

Unit Testing Plan for PTS System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

Team 6

Date

2014-11-30

Team Information

김창규(201110202)

김민우(201111339)

배장길(201111359)

전진영(201111382)

Table of Contents

1	Introduction	4
1.1	Objectives.....	4
1.2	Background	4
1.3	Scope.....	4
1.4	Project plan	4
1.5	Configuration management plan.....	4
1.6	References.....	4
2	Test items	5
3	Features to be tested.....	5
4	Features not to be tested	6
5	Approach.....	7
6	Item pass/fail criteria	7
7	Unit test design specification.....	7
7.1	Test design specification identifier	7
7.2	Features to be tested	7
7.3	Approach refinements.....	8
7.4	Test identification	8
7.5	Feature pass/fail criteria	9
8	Unit test case specification.....	9
8.1	Test case specification identifier	9
8.2	Test items	15

8.3	Input specifications.....	15
8.4	Output specifications.....	15
9	Testing tasks	15
10	Environmental needs	16
11	Unit Test deliverables.....	16
12	Schedules	16

1 Introduction

1.1 Objectives

Public Transportation System (PTS)에서 사용될 SW의 unit test를 위해 필요한 활동과 기준에 대한 정의, 환경적인 요구사항, test도구들에 관해 세부적으로 명세한 문서이다.

1.2 Background

Public Transportation System은 카드 태그, 운행 요금을 정산하는 시스템이다. 이 시스템은 시스템의 작동에 관련된 다양한 상태 값을 가지고 있다.

Unit test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 test이며, 시스템에 관련된 데이터 및 프로세스들이 요구사항을 만족하고 제대로 작동하는지 확인할 수 있는 기본적인 테스트 방법이다.

1.3 Scope

Public Transportation System 중에서 하드웨어와 관련 없이 작동할 수 있는 모듈을 대상으로 Unit test를 수행한다. 이와 관련하여 test를 수행하기 위한 자원과 절차, 접근방법, 기술적인 부분, 환경 및 도구 등을 해당 문서에서 정의한다. 그리고 이를 바탕으로 정의된 요구사항이 만족하는지를 test한다.

1.4 Project plan

1.5 Configuration management plan

1.6 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)

IEEE Std. 8301998

Lab. introduction #1 (SASD)

(<http://dslab.konkuk.ac.kr/Class/2014/14SE/Lecture%20Note/Introduction%20to%20SASD.pdf>)

실습 수업 주제(SRS)

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

2 Test items

PTS를 구성하는 최소단위의 모듈들이 unit test의 대상이 된다. 각 모듈의 요구사항을 만족하는지를 test하며, test item은 다음 자료들로부터 작성되었다.

- Functionality of modules - 2014SE_B_TP2_T6_SRA_Rev_2.3 : Process specification
- Module interface - 2014SE_B_TP2_T6_SRA_Rev_2.3 : Process specification, structure chart

3 Features to be tested

- Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는지를 test한다.
- Modules in SDS : 각 모듈이 가지고 있는 데이터 인터페이스를 test한다. Table 1 테스트할 Process(DFD) 리스트 의 함수이름 참조

Table 1 테스트할 Process(DFD) 리스트

<Card Tagger>		
함수번호	함수이름	함수설명
1	Card Reader Interface	카드를 읽어서 단말기ID와 Time Tick로부터 들어온 태그 시간을 Data store에 저장
2.1	Get on/off Process	카드정보(시간, 교통수단, 승/하차, 잔액, 탑승 단말기 정보)와 단말기 태그 ID를 받아서 승차인지 하차인지의 데이터(On/off info)와 Tag ID, Tag Time 전송
2.2	Transfer Process	카드정보(시간, 교통수단, 승/하차, 잔액, 탑승 단말기 정보)와 단말기 태그 ID를 받아서 필요한 정보를 써서 환승 여부를 판별해서 output으로 전송
2.3	Balance Process	카드정보(시간, 교통수단, 승/하차, 잔액, 탑승 단말기 정보)와 단말기 태그 ID를 받아서 미정산 여부를 확인 후 전송
2.4	T.D Process	카드정보(시간, 교통수단, 승/하차, 잔액, 탑승 단말기 정보)와 단말기 태그 ID를 받아 필요한 정보를 처리하여 버스나 지하철의 단말기인지 판별하여 데이터 전송
2.5	F.T Process	카드정보(시간, 교통수단, 승/하차, 잔액, 탑승 단말기 정보)와 단말기 태그 ID를 받아 부과될 요금과 Writer info를 전송

3.1	B.D Control	Charge(잔액 - 결제금액)을 받아서 Charge값이 0이상 인지 검사 후 Trigger신호를 보냄
3.2	Saving	Trigger 신호를 받아서 데이터(승/하차, 교통수단,결제금액, 태그 시간)를 단말기에 저장
3.5	Send Process	Process 3.1에서 Trigger를 받아서 Card interface에 Writer Info, Charge를 전송
4	LED Interface	들어온 정보들을 종합해서 카드에 기록

<Calculating>		
함수번호	함수이름	함수설명
1.1	Classify Process	분류를 시작하는 데이터를 받아서 Daily Data(태그 시간, 교통수단, 승/하차, 결제 금액, 탑승 단말기 정보)를 가져와서 교통수단에 따라서 데이터를 분류한다.
1.2	Added Charge	분류된 버스요금과 지하철 요금을 각각 합산해서 버스요금 총액(시간, 교통수단, 정산금액)과 지하철 요금 총액(시간, 교통수단, 정산금액)으로 나누어 저장
1.6	Time Check	정산 시작하라는 신호 (2분30초), 정산완료 (하루 끝 3분) 신호를 보낸다
1.7	Data Receive Process	정산 시작 시 단말기로부터 일일 누적 결제 기록을 받아온다.

4 Features not to be tested

- Process in SRA : 외부장치 드라이버, 단순 데이터 전달 프로세스 등은 Test에서 제외한다.
- Modules in SDS : Table 2 테스트 하지 않을 Process(DFD) 리스트 의 함수이름 참조.

Table 2 테스트 하지 않을 Process(DFD) 리스트

<Card Tagger>		
함수번호	함수이름	함수설명
-	Card Reader	Card에 데이터를 읽어온다.
-	Card&TagIDData&Tag time	입력받은 카드정보, 시간, 단말기ID를 저장한다.
-	Daily data	단말기에서 저장되어있는 정보(태그 시간, 교통수단, 승/하차, 결제 금액, 탑승 단말기 정보)를 가지고 있다.
-	Calculating System	Calculating System을 의미
3.3	B.D Possible	Process 3.1에서 Trigger를 받아 Approval(탑승 가능) 신

		호를 LED에 전송
3.4	B.D Impossible	3.1에서 Trigger를 받아 Disapproval(탑승 불가) 신호를 LED에 전송
5	Card Writer Interface	들어온 Charge가 음의 값인지 양의 값인지 확인해서 승차를 허용 후 Daily data에 저장, Process 3에 Charge를 전송
-	Card	사용자의 Card(프로그래밍상으론 txt로 대체)
-	LED	화면에 정보를 표시한다.

<Calculating>		
함수번호	함수이름	함수설명
-	Card Tagger	Card Tagger시스템을 의미한다.
-	Overall Bus Charge	시간(YYYYMMDD), 교통수단, 결제 금액을 저장한다.
-	Overall Sub Charge	시간(YYYYMMDD), 교통수단, 결제 금액을 저장한다.
2	Send Bus Company	Overall Bus Charge를 받아서 버스 회사에 전달
3	Send Sub Company	Overall Sub Charge를 받아서 지하철 회사에 전달
-	Bus Company	버스회사를 의미한다.
-	Sub Company	지하철회사를 의미한다.
1.3	Delete Data	정산완료 신호를 받아서 Daily Data를 초기화
1.4	Tag Restart	정산완료 신호를 받아 카드 리더기에 태그 승인 신호 전송
1.5	Tag stop	정산시작 신호를 받아 카드 리더기에 태그 받는 것을 중지 신호 전송

5 Approach

Public Transportation System의 Program source code를 C표준 라이브러리 함수를 사용하여 unit test 실행.

6 Item pass/fail criteria

Functional test pass/fail criteria : 각 모듈은 요구사항을 모두 만족하여야 한다.

7 Unit test design specification

7.1 Test design specification identifier

7.2 Features to be tested

7.2.1 Processes in SRA

Table1 테스트할 Process(DFD) 리스트 참조

7.3 Approach refinements

7.4 Test identification

Table 3 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
-------------	------------------	--

7.5 Feature pass/fail criteria

PTS의 각 모듈(프로세스)은 SRA에 정의 되어 있는 요구사항 (입력/출력 및 동작)을 모두 만족해야한다. 각 모듈(프로세스)의 입력/출력 및 동작은 SRA의 process description 항목 및 State Transition Diagram을 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

Table 4 Test Case Identification

<Card Tagger>		
Identifier	Input specification	Output specification
PTS.UTC_000_000	Card1.txt- >TAGGEDINFO{????, ??, ??,??, ??, ???, ???, ?????, ?????, ??};	TAGGEDINFO{2014, 11 , 21,01,01, ???, ???, ?????, ?????, ??};
PTS.UTC_000_001	Card2.txt- >TAGGEDINFO{????, ??, ??, ??, ??, ???, ???, ?????, ?????, ??};	TAGGEDINFO{2014, 11 , 21,01, 01, ???, ???, ?????, ?????, ??};
PTS.UTC_000_002	Card3.txt- >TAGGEDINFO{????, ??, ??, ??, ??, ???, ???, ?????, ?????, ??};	TAGGEDINFO{2014, 11 , 21,01, 01, ???, ???, ?????, ?????, ??};
PTS.UTC_000_003	Card4.txt- >TAGGEDINFO{????, ??, ??, ??, ??, ???, ???, ?????, ?????, ??};	TAGGEDINFO{2014, 11 , 21,01, 01, ???, ???, ?????, ?????, ??};
PTS.UTC_000_004	Card5.txt- >TAGGEDINFO{????, ??, ??, ??, ??, ???, ???, ?????, ?????, ??};	TAGGEDINFO{2014, 11 , 21,01, 01, ???, ???, ?????, ?????, ??};
PTS.UTC_000_005	Card6.txt- >TAGGEDINFO{????, ??, ??, ??, ??, ???, ???, ?????, ?????, ??};	TAGGEDINFO{2014, 11 , 21,01, 01, ???, ???, ?????, ?????, ??};
PTS.UTC_000_006	Card???.txt- >TAGGEDINFO{????, ??, ??, ??, ??, ???, ???, ?????, ?????, ??};	함수 미실행

PTS.UTC_001_000	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?};</p>	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, ? , ?};</p>
PTS.UTC_001_001	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?};</p>	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,OUT, BUS_0, ? , ?};</p>
PTS.UTC_001_002	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,?, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?};</p>	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,?, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,?, BUS_0, ? , ?};</p>
PTS.UTC_002_000	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 22,01,02, BUS,IN, BUS_0, ? , ?};</p>	tginfo->transfer=false
PTS.UTC_002_001	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 20,01,02, BUS,IN, BUS_0, ? , ?};</p>	tginfo->transfer=false
PTS.UTC_002_002	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,03,02, SUB,IN, SUB_0, ? , ?};</p>	tginfo->transfer=false
PTS.UTC_002_003	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,1000, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,01,02, BUS,IN, BUS_0, ? , ?};</p>	tginfo->transfer=false
PTS.UTC_002_004	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,OUT, SUB_0,0, 10};</p> <p>tginfo->tagInfo = {2014, 11 , 21,01,02, BUS,IN, BUS_0, ? , ?};</p>	tginfo->transfer=false
PTS.UTC_002_005	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,0, 10};</p> <p>tginfo->tagInfo = {2014, 11 ,</p>	tginfo->transfer=false

	21,01,02, SUB,IN, SUB_0, ? , ?};	
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->tagInfo = { 2014, 11 , 22,01,01,BUS,IN , BUS_0, ? , ?};	tginfo->yetFee = true
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 ,	tginfo->yetFee = false

	21,01,01, BUS,IN , BUS_0,1000, 10}; tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,OUT , BUS_0, ? , ?};	
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 , ?}; tginfo.transfer = { ? , ? ,false, true, ? , ? , ?} tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_0,5000, 10};	wrinfo.fee = 1650;
PTS.UTC_005_003	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 , ?};	wrinfo.fee = 1750;
PTS.UTC_005_004	tginfo.transfer = { ? , ? ,false, false, ? , ? , ?} tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, true , ?};	wrinfo.fee = 1050;
PTS.UTC_005_005	tginfo.transfer = { ? , ? ,true,	wrinfo.fee = 1150;

	<p>false, ?, ? , ?}</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,OUT, BUS_0, true , ?};</p> <p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,5000, 10};</p>	
PTS.UTC_005_006	<p>tginfo.transfer = { ?, ? ,true, true, ?, ? , ?}</p> <p>tginfo->tagInfo = {2014, 11 , 21,02,02, SUB,IN, SUB_0, true , ?};</p> <p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,OUT, BUS_0,5000, 10};</p>	wrinfo.fee = 0;
PTS.UTC_005_007	<p>tginfo.transfer = { ?, ? ,false, true, ?, ? , ?}</p> <p>tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,IN, SUB_0, true , ?};</p> <p>tginfo->cardInfo = {2014, 11 , 21,01,01, BUS,IN, BUS_0,5000, 10};</p>	wrinfo.fee = 1750;
PTS.UTC_005_008	<p>tginfo.transfer = { ?, ? ,false, false, ?, ? , ?}</p> <p>tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,IN, SUB_0, true , ?};</p> <p>tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_0,5000, 10};</p>	wrinfo.fee = 1650;
PTS.UTC_005_009	<p>tginfo.transfer = { ?, ? ,false, false, ?, ? , ?}</p> <p>tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,IN, SUB_0, true , ?};</p> <p>tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,OUT, SUB_0,5000, 10};</p>	wrinfo.fee = 1050;
PTS.UTC_005_010	<p>tginfo.transfer = { ?, ? ,false, false, ?, ? , ?}</p> <p>tginfo->tagInfo = {2014, 11 , 21,04,02, SUB,OUT, SUB_0, true , ?};</p> <p>tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_2,5000, 10};</p>	wrinfo.fee = 1250;
PTS.UTC_005_011	<p>tginfo->cardInfo = {2014, 11 , 21,01,01, SUB,IN, SUB_2,5000, 10};</p>	tginfo->board_bool = true;
PTS.UTC_005_012	<p>tginfo->cardInfo = {2014, 11 ,</p>	tginfo->board_bool = false;

	21,01,01, SUB,IN, SUB_2,500, 10};	
PTS.UTC_006_000	tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, true , ?};	trigger = true; sendApproval saveData sendWriterInfo 함수 호출
PTS.UTC_006_001	tginfo->tagInfo = {2014, 11 , 21,02,02, BUS,IN, BUS_0, false , ?};	trigger = false; sendApproval saveData sendWriterInfo 함수 호출
PTS.UTC_007_000	cardinfo taginfo writing.info	BUS_0.txt
PTS.UTC_007_001	cardinfo taginfo writing.info	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), 결제요금(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

8.2 Test items

Table 3 Test Design Identification 참조

8.3 Input specifications

Table 4 Test Case Identification 참조

8.4 Output specifications

Table 4 Test Case Identification 참조

9 Testing tasks

Table 5 Testing tasks & Schedule

Task	Predecessor tasks	Special skills	Effort	Finish data
1)Unit Test Plan 작성	PTS 구현		4	
2)Test design specification	Task 1	PTS 이해	4	
3)Test case specification	Task 2	PTS 이해	3	
4)Test Execution	Task 3	PTS 이해	2	
5)Test result report	Task 4		1	

6)개발팀에 Report 전달	Task 5		1	
------------------	--------	--	---	--

10 Environmental needs

PTS의 Unit test를 위한 환경적 요구사항은 다음과 같다.

(1) Hardware & Platform, Cygwin, Vi, GCC

11 Unit Test deliverables

11.1 Unit test plan

11.2 Unit test design specification

11.3 Unit test case specification

11.4 Unit test summary report

12 Schedules

Table 5 Testing tasks & Schedule 참조