

# System Testing Plan for PTS System

- Test Plan
- Test Design Specification
- Test Cases Specification

## Project Team

Team 6

Latest update on:

**2014-12-04**

---

## 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
2.1	Software requirements specification _____	5
2.2	Software requirements analysis _____	5
3	Features to be tested _____	5
4	Features not to be tested _____	6
5	Approach _____	6
6	Item pass/fail criteria _____	6
7	System test design specification _____	6
7.1	Test design specification identifier _____	6
7.2	Features to be tested _____	6
7.3	Approach refinements _____	6
7.4	Test identification _____	7
7.5	Feature pass/fail criteria _____	7
8	System test case specification _____	8
8.1	Test case specification identifier _____	8
8.2	Test items _____	11
8.3	Input specifications _____	11

8.4	Output specifications	11
9	Testing tasks	11
10	Environmental needs	11
11	System test deliverables	12
12	Schedules	12

## 1 Introduction

### 1.1 Objectives

Public Transportation System(PTS)을 System test하기 위해 필요한 활동과 기준에 대한 정의, 환경적인 요구사항, test도구들에 관해 세부적으로 명시한 문서이다.

### 1.2 Background

Public Transportation System(PTS)는 카드 태그 및 운행 요금을 정산하는 시스템이다. 이 시스템은 시스템의 작동에 관련된 다양한 상태 값을 가지고 있다.

System test는 시스템이 동작함에 있어 제시된 SRS및 SRA에 명시한 기능들이 제대로 동작하는지 확인하는 기본적인 테스트 방법이다.

### 1.3 Scope

Public Transportation System(PTS)중 하드웨어와 관련 없이 작동할 수 있는 모듈들을 대상으로 System test를 실시한다. 이와 관련하여 test를 수행하기 위한 자원과 절차, 접근방법 기술적인 부분, 환경 및 도구 등을 해당 문서에서 정의한다. 그리고 이를 바탕으로 기존에 제시된 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](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](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

### 2.1 Software requirements specification

실습 수업 주제(SRS) 참조

### 2.2 Software requirements analysis

2014SE\_B\_TP2\_T6\_SRA\_Rev\_2.3 참조

## 3 Features to be tested

- Process in SRA/SRS : Function이 SRS 및 SRA에 명시되어있는 요구사항을 만족하는지를 test 한다.
- Table 1 테스트 할 function 리스트 참조

**Table 1 테스트할 function 리스트**

Identifier	Function	Function detail
PTS_STC_000	버스 승차	버스 단말기에서 사용자 카드.txt에 승차태그 기록 승차 가능 요금 계산 승차 정보 단말기에 출력
PTS_STC-001	버스 하차	하차 태그 시 환승 여부 확인 환승일 경우 환승 요금 계산 및 부과 하차 정보 사용자 카드.txt에 기록 하차 정보 단말기에 출력
PTS_STC_002	버스 환승	교통 카드 태그시 하차시간확인 하차시간, 승차시간 비교에 따른 환승여부 확인 환승 교통 수단 확인 환승 정보 단말기 출력
PTS_STC_003	지하철 승차	지하철 단말기에서 사용자 카드.txt에 승차태그 기록 승차 가능 요금 계산 승차 정보 단말기에 출력
PTS_STC_004	지하철 하차	하차 태그 시 환승 여부 확인 환승 일 경우 추가 구간 요금 계산 및 부과 하차 정보 사용자 카드.txt에 기록 하차 정보 단말기에 출력
PTS_STC_005	지하철 환승	교통 카드 태그시 하차시간확인

		하차시간, 승차시간 비교에 따른 환승여부 확인 환승 교통 수단 확인 환승 정보 단말기 출력
PTS_STC_006	미환승	버스/지하철 각 요금을 각 회사로 분배
PTS_STC_007	환승	환승에 따른 요금계산여부 확인 계산된 요금 분배여부 확인
PTS_STC_008	미정산	사용자 카드.txt에서 미정산 여부 확인
PTS_STC_009	정산	각 회사에 전달되는 금액 출력 정산 완료 신호 전송 정산 시스템 초기화

4 Features not to be tested

- Process in SRA/SRS : System의 동작을 확인하기 위한 Function은 Test에서 제외한다.
- Table 2 테스트 하지 않을 function 리스트 참조

**Table 2 테스트 하지 않을 function 리스트**

Identifier	Function	Function detail
PTS_STC_000	단말기	각 단말기 별로 고유한 식별자를 가지고 있어서 구분할수 있다.

5 Approach

Public Transportation System을 Cygwin환경에서 구동

6 Item pass/fail criteria

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

7 System test design specification

7.1 Test design specification identifier

7.2 Features to be tested

Table 1 테스트할 Function리스트 참조

7.3 Approach refinements

## 7.4 Test identification

**Table 3 Test Design Identification**

Identifier	Function	Function detail
PTS_STC_000	버스 승차	버스 단말기에서 사용자 카드.txt에 승차태그 기록 승차 가능 요금 계산 승차 정보 단말기에 출력
PTS_STC_001	버스 하차	하차 태그 시 환승 여부 확인 환승일 경우 환승 요금 계산 및 부과 하차 정보 사용자 카드.txt에 기록 하차 정보 단말기에 출력
PTS_STC_002	버스 환승	교통 카드 태그시 하차시간확인 하차시간, 승차시간 비교에 따른 환승여부 확인 환승 교통 수단 확인 환승 정보 단말기 출력
PTS_STC_003	지하철 승차	지하철 단말기에서 사용자 카드.txt에 승차태그 기록 승차 가능 요금 계산 승차 정보 단말기에 출력
PTS_STC_004	지하철 하차	하차 태그 시 환승 여부 확인 환승 일 경우 추가 구간 요금 계산 및 부과 하차 정보 사용자 카드.txt에 기록 하차 정보 단말기에 출력
PTS_STC_005	지하철 환승	교통 카드 태그시 하차시간확인 하차시간, 승차시간 비교에 따른 환승여부 확인 환승 교통 수단 확인 환승 정보 단말기 출력
PTS_STC_006	미환승	버스/지하철 각 요금을 각 회사로 분배
PTS_STC_007	환승	환승에 따른 요금계산여부 확인 계산된 요금 분배여부 확인
PTS_STC_008	미정산	사용자 카드.txt에서 미정산 여부 확인
PTS_STC_009	정산	각 회사에 전달되는 금액 출력 정산 완료 신호 전송 정산 시스템 초기화

## 7.5 Feature pass/fail criteria

PTS는 SRS 및 SRA에 정의 되어 있는 요구사항(입력/출력 및 동작)을 모두 만족해야 한

다. PTS의 입력/출력 및 동작은 SRA의 process description 항목 및 State Transition Diagram을 참조한다.

## 8 System test case specification

### 8.1 Test case specification identifier

**Table 4 Test Case Identification**

Identifier	Function	Valid / Invalid value
PTS_STC_000_000	Tag == IN / Bus tagger -> UserCard.txt = {YYYY,MM,DD,HH,MM,???,???,Balance,???,count}	Bus tagger -> UserCard.txt = {YYYY,MM,DD,HH,MM, <b>BUS,IN</b> ,Balance, <b>BUS_0</b> +count}
PTS_STC_000_001	UserCard.txt -> Balance >= 1050	True
PTS_STC_000_002	UserCard.txt -> Balance <= 1050	False
PTS_STC_000_003	Usercard.txt -> Tagger	Tagger = {2014,12,04,08,50,BUS,IN,2100,BUS_00}
PTS_STC_001_000	Tag == OUT	Check Transfer == True
PTS_STC_001_001	Tag == OUT	Check Transfer == false
PTS_STC_001_002	Check Transfer == True	Tagger = {2014,12,04,08,50,BUS, <b>OUT,0</b> ,BUS_00}
PTS_STC_001_003	Check Transfer == false	Tagger = {2014,12,04,08,50,BUS, <b>OUT,1050</b> ,BUS_00}
PTS_STC_001_004	Tagger -> UserCard.txt	UserCard.txt = {2014,12,04,08,50,BUS, <b>OUT,1050</b> ,BUS_00}
PTS_STC_001_005	Tagger -> LED Interface	LED Interface = {2014,12,04,08,50,BUS, <b>OUT,2100</b> ,BUS_00}
PTS_STC_002_000	Tag == IN UserCard.txt -> Tagger = {2014,12,04, <b>08,50</b> ,BUS,IN,2100,BUS_00}	Tagger = {2014,12,04,08,50,BUS,IN,2100,BUS_00}



PTS_STC_002_001	(Tagger.prevTime + time) >= Tagger.nextTime	Transfer == True
PTS_STC_002_002	(Tagger.prevTime + time) <= Tagger.nextTime	Transfer == False
PTS_STC_002_003	Check Bus/Sub	Tagger = {2014,12,04,08,50, <b>BUS</b> ,IN,2100,BUS_00}
PTS_STC_002_004	Tagger -> LED Interface	LED Interface = {2014,12,04,08,50, <b>BUS</b> ,OUT,2100,BUS_00}
PTS_STC_003_000	Tag == IN / Sub tagger -> UserCard.txt = {YYYY,MM,DD,HH,MM,???,???,Balance,???,count}	Sub tagger -> UserCard.txt = {YYYY,MM,DD,HH,MM, <b>SUN</b> ,IN,Balance, <b>SUB</b> _00}
PTS_STC_003_001	UserCard.txt -> Balance >= 1050	True
PTS_STC_003_002	UserCard.txt -> Balance <= 1050	False
PTS_STC_003_003	Usercard.txt -> Tagger 출력	Tagger = {2014,12,04,08,50,SUB,IN,2100,SUB_00}
PTS_STC_004_000	Tag == OUT	Check Transfer == True
PTS_STC_004_001	Tag == OUT	Check Transfer == false
PTS_STC_004_002	Check Transfer == True	Tagger = {2014,12,04,08,50,SUB, <b>OUT</b> ,1100,BUS_00}
PTS_STC_004_003	Check Transfer == false	Tagger = {2014,12,04,08,50,SUB, <b>OUT</b> ,2100,BUS_00}
PTS_STC_004_004	Tagger -> UserCard.txt	UserCard.txt = {2014,12,04,08,50,SUB, <b>OUT</b> ,2100,BUS_00}
PTS_STC_004_005	Tagger -> LED Interface	LED Interface = {2014,12,04,08,50,SUB, <b>OUT</b> ,2100,BUS_0,0}
PTS_STC_005_000	Tag == IN UserCard.txt -> Tagger = {2014,12,04, <b>08</b> ,50,SUB,IN,2100,BUS_00}	Tagger = {2014,12,04,08,50,SUB,IN,2100,BUS_00}
PTS_STC_005_001	(Tagger.prevTime + time) >=	Transfer == True

	Tagger.nextTime	
PTS_STC_005_002	(Tagger.prevTime + time) <= Tagger.nextTime	Transfer == False
PTS_STC_005_003	Check Bus/Sub	Tagger = {2014,12,04,08,50, <b>BUS</b> ,IN,2100,BUS_00}
PTS_STC_005_004	Tagger -> LED Interface	LED Interface = {2014,12,04,08,50, <b>SUB</b> ,OUT,2100,BUS_00}
PTS_STC_006_000	Bus Charge -> Bus Company = {YYYY,MM,DD,TRANSFER,Charge}	Bus Charge -> Bus Company = {2014,12,04,BUS,125400}
PTS_STC_006_001	Sub Charge -> Sub Company = {YYYY,MM,DD,TRANSFER,Charge}	Sub Charge -> Sub Company = {2014,12,04,Sub,214500}
PTS_STC_007_000	Bus -> Sub All charge=(Bus+Sub)+Sub Charge=Bus+sub	Bus = (Bus+Sub)/(All Charge*Charge) Sub = Sub/(Allcharge*charge)
PTS_STC_007_001	Sub -> Bus All charge=(Sub+Bus)+Bus Charge=Sub+Bus	Sub = (Sub+Bus)/(All Charge*Charge) Bus = Bus/(Allcharge*charge)
PTS_STC_007_002	Bus -> Bus Company = {YYYY,MM,DD,TRANSFER,Charge}	Bus -> Bus Company = {2014,12,04,BUS,125400}
PTS_STC_007_003	Sub -> Sub Company = {YYYY,MM,DD,TRANSFER,Charge}	Sub -> Sub Company = {2014,12,04,Sub,214500}
PTS_STC_008_000	Tagger = {2014, <b>12,04</b> ,08,50, <b>BUS</b> ,IN,2100,BUS_00} Tagger = {2014, <b>12,05</b> ,08,50, <b>BUS</b> ,IN,2100,BUS_00}	Add charge
PTS_STC_008_001	Tagger = {2014, <b>12,04</b> ,08,50, <b>SUB</b> ,IN,2100,BUS_00} Tagger = {2014, <b>12,05</b> ,08,50, <b>BUS</b> ,IN,2100,BUS_00}	Add Charge
PTS_STC_008_002	Tagger = {2014,12,04, <b>08,50</b> , <b>SUB</b> ,IN,2100,SUB_00}	Add Charge

	Tagger = {2014,12,04,14,50,SUB,IN,2100,SUB_00}	
PTS_STC_009_000	Bus Company -> BusCharge.txt = {2014,12,04,BUS,125400}	BusCharge.txt = {2014,12,04,BUS,125400}
PTS_STC_009_001	Sub Company -> SubCharge.txt = {2014,12,04,SUB,214500}	Sub Company ->SubCharge.txt = {2014,12,04, SUB,214500}
PTS_STC_009_002	Trigger.txt -> Tagger = {0}	Tagger Restart
PTS_STC_009_003	Trigger.txt -> Tagger = {1}	Tagger Stop
PTS_STC_009_004	Trigger.txt -> Tagger = {2}	Tagger data Delete

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)System Test Plan 작성	PTS 동작확인		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 System test을 위한 환경적 요구사항은 다음과 같다.

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

11 System test deliverables

12 Schedules