Unit Testing Plan

for Public Transportation System

- **Test Plan**
- **Test Design Specification**
- **Test Cases Specification**

Project Team

Team4

Date

2014-11-20

Team Information
200910814 강기웅
201214146 김남형
201214147 박현승
201214151 정재명

Table of Contents

1 Introduction 4

1.1 Objectives 4

1.2 Background 4

1.3 Scope 4

1.4 Project plan 5

1.5 Configuration management plan 5

1.6 References 5

2 Test items 5

3 Features to be tested 8

4 Features not to be tested 9

5 Approach 11

6 Item pass/fail criteria 11

7 Unit test design specification 11

7.1 Test design specification identifier 11

7.2 Features to be tested 11

7.3 Approach refinements 11

7.4 Test identification 11

7.5 Feature pass/fail criteria 14

8 Unit test case specification 14

8.1 Test case specification identifier 14

8.2 Test items 18

8.3 Input specifications 18

8.4 Output specifications 18

9 Environmental needs 18

10 Unit Test deliverables 19

11 Schedules 19

1 Introduction

1.1 Objectives

이 문서는 TEAM4의 PTS System의 unit test를 수행하기 위해 작성된 계획 문서이며, 본 system이 제대로 작동하는지를 살펴보기 위해 필요한 요소들을 정리해 놓은 문서이다. 앞서 단계에서 작성했던 SRA_Team4_ver3을 바탕으로 test할 부분을 선정하였고, 각 기능이 제대로 수행되는지 확인하는 것에 중점을 두어 작성하였다. Test를 수행하기 위해 필요한 활동 및 자원을 정의하고, test approach 및 techniques를 정의한다. 또한 test를 위한 환경적인 요구사항 및 test **도구들을** 정의한다.

1.2 Background

PTS system은 대중교통 시스템으로, 사용자의 태그 행위를 통하여 인식되는 카드의 정보를 통해 요금 계산과 정산을 제어하는 시스템이다.

Unit test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 test이며, 시스템의 성능을 좌우하는 요소들이 요구사항을 만족하는지를 확인할 수 있는 기본적인 test approach이다.

테스트 방법은 기능 위주의 블랙박스 테스트 방법으로 진행 되었다.

각각의 케이스를 독립적으로 수행하여 입력이 불가능한 상태 혹은 입력이 가능한 상태의 입력을 일부러 입력 그에 따라 프로그램이 정상적으로 실행이 가능한지, 혹은 적절한 에러메시지를 반환하는지, 혹은 프로그램이 실행 중에 강제 종료되지 않는지를 확인한다.

1.3 Scope

이 계획 문서는 PTS System의 unit test를 수행하기 위한 모든 것을 포함한다. PTS의 unit test를 수행하기 위한 자원과 절차, test approach와 technique와 필요로 하는 환경 및 도구 등을 정의한다 PTS의 unit test는 시스템을 구성하는 최소 단위의 모듈들을 대상으로 하며, 구현된 모듈이 요구사항을 만족하는지를 test 한다.

1.4 Project plan

1.5 Configuration management plan

PTS System의 Program Source code 및 Unit test를 위한 Test Code는 Cygwin 환경에서 이루어지며, Program Source code의 변경 및 수정사항은 지속적으로 통합되고 test된다.

(1) Program Source code의변경
Program Source code에 변경 및 수정 발생시, 이를 통합하고 수동적으로 Unit test를수행한다.

(2) 일정주기
Program Source code는 일정주기를 가지고 팀원들과 build 및 unit test를 수행한다.

1.6 References

SDS_Team4_ver2

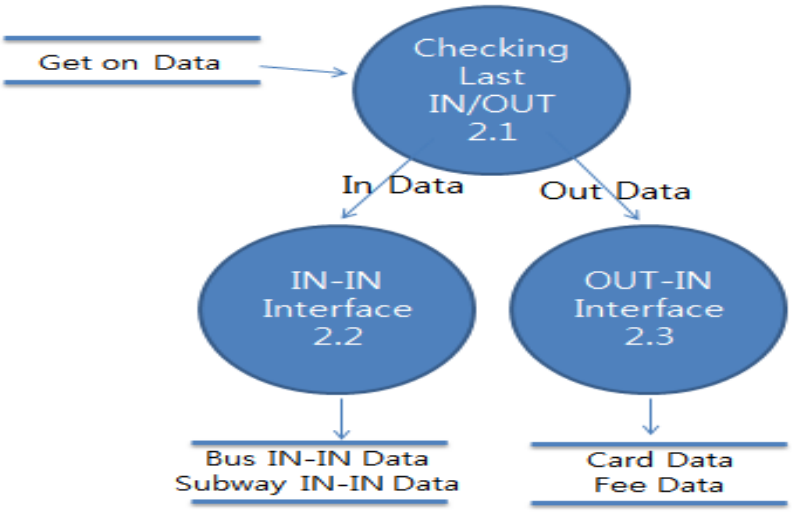
SRA_Team4_ver3

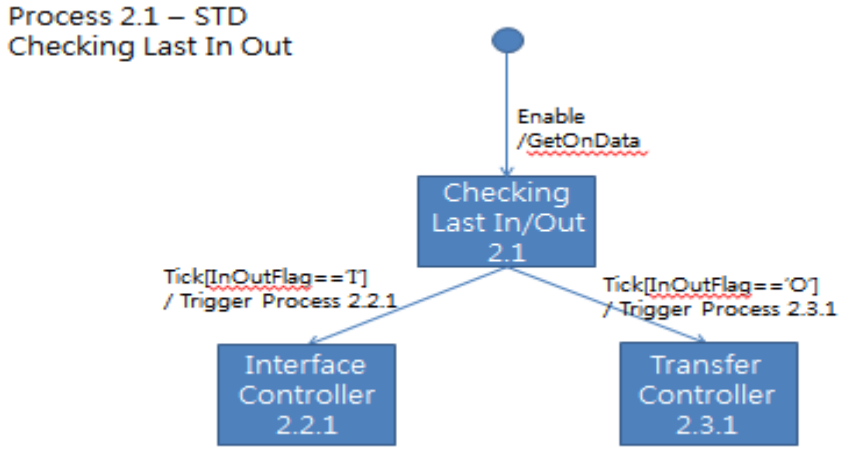
DS-2014SE-PTS-SRS-1.0

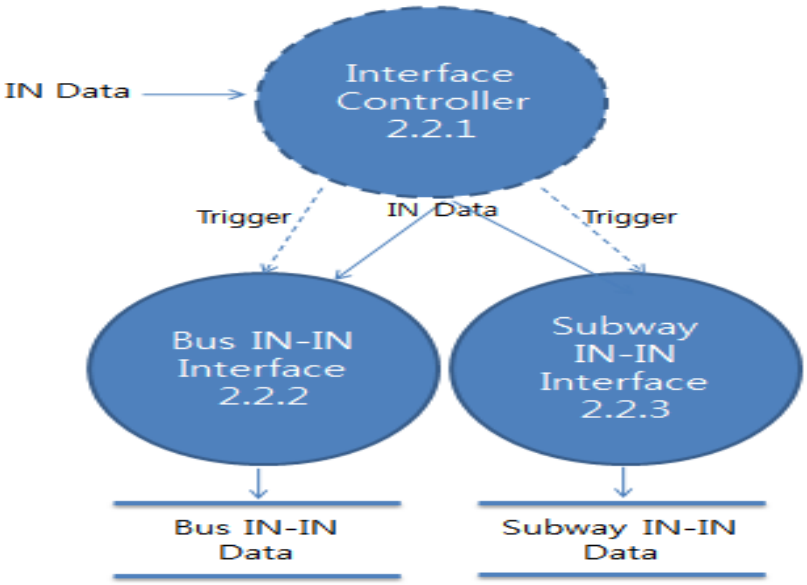
2 Test items

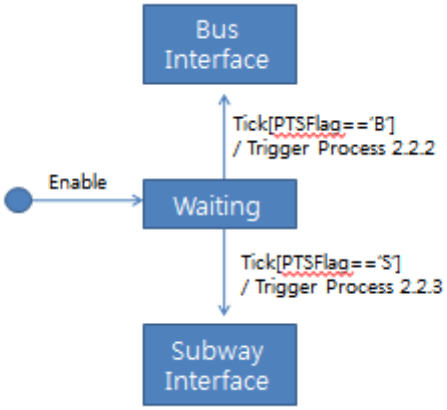
PTS를 구성하는 최소 단위의 모듈들이 unit test의 대상이 된다. 각 모듈의 요구사항을 만족하는지를 test하며, test item은 다음 자료들로부터 작성되었다.

(1) Functionality of modules –SRA_Team4_ver3참고

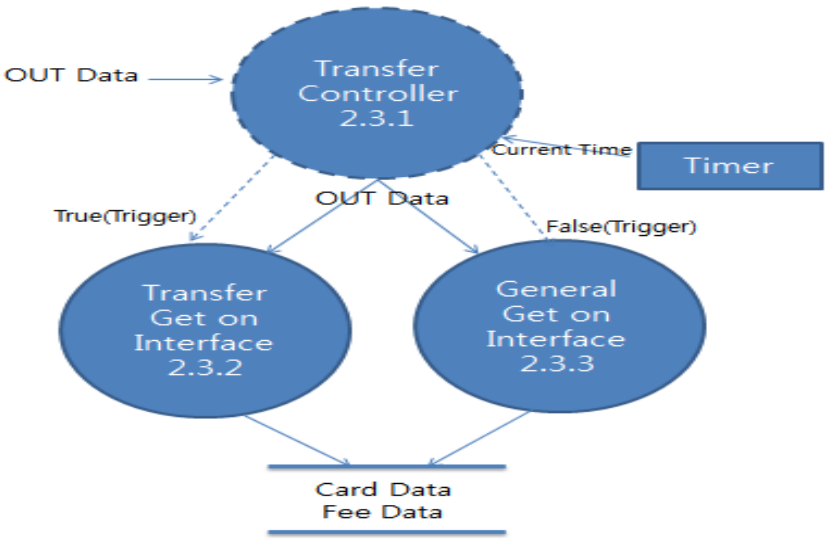

Get on Data
Checking Last IN/OUT 2.1
In Data
Out Data
IN-IN Interface 2.2
OUT-IN Interface 2.3
Bus IN-IN Data
Subway IN-IN Data
Card Data
Fee Data


Process 2.1 – STD
Checking Last In Out
Enable
/GetOnData
Checking Last In/Out 2.1
Tick[InOutFlag=='I']
/ Trigger Process 2.2.1
Tick[InOutFlag=='O']
/ Trigger Process 2.3.1
Interface Controller 2.2.1
Transfer Controller 2.3.1

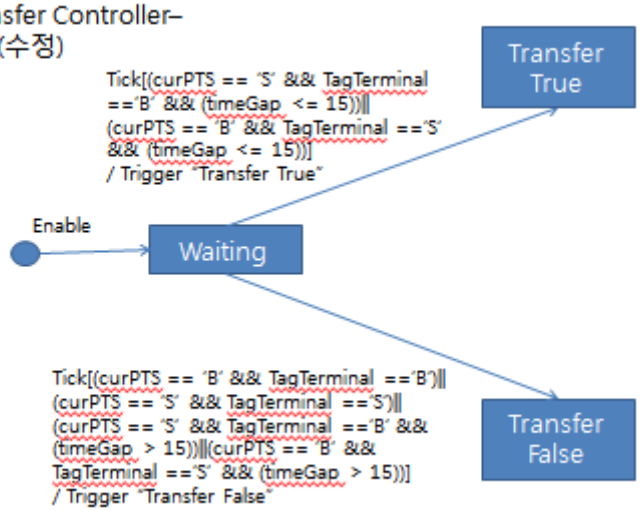

IN Data
Interface Controller 2.2.1
Trigger
IN Data
Trigger
Bus IN-IN Interface 2.2.2
Subway IN-IN Interface 2.2.3
Bus IN-IN Data
Subway IN-IN Data

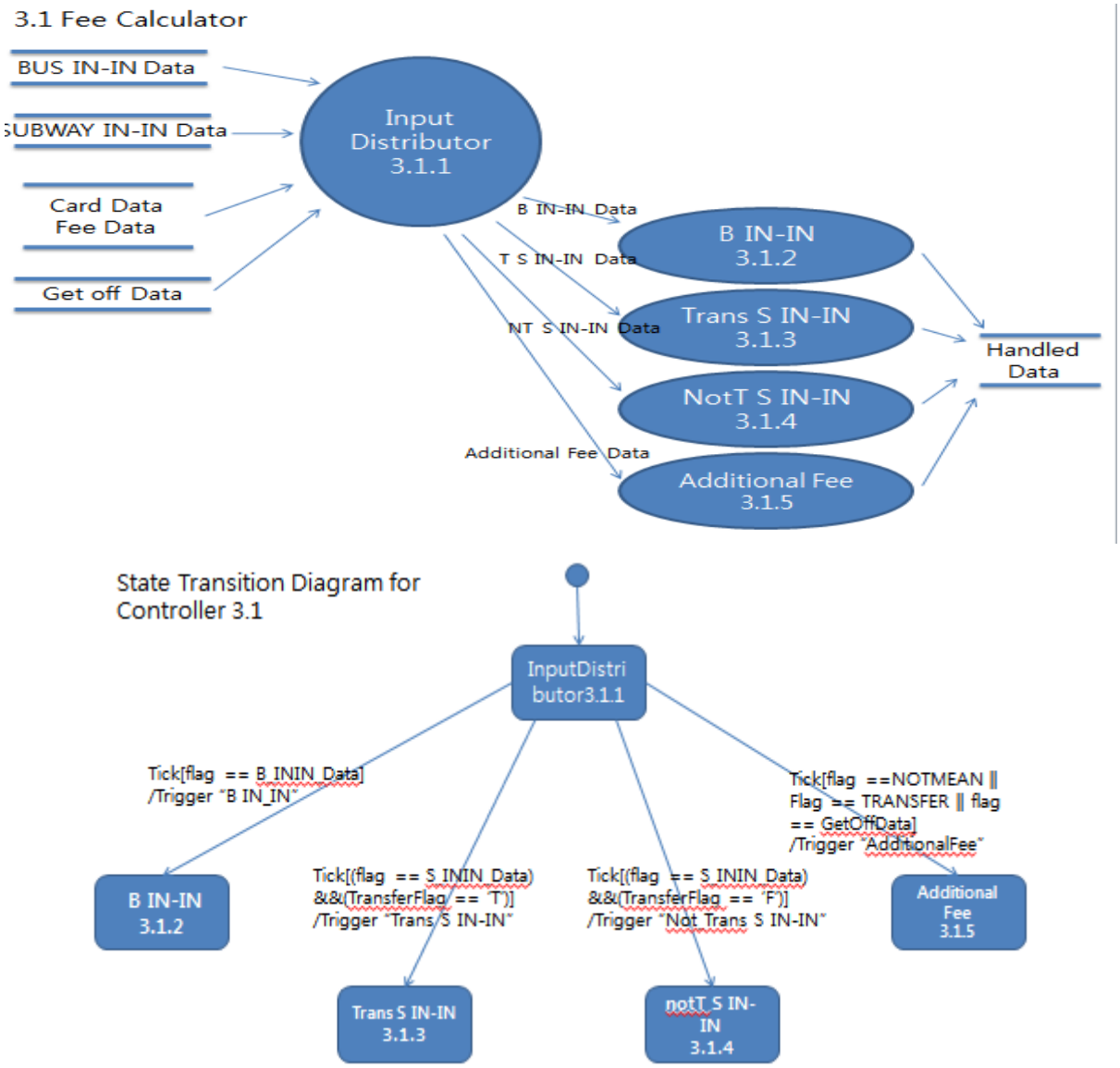


< OUT-IN Interface 2.3 >



Process 2.3.1
Transfer Controller–
STD(수정)





(2) Module interface – SRA_Team4_ver3 : Process specification, structure chart

3 Features to be tested

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

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

Table 1 테스트할 Process (DFD) 리스트

ID	Name	Description
2.1	Checking Last IN/OUT	Card Data 로부터 받은 정보를 바탕으로 최근 데이터가 IN인지 Out인지를 판단하여 데이터를 구분하여 전송한다.
2.2.1	Interface Controller	Input으로 들어온 IN Data를 읽고 가장 최근에 사용한 교통수단의 종류에 따라 Trigger 신호를 전달한다.
2.3.1	Transfer Controller	입력받은 데이터를 통해 환승 유무를 판단한다. 환승이면 True 시그널을, 아니라면 False 시그널을 전달한다.
3.1	InputDistributor	Subway IN-IN Data가 입력되면 Transfer Controller에 Request하여 환승여부를 확인 받는다. 환승 확인이 되면T S IN-IN Data가 된다. 환승이 아니면 NT S IN-IN Data가 된다. Bus IN-IN Data의 경우 환승 여부와 상관없이 미정산 요금을 부과한다. 이외의 경우는 해당 경우에 알맞은 추가요금 계산을 위해 Additional Fee Data로 가공되어 해당 프로세스에 보내진다.
3.1.2	B IN-IN	이번 승차 전에 버스를 탑승하고 하차 시 카드를 태그 하지 않은 Data이다. 미 정산 요금 700원을 부과하고 Handled Data로 가공하여 출력한다.
3.1.3.	Trans S IN-IN	이번 승차 전에 환승한 지하철을 탑승하고 하차 시 카드를 태그 하지 않은 Data이다. 미 정산 요금 600원을 부과하고 Handled Data로 가공하여 출력한다.
3.1.4	notT S IN-IN	이번 승차 전에 지하철을 탑승하고 하차 시 카드를 태그하지 않은 Data이다. 미 정산 요금 200원을 부과하고 Handled Data로 가공하여 출력한다 .
3.1.5	Additional Fee	앞선 경우의 수에 해당하지 않는 데이터의 추가요금 계산 과정을 수행하는 프로세스. 추가요금을 계산한 후 Handled Data로 가공하여 출력한다.
Adjustment System		
ID	**Name**	**Description**
2.1	Non-Ad-Calculator	DB Data를 받아, 미 정산 요금이 있는 경우, 경우의 수 대로 계산하여 Subway, Bus Fee에 합산한다.
2.2	GeneralCalculator	DB Data를 받아, 일반 탑승인 경우로 Subway, Bus Fee에 합산한다.
2.3	TransferCalculator	DB Data를 받아, 환승 한 경우의 Data를 계산법 대로 계산하여 Subway Bus Fee에 합산한다.

4 Features not to be tested

(1) Processes in SRA: 외부장치드라이버, 단순 데이터 전달 프로세스 등은 test에서제외한다.

(2) Modules in SDS: <Table2 테스트하지 않을 Process (DFD) 리스트>의 Process name 참조

Table 1 테스트하지 않을 Process (DFD) 리스트

Terminal System		
ID	**Name**	**Description**
1.2	DB Initialization	MainController에서 DB initialization요청을 받아 DB를 초기화한다.
1.3.1	OffSensor Handler	Main Controller 에서 오는 Enable/Disable 신호에 따라 동작한다. Card의 Tag를 인식하고 Card Data Input을 출력하거나 Update Card Data를 받아 Card에 해당 데이터를 기록한다.
1.3.2	Input Distributor	Tag되어 읽어온 Card Data Input을 Get off Data로 가공하여 출력하거나 After Calculated Data를 가공하여 Update Card Data로 출력한다.
1.4.1	OnSensor Handler	Process 1.3.1 – OffSensor Handler 와 동일
1.4.2	Input Distributor	Process 1.3.2 – Input Distributor 와 동일
2.2.2	Bus IN-IN Interface	Trigger 신호에 의해 동작하며 IN Data를 가공하여 Bus IN-IN Data로 출력한다.
2.2.3	Subway IN-IN Interface	Trigger 신호에 의해 동작하며 IN Data를 가공하여 Subway IN-IN Data로 출력한다.
2.3.2	Transfer Get On Interface	Trigger에 의해 동작되면 OUT Data를 토대로 해당 카드의 환승 처리 프로세싱을 수행한 후 Card Data와 Fee Data를 출력한다.
2.3.3	General Get On Interface	Trigger에 의해 동작되면 OUT Data를 토대로 해당 카드의 정상 승차 프로세싱을 수행한 후 Card Data와 Fee Data를 출력한다.
3.3	Display Interface	Impossible Data를 토대로 Display에 잔액 부족을 출력하거나, Display Data를 토대로 Display에 관련 정보를 출력해준다.
4.1	Output Handler	Input 으로 들어온 Data를 해당하는 output에 맞게 가공 및 출력해준다.
4.2	DB Writer	DB Data를 단말기 Database에 저장한다.
Adjustment System		
ID	**Name**	**Description**
1.1	DB Read	DB에서 Data를 읽어온다.

3.1	OutputHandler	Data를 DB에 Write, Display에 출력, 각 회사에 전송한다.

5 Approach

PTS System의 Program Source code 및 Unit test를 위한 Test Code는 Cygwin 환경에서 이루어지며, Program Source code의 변경 및 수정사항은 지속적으로 통합되고 test된다.

(1)Brute force testing : 각 모듈의 요구사항을 만족하는지를 확인할 수 있는 test case를 작성한다. 그 이외의 예외사항에 대해서는 test하지 않는다.

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

PTS_UTC_000_000

7.2 Features to be tested

processes in SRA

<Table 1 테스트할Process (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

Table 3 PTS Test Design Identification

Identifier	Feature(Process ID in DFD)	Valid / Invalid value
PTS_UTC_210_000	2.1.0 Checking Last In Out	Checking Last In/Out 상태에서 InOutFlag == 'I' 입력이 들어온다

PTS_UTC_210_001	2.1.0Checking Last In Out	Checking Last In/Out 상태에서 InOutFlag == 'O' 입력이 들어온다.
PTS_UTC_221_000	2.2.1Interface Controller	Waiting 상태에서 PTSFlag == 'B' 입력이 들어온다.
PTS_UTC_221_001	2.2.1Interface Controller	Waiting 상태에서 PTSFlag == 'S' 입력이 들어온다.
PTS_UTC_231_000	2.3.1 Transfer Controller	Waiting 상태에서 curPTS == 'S' 일 때, TagTerminal =='B', timeGap <= 15 입력이 들어온다
PTS_UTC_231_001	2.3.1 Transfer Controller	Waiting 상태에서 curPTS == 'S' 일 때, TagTerminal =='S', timeGap <= 15 입력이 들어온다
PTS_UTC_231_002	2.3.1 Transfer Controller	Waiting 상태에서 curPTS == 'B' 일 때, TagTerminal =='S', timeGap <= 15 입력이 들어온다
PTS_UTC_231_003	2.3.1 Transfer Controller	Waiting 상태에서 curPTS == 'B' 일 때, TagTerminal =='B', timeGap <= 15 입력이 들어온다
PTS_UTC_310_000	3.1.0 InputDistributor	InputDistributor 상태에서 flag == B_ININ_Data 입력이 들어온다.
PTS_UTC_310_001	3.1.0 InputDistributor	InputDistributor 상태에서 flag == S_ININ_Data 일 때, TransferFlag =='T' 입력이 들어온다.
PTS_UTC_310_002	3.1.0 InputDistributor	InputDistributor 상태에서 flag == S_ININ_Data 일 때, TransferFlag =='F' 입력이 들어온다.
PTS_UTC_310_003	3.1.0 InputDistributor	InputDistributor 상태에서 flag ==NOTMEAN 입력이 들어온다
PTS_UTC_310_004	3.1.0 InputDistributor	InputDistributor 상태에서 flag ==TRANSFER 입력이 들어온다
PTS_UTC_310_005	3.1.0 InputDistributor	InputDistributor 상태에서 flag == GetOffData 입력이 들어온다
PTS_UTC_310_006	3.1.2 B IN-IN	Trigger입력이 들어오면 B IN-IN을 작동한다.
PTS_UTC_310_007	3.1.2 B IN-IN	Trigger입력이 들어오면 B IN-IN을 작동한다. (006에 대한 표본값 추가테스트)
PTS_UTC_310_008	3.1.3 Trans S IN-IN	Trigger입력이 들어오면 Trans S IN-IN을 작동한다.
PTS_UTC_310_009	3.1.3 Trans S IN-IN	Trigger입력이 들어오면 Trans S IN-IN을 작동한다. (008에 대한 표본값 추가테스트)
PTS_UTC_310_010	3.1.4 notT S IN-IN	Trigger입력이 들어오면 notT S IN-IN을 작동한다.
PTS_UTC_310_011	3.1.4 notT S IN-IN	Trigger입력이 들어오면 notT S IN-IN을 작동한다. (010에 대한 표본값 추가테스트)
PTS_UTC_315_000	3.1.5 Additional Fee	Flag == TRANSFER 입력이 들어온다.
PTS_UTC_315_001	3.1.5 Additional Fee	Flag == NOTMEAN 입력이 들어온다.
PTS_UTC_315_002	3.1.5 Additional Fee	TransferFlag == 'T' 일때, flag ==GetOffData 입력이 들어온다.

PTS_UTC_315_003	3.1.5 Additional Fee	TransferFlag == 'T' 일때, flag ==GetOffData 입력이 들어온다. (002에 대한 표본값 추가테스트)
PTS_UTC_315_004	3.1.5 Additional Fee	TransferFlag == 'F' 일때, flag ==GetOffData 입력이 들어온다.
PTS_UTC_315_005	3.1.5 Additional Fee	TransferFlag == 'F' 일때, flag ==GetOffData 입력이 들어온다. (005에 대한 표본값 추가테스트)
PTS_UTC_320_000	3.2.0 Balance check	Balance check 상태에서 Sm(subtract money) =-1이 들어온다.
PTS_UTC_320_001	3.2.0 Balance check	Balance check 상태에서 Sm(subtract money) >= 0 이 들어온다.

Table 3 ADJ Test Design Identification

Identifier	Feature(Process ID in DFD)	Valid / Invalid value
ADJ_UTC_210_000	2.1.0 Non_Ad_Calculator	Non_Ad_Calculator 상태에서 tData[Index].nonAdFee == 600 입력이 들어온다
ADJ_UTC_210_001	2.1.0 Non_Ad_Calculator	Non_Ad_Calculator 상태에서 tData[Index].nonAdFee == 700 입력이 들어온다
ADJ_UTC_210_002	2.1.0 Non_Ad_Calculator	Non_Ad_Calculator 상태에서 tData[Index].nonAdFee == 200 입력이 들어온다
ADJ_UTC_210_003	2.1.0 Non_Ad_Calculator	Non_Ad_Calculator 상태에서 tData[Index].nonAdFee == 0 입력이 들어온다
ADJ_UTC_220_000	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_220_001	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID !=6666 입력이 들어온다
ADJ_UTC_220_002	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_220_003	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID !=6666 입력이 들어온다
ADJ_UTC_220_004	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_220_005	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID !=6666 입력이 들어온다

ADJ_UTC_220_006	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_220_007	2.2.0 TransferCalculator	TransferCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID !=6666 입력이 들어온다
ADJ_UTC_230_000	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_230_001	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID !=6666 입력이 들어온다
ADJ_UTC_230_002	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_230_003	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='O'이고 Data[Index].TerminalID !=6666 입력이 들어온다
ADJ_UTC_230_004	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_230_005	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'T' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID !=6666 입력이 들어온다
ADJ_UTC_230_006	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID ==6666 입력이 들어온다
ADJ_UTC_230_007	2.3.0 GeneralCalculator	GeneralCalculator 상태에서 TransferFlag == 'F' 일 때, InOutFlag =='I'이고 Data[Index].TerminalID !=6666 입력이 들어온다

7.5Feature 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

Table 4 PTS Test Case Identification

Test case Identifier	Input specification	Output specification
PTS_UTC_210_000	State ==Checking Last In Out/ InOutFlag == 'I'	State == Interface Controller /Trigger "Interface Controller"
PTS_UTC_210_001	State ==Checking Last In Out/ InOutFlag == 'O'	State == Transfer Controller /Trigger "Transfer Controller."
PTS_UTC_221_000	State == Waiting /PTSFlag == 'B'	State == Bus IN-IN Interface /Trigger "Bus IN-IN Interface"
PTS_UTC_221_001	State == Waiting /PTSFlag == 'S'	State == Subway IN-IN Interface /Trigger "Subway IN-IN Interface"
PTS_UTC_231_000	State == Waiting /curPTS == 'S'/ TagTerminal =='B' / timeGap <=15	State == Transfer Get On Interface /Trigger "Transfer True"
PTS_UTC_231_001	State == Waiting /curPTS == 'S'/ TagTerminal =='S' / timeGap <=15	State == General Get on Interface /Trigger "Transfer False"
PTS_UTC_231_002	State == Waiting /curPTS == 'B'/ TagTerminal =='S' / timeGap <=15	State == Transfer Get on Interface /Trigger "Transfer True"
PTS_UTC_231_003	State == Waiting /curPTS == 'B'/ TagTerminal =='B' / timeGap <=15	State == General Get on Interface /Trigger "Transfer False"
PTS_UTC_231_004	State == Waiting /curPTS == 'B'/ TagTerminal =='S' / timeGap > 15	State == General Get on Interface /Trigger "Transfer False"
PTS_UTC_231_005	State == Waiting /curPTS == 'S'/ TagTerminal =='B' / timeGap > 15	State == General Get on Interface /Trigger "Transfer False"
PTS_UTC_310_000	State == InputDistributor / Flag == B_ININ_Data	Trigger 3.1.2 B IN-IN
PTS_UTC_310_001	State == InputDistributor /Flag == S_ININ_Data/TransferFlag == 'T'	Trigger 3.1.3 Trans_S IN-IN
PTS_UTC_310_002	State == InputDistributor /Flag == S_ININ_Data/TransferFlag == 'F'	Trigger 3.1.4 notT S IN-IN
PTS_UTC_310_003	State == InputDistributor / Flag == NOTMEAN	Trigger 3.1.5 Additional Fee
PTS_UTC_310_004	State == InputDistributor / Flag == TRANSFER	Trigger 3.1.5 Additional Fee
PTS_UTC_310_005	State == InputDistributor / Flag == GetOffData	Trigger 3.1.5 Additional Fee
PTS_UTC_312_006	Trigger in /State == B IN-IN/ Balance >= 1750	Trigger "Balance OK" CardData.balance = CardData.balance – 1750
PTS_UTC_313_007	Trigger in /State == B IN-IN/ Balance < 1750	Trigger "Balance Not OK" -1 return

PTS_UTC_313_008	Trigger in /State == Trans_S IN-IN Balance >= 1650	Trigger "Balance OK" CardData.balance = CardData.balance – 1650
PTS_UTC_313_009	Trigger in /State == Trans_S IN-IN Balance < 1650	Trigger "Balance Not OK" -1 return
PTS_UTC_313_010	Trigger in /State == NOT_Trans S IN-IN Balance >=1250	Trigger "Balance OK" CardData.balance = CardData.balance – 1250
PTS_UTC_313_011	Trigger in /State == NOT_Trans S IN-IN Balance < 1250	Trigger "Balance Not OK" -1 return
PTS_UTC_315_000	State == Additional Fee/ Flag == TRANSFER	Trigger "TRANSFER" FeeData.basicFee = 0
PTS_UTC_315_001	State == Additional Fee/ Flag == NOTMEAN	Trigger "NOTMEAN" FeeData.basicFee = 1050
PTS_UTC_315_002	State == Additional Fee/ Flag == GetOffData/TransferFlag == 'T' /whatTerm == 6666 / timeGap < 30	Trigger "TRANSFERBUSGETOFF" FeeData.addFee = 0;
PTS_UTC_315_003	State == Additional Fee/ Flag == GetOffData/ TransferFlag == 'T' / whatTerm == 1111 / CardData.TerminalID == 1111	Trigger "TRANSFERSUBGETOFF" FeeData.addFee = 600;
PTS_UTC_315_004	State == Additional Fee/Flag == GetOffData/ TransferFlag == 'F'/ whatTerm == 6666	Trigger "GENBUSGETOFF" FeeData.addFee = 0;
PTS_UTC_315_005	State == Additional Fee/ Flag == GetOffData/TransferFlag == 'F' whatTerm == 1111 /CardData.TerminalID == 1111	Trigger "GENSUBGETOFF" FeeData.addFee = 200;
PTS_UTC_320_000	State == Balance check/ CardData.Balace == 1500 totalFee == 1050	Trigger "BALANCE OK" Return 0
PTS_UTC_320_001	State == Balance check/ CardData.Balace == 1500 totalFee == 1650	Trigger "BALANCE NOT OK" Return -1

Table 4 ADJ Test Case Identification

Identifier	Iutput specification	Output specification
ADJ_UTC_210_000	State == Non_Ad_Calculator/ tData[Index].nonAdFee == 600	State == Non_Ad_Calculator / Tmp.total_bus_fee = -408 Tmp.total_subway_fee = 1008

ADJ_UTC_210_001	State == Non_Ad_Calculator/ tData[Index].nonAdFee ==700	State == Non_Ad_Calculator / Tmp.total_bus_fee = 1429 Tmp.total_subway_fee = -729
ADJ_UTC_210_002	State == Non_Ad_Calculator/ tData[Index].nonAdFee == 200	State == Non_Ad_Calculator / Tmp.total_subway_fee = 1008 Tmp.total_bus_fee = -408
ADJ_UTC_210_003	State == Non_Ad_Calculator/ tData[Index].nonAdFee == 0	State == Non_Ad_Calculator / Trigger 2.2.0 TransferCalculator
ADJ_UTC_220_000	State == TransferCalculator / TransferFlag == 'T'/ InOutFlag == 'O'/ Data[Index].TerminalID ==6666/ basic fee =1050 /Add = 550	State == TransferCalculator / FeeData.total_bus_fee + FeeData.total_subway_fee = 550
ADJ_UTC_220_001	State == TransferCalculator / TransferFlag == 'T'/ InOutFlag == 'O'/ Data[Index].TerminalID !=6666/ basic fee =1050 /Add = 550	State == TransferCalculator / FeeData.total_bus_fee + FeeData.total_subway_fee = 550
ADJ_UTC_220_002	State == TransferCalculator / TransferFlag == 'T'/ InOutFlag == 'I'/ Data[Index].TerminalID ==6666	State == TransferCalculator / Trigger 2.3.0 GeneralCalculator
ADJ_UTC_220_003	State == TransferCalculator / TransferFlag == 'T'/ InOutFlag == 'I'/ Data[Index].TerminalID !=6666	State == TransferCalculator / Trigger 2.3.0 GeneralCalculator
ADJ_UTC_220_004	State == TransferCalculator / TransferFlag == 'F'/ InOutFlag == 'O'/ Data[Index].TerminalID ==6666	State == TransferCalculator / Trigger 2.3.0 GeneralCalculator
ADJ_UTC_220_005	State == TransferCalculator / TransferFlag == 'F'/ InOutFlag == 'O'/ Data[Index].TerminalID !=6666	State == TransferCalculator / Trigger 2.3.0 GeneralCalculator
ADJ_UTC_220_006	State == TransferCalculator / TransferFlag == 'F'/ InOutFlag == 'I'/ Data[Index].TerminalID ==6666	State == TransferCalculator / Trigger 2.3.0 GeneralCalculator
ADJ_UTC_220_007	State == TransferCalculator / TransferFlag == 'F'/ InOutFlag == 'I'/ Data[Index].TerminalID !=6666	State == TransferCalculator / Trigger 2.3.0 GeneralCalculator
ADJ_UTC_230_000	State == GeneralCalculator / TransferFlag == 'F'/ InOutFlag == 'O'/ Data[Index].TerminalID ==6666/ basicfee = 1050	State == GeneralCalculator / FeeData.total_bus_fee = 1050 FeeData.total_subway_fee = 0

ADJ_UTC_230_001	State == GeneralCalculator / TransferFlag == 'F'/ InOutFlag == 'O'/ Data[Index].TerminalID !=6666 / basicfee = 1050	State == GeneralCalculator / FeeData.total_bus_fee = 0 FeeData.total_subway_fee = 1050
ADJ_UTC_230_002	State == GeneralCalculator / TransferFlag == 'T'/ InOutFlag == 'O'/ Data[Index].TerminalID ==6666	State == GeneralCalculator / Trigger 2.3.0 OutputHandler
ADJ_UTC_230_003	State == GeneralCalculator / TransferFlag == 'T'/ InOutFlag == 'O'/ Data[Index].TerminalID !=6666	State == GeneralCalculator Trigger 2.3.0 OutputHandler /
ADJ_UTC_230_004	State == GeneralCalculator / TransferFlag == 'T'/ InOutFlag == 'I'/ Data[Index].TerminalID ==6666	State == GeneralCalculator / Trigger 2.3.0 OutputHandler
ADJ_UTC_230_005	State == GeneralCalculator / TransferFlag == 'T'/ InOutFlag == 'I'/ Data[Index].TerminalID !=6666	State == GeneralCalculator Trigger 2.3.0 OutputHandler /
ADJ_UTC_230_006	State == GeneralCalculator / TransferFlag == 'F'/ InOutFlag == 'I'/ Data[Index].TerminalID ==6666	State == GeneralCalculator / Trigger 2.3.0 OutputHandler
ADJ_UTC_230_007	State == GeneralCalculator / TransferFlag == 'F'/ InOutFlag == 'I'/ Data[Index].TerminalID !=6666	State == GeneralCalculator Trigger 2.3.0 OutputHandler /

8.2 Test items

<Table 3 Test Design Identification> 참조

8.3 Input specifications

<Table 4 Test Design Identification> 참조

8.4 Output specifications

<Table 4 Test Design Identification> 참조

9 Environmental needs

Public Transportation System의 unit test를 위한 환경적 요구사항은 다음과 같다.

IDE : VisualStudio

Compiler : GCC 환경

10 Unit Test deliverables

10.1 Unit test plan

10.2 Unit test design specification

10.3 Unit test case specification

10.4 Unit test report

11 Schedules

Task	Previous Task
(1) **Unit Test Plan 작성**	SRA_Team4 작성 SDS_Team4 작성 PTS구현
(2) **Test design specification**	Task 1
(3) **Test case specification**	Task 2
(4) **test result report**	Task 3

s