

Rev. for Test results

Public Transportation System

- Test Cases Specification
- Test Summary Report

Project Team

Team 4

Latest update on:

2014-12-011

노은방 200811428

김상민 200910044

박수민 201111353

한별 201214217

1 Introduction

1.1 Objectives

본 문서는 2014년 2학기 소프트웨어 공학 개론 수업에서 T4가 개발한 Public Transportation System (PTS)의 테스트 결과 대응서이다.

1.2 References

T4_STP_ver.1

2 System test case Rev. – watch table 1 test result Rev.

3 '강남역' 문제 대응

4 정산 대응

Table 1 Test Result Rev.

Test case Identifier	Input specification	Output specification	Pass/ Fail
PTS.STC.000.000	(버스)프로그램 실행	(버스)이름입력, 탑승(0) 하차(1)	P
PTS.STC.000.001	(지하철)프로그램 실행	(지하철)이름입력	P
PTS.STC.000.002	(정산)프로그램 실행,3분	(정산)	P
PTS.STC.000.003	(초기화)프로그램 실행	(초기화)초기화!	P
PTS.STC.100.000	(사용자 태그) 이름(최초), 최초충전금액	(사용자태그)Inout::n, Transportaion :n, balance : 999999, 탑승 (0) 하차(1) : 0	P
PTS.STC..100.001	(사용자태그) 이름	(사용자태그)Inout::user info.In_out, Transportaion : user Info.transportaition balance : 999999, 탑승 (0) 하차(1) : 0	P
PTS.STC.101.000	(승하차분류)0, 승차일때이다.	(승하차분류) 승차 실행	P
PTS.STC.101.001	(승하차분류)1 하차일때이다.	(승하차분류) 하차 실행	P
PTS.STC.102.000	(승차)0,	(승차) IN, 시간 : 00:00, 금액 : 999999 잔액 : 999999	P
PTS.STC.102.001	(승차)0, user info.balance<500 요금부족의 테스트경우로, 지하철에서 버스로 환승한 경우 잔액 테스트 해야 하는 금액을 500원으로 설정한 상태이다. (하루 3분이고, 3분을 모두 타는 건 정산시간때문에 불가능, 500원이	(승차) 요금이 부족하여 승차할 수 없습니다!!!	P

	환승 최대요금이라는건 아님),		
PTS.STC.102.002	(승차)0, user info.balance<1050 최초 승차 및 초기화 후 버스 탑승시 기본금액인 1050원이 부족한 경우이다.	(승차) 요금이 부족하여 승차할 수 없습니다!!!	P
PTS.STC.102.003	(승차)0, user info.balance<1650 버스에서 지하철 환승 했을 때 미정산 으로 내린경우 다음 버스 승차때 600 원이 추가되어 부가됐는데 그 금액이 모자란 경우이다.	(승차) 요금이 부족하여 승차할 수 없습니다!!!	P
PTS.STC.102.004	(승차)0, user info.balance<1750 지하철에서 버스로 환승하고 미정산으 로 내렸을 때 다음 버스 승차때 환승 최대금액 700원이 추가되어 부가됐는 데 금액이 모자란 경우이다.	(승차) 요금이 부족하여 승차할 수 없습니다!!!	P
PTS.STC.102.005	(승차)0, user info.balance<1250 지하철에서 하차시 미정산으로 내렸을 때 다음 버스 승차시 200원이 추가되 어 부가되었고, 그 금액이 모자란경우 혹은 지하철 최초탑승	(승차) 요금이 부족하여 승차할 수 없습니다!!!	P
PTS.STC.102.010	(승차)0, user info.balance>=500 위와 동일하지만 금액이 충분한 경우	(승차) IN, 시간 : 00:00, 금액 : 0 잔액 : 999999	P
PTS.STC.102.011	(승차)0, user info.balance>=1050	(승차) IN, 시간 : 00:00, 금액 : 1050 잔액 : 999999	P
PTS.STC.102.012	(승차)0, user info.balance>=1650	(승차) IN, 시간 : 00:00, 금액 : 1650 잔액 :	P

		999999	
PTS.STC.102.013	(승차)0, user info.balance>=1750	(승차) IN, 시간 : 00:00, 금액 : 1750 잔액 : 999999	P
PTS.STC.102.014	(승차)0, user info.balance>=1250	(승차) IN, 시간 : 00:00, 금액 : 1250 잔액 : 999999	P
PTS.STC.103.000	(하차)1, user info, current info 하차시 환승일 경우 시간당 금액 확인 환승 아닐경우 0원	(하차) OUT, 시간 : 00:00, 금액 : 999999 잔액 : 999999	P
PTS.STC.104.000	(초기화), secTime = 3:00 3분이 되서 초기화 되는 금액 확인	(초기화) 초기화!	P
PTS.STC.200.000	(사용자 태그) 이름(최초), 최초충전금액 사용자 이름이 입력 되었을 때 이름이 없는 경우(신규) 최초 금액 입력	(사용자 태그)Transport :n inout :n balance : 999999, 탑승 (0) 하차(1)	P
PTS.STC.200.001	(사용자태그) 이름 사용자 이름이 입력 되었을 때 이름이 존재해서 해당 유저정보를 불러온경우	(사용자 태그)TAG TIME 00 : 00, ====user information====, lasttagtime : 00 : 00 transport : METRO input : BOARD, balance: 999999, terminal : X_9, 탑승(0) 하차(1)	P
PTS.STC.201.000	(승하차분류)0 0은 승차	(승하차분류) 승차 실행	P
PTS.STC.201.001	(승하차분류)1 1은 하차	(승하차분류) 하차 실행	P
PTS.STC.202.000	(승차)0, 지하철에서 승차시 각 역을 선택한다	(승차)건대(b), 강남(c) 신림(d) 합정(e), 동역사(f)	P
PTS.STC.202.001	(승차)b,	(승차)건대, 9999원, 승차 고고~~	P

PTS.STC.202.002	(승차)c,	(승차)강남 9999원, 승차 고고~~	P
PTS.STC.202.003	(승차)d,	(승차)신림 9999원, 승차 고고~~	P
PTS.STC.202.004	(승차)e,	(승차)합정, 9999원, 승차 고고~~	P
PTS.STC.202.005	(승차)f,	(승차)동역사, 9999원, 승차 고고~~	P
PTS.STC.202.000	(승차) b c d e f, user info.balance<Currentino..balance + p600 버스에서 지하철로 환승하고 미정산으로 내렸을 때 600원 추가 된 금액이 모자란 경우 (지하철 탑승금액 1250원 까지 검색에 포함한다)	(승차)요금이 부족해서 승차하실 수 없습니다!!!	p
PTS.STC.202.001	(p승차) b c d e f, user info.balance<Currentino..balance + 200 지하철에서 하차시 미정산으로 내렸을 때 다음 지하철 승차시 200원이 추가 된 금액이 모자란경우	(승차)요금이 부족해서 승차하실 수 없습니다!!!	P
PTS.STC.202.002	(승차) b c d e f, user info.balance<Currentino..balance + 700 지하철에서 버스로 환승하고 미정산으로 내렸을 때 다음 지하철 승차 때 환승 최대금액 700원이 추가된 경우	(승차)요금이 부족해서 승차하실 수 없습니다!!!	P
PTS.STC.202.003	(승차) b c d e f, user info.balance>= Currentino..balance + 600	(승차)역이름 9999원, 승차 고고~~	P

	위와 동일하지만, 금액이 충분한경우		
PTS.STC.202.004	(승차) b c d e f, user info.balance >= Currentino..balance + 200	(승차)역이름, 9999원, 승차 고고~~	P
PTS.STC.202.005	(승차) b c d e f, user info.balance >= Currentino..balance + 700	(승차)역이름, 9999원, 승차 고고~~	P
PTS.STC.203.000	(하차)1 하차시 어느역에서 하차할 것인지 확인한다. 하차하차 등의 예외도 여기서 처리된다.	(하차)건대(b), 강남(c) 신림(d) 합정(e), 동역사(f)	P
PTS.STC.203.001	(하차)b,	(하차)건대, 하차시 추가요금 : 999999원	P
PTS.STC.203.002	(하차)c,	(하차)강남 하차시 추가요금 : 999999원	P
PTS.STC.203.003	(하차)d,	(하차)신림 하차시 추가요금 : 999999원	P
PTS.STC.203.004	(하차)e,	(하차)합정, 하차시 추가요금 : 999999원	P
PTS.STC.203.005	(하차)f,	(하차)동역사, 하차시 추가요금 : 99999원	P
PTS,STC,204.000	(초기화), secTime = 3:00 3분이 되고 지하철의 정보(파일)이 초기화 되는지 확인	(초기화) 초기화!	P
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	p
PTS,STC,300.001	(전송)n번째 : Lt = 00 : 00 n번째 : tr = 0 1 n번째 : fee = 99999	(전송) busCompany.txt, subwayCompany.txt	P

	<p>n번째 : terminal = a b c d e f_n 정산시 각 터미널정보별로 정산을 시행한다. 승차하차의 경우 합쳐서 계산하고 승차로 종료된 경우 승차 하나로만 계산한다.</p>		
--	--	--	--

Test Case '강남역'

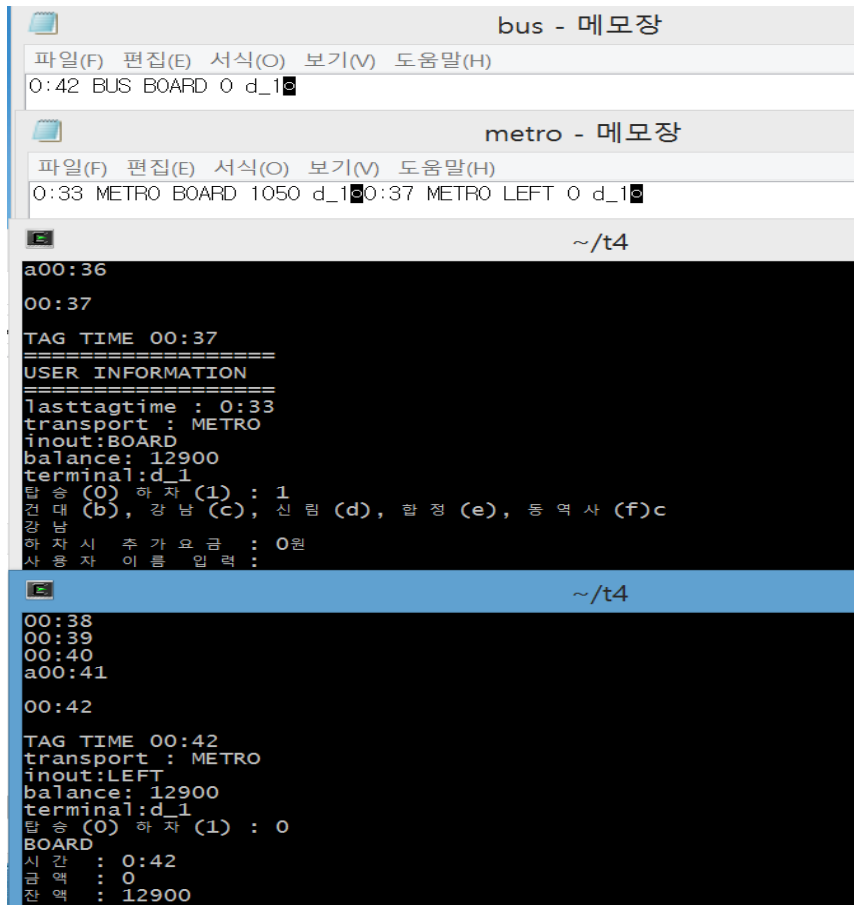
버스에서 내려서 강남역으로 환승하는 경우

The screenshot displays a test environment with three windows: 'bus - 메모장', 'metro - 메모장', and a terminal window. The bus log shows boarding at 0:16 and leaving at 0:20. The metro log shows boarding at 0:7 and another boarding at 0:25. The terminal output shows the current time as 00:23, followed by 'a00:24' and '00:25'. It then displays 'TAG TIME 00:25' and 'USER INFORMATION' with details: lasttagtime: 0:20, transport: BUS, inout: LEFT, balance: 17700, terminal: a_1. A menu is shown with options for boarding (0), transfer (1), station (b), Gangnam (c), Shinlim (d), transfer (e), and station (f). The user has selected 'c' for Gangnam.

강남역에서 최초로 탑승을 하는 경우

The screenshot displays a test environment with two windows: 'bus - 메모장' and a terminal window. The bus log shows boarding at 0:19 and leaving at 0:23. The metro log shows boarding at 0:15 and another boarding at 0:42. The terminal output shows the current time as 00:40, followed by 'a00:41' and '00:42'. It then displays 'TAG TIME 00:42' and 'USER INFORMATION' with details: lasttagtime: 0:23, transport: BUS, inout: LEFT, balance: 14700, terminal: a_1. A menu is shown with options for boarding (0), transfer (1), station (b), Gangnam (c), Shinlim (d), transfer (e), and station (f). The user has selected 'c' for Gangnam.

이전에 지하철에서 탑승을 하고 다시 강남역에서 내린 경우.



Test case 정산

(버스->지하철->버스 순으로 환승이 2번잇는경우)

버스에서 탑승금액과 하차금액은 더하고 지하철에서도 역시 탑승과 하차금액을 더한다. 다만, 탑승으로 끝난 미정산의 경우 더해주진 않는다.

결과적으로 버스2개와 지하철 2개에 대한 정산으로 바뀌고, 이에 대해 정산을 시행해 준다.

```

busCompany - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
0:58,BUS,1383|1:18,BUS,71|

metroCompany - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
0:52,METRO,1050|1:7,METRO,496|

bus - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
0:58 BUS BOARD 1250 a_1|1:3 BUS LEFT 0 a_1|1:18 BUS BOARD 0 a_1|1:57 BUS LEFT 100 a_1|

metro - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
0:52 METRO BOARD 1050 c_1|1:7 METRO BOARD 0 a_1|1:11 METRO LEFT 600 a_1|

~/t4
0번 째 : lasttagtime = 0:52
0번 째 : 버스 (0) / 지하철 (1) = 1
0번 째 : fee = 1050
0번 째 : terminal = c_1

1번 째 : lasttagtime = 0:58
1번 째 : 버스 (0) / 지하철 (1) = 0
1번 째 : fee = 1383
1번 째 : terminal = a_1

2번 째 : lasttagtime = 1:7
2번 째 : 버스 (0) / 지하철 (1) = 1
2번 째 : fee = 496
2번 째 : terminal = a_1

3번 째 : lasttagtime = 1:18
3번 째 : 버스 (0) / 지하철 (1) = 0
3번 째 : fee = 71
3번 째 : terminal = a_1
  
```