01,02, BUS,IN, BUS_0, ? , ?}; </pre>	tginfo->transfer=false
PTS.UTC_002_001	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 20,01,02, BUS,IN, BUS_0, ? , ?}; </pre>	tginfo->transfer=false

PTS.UTC_002_002	tginfo->cardInfo = {2014, 11 , 21, 01,01 , BUS,OUT, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21, 03,02 , SUB,IN, SUB_0, ? , ?};	tginfo->transfer=false
PTS.UTC_002_003	tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT , BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,01,02, BUS,IN , BUS_0, ? , ?};	tginfo->transfer=false
PTS.UTC_002_004	tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,OUT , SUB_0, 0 , 10}; tginfo->tagInfo = {2014, 11 , 21,01,02, BUS,IN, BUS_0, ? , ?};	tginfo->transfer=false
PTS.UTC_002_005	tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT , BUS_0, 0 , 10}; tginfo->tagInfo = {2014, 11 , 21,01,02, SUB,IN, SUB_0, ? , ?};	tginfo->transfer=false
PTS.UTC_002_006	tginfo->cardInfo = {2014, 11 , 21,01,01, SUB, IN ,SUB_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS, IN , BUS_0, ? , ?};	tginfo->transfer=false
PTS.UTC_002_007	tginfo->cardInfo = {2014, 11 , 21,01,01, SUB, OUT ,SUB_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS, OUT , BUS_0, ? , ?};	tginfo->transfer=false
PTS.UTC_002_008	tginfo->cardInfo = {2014, 11 , 21,01,01, SUB, IN ,SUB_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS, OUT , BUS_0, ? , ?};	tginfo->transfer=false
PTS.UTC_002_009	tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,01,02, SUB,IN, SUB_0, ? , ?};	tginfo->transfer=true
PTS.UTC_003_000	tginfo->cardInfo = { 2014, 11 , 21 ,01,01, BUS,IN , BUS_0,1000, 10}; tginfo->tagInfo = { 2014, 11 , 21 ,02,02, BUS,IN , BUS_0, ? , ?};	tginfo->yetFee = true
PTS.UTC_003_001	tginfo->cardInfo = { 2014, 11 , 21 ,01,01,BUS,IN, BUS_0,1000, 10};	tginfo->yetFee = true

	tginfo->tagInfo = { 2014, 11 , 22,01,01,BUS,IN , BUS_0, ? , ?};	
PTS.UTC_003_002	tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN , BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 22,01,01, BUS,IN , BUS_0, ? , ?};	tginfo->yetFee = false
PTS.UTC_003_003	tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT , BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,OUT , BUS_0, ? , ?};	tginfo->yetFee = false
PTS.UTC_003_004	tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT , BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN , BUS_0, ? , ?};	tginfo->yetFee = false
PTS.UTC_003_005	tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN , BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,OUT , BUS_0, ? , ?};	tginfo->yetFee = false
PTS.UTC_004_000	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , BUS_0, ? , ?};	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , BUS , BUS_0, ? , ?};
PTS.UTC_004_001	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , SUB_0, ? , ?};	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , SUB , SUB_0, ? , ?};
PTS.UTC_004_002	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , SUB_1, ? , ?};	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , SUB , SUB_1, ? , ?};
PTS.UTC_004_003	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , SUB_2, ? , ?};	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , SUB , SUB_2, ? , ?};
PTS.UTC_004_004	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , SUB_3, ? , ?};	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , SUB , SUB_3, ? , ?};
PTS.UTC_004_005	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , SUB_4, ? , ?};	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , SUB , SUB_4, ? , ?};
PTS.UTC_004_006	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , BUS_5, ? , ?};	tginfo->tagInfo = {2014, 11 , 21,02,02, ? , ? , BUS_5, ? , ?};
PTS.UTC_005_001	tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, true , ?}; tginfo.transfer = { ? , ? , true, true, ? , ? , ? }	wrinfo.fee = 0;
PTS.UTC_005_002	tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, true , ?};	wrinfo.fee = 1650;

	<pre> tginfo.transfer = { ?, ? ,false, true, ?, ? , ?} tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_0,5000, 10}; </pre>	
PTS.UTC_005_003	<pre> tginfo.transfer = { ?, ? ,false, true, ?, ? , ?} tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,5000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, true , ?}; </pre>	wrinfo.fee = 1750;
PTS.UTC_005_004	<pre> tginfo.transfer = { ?, ? ,false, false, ?, ? , ?} tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, true , ?}; </pre>	wrinfo.fee = 1050;
PTS.UTC_005_005	<pre> tginfo.transfer = { ?, ? ,true, false, ?, ? , ?} tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,OUT, BUS_0, true , ?}; tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,5000, 10}; </pre>	wrinfo.fee = 1150;
PTS.UTC_005_006	<pre> tginfo.transfer = { ?, ? ,true, true, ?, ? , ?} tginfo->tagInfo = {2014, 11 , 21,02,02, SUB,IN, SUB_0, true , ?}; tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,5000, 10}; </pre>	wrinfo.fee = 0;
PTS.UTC_005_007	<pre> tginfo.transfer = { ?, ? ,false, true, ?, ? , ?} tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,IN, SUB_0, true , ?}; tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,5000, 10}; </pre>	wrinfo.fee = 1750;
PTS.UTC_005_008	<pre> tginfo.transfer = { ?, ? ,false, false, ?, ? , ?} tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,IN, SUB_0, true , ?}; tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_0,5000, 10}; </pre>	wrinfo.fee = 1650;

PTS.UTC_005_009	<pre> tginfo.transfer = { ?, ? ,false, false, ?, ? , ?} tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,IN, SUB_0, true , ?}; tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,OUT, SUB_0,5000, 10}; </pre>	wrinfo.fee = 1050;
PTS.UTC_005_010	<pre> tginfo.transfer = { ?, ? ,false, false, ?, ? , ?} tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,OUT, SUB_0, true , ?}; tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_2,5000, 10}; </pre>	wrinfo.fee = 1250;
PTS.UTC_005_011	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_2,5000, 10}; </pre>	tginfo->board_bool = true;
PTS.UTC_005_012	<pre> tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_2,500, 10}; </pre>	tginfo->board_bool = false;
PTS.UTC_006_000	<pre> tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, true , ?}; </pre>	<pre> trigger = true; sendApproval saveData sendWriterInfo 함수 호출 </pre>
PTS.UTC_006_001	<pre> tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, false , ?}; </pre>	<pre> trigger = false; sendApproval saveData sendWriterInfo 함수 호출 </pre>
PTS.UTC_007_000	<pre> cardinfo taginfo writing.info </pre>	BUS_0.txt
PTS.UTC_007_001	<pre> cardinfo taginfo writing.info </pre>	SUB_0~5.txt
PTS.UTC_008_000	Charge , WRITTERINFO wrinfo	Checkboarding(int charge, WRITTERINFO tginfo) 함수 호출
PTS.UTC_009_000	WRITTERINFO wrinfo ,charge	화면에 tginfo->tagInfo.year, tginfo->tagInfo.month, tginfo->tagInfo.day, tginfo->tagInfo.hour, tginfo->tagInfo.minute, 교통수단, 탑승또는하차,inputinfo.fee, inputinfo.balance 를 printf를 써서 출력

<Calculating>		
Identifier	Input specification	Output specification
PTS.UTC_010_000	Hour,minute hour==20 && minute==0	classifyData, receive data 호출
PTS.UTC_010_001	Hour,minute 0<=hour<20,	어떠한 함수도 호출 하지 않음
PTS.UTC_010_002	Hour,minute Hour ==0,minute==0	deleteData 및 restartTag 호출
PTS.UTC_011_000	(BUS.txt SUB1.txt SUB2.txt SUB3.txt SUB4.txt SUB5.txt) read success	Structure Node(Linked List) Node.Data = { 시간 (yyyymmddhhmmss), 교통수단, 승하차,결제요금, 단말기ID}
PTS.UTC_011_001	(BUS.txt SUB1.txt SUB2.txt SUB3.txt SUB4.txt SUB5.txt) read fail	NULL
PTS.UTC_012_000	text파일을 읽어온 structure Node(linked list의 구조)	structure Node 2개 BusList.Data, SubList.Data {시간(yyyymmddhhmmss), BUS or SUB, 승하차(IN or OUT), 결 제요금(int), 단말기 ID(char*)} added charge 함수 호출
PTS.UTC_013_000	structure CLASSIFIEDDATA(BUSLIST,SUBLIST를 가지고 있는 structure)중 charge >= 0	struct OVERALLBUSCHARGE {시간(yyyymmdd),"BUS", 결제요금(int) } struct OVERALLSUBCHARGE {시간, "SUB", 결제요금 } 두 스트럭트가반환되며 sendBusCharge, sendSubcharge 함수 호출
PTS.UTC_013_001	structure CLASSIFIEDDATA(BUSLIST,SUBLIST를 가지고 있는 structure)중 charge < 0	어떠한 함수도 호출 하지 않음

2.3 Input specifications

Table1 Test Design Identification 참조

2.4 Output specifications

Table1 Test Design Identification 참조

3 Environmental needs

Hardware & Platform, Cygwin, Vi, GCC

4 Unit test summary report

4.1 Test summary report identifier

- 그림 1 Card Tagger Unit test Result 참조

PTS.UTC_000_000	passed
PTS.UTC_000_001	Passed
PTS.UTC_000_002	passed
PTS.UTC_000_003	Passed
PTS.UTC_000_004	passed
PTS.UTC_000_005	passed
PTS.UTC_000_006	passed
PTS.UTC_001_000	Passed
PTS.UTC_001_001	passed
PTS.UTC_001_002	Passed
PTS.UTC_002_000	passed
PTS.UTC_002_001	passed
PTS.UTC_002_002	passed
PTS.UTC_002_003	Passed
PTS.UTC_002_004	passed
PTS.UTC_002_005	Passed
PTS.UTC_002_006	passed
PTS.UTC_002_007	passed
PTS.UTC_002_008	passed
PTS.UTC_002_009	Passed
PTS.UTC_003_000	passed
PTS.UTC_003_001	Passed
PTS.UTC_003_002	passed
PTS.UTC_003_003	passed
PTS.UTC_003_004	passed
PTS.UTC_003_005	Passed

PTS.UTC_004_000	passed
PTS.UTC_004_001	Passed
PTS.UTC_004_002	passed
PTS.UTC_004_003	passed
PTS.UTC_004_004	passed
PTS.UTC_004_005	Passed
PTS.UTC_004_006	passed
PTS.UTC_005_001	Passed
PTS.UTC_005_002	passed
PTS.UTC_005_003	passed
PTS.UTC_005_004	passed
PTS.UTC_005_005	Passed
PTS.UTC_005_006	passed
PTS.UTC_005_007	Passed
PTS.UTC_005_008	passed
PTS.UTC_005_009	passed
PTS.UTC_005_010	passed
PTS.UTC_005_011	Passed
PTS.UTC_005_012	passed
PTS.UTC_006_000	Passed
PTS.UTC_006_001	passed
PTS.UTC_007_000	passed
PTS.UTC_007_001	Passed
PTS.UTC_008_000	Passed
PTS.UTC_009_000	passed

-

- 그림 2 Calculating Unit test result

PTS.UTC_010_000	Passed
PTS.UTC_010_001	passed
PTS.UTC_010_002	Passed
PTS.UTC_011_000	Passed
PTS.UTC_011_001	passed
PTS.UTC_012_000	Passed
PTS.UTC_013_000	passed
PTS.UTC_013_001	passed

그림 1 Card Tagger Unit test Result

그림 2 Calculating Unit test result

4.2 Evaluation

Total test case : 61개

Passed : 61개

Failed : 0개