

Unit Testing Plan

for Public Transportaion System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

AT2 Team

Date

2014-11-21

Team Information

201011337 변영석

201011349 이규진

201011365 정재혁

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.....	6
4	Features not to be tested	7
5	Approach.....	9
6	Item pass/fail criteria	9
7	Unit test design specification.....	9
7.1	Test design specification identifier	9
7.2	Features to be tested	9
7.3	Approach refinements.....	9
7.4	Test identification	10
7.5	Feature pass/fail criteria	12
8	Unit test case specification.....	12
8.1	Test case specification identifier	13
8.2	Test items	15

8.3 Input specifications..... 15

8.4 Output specifications..... 16

9 Testing tasks 17

10 Environmental needs 17

11 Unit Test deliverables..... 18

12 Schedules 18

1 Introduction

1.1 Objectives

Public Transportation System의 Unit test를 위해 필요한 활동과 기준에 대한 정의, 환경적인 요구사항, test 도구들에 관해 세부적으로 명시한다.

1.2 Background

Public Transportation System은 대중 교통 시스템으로 사용자가 카드를 이용하여 센스에 태그 했을 때 입력 값에 따라서 알맞은 요금을 계산하여 그 결과를 다시 그 카드에 출력하는 시스템이다. 또한 그 요금을 매일 1회씩 정산하여 버스와 지하철 회사에 결과를 보내준다.

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

이 계획 문서는 Public Transportation System(이하 PTS)의 unit test를 수행하기 위한 모든 것을 포함한다. EDLS의 unit test를 수행하기 위한 자원과 절차, test approach와 technique과 필요로 하는 환경 및 도구 등을 정의한다. PTS의 unit test는 시스템을 구성하는 최소 단위의 모듈들을 대상으로 하며, 구현된 모듈이 요구사항을 만족하는지를 test한다.

1.4 Project plan

1.5 Configuration management plan

1.6 References

[2014SE_A][T2]SRA_2_1

[2014SE_A][T2]SDS

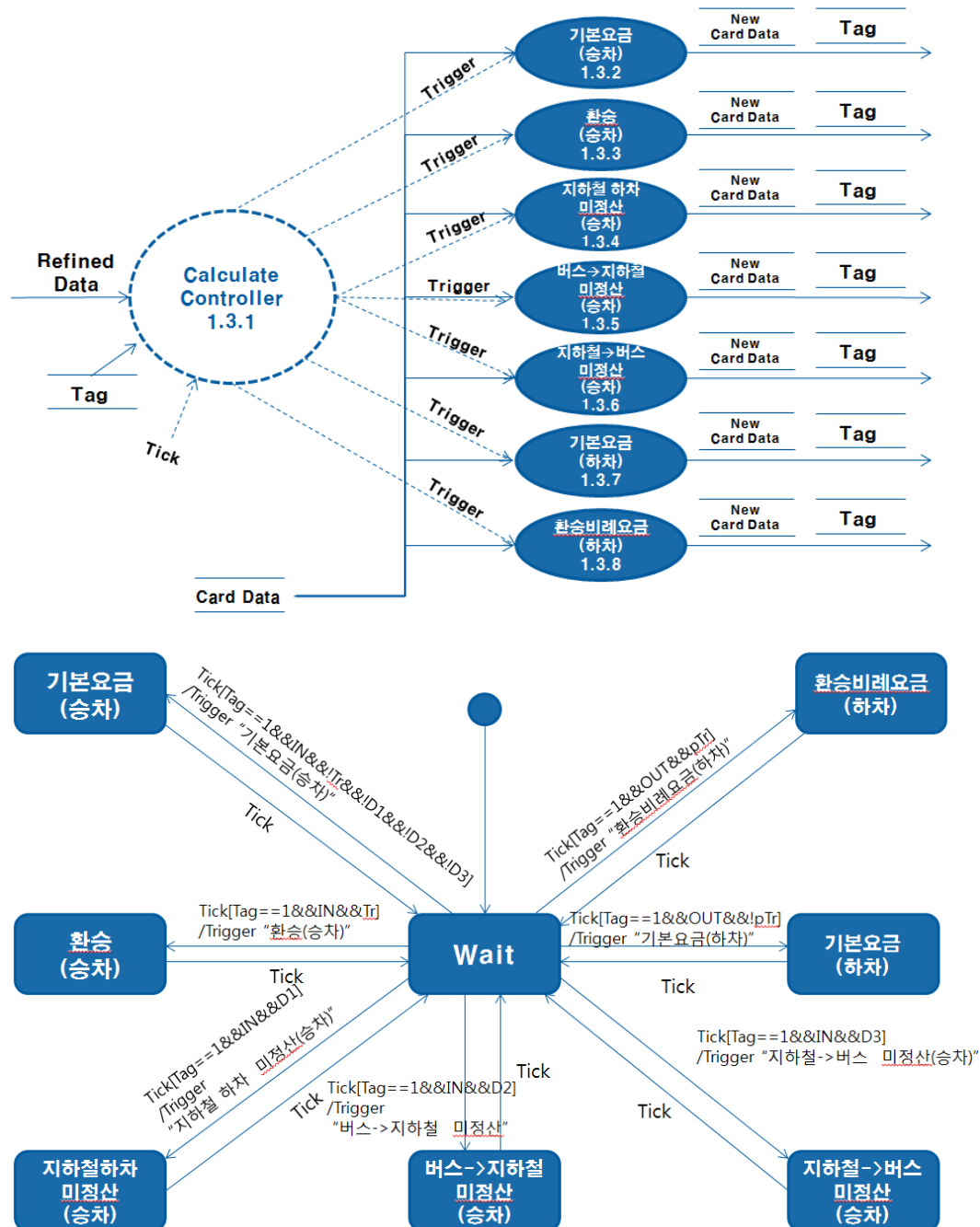
DS-2014SE-PTS-SRS-1.0

2 Test items

EDLS를 구성하는 최소 단위의 모듈들이 unit test의 대상이 된다. 각 모듈의 요구사항을 만족하는지를 test하며, test item은 다음 자료들로부터 작성되었다.

(1) Functionality of modules – [2014SE_A][T2]SRA_2_1 : Process specification

아래 그림은 일부를 참조한 것이다.



3 Features to be tested

(1) Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는지를 test한다.

(2) Modules in SDS : 각 모듈이 가지고 있는 데이터 인터페이스를 test한다. <Table 1,2,3 테스트 할 Process(DFD) 리스트>의 Process name 참조

1) Table 1 테스트 할 Bus System Process (DFD) 리스트

ID	Name	Description
1.2	Refine Process	Card Data와 Tag를 통해 계산을 하는데 필요한 정보만 담은 Refined Data를 보낸다.
1.3.1	Calculate Controller	Refined Data를 받아 필요한 프로세스의 Trigger를 발동한다.
2.1.1	Output Contorller	계산된 Card Data를 받아 알맞은 Display에 Trigger를 발동하고 Fair Recorder Command, Card Recorder Command를 알맞게 Control한다.
2.1.4	Tag	잔액이 충분한 교통카드가 태그 하였을 때의 Display를 출력하는 Command를 보낸다.

2) Table 2 테스트 할 Subway System Process (DFD) 리스트

ID	Name	Description
1.2	Refine Process	Card Data와 Tag를 통해 계산을 하는데 필요한 정보만 담은 Refined Data를 보낸다.
1.7.1	Calculate Controller	Refined Data를 받아 필요한 프로세스의 Trigger를 발동한다.
2.1.1	Output Contorller	계산된 Card Data를 받아 알맞은 Display에 Trigger를 발동하고 Fair Recorder Command, Card Recorder Command를 알맞게 Control한다.
2.1.4	Tag	잔액이 충분한 교통카드가 태그 하였을 때의 Display를 출력하는 Command를 보낸다.

3) Table 3 테스트 할 Adjust System Process (DFD) 리스트

ID	Name	Description
1.2	Sorting Process	List Data를 탑승 단말기 정보를 이용해 정렬하여 정렬된 Data를 내보낸다.
2.1	Main Control	들어온 List Data를 통해 Display 및 각각의 Send Command를 알맞게 Control한다.

4 Features not to be tested

(1) Processes in SRA: 외부 장치 드라이버, 단순 데이터 전달 프로세스 등은 test에서 제외한다.

(2) Modules in SDS: <Table 4,5,6 테스트하지 않을 Process(DFD)리스트>의 Processes name 참조

1) Table 4 테스트하지 않을 Bus System Process (DFD) 리스트

ID	Name	Description
1.1	Bus Sensor Interface	Bus Sensor에 교통카드가 Tag가 되면 Sensor로부터 들어온 카드기록을 통해 Card Data와 Tag를 보내준다.
1.3.3	환승(승차)	Trigger를 받아 환승을 하고 승차한 카드의 계산된 Card Data를 보낸다.
1.3.4	지하철 하차 미정산(승차)	Trigger를 받아 지하철 승차 후 미정산한 카드의 요금을 계산한 뒤 계산된 Card Data를 보낸다.
1.3.5	버스->지하철 미정산(승차)	Trigger를 받아 버스에서 지하철로 환승한 뒤 미정산한 카드의 요금을 계산한 뒤 계산된 Card Data를 보낸다.
1.3.6	지하철->버스 미정산(승차)	Trigger를 받아 지하철에서 버스로 환승한 뒤 미정산한 요금을 계산한 뒤 계산된 Card Data를 보낸다.
1.3.7	기본요금(하차)	Trigger를 받아 기본요금을 계산한 뒤 계산된 Card Data를 보낸다.
2.2	Display Interface	Command로 Display를 실행한다.
2.3	Fare Recorder Interface	Command로 Fare Record를 기록한다.
2.4	Card Recorder Interface	Command로 Card Record를 기록한다.
2.1.2	No-Tag	태그가 일어나지 않은 상황에서 평소의 Display를 출력하는 Command를 보낸다.

2.1.3	Balance Lack	잔액이 부족한 교통카드가 태그 하였을 때의 Display를 출력하는 Command를 보낸다.
--------------	--------------	---

2) Table 5 테스트하지 않을 Subway System Process (DFD) 리스트

ID	Name	Description
1.1	Subway Sensor Interface(건대입구)	Subway Sensor(건대입구)에 교통카드가 Tag가 되면 Sensor로부터 들어온 카드기록을 통해 Card Data와 Tag를 보내준다.
1.2	Subway Sensor Interface(강남)	Subway Sensor(강남)에 교통카드가 Tag가 되면 Sensor로부터 들어온 카드기록을 통해 Card Data와 Tag를 보내준다.
1.3	Subway Sensor Interface(신림)	Subway Sensor(신림)에 교통카드가 Tag가 되면 Sensor로부터 들어온 카드기록을 통해 Card Data와 Tag를 보내준다.
1.4	Subway Sensor Interface(합정)	Subway Sensor(합정)에 교통카드가 Tag가 되면 Sensor로부터 들어온 카드기록을 통해 Card Data와 Tag를 보내준다.
1..5	Subway Sensor Interface(동대문역사문화공원)	Subway Sensor(동대문역사문화공원)에 교통카드가 Tag가 되면 Sensor로부터 들어온 카드기록을 통해 Card Data와 Tag를 보내준다.
1.6	Refine Process	Card Data와 Tag를 통해 계산을 하는데 필요한 정보만 담은 Refined Data를 보낸다.
2.2	Display Interface	Command로 Display를 실행한다.
2.3	Fare Recorder Interface	Command로 Fare Record를 기록한다.
2.4	Card Recorder Interface	Command로 Card Record를 기록한다.
2.1.2	No-Tag	태그가 일어나지 않은 상황에서 평소의 Display를 출력하는 Command를 보낸다.
2.1.3	Balance Lack	잔액이 부족한 교통카드가 태그 하였을 때의 Display를 출력하는 Command를 보낸다.

3) Table 6 테스트하지 않을 Adjust System Process (DFD) 리스트

ID	Name	Description
1.1	List Sensor Interface	List Receiver Input을 통해 받은 각각의 List를 합쳐 List Data로 내보낸다.
2.2	Display Interface	Command로 Display를 실행한다.
2.3	Bus Company Interface	Command로 Bus Send를 실행한다.
2.4	Subway Company Interface	Command로 Subway Send를 실행한다.

5 Approach

Digital Door Lock System의 program source code 및 unit test를 위한 test code는 CTIP(Continuous Testing & Integration Platform) 환경에서 이루어지며, program source code/test code 의 변경 및 수정 사항은 지속적으로 통합되고 test 된다.

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

BUS.UTD.000.000

SUB.UTD.000.000

ADJ.UTD.000.000

7.2 Features to be tested

7.2.1 Processes in SRA

<Table 1,2,3 테스트 할 Process(DFD) 리스트> 참조

7.3 Approach refinements

7.3.1 Brute force testing

PTS의 각 모듈이 요구사항을 만족하는지를 확인하기 위하여, 요구사항에 정의된 내용에 기반하여 test case를 작성한다. 그 이외의 예외 상황에 대해서는 test case를 작성하지 않는다.

7.4 Test identification

1) Bus System

Identifier	Feature(Process ID in DFD)	value
BUS.UTC.120.000	1.2 Refine Process	newCD.inout == "OUT" && newCD.transport == "Metro" && CurTime - newCD.time < 200 Input
BUS.UTC.120.001	1.2 Refine Process	newCD.inout == "OUT" && newCD.transport == "Metro" && CurTime - newCD.time > 200 Input
BUS.UTC.120.002	1.2 Refine Process	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Metro" && oldCD.balance - newCD.balance == 1050
BUS.UTC.120.003	1.2 Refine Process	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Metro" && oldCD.balance - newCD.balance != 1050
BUS.UTC.120.004	1.2 Refine Process	newCD.sensorInfo != oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Metro"
BUS.UTC.120.005	1.2 Refine Process	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "OUT" && newCD.transport == "Metro"
BUS.UTC.120.006	1.2 Refine Process	newCD.inout == "IN" && newCD.transport == "BUS" && oldCD.balance == newCD.balance
BUS.UTC.120.007	1.2 Refine Process	newCD.inout == "IN" && newCD.transport == "BUS" && oldCD.balance != newCD.balance
BUS.UTC.131.000	1.3.1 Calculate Controller	Tag == 1 && Refined Data의 inout == "IN" && Tr == FALSE && d1 == FALSE && d2 == FALSE && d3 == FALSE Input
BUS.UTC.131.001	1.3.1 Calculate Controller	Tag == 10이고 Refined Data의 inout == "IN", Tr == TRUE Input

BUS.UTC.131.002	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "IN", d1 = TRUE Input
BUS.UTC.131.003	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "IN", d2 = TRUE Input
BUS.UTC.131.004	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "IN", d3 = TRUE Input
BUS.UTC.131.005	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "OUT" pTr == FALSE Input
BUS.UTC.131.006	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "OUT" pTr == TRUE Input
BUS.UTC.211.000	2.1.1 Output Controller	Tag == 1 이고 Balance < Fare Input
BUS.UTC.211.001	2.1.1 Output Controller	Tag == 1 이고 Balance > Fare Input

2) Subway System

Identifier	Feature(Process ID in DFD)	value
SUB.UTC.160.000	1.6 Refine Process	newCD.inout == "OUT" && newCD.transport == "Bus" && CurTime - newCD.time < 200 Input
SUB.UTC.160.001	1.6 Refine Process	newCD.inout == "OUT" && newCD.transport == "Bus" && CurTime - newCD.time > 200 Input
SUB.UTC.160.002	1.6 Refine Process	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Bus" && oldCD.balance - newCD.balance == 1050
SUB.UTC.160.003	1.6 Refine Process	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Bus" && oldCD.balance - newCD.balance != 1050
SUB.UTC.160.004	1.6 Refine Process	newCD.sensorInfo != oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Bus"
SUB.UTC.160.005	1.6 Refine Process	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "OUT" && newCD.transport == "Bus"
SUB.UTC.160.006	1.6 Refine Process	newCD.inout == "IN" && newCD.transport ==

		"Metro" && oldCD.balance == newCD.balance
SUB.UTC.160.007	1.6 Refine Process	newCD.inout == "IN" && newCD.transport == "Metro" && oldCD.balance != newCD.balance
SUB.UTC.171.000	1.3.1 Calculate Controller	Tag == 1 && Refined Data의 inout == "IN" && Tr == FALSE && d1 == FALSE && d2 == FALSE && d3 == FALSE Input
SUB.UTC.171.001	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "IN", Tr == TRUE Input
SUB.UTC.171.002	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "IN", d1 = TRUE Input
SUB.UTC.171.003	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "IN", d2 = TRUE Input
SUB.UTC.171.004	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "IN", d3 = TRUE Input
SUB.UTC.171.005	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "OUT" pTr == FALSE Input
SUB.UTC.171.006	1.3.1 Calculate Controller	Tag == 1이고 Refined Data의 inout == "OUT" pTr == TRUE Input
SUB.UTC.211.000	2.1.1 Output Controller	Tag == 1 이고 Balance < Fare Input
SUB.UTC.211.000	2.1.1 Output Controller	Tag == 1 이고 Balance > Fare Input

3) Adjust System

Identifier	Feature(Process ID in DFD)	value
ADJ.UTC.120.000	1.2.0 SortingProcess	Buffer != NULL
ADJ.UTC.212.000	2.1.2 Adjust	TF != NULL

7.5 Feature pass/fail criteria

PTS의 각 모듈(프로세스)은 SRA에 정의되어 있는 요구사항 (입력 / 출력 및 동작)을 모두 만족해야 한다. 각 모듈(프로세스)의 입력 / 출력 및 동작은 SRA의 process description 항목을 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

1) Bus System

Identifier	Input Specification	Output Specification
BUS.UTC.120.000	newCD.inout == "OUT" && newCD.transport == "Bus" && CurTime - newCD.time < 200 Input	TR = TRUE
BUS.UTC.120.001	newCD.inout == "OUT" && newCD.transport == "Bus" && CurTime - newCD.time > 200 Input	TR = FALSE
BUS.UTC.120.002	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Bus" && oldCD.balance - newCD.balance == 1050	PTR = TRUE D1 = TRUE
BUS.UTC.120.003	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Bus" && oldCD.balance - newCD.balance != 1050	PTR = TRUE D2 = TRUE
BUS.UTC.120.004	newCD.sensorInfo != oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Bus"	PTR = FALSE D2 = TRUE
BUS.UTC.120.005	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "OUT" && newCD.transport == "Bus"	PTR = TRUE D1 = FALSE D2 = FALSE
BUS.UTC.120.006	newCD.inout == "IN" && newCD.transport == "Metro" && oldCD.balance == newCD.balance	D3 = TRUE
BUS.UTC.120.007	newCD.inout == "IN" && newCD.transport == "Metro" && oldCD.balance != newCD.balance	D3 = FALSE
BUS.UTC.131.000	Tag == 1 && inout == "IN" && Tr == FALSE && d1 == FALSE && d2 == FALSE && d3 == FALSE Input	Fare = 1050
BUS.UTC.131.001	Tag == 1 && inout == "IN" && Tr == TRUE Input	Fare = 0
BUS.UTC.131.002	Tag == 1 && inout == "IN" && d1 = TRUE Input	Fare = 1250
BUS.UTC.131.003	Tag == 1 && inout == "IN" && d2 = TRUE Input	Fare = 1650
BUS.UTC.131.004	Tag == 1 && inout == "IN" && d3 = TRUE Input	Fare = 1750
BUS.UTC.131.005	Tag == 1 && inout == "OUT" && pTr == FALSE Input	Fare = 0
BUS.UTC.131.006	Tag == 1 && inout == "OUT" && pTr == TRUE Input && CurTime - pTagTime < 400	Fare = 0
BUS.UTC.131.007	Tag == 1 && inout == "OUT" && pTr == TRUE Input && CurTime - pTagTime = 400	Fare = 100

BUS.UTC.211.000	Tag == 1 && Balance < Fare Input	Balance_Lack();
BUS.UTC.211.001	Tag == 1 && Balance > Fare Input	Tag();

2) Subway System

Identifier	Input Specification	Output Specification
SUB.UTC.120.000	newCD.inout == "OUT" && newCD.transport == "Metro" && CurTime - newCD.time < 200 Input	TR = TRUE
SUB.UTC.120.001	newCD.inout == "OUT" && newCD.transport == "Metro" && CurTime - newCD.time > 200 Input	TR = FALSE
SUB.UTC.120.002	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Metro" && oldCD.balance - newCD.balance == 1050	PTR = TRUE D1 = TRUE
SUB.UTC.120.003	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Metro" && oldCD.balance - newCD.balance != 1050	PTR = TRUE D2 = TRUE
SUB.UTC.120.004	newCD.sensorInfo != oldCD.sensorInfo && newCD.inout == "IN" && newCD.transport == "Metro"	PTR = FALSE D2 = TRUE
SUB.UTC.120.005	newCD.sensorInfo == oldCD.sensorInfo && newCD.inout == "OUT" && newCD.transport == "Metro"	PTR = TRUE D1 = FALSE D2 = FALSE
SUB.UTC.120.006	newCD.inout == "IN" && newCD.transport == "BUS" && oldCD.balance == newCD.balance	D3 = TRUE
SUB.UTC.120.007	newCD.inout == "IN" && newCD.transport == "BUS" && oldCD.balance != newCD.balance	D3 = FALSE
SUB.UTC.131.000	Tag == 1 && inout == "IN" && Tr == FALSE && d1 == FALSE && d2 == FALSE && d3 == FALSE Input	Fare = 1050
SUB.UTC.131.001	Tag == 1 && inout == "IN" && Tr == TRUE Input	Fare = 0
SUB.UTC.131.002	Tag == 1 && inout == "IN" && d1 = TRUE Input	Fare = 1250
SUB.UTC.131.003	Tag == 1 && inout == "IN" && d2 = TRUE Input	Fare = 1650
SUB.UTC.131.004	Tag == 1 && inout == "IN" && d3 = TRUE Input	Fare = 1750
SUB.UTC.131.005	Tag == 1 && inout == "OUT" && pTr == FALSE Input	Fare = 0
SUB.UTC.131.006	Tag == 1 && inout == "OUT" && pTr == TRUE Input && CurTime - pTagTime < 400	Fare = 0

SUB.UTC.131.007	Tag == 1 && inout == "OUT" && pTr == TRUE Input && CurTime - pTagTime = 400	Fare = 100
SUB.UTC.211.000	Tag == 1 && Balance < Fare Input	Balance_Lack();
SUB.UTC.211.001	Tag == 1 && Balance > Fare Input	Tag();

3) Adjust System

Identifier	Input Specification	Output Specification
ADJ.UTC.120.000	Buffer != NULL	TF != NULL
ADJ.UTC.212.000	TF !=NULL	Total_BusFare Total_SubwayFare

8.2 Test items

7.4 Test identification의 value항목 참조

8.3 Input specifications

1) Bus System

Input	Input Specification
CardData.time	String
CardData.transport	String
CardData.inout	String
CardData.balance	int
CardData.sensorInfo	String
RefinedData.inout	String
RefinedData.tr	char
RefinedData.ptr	char
RefinedData.d1	char
RefinedData.d2	char
RefinedData.d3	char
RefinedData.balance	Int
RefinedData.pTagTime	Int
RefinedData.pSensorInfo	int
Tag	int

inout	String
-------	--------

2) Subway System

Input	Input Specification
CardData.time	String
CardData.transport	String
CardData.inout	String
CardData.balance	int
CardData.sensorInfo	String
RefinedData.inout	String
RefinedData.tr	char
RefinedData.ptr	char
RefinedData.d1	char
RefinedData.d2	char
RefinedData.d3	char
RefinedData.balance	Int
RefinedData.pTagTime	Int
RefinedData.pSensorInfo	int
Tag	int
inout	String

3) Adjust System

Input	Input Specification
Buffer	String
TF	Struct record{String, String, String,int, String,}

8.4 Output specifications

1) Bus System

Output	Output Specification
CardRecord	txt
FareRecord	txt

2) Subway System

Output	Output Specification
CardRecord	txt
FareRecord	Txt

3) Adjust System

Output	Output Specification
TF	Struct record{String, String, String,int, String,}
Total_BusFare	Int
Total_SubwayFare	int

9 Testing tasks

Task	Predecessor tasks	Special Skills	Effort	Finish data
1 Unit Test Plan 작성	PTS 구현			
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			

10 Environmental needs

PTS의 unit test를 위한 환경적 요구사항은 다음과 같다.

- (1) Hardware & Platform, gcc compiler/linker
- (2) CTIP(Continuous Testing & Integration Platform) Environment

Test tools

CUnit unit test framework for C

11 Unit Test deliverables

12 Schedules

9 Testing tasks 참조