

System Testing Plan for Public Transportation System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

Team 4

Latest update on:

2014-12-05

노은방 200811428

김상민 200910044

박수민 201111353

한별 201214217

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	4
2.1	Software requirements specification	4
2.2	Software requirements analysis	6
3	Features to be tested	8
4	Features not to be tested	8
5	Approach	8
6	Item pass/fail criteria	8
7	System test design specification	8
7.1	Test design specification identifier	8
7.2	Features to be tested	8
7.3	Approach refinements	10
7.4	Test identification	10
7.5	Feature pass/fail criteria	10
8	System test case specification	10
8.1	Test case specification identifier	10
8.2	Test items	10
8.3	Input specifications	11

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

1 Introduction

1.1 Objectives

본 문서는 2014년 2학기 소프트웨어 공학 개론 수업에서 T4이 개발한 Public Transportation System (PTS) 의 System Testing Plan (STP)를 위한 문서이다.

1.2 Background

PTS의 System Testing (ST)를 위하여 PTS의 Software Requirement Specification (SRS)에 대한 이해가 필요하다. SRS는 2014 소프트웨어 공학 개론의 수업 내용인 SA 기법으로 작성되었다.

1.3 Scope

본 STP는 PTS가 가진 버스와 지하철 그리고 정산 시스템을 이용하여 수행할 수 있는 행위에 대한 ST를 명세하고 있다.

1.4 Project plan

완성된 PTS의 Software를 ST하기 위한 본 계획서를 작성한다.

본 계획서에 작성된 내용에 기반하여 ST를 수행한다.

St에 대한 결과를 개발팀에 통보한다.

1.5 Configuration management plan

SASD를 이용하여 SW를 구현한 후, System Testing (ST)를 수행한다.

ST의 결과를 대상 SW에 반영한다.

1.6 References

SRA3

2 Test items

2.1 Software requirements specification

2.1.1 Bus Function

(1) 승차

Tag해서 User정보를 받아온다

요금 계산한다.

요금이 부족할 시 요금 부족을 출력한다.

요금이 충분할 시 승차하고 시간과 요금 그리고 잔액을 출력한다.

(2) 하차

Tag 해서 User정보에 단말기 정보를 입력한다.

추가요금과 시간 그리고 잔액 출력한다.

(3) 저장

단말기 정보를 저장한다.

(4) 초기화

3분이 되었을 시 초기화한다.

2.1.2 Subway Function

(1) 승차

Tag해서 User정보를 받아온다

요금 계산한다.

요금이 부족할 시 요금 부족을 출력한다.

요금이 충분할 시 승차하고 시간과 요금 그리고 잔액을 출력한다.

(2) 하차

Tag 해서 User정보에 단말기 정보를 입력한다.

추가요금과 시간 그리고 잔액 출력한다.

(3) 저장

단말기 정보를 저장한다.

(4) 초기화

3분이 되었을 시 초기화한다.

2.1.3 Adjustment Function

(1) 정산

버스와 지하철에게 정보를 받아서 요금을 총 정산한다.

(2) 전송

버스 회사와 지하철 회사에 각 정산된 요금을 전송한다.

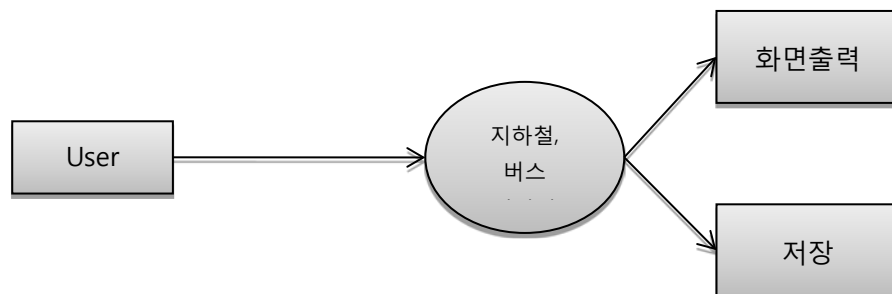
(3) 초기화

초기화

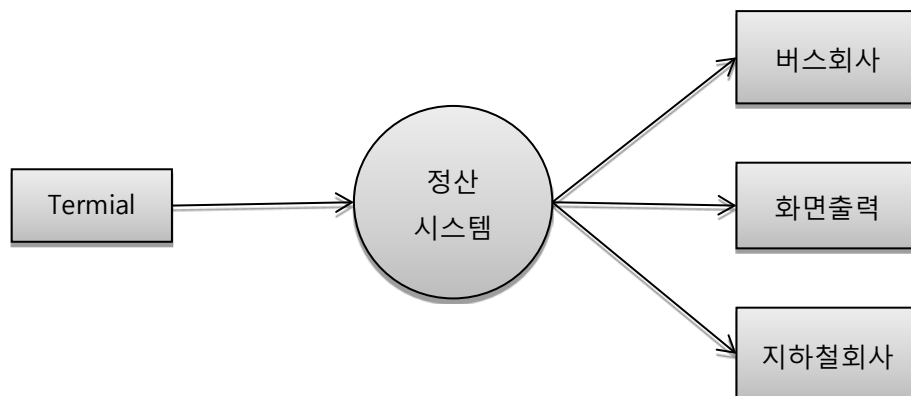
2.2 Software requirements analysis

2.2.1 Basic System Context Diagram

-지하철, 버스



-정산 시스템



2.2.2 Event List

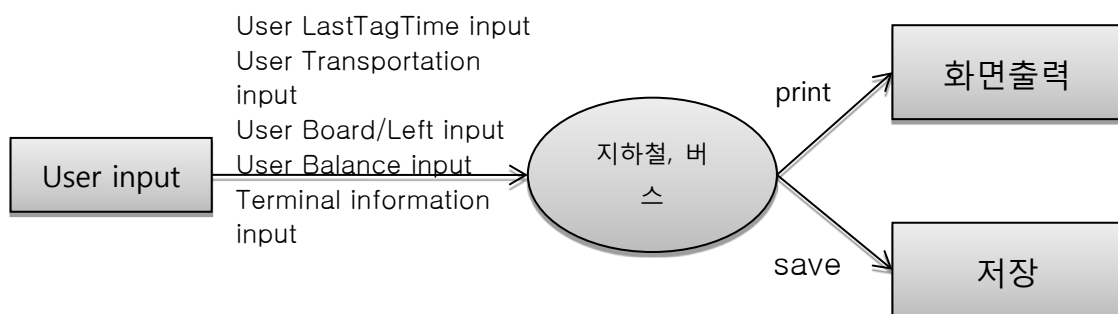
-버스, 지하철

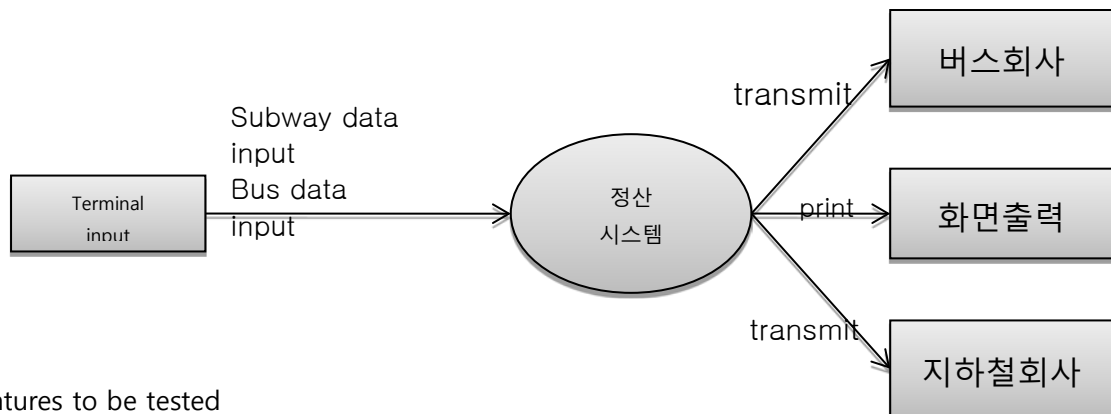
Input/ Output Event	Description
User LastTagTime input	태그가 되었을 때 시간 정보를 단말기로 보내준다
User Transportation input	태그가 되었을 때 교통수단 정보를 단말기로 보내준다
User Board/Left input	태그가 되었을 때 승차/하차 정보를 단말기로 보내준다
User Balance input	태그가 되었을 때 잔액 정보를 단말기로 보내준다
Terminal information input	태그가 되었을 때 이전 단말기의 정보를 단말기로 보내준다
Print	화면에 금액을 출력하거나 금액 부족을 출력한다
Save	각 단말기의 저장소와 유저 카드에 저장을 시킨다.

-정산

Input/ Output Event	Description
Subway data input	지하철에서의 시간 정보, 교통수단 정보, 승차/하차 정보, 잔액 정보 그리고 이전 단말기의 정보를 정산시스템으로 보내준다
Bus data input	버스에서의 시간 정보, 교통수단 정보, 승차/하차 정보, 잔액 정보 그리고 이전 단말기의 정보를 정산시스템으로 보내준다
Transmit	정산금액을 각 버스/지하철 회사에 전송(저장)시킨다.
Print	화면에 정산금액을 출력해준다.

2.2.3 System Context Diagram





3 Features to be tested

4 Features not to be tested

5 Approach

요구사항에 정의되어 있는 기능들을 확인하기 위한 시나리오를 작성한다. 작성한 시나리오에 기반하여 Testing design 및 Test cases를 작성하여 Testing을 수행한다.

6 **Item** pass/fail criteria

7 System test design specification

7.1 Test design specification identifier

PTS.STC.Number

7.2 Features to be tested

7.2.1 버스를 타면 버스 단말기가 실행되고 지하철을 타면 지하철 단말기가 실행된다.
또한 3분이 되었을 때 정산이 작동되고 후에 초기화된다.(PTS.STC.000)

7.2.2 버스 단말기

(1) 사용자 태그(입력) (PTS.STC.100)

A. 사용자 정보를 입력 받는다.

(2) 승 하차 분류(PTS.STC.101)

A. User 정보를 Input 받는다.

B. 승차인지 하차인지 입력한다.

(3) 승차(PTS.STC.102)

- A. 정보에 따라 요금 계산을 한다.
- B. 요금 충분, 부족을 판단한다.
- C. 충분할 경우 요금과 시간 잔액을 출력하고 저장한다.
- D. 부족할 경우 부족 메시지를 출력한다.

(2) 하차(PTS.STC.103)

- A. User 정보를 Input 받는다.
- B. 정보에 따라 요금 계산을 한다.
- C. 요금과 잔액 시간을 저장 및 출력한다.

(3) 초기화(PTS.STC.104)

- A. 3분이 경과 했을 때 초기화한다.

7.2.3 지하철 단말기

(1) 사용자 태그(입력)(PTS.STC.200)

- A. 사용자 정보를 입력 받는다.

(2) 승 하차 분류(PTS.STC.201)

- B. User 정보를 Input 받는다.
- C. 승차인지 하차인지 입력 받는다

(2) 승차(PTS.STC.202)

- A. 승차할 역을 고른다.
- B. 정보에 따라 요금 계산을 한다.
- C. 요금 충분, 부족을 판단한다.
- D. 충분할 경우 요금과 시간 잔액을 출력하고 저장한다.

E. 부족할 경우 부족 메시지를 출력한다.

(4) 하차(PTS.STC.203)

- A. 하차할 역을 고른다.
- B. User 정보를 Input 받는다.
- C. 정보에 따라 요금 계산을 한다.
- D. 요금과 잔액 시간을 저장 및 출력한다.

(5) 초기화(PTS.STC.204)

- A. 3분이 경과 했을 때 초기화한다.

7.2.4 정산

(1) 정산(PTS.STC.300)

- A. 버스와 지하철에게 정보를 받아서 요금을 총 정산한다.

(2) 전송(PTS.STC.301)

- A. 버스 회사와 지하철 회사에 각 정산된 요금을 전송한다.

7.3 Approach refinements

PTS는 SRS의 Specific Requirements에 나온 모든 동작을 명세 된 내용과 동일하게 수행해야 한다. 각 시나리오들이 정해진 순서의 입력을 받았을 때, SRS에 명세 된 동작을 수행해야 한다.

7.4 Test identification

PTS,STC,Number,Number

7.5 Feature pass/fail criteria

8 System test case specification

8.1 Test case specification identifier

8.2 Test items

8.3 Input specifications

8.4 Output specifications

9 Testing tasks

Table 1 Testing tasks & Schedule

Task	Predecessor tasks	Special skills	Effort	Finish date
(1) STP작성	SRA 작성 SDS 작성 PTS 구현		3	
(2) Test design specification	Task 1	PTS에 대한 이해		
(3) Test case specification	Task 2	PTS에 대한 이해		
(4) Test execution	Task 3	Test code 작성 Test tools에 대한 이해		
(5) Test result report	Task 4		1	
(6) 개발팀에 report 전달	Task 5		1	

10 Environmental needs

PTS를 실행하기 위한 PC-Cygwin 필요

11 System test deliverables

11.1 System test plan

11.2 System test design specification

11.3 System test case specification

11.4 System test summary report

12 Schedules

< Table 1 Testing tasks & Schedule > 참조

Table 2 System Test Design Identification

Identifier	Feature	Valid/ Invalid value
PTS.STC.000	지하철, 버스 선택. 3분 초기화	지하철을 탑승하면 지하철이 시작되고 버스를 탑승하면 버스가 시작된다. 3분이 경과되면 정산과 초기화가 진행된다.
PTS.STC.100	(버스)사용자 태그(입력)	사용자 정보가 입력된다.
PTS.STC.101	(버스)승하차 분류	승차는 승차로 하차는 하차로 처리된다.
PTS.STC.102	(버스)승차	충분할 경우 요금과 시간과 잔액을 출력 및 저장하고 부족할 경우 부족 메시지만 출력
PTS.STC.103	(버스)하차	요금과 잔액 시간을 저장 및 출력한다.
PTS.STC.104	(버스)초기화	3분 경과 했을 때 초기화 된다.
PTS.STC.200	(지하철)사용자 태그(입력)	사용자 정보가 입력된다.
PTS.STC.201	(지하철)승하차 분류	승차는 승차로 하차는 하차로 처리된다.
PTS.STC.202	(지하철)승차	충분할 경우 요금과 시간과 잔액을 출력 및 저장하고 부족할 경우 부족 메시지만 출력
PTS.STC.203	(지하철)하차	요금과 잔액 시간을 저장 및 출력한다.
PTS.STC.204	(지하철)초기화	3분 경과 했을 때 초기화 된다.
PTS.STC.300	(정산)정산	버스와 지하철에게 정보를 받아서 요금을 총 정산한다.
PTS.STC.301	(정산)전송	버스 회사와 지하철 회사에 각 정산된 요금을 전송한다.
PTS.STC.302	(정산)초기화	초기화 한다,

Table 3 Test Cases Identification

Test case Identifier	Input specification	Output specification
PTS.STC.000.000	(버스)프로그램 실행	(버스)이름입력, 탑승(0) 하차(1)
PTS.STC.000.001	(지하철)프로그램 실행	(지하철)이름입력
PTS.STC.000.002	(정산)프로그램 실행,3분	(정산)
PTS.STC.000.003	(초기화)프로그램 실행	(초기화)초기화!
PTS.STC.100.000	(사용자 태그) 이름(최초), 최초충전금액	(사용자태그)Inout::n, Transportaion :n, balance : 999999, 탑승 (0) 하차 (1) : 0
PTS.STC..100.001	(사용자태그) 이름	(사용자태그)Inout::user info.In_out, Transportaion : user Info.transportaition balance : 999999, 탑승 (0) 하차(1) : 0
PTS.STC.101.000	(승하차분류)0,	(승하차분류) 승차 실행
PTS.STC.101.001	(승하차분류)1	(승하차분류) 하차 실행
PTS.STC.102.000	(승차)0,	(승차) IN, 시간 : 00:00, 금액 : 999999 잔액 : 999999
PTS.STC.102.001	(승차)0, user info.balance<500	(승차) 요금이 부족하여 승차할 수 없습니다!!!
PTS.STC.102.002	(승차)0, user info.balance<1050	(승차) 요금이 부족하여 승차할 수 없습니다!!!
PTS.STC.102.003	(승차)0, user info.balance<1650	(승차) 요금이 부족하여 승차할 수 없습니다!!!
PTS.STC.102.004	(승차)0, user info.balance<1750	(승차) 요금이 부족하여 승차할 수 없습니다!!!
PTS.STC.102.005	(승차)0, user info.balance<1250	(승차) 요금이 부족하여 승차할 수 없습니다!!!
PTS.STC.102.006	(승차)0, user info.balance>=500	(승차) IN, 시간 : 00:00, 금액 : 0 잔액 : 999999
PTS.STC.102.007	(승차)0, user info.balance>=1050	(승차) IN, 시간 : 00:00, 금액 : 1050 잔액 : 999999
PTS.STC.102.008	(승차)0, user info.balance>=1650	(승차) IN, 시간 : 00:00, 금액 : 1650 잔액 : 999999

PTS.STC.102.009	(승차)0, user info.balance>=1750	(승차) IN, 시간 : 00:00, 금액 : 1750 잔액 : 999999
PTS.STC.102.010	(승차)0, user info.balance>=1250	(승차) IN, 시간 : 00:00, 금액 : 1250 잔액 : 999999
PTS.STC.103.000	(하차)1, user info, current info	(하차) OUT, 시간 : 00:00, 금액 : 999999 잔액 : 999999
PTS.STC.104.000	(초기화), secTime = 3:00	(초기화) 초기화!
PTS.STC.200.000	(사용자 태그) 이름(최초), 최초충전금액	(사용자 태그)Transport :n inout :n balance : 999999, 탑승 (0) 하차(1)
PTS.STC.200.001	(사용자태그) 이름	(사용자 태그)TAG TIME 00 : 00, ====user information====, lasttagtime : 00 : 00 transport : METRO input : BOARD, balance: 999999, terminal : X_9, 탑승(0) 하차(1)
PTS.STC.201.000	(승하차분류)0	(승하차분류) 승차 실행
PTS.STC.201.001	(승하차분류)1	(승하차분류) 하차 실행
PTS.STC.202.000	(승차)0,	(승차)건대(b), 강남(c) 신림(d) 합정(e), 동역사(f)
PTS.STC.202.001	(승차)b,	(승차)건대, 9999원, 승차 고고~~
PTS.STC.202.002	(승차)c,	(승차)강남 9999원, 승차 고고~~
PTS.STC.202.003	(승차)d,	(승차)신림 9999원, 승차 고고~~
PTS.STC.202.004	(승차)e,	(승차)합정, 9999원, 승차 고고~~
PTS.STC.202.005	(승차)f,	(승차)동역사, 9999원, 승차 고고~~
PTS.STC.202.000	(승차) b c d e f, user info.balance<Currentino..balance + 600	(승차)요금미 부족해서 승차하실 수 없습니다!!!
PTS.STC.202.001	(승차) b c d e f, user info.balance<Currentino..balance + 200	(승차)요금미 부족해서 승차하실 수 없습니다!!!
PTS.STC.202.002	(승차) b c d e f, user info.balance<Currentino..balance + 700	(승차)요금미 부족해서 승차하실 수 없습니다!!!
PTS.STC.202.003	(승차) b c d e f, user info.balance>= Currentino..balance + 600	(승차)역이름 9999원, 승차 고고~~
PTS.STC.202.004	(승차) b c d e f, user info.balance>=Currentino..balance +200	(승차)역이름, 9999원, 승차 고고~~
PTS.STC.202.005	(승차) b c d e f, user info.balance>= Currentino..balance + 700	(승차)역이름, 9999원, 승차 고고~~

PTS.STC.203.000	(하차)1	(하차)건대(b), 강남(c) 신림(d) 합정(e), 동역사(f)
PTS.STC.203.001	(하차)b,	(하차)건대, 하차시 추가요금 : 999999원
PTS.STC.203.002	(하차)c,	(하차)강남 하차시 추가요금 : 999999원
PTS.STC.203.003	(하차)d,	(하차)신림 하차시 추가요금 : 999999원
PTS.STC.203.004	(하차)e,	(하차)합정, 하차시 추가요금 : 999999원
PTS.STC.203.005	(하차)f,	(하차)동역사, 하차시 추가요금 : 999999원
PTS,STC,204.000	(초기화), secTime = 3:00	(초기화) 초기화!
PTS,STC,300.000	(정산) secTime = 3:00	(정산) n번째 : Lt = 00 : 00 n번째 : tr = 0 1 n번째 : fee = 99999 n번째 : terminal = a b c d e f_n
PTS,STC,300.001	(전송)n번째 : Lt = 00 : 00 n번째 : tr = 0 1 n번째 : fee = 99999 n번째 : terminal = a b c d e f_n	(전송) busCompany.txt, subwayCompany