**System Test Report  
for Public Transportation System**

**• Test Cases Specification**

**• Test Summary Report**

**Project Team**

**Team 3**

Latest update on:

**2014-12-10**

**Team Information**

**201311269 김제헌**

**201311275 박상희**

**201311276 박형민**

**201311287 엄현식**

**Table of Contents**

[**1** **Introduction** 4](#_Toc405430441)

[**1.1** **Objectives** 4](#_Toc405430442)

[**1.2** **References** 4](#_Toc405430444)

[**2** **System test case specification** 4](#_Toc405430447)

[**2.1** **Test case specification identifier** 4](#_Toc405430448)

[**2.2** **Test items** 4](#_Toc405430450)

[**2.3** **Input specifications** 4](#_Toc405430452)

[**2.4** **Output specifications** 4](#_Toc405430454)

[**3** **Environmental needs** 4](#_Toc405430456)

[**4** **System test summary report** 4](#_Toc405430458)

[**4.1** **Test summary report identifier** 4](#_Toc405430459)

[**4.2** **Evaluation** 4](#_Toc405430461)

1. **Introduction**
   1. **Objectives**

본문서는 System test를 위한 design 문서이다. 요구사항으로부터 도출한 test 요소들을 실제 test가 가능한 case로 만들기 위한 design 문서이다.

* 1. **References**

DS-2014SE-PTS-SRS-1.0

T3-2014-PTS-SRA-1.3

1. **System test case specification**
   1. **Test case specification identifier**

**<Table 1 Test Case Identification> 참조.**

* 1. **Test items**

**<Table 2 Test Design Identification> 참조.**

* 1. **Input specifications**

**<Table 1 Test Case Identification> 참조.**

* 1. **Output specifications**

**<Table 1 Test Case Identification> 참조.**

1. **Environmental needs**

PTS를 실행하기 위한 PC – Cygwin & GCC Compiler 필요.

1. **System test summary report**
   1. **Test summary report identifier**

**<Table 3 Result Identification of the System Test> 참조.**

* 1. **Evaluation**

정산 시스템 부분이 한 테스터의 환경에서만 실행이 잘 안 되는 현상이 일어난다.

개발에 문제인지 환경에 문제인지 아직 파악이 안되었다.

**<Table 1 Test Cases Identification>**

|  |  |  |
| --- | --- | --- |
| Test Case Identifier | Input Specification | Output Specification |
| PTS.STC.000.000 |  |  |
| PTS.STC.001.000 |  |  |
| PTS.STC.100.000 |  |  |
| PTS.STC.200.000 | 지하철 일반 승/하차 시 지하철 역 간격이 1정거장이내 | 0원 부과 |
| PTS.STC.200.001 | 지하철 일반 승/하차 시 지하철 역 간격이 2정거장 일 때 | 200원부과 |
| PTS.STC.201.000 | 하차 태그 했을 때 다음 승차 태그 시간과 15초이내고, 교통수단이 달라졌을 때 | 환승 상태 |
| PTS.STC.202.000 | 버스를 환승 요금으로 탔을 때 | 30초마다 하차요금이 100원씩 증가 |
| PTS.STC.203.000 | 지하철을 환승 요금으로 탔을 때 | 1정거장마다 하차요금이 300원씩 증가 |
| PTS.STC.204.000 | 전에 미정산 요금부과 없이 정상 승차 처리시, 기존 카드에 1050원 보다 적은 금액이 들어있다. | “잔액 부족”이란 문구를 출력하고 승차되지 않은 채 초기화면으로 돌아간다. |
| PTS.STC.204.001 | 전에 지하철 미정산 요금부과상태에서 정상 승차 처리시, 기존 카드에 1250원 보다 적은 금액이 들어있다. | “잔액 부족”이란 문구를 출력하고 승차되지 않은 채 초기화면으로 돌아간다. |
| PTS.STC.204.002 | 지하철에서 버스로 환승 할 때,기존 카드에 700원 보다 적은 금액이 들어 있다. | “잔액 부족”이란 문구를 출력하고 승차되지 않은 채 초기화면으로 돌아간다. |
| PTS.STC.204.003 | 버스에서 지하철로 환승 할 때 기존 카드에 600원 보다 적은 금액이 들어 있다. | “잔액 부족”이란 문구를 출력하고 승차되지 않은 채 초기화면으로 돌아간다. |
| PTS.STC.204.004 | 전에 지하철 환승 미정산 요금부과 상태에서 정상 승차처리시, 기존 카드에 1650원 보다 적은 금액이 들어있다. | “잔액 부족”이란 문구를 출력하고 승차되지 않은 채 초기화면으로 돌아간다. |
| PTS.STC.204.005 | 전에 버스 환승 미정산 요금부과 상태에서 정상 승차처리시, 기존 카드에 1750원 보다 적은 금액이 들어있다. | “잔액 부족”이란 문구를 출력하고 승차되지 않은 채 초기화면으로 돌아간다. |
| PTS.STC.204.006 | 카드잔액이 부과금액보다 많다. | 카드의 잔액에서 부과금액을 빼고 카드의 상태를 승차 상태로 바꿔준다. 현재 타고 있는 교통의 정보를 입력시켜준다. |
| PTS.STC.205.000 | 지하철승차 -> 지하철하차 -> 버스승차 -> 승차f | 1750원 |
| PTS.STC.205.001 | 지하철승차 -> 승차 | 1250원 |
| PTS.STC.205.002 | 버스승차 -> 버스하차 -> 지하철승차 -> 승차 | 1650원 |
| PTS.STC.300.000 | 실행한지 3분마다 | 정산 실행 |
| PTS.STC.300.001 | 실행한지 3분마다가 아니면 | 정산을 실행하지 않는다. |
| PTS.STC.301.000 | 하루(3분)종료 시, 버스와 지하철의 일별 요금 목록 | 버스와 지하철에 각각 수익을 배분하는지에 대한 확인 |
| PTS.STC.302.000 | 정산 완료 신호 | 탑승되어 있는 승객은 미 정산으로 처리 |
| PTS.STC.303.000 | 정산 완료 신호 | 정산 시 소수점 이하는 반올림한다. |
| PTS.STC.304.000 | 하루(3분)이 종료될 시,  버스와 지하철 단말기의 각 단말기 파일로부터 하루치 결제 정보(일별 요금 목록)를 전달 받는다 | 정산 결과(데이터)를 출력  버스 : ㅁㅁ원  지하철 : ㅇㅇ원 |
| PTS.STC.305.000 | 환승, 미정산과 관계 없이 그냥 버스만 탑승한 상황 | 정산 금액 버스 += 버스 기본 요금(1050원) |
| PTS.STC.305.001 | 환승, 미정산과 관계 없이 그냥 버스만 탑승한 상황으로 0~1개역을 이동한 상황 | 정산 금액 지하철 += 지하철 기본 요금(1050원) |
| PTS.STC.305.002 | 환승, 미정산과 관계 없이 그냥 지하철만 탑승한 상황으로 2개역을 이동한 상황 | 정산 금액 지하철 += 지하철 기본 요금(1250원) |
| PTS.STC.305.003 | 미정산과 관계 없이  버스지하철 환승한 상황으로 n개역을 이동한 상황 | 정산 금액 버스 +=  (1050+300\*n)\*(1050+300\*n)/((1050+300\*n)+300\*n)원  정산 금액 지하철 +=  300\*n\*(1050+300\*n)/  ((1050+300\*n)+300\*n)원 |
| PTS.STC.305.004 | 미정산과 관계 없이  지하철버스 환승한 상황으로 탑승후 0~n초일 때 하차태그한 상황 | 정산 금액 버스 +=  (1050+n/30\*100)\* (n/30\*100)/((1050+n/30\*100)+n/30\*100)원  정산 금액 지하철 +=  (1050+n/30\*100)\* (1050+n/30\*100)/  ((1050+n/30\*100)+n/30\*100)원 |
| PTS.STC.305.005 | 전에 정상처리로 지하철을 탄 상황에서 하차태그를 찍지 않고 내려 미정산 요금이 있을 때, 버스를 탄 상황 | 정산 금액 버스 += 버스 기본 요금(1050원)  정산 금액 지하철 += 지하철 미정산 요금(200원) |
| PTS.STC.305.006 | 전에 정상처리로 지하철을 탄 상황에서 하차태그를 찍지 않고 내려 미정산 요금이 있을 때, 지하철을 탄 상황 | 정산 금액 지하철 += 지하철 기본 요금(1050원)  정산 금액 지하철 += 지하철 미정산 요금(200원) |
| PTS.STC.305.007 | 전에 환승처리로 지하철을 탄 상황에서 하차태그를 찍지 않고 내려 미정산 요금이 있을 때, 버스를 탄 상황 | 정산 금액 버스 += 버스 기본 요금(1050원)  정산 금액 지하철 += 지하철 최대 미정산 요금(600원) |
| PTS.STC.305.008 | 전에 환승처리로 버스를 탄 상황에서 하차태그를 찍지 않고 내려 미정산 요금이 있을 때, 버스를 탄 상황 | 정산 금액 버스 += 지하철 기본 요금(1050원)  정산 금액 지하철 += 버스 최대 미정산 요금(700원) |
| PTS.STC.305.009 | 미정산과 관계 없이  버스지하철버스 환승한 상황 | \*\*(총요금 = (버스1 + 지하철 + 버스2) + (지하철 + 버스2) + (버스2)) \*\*(부과요금 = 버스1 + 지하철 + 버스2) 정산 금액 버스 += (버스1 + 지하철 + 버스2) / (총요금) \* (부과요금) 정산 금액 지하철 += (지하철 + 버스2) / (총요금) \* (부과요금)  정산 금액 버스 += (버스2) / (총요금) \* (부과요금) |
| PTS.STC.305.0010 | 미정산과 관계 없이  지하철버스지하철 환승한 상황 | \*\*(총요금 = (지하철1 + 버스 + 지하철2) + (버스 + 지하철2) + (지하철2)) \*\*(부과요금 = 지하철1 + 버스 + 지하철2) 정산 금액 버스 += (지하철1 + 버스 + 지하철2) / (총요금) \* (부과요금) 정산 금액 지하철 += (버스 + 지하철2) / (총요금) \* (부과요금)  정산 금액 지하철 += (지하철2) / (총요금) \* (부과요금) |
| PTS.STC.305.0011 | 미정산과 관계 없이  버스지하철버스지하철 환승한 상황 | \*\*(총요금 = (버스1 + 지하철1 + 버스2 + 지하철2) + (지하철1 + 버스2 + 지하철2) + (버스2 + 지하철2) + (지하철2)) \*\*(부과요금 = 버스1 + 지하철1 + 버스2 + 지하철2) 정산 금액 버스 += (버스1 + 지하철1 + 버스2 + 지하철2) / (총요금) \* (부과요금) 정산 금액 지하철 += (지하철1 + 버스2 + 지하철2) / (총요금) \* (부과요금)  정산 금액 버스 += (버스2 + 지하철2) / (총요금) \* (부과요금) 정산 금액 지하철 += (지하철2) / (총요금) \* (부과요금) |
| PTS.STC.305.012 | 미정산과 관계 없이  지하철버스지하철버스 환승한 상황 | \*\*(총요금 = (지하철1 + 버스1 + 지하철2 + 버스2) + (버스1 + 지하철2 + 버스2) + (지하철2 + 버스2) + (버스2))  \*\*(부과요금 = 지하철1 + 버스1 + 지하철2 + 버스2)  정산 금액 지하철 += (지하철1 + 버스1 + 지하철2 + 버스2) / (총요금) \* (부과요금)  정산 금액 버스 += (버스1 + 지하철2 + 버스2) / (총요금) \* (부과요금)  정산 금액 지하철 += (지하철2 + 버스2) / (총요금) \* (부과요금)  정산 금액 버스 += (버스2) / (총요금) \* (부과요금) |
| PTS.STC.306.000 | 정산 완료 시 | 버스회사 및 지하철 회사로 정산한 금액을 각각 전송 |
| PTS.STC.307\_000 | 정산 완료 시 | 버스회사 및 지하철 회사로 정산 완료 신호를 각각 전송 |
| PTS.STC.400.000 | CardID | 승차 시 항상 카드를 태그하며 하차 시 카드를 태그 할 수도 있고 안 할 수도 있다. |
| PTS.STC.500.000 | 정산 완료 신호 | PTS 작동 |
| PTS.STC.500.001 | 정산 완료 신호가 없다. | PTS 작동되지 않는다. |
| PTS.STC.600.000 | 정산 완료 신호 | 사용자 카드 정보를 제외한 모든 정보가 초기화 |
| PTS.STC.700.000 | 파일 저장 | 교통카드 태그 행위 시 해당 파일에 제대로 입력이 되었는지에 대한 확인 |
| PTS.STC.701.000 | Tag | Updated 사용자 카드 |
| PTS.STC.701.001 | Tag | Updated 단말기 파일 |
| PTS.STC.704.000 | 버스에서 정상 탑승 태그시 환승이 아닐경우, | 버스에서 탑승했다는 정보를 교통카드와 단말기 누적 결재 기록에 기록 |
| PTS.STC.704.001 | 건대입구역에서 정상 탑승 태그시 환승이 아닐경우, | 건대입구역에서 탑승했다는 정보를 교통카드와 단말기 누적 결재 기록에 기록 |
| PTS.STC.704.002 | 강남역에서 정상 탑승 태그시 환승이 아닐경우, | 강남역에서 탑승했다는 정보를 교통카드와 단말기 누적 결재 기록에 기록 |
| PTS.STC.704.003 | 신림역에서 정상 탑승 태그시 환승이 아닐경우, | 신림역에서 탑승했다는 정보를 교통카드와 단말기 누적 결재 기록에 기록 |
| PTS.STC.704.004 | 합정역에서 정상 탑승 태그시 환승이 아닐경우, | 합정역에서 탑승했다는 정보를 교통카드와 단말기 누적 결재 기록에 기록 |
| PTS.STC.704.005 | 동대문역사문화공원역에서 정상 탑승 태그시 환승이 아닐경우, | 동대문역사문화공원역에서 탑승했다는 정보를 교통카드와 단말기 누적 결재 기록에 기록 |
| PTS.STC.705.000 | 버스에서 정상 탑승 태그시 환승인 경우, | 카드에 저장되어 있는 이전 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록 |
| PTS.STC.705.001 | 건대입구역에서 정상 탑승 태그시 환승인 경우, | 카드에 저장되어 있는 이전 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록 |
| PTS.STC.705.002 | 강남역에서 정상 탑승 태그시 환승인 경우, | 카드에 저장되어 있는 이전 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록 |
| PTS.STC.705.003 | 신림역에서 정상 탑승 태그시 환승인 경우, | 카드에 저장되어 있는 이전 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록 |
| PTS.STC.705.004 | 합정역에서 정상 탑승 태그시 환승인 경우, | 카드에 저장되어 있는 이전 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록 |
| PTS.STC.705.005 | 동대문역사문화공원역에서 정상 탑승 태그시 환승인 경우, | 카드에 저장되어 있는 이전 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록 |
| PTS.STC.706.000 | 각 파일로부터 입력을 받는다. | 만약 매번 각 파일로부터 입력받은 값으로부터 올바르게 정산이 되었다면, 그것을 통해, 각각의 일별 요금 목록을 버스와 지하철 단말기로부터 입력 받는지를 증명할 수 있다. |
| PTS.STC.800.000 | Card Information | price, cash, Tag time, Daily Charge List, Card information after charge |

**<Table 2 System Test Design Identification>**

|  |  |  |
| --- | --- | --- |
| Identifier | Feature | Vaild/Invaild |
| PTS.STC.000 | 기본 구성 | 버스 1대와 지하철 5개역 (건대입구, 왕십리(동대문 역사 문화공원), 합정, 신림, 강남)이 있는지 확인 한다. |
| PTS.STC.001 | 기본 구성 | PTS가 총 3가지(버스용 단말기, 지하철용 단말기, 정산 시스템)로 구성 되었는지 확인 |
| PTS.STC.002 | 기본 구성 | 하드웨어 인터페이스는 고려하지 않는다. |
| PTS.STC.003 | 기본 구성 | 카드 인식은 태그(카드ID를 입력하는 행위)를 통해 수행되는지에 대한 확인 |
| PTS.STC.004 | 기본 구성 | 각 역마다 하나의 지하철 단말기가 독립적으로 작동하는지에 대한 확인 |
| PTS.STC.100 | 태그 | 탑승 태그와 하차 태그가 가능한지의 확인 |
| PTS.STC.200 | 기본요금 | 1개역 이하를 이동하면 기본료(1050원)가 부과되며 두 개역을 이동하면 추가 요금(200원)이 부과되는지 확인 |
| PTS.STC.201 | 환승 조건 | 하차 태그 했을 경우에만 버스와 지하철 간의 정해진 시간(15초)내에 환승이 가능한지 확인. |
| PTS.STC.202 | 지하철->버스 환승 요금 | 지하철->버스 단위 시간 당 버스의 환승 요금이 추가로 부과(30초당 100원/최대 700원) 되는지에 대한 확인 |
| PTS.STC.203 | 버스->지하철 환승 요금 | 버스->지하철 역 당 지하철의 환승 요금(역당 300원/최대 600원)이 추가로 부과되는지에 대한 확인 |
| PTS.STC.204 | 잔액 부족 | 잔액이 모자를 경우 태우지 않는지에 대한 확인(초기 탑승 시 : 기본요금인 1050원 / 환승 시 : [버스: 환승 최고 부과금액인 700원/지하철 환승 최고 부과금액인 600원] / \*미정산 요금이 있을 시 : 기본요금(1050원)+ 미 정산 요금(200원or600원or700원) ) |
| PTS.STC.205 | 환승 요금 결정 조건 | 지하철에서 버스 환승 후 하차 시 단말기를 태그 하지 않았을 경우 환승 최고 요금인 700원, 지하철 하차 후 미정산 금액은 200원, 버스에서 지하철 환승 후 미정산 금액은 지하철 환승 최대 요금인 600원을 부과하는지에 대한 확인 |
| PTS.STC.300 | 정산 주기 | 정산은 하루(3분)에 한번만 이뤄지는 지에 대한 확인. |
| PTS.STC.301 | 수익 배분 | 버스와 지하철에 각각 수익을 배분하는지에 대한 확인 |
| PTS.STC.302 | 정산 여부 | 하루(3분) 종료 시 탑승되어 있는 승객은 미정산으로 처리되는지에 대한 확인 |
| PTS.STC.303 | 정산 제약 조건 | 정산 시 소수점 이하는 반올림하는지에 대한 확인 |
| PTS.STC.304 | 정산 시스템 인터페이스 | 정산 시스템의 사용자 인터페이스에서 버스와 지하철 단말기의 일별 요금 목록의 입력을 받고 정산 결과(데이터)를 출력 하는지에 대한 확인 |
| PTS.STC.305 | 정산 조건 | 정해진 정산 방법을 따르는지에 대한 확인 |
| PTS.STC.306 | 정산 결과 전송 | 정산한 금액을 각 회사(버스, 지하철)에 전송하는지에 대한 확인 |
| PTS.STC.307 | 정산 완료 신호 전송 | 정산 완료 신호를 전송하는지에 대한 확인 |
| PTS.STC.400 | 사용자의 행동 | 승차 시 카드를 태그하며 하차 시 카드를 태그 할 수도 있고 안 할 수도 있는지에 대한 확인 |
| PTS.STC.500 | 운행 시작 조건 | 운행이 시작하기 전까지 정산이 반드시 이루어 지는지에 대한 확인 |
| PTS.STC.600 | 단말기 정보 초기화 | 정산 후 사용자 카드 정보를 제외한 모든 정보가 초기화 되는지에 대한 확인 |
| PTS.STC.700 | 파일 저장 | 교통카드 태그 행위 시 해당 파일에 제대로 입력이 되었는지에 대한 확인 |
| PTS.STC.701 | 파일 저장 | 계산된 부과 요금과 기타 정보를 사용자 카드와 단말기 파일에 기록하는지에 대한 확인 |
| PTS.STC.702 | 파일 저장 | Data 구분은 쉼표(,)를 사용하는지에 대한 확인 |
| PTS.STC.703 | 파일 저장 | 각 단말기는 탑승 단말기 정보 (단말기 고유 ID + 하루 동안 태그 된 Count)를 가지고 있는지에 대한 확인 |
| PTS.STC.704 | 파일 저장 | 사용자가 단말기에 카드를 태그 할 경우 단말기는 환승 확인을 확인 후 환승이 아닐 경우 해당 탑승 단말기 정보를 교통카드와 단말기 누적 결재 기록에 기록되는지에 대한 확인 |
| PTS.STC.705 | 파일 저장 | 사용자가 단말기에 카드를 태그 할 경우 단말기는 환승 확인을 확인 후 환승일 경우 카드에 저장되어 있는 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록하는지에 대한 확인 |
| PTS.STC.706 | 파일 입력 | 각각의 일별 요금 목록을 버스와 지하철 단말기로부터 입력 받는지에 대한 확인 |
| PTS.STC.800 | PTS 인터페이스 | PTS의 사용자 인터페이스에서 교통카드 정보를 입력 받고 사용금액, 잔액/시간, 일별 요금 목록, 거래 후 교통 카드 정보를 출력하는지에 대한 확인 |

**<Table 3 Result Identification of the System Test>**

|  |  |
| --- | --- |
| Identifier | PASS/FAIL |
| PTS.STC.000.000 | PASS |
| PTS.STC.001.000 | PASS |
| PTS.STC.100.000 | PASS |
| PTS.STC.200.000 | PASS |
| PTS.STC.200.001 | PASS |
| PTS.STC.201.000 | PASS |
| PTS.STC.202.000 | PASS |
| PTS.STC.203.000 | PASS |
| PTS.STC.204.000 | PASS |
| PTS.STC.204.001 | PASS |
| PTS.STC.204.002 | PASS |
| PTS.STC.204.003 | PASS |
| PTS.STC.204.004 | PASS |
| PTS.STC.204.005 | PASS |
| PTS.STC.204.006 | PASS |
| PTS.STC.205.000 | PASS |
| PTS.STC.205.001 | PASS |
| PTS.STC.205.002 | PASS |
| PTS.STC.300.000 | PASS |
| PTS.STC.300.001 | PASS |
| PTS.STC.301.000 | PASS |
| PTS.STC.302.000 | PASS |
| PTS.STC.303.000 | PASS |
| PTS.STC.304.000 | PASS |
| PTS.STC.305.000 | PASS |
| PTS.STC.305.001 | PASS |
| PTS.STC.305.002 | PASS |
| PTS.STC.305.003 | PASS |
| PTS.STC.305.004 | PASS |
| PTS.STC.305.005 | PASS |
| PTS.STC.305.006 | PASS |
| PTS.STC.305.007 | PASS |
| PTS.STC.305.008 | PASS |
| PTS.STC.305.009 | PASS |
| PTS.STC.305.010 | PASS |
| PTS.STC.305.011 | PASS |
| PTS.STC.305.012 | PASS |
| PTS.STC.306.000 | PASS |
| PTS.STC.307.000 | PASS |
| PTS.STC.400.000 | PASS |
| PTS.STC.500.000 | PASS |
| PTS.STC.500.001 | PASS |
| PTS.STC.600.000 | PASS |
| PTS.STC.700.000 | PASS |
| PTS.STC.701.000 | PASS |
| PTS.STC.701.001 | PASS |
| PTS.STC.702.000 | FAIL |
| PTS.STC.703.001 | FAIL |
| PTS.STC.704.000 | PASS |
| PTS.STC.704.001 | PASS |
| PTS.STC.704.002 | PASS |
| PTS.STC.704.003 | PASS |
| PTS.STC.704.004 | PASS |
| PTS.STC.704.005 | PASS |
| PTS.STC.705.000 | FAIL |
| PTS.STC.705.001 | FAIL |
| PTS.STC.705.002 | FAIL |
| PTS.STC.705.003 | FAIL |
| PTS.STC.705.004 | FAIL |
| PTS.STC.705.005 | FAIL |
| PTS.STC.706.000 | PASS |
| PTS.STC.800.000 | PASS |

Comment

기본적으로는 거의 기본적인 조건은 거의 다 만족을 했다. 정산부분도 잘 돌아가기는 하나 엄밀하게 잘 돌아가는지에 대해서는 테스팅할 시간이 부족해서 일단 보류로써 빈칸으로 남기었다.

확실하게 FAIL표시가 난 것들에 대해서 언급해 보겠다.

PTS.STC.303.000와 PTS.STC.702.000이 FAIL인데, 이는 개발할 당시에는 파악하지 못했던 것으로, 이번에 SRS을 꼼꼼히 파악해보면서 비교해보다가 발견하게 된 점들이다. 소수점 이하 반올림은 추후에 수정할 것이고, 파일 양식으로는 ‘,’없이 지금도 잘 돌아가지만, 꼭 이 양식을 지켜야 한다면, 추후에 파싱을 하면서 수정할 예정이다.

PTS.STC.703.001부분은 각 단말기가 해당 단말기의 고유 정보로써, 단말기의 고유 ID와 하루 동안 태그 된 Count를 가지고 있는지에 대한 여부를 어겼다는 것인데, ‘하루 동안 태그 된 Count’를 포함하지 못해서 발생한 문제였다. 놓쳤다면 놓쳤다고 볼 수도 있지만, 나중에 파악했을 때는 해당 count값이 단순히 보여주거나 통계를 위해서가 아니라면 도저히 필요가 없을 뿐더러(count가 없어도 잘 돌아간다), 그것 하나만을 현재 단말기 파일의 구조에 포함시기에는 적합하지 않다고 판단해서 파일에 별도로 넣지 않았다. 만약 하루동안 단말기에 태그 된 Count값이 정말로 필요하다면 파일에 별도로 저장을 하지 않더라도 파일을 열고 태그된 횟수를 셈으로써 얼마든지 count값을 얻어낼 수 있다고 판단했기 때문에 count값을 포함하지 않았던 것이다.

마지막으로 PTS.STC.705.000 ~ PTS.STC.705.005은 각 역(버스 및 지하철) 별로 정상적으로 탑승태그를 하는데 환승인 경우, 카드에 저장되어 있는 이전 탑승 단말기 정보를 읽어와 교통카드와 단말기 누적 결제 기록에 기록하는지 확인하는 목록이다.

이 부분도 개발 도중에 SRS를 파악하다가 확인했다가 무시했던 부분으로, 원인은 그렇게는 구현할 수 없다고 판단했기 때문이다. 엄밀하게 말하면, 그런식으로 저장되도록 PTS부분에서 구현하는 것은 문제가 아니나 그렇게 구현했을 시, 그렇게 저장된 자료를 바탕으로 FCS부분에서 정산을 할 때 어떻게 자료들을 거슬러 올라갈 것인가 하는 부분이 알고리즘적인 측면에서 막혀서 포기했다. 현재는 환승이라고 이전 탑승 단말기 정보(CRID)를 저장하는 것이 아닌, 환승여부에 관계없이 해당 탑승 단말기 정보(CRID)를 단말기의 누적기록으로 저장한다.

\*수정한 결과(PASS)

처음에는 단말기 정보를 저장할 때 시그윈 환경에서는 한 줄로 저장하여 기존의 파일을 불러오는 방식과 맞지 않아 segmentation fault가 일어나서 Fail이 났습니다. 그래서 파일을 불러오는 방식을 바꾸었습니다. 또한 time.h에 있는 mktime함수의 경우 데이터의 손실이 일어나 정렬이 잘 이루어지지 않았습니다. 그래서 maketime이라는 함수를 만들어서 데이터의 손실이 일어나지 않도록 하여 정렬이 잘 이루어져 정산이 잘 작동하게 했습니다.

결국 우리 팀의 결과는 SRS에서 정말로 놓쳐서라던가, SRS에서 명시는 되어있지만 불필요할 뿐더러 별도로 언제든지 값을 얻을 수 있어서라든가, 우리가 판단하게에 SRS에 명시된 대로 구현하면 답이 없다고 판단해서 피한 부분을 제외하면 모두 SRS에 명시된 대로 돌아간다고 볼 수 있다.

정산부분은 추후에 더 정밀한 테스트를 통해 오류여부를 점검해볼 것이다.