UTP, Implementation & UTR for Public Transport System

박재원 201011332
김철진 201211032
장계인 201312412

TEAM 7

Contents

Modification

Modification

Added Variable

- existInfo(Boolean): Total Payment History Data에 승차시 기록이 있으면 T, 없으면 F
- SameID(Boolean): 단말기 고유ID의 앞 숫자가 같은 경우
  Ex) 버스->5100, 지하철->7001, 7002, …
- Trans(Boolean): 이전 하차시 태그 시간과 현재 태그 시간의 차가 15이내이고 SameID가 F면 T, 그렇지 않으면 F
- exID(Boolean): 카드에 기록된 직전 단말기 정보가 버스면 T, 지하철 이면 F

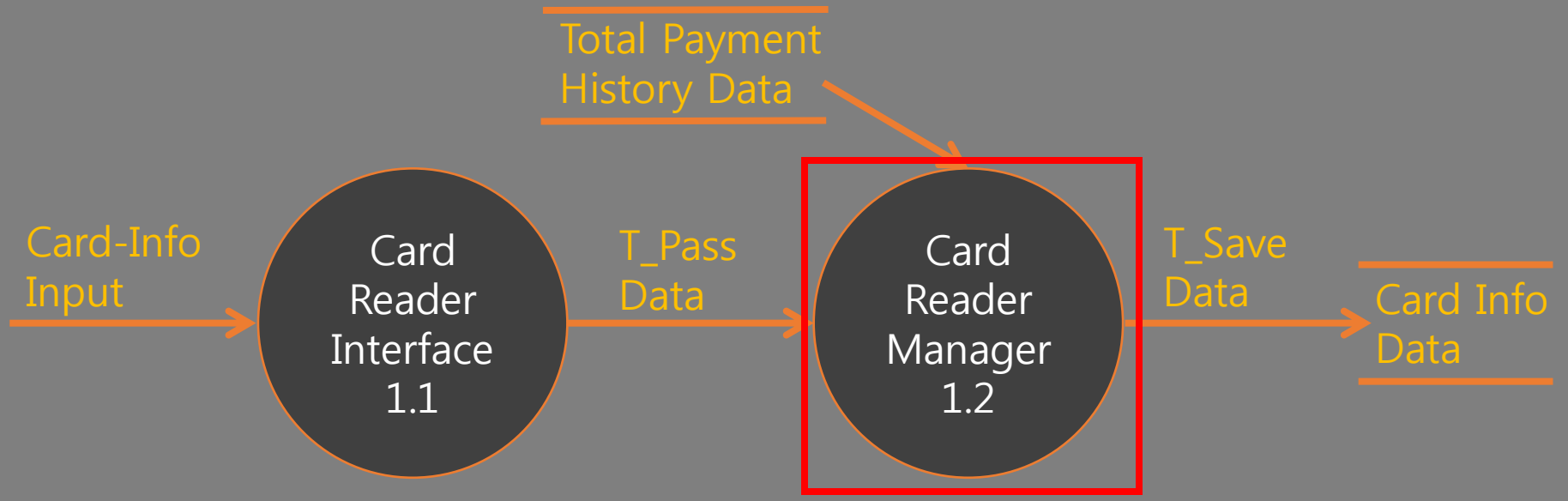
Modification

Modified Data

Data Name	Description	Format / Type
T_Pass Data	Card Reader Interface에서 받아들인 데이터를 Txt파일 형태로 정리한 중간단계의 데이터	Data Structure(int, Boolean)
T_Save Data	Card Info Input을 5가지 Type으로 분류한 것과 5가지를 기준으로 값이 정해진 existInfo, Trans, SameID, exID를 포함하는 데이터	LastTime : int, Mot : Boolean InOut : Boolean, Balance : double TerminalInfo : int, existInfo : Boolean sameID : Boolean, Trans : Boolean exID : Boolean
T_Load Data	Card Info Data에 저장된 데이터	LastTime : int, Mot : Boolean InOut : Boolean, Balance : double TerminalInfo : int, existInfo : Boolean sameID : Boolean, Trans : Boolean exID : Boolean
Card Info Data	Card Info Input을 5가지 Type으로 분류한 것과 5가지를 기준으로 값이 정해진 existInfo, Trans, SameID, exID를 포함하는 데이터를 일시적으로 저장하는 장소	Data Store

Modification

Modified Process



Reference No.	**1.2**
Name	Card Reader Manager
Input	T_Pass Data, Total Payment history data, Tick
Output	Card-Info data(LastTime, Mot, InOut, Balance, TerminalInfo, existInfo, Trans, SameID, exID)
Process Description	Card-Info txt 파일 내의 시간, 교통수단, 승차/하차, 잔액, 탑승 단말기 정보를 각각 LastTime, Mot(T=Bus, F=Metro), InOut(In=승차, Out=하차), Balance, exID에 저장한다. existInfo는 승차시 Card-Info와 동일한 정보가 Total Payment history에 있으면 T 그렇지 않으면 F가 된다. SameID는 exID와 현재 태그하는 단말기의 ID의 앞 숫자가 같으면 T 그렇지 않으면 F가 된다. Trans는 현재시간과 LastTime의 차이가 15초 이내이고 SameID가 T이면 T이고 그렇지 않으면 F가 된다. exID는 직전 단말기 정보가 버스면 T, 지하철이면 F이다.

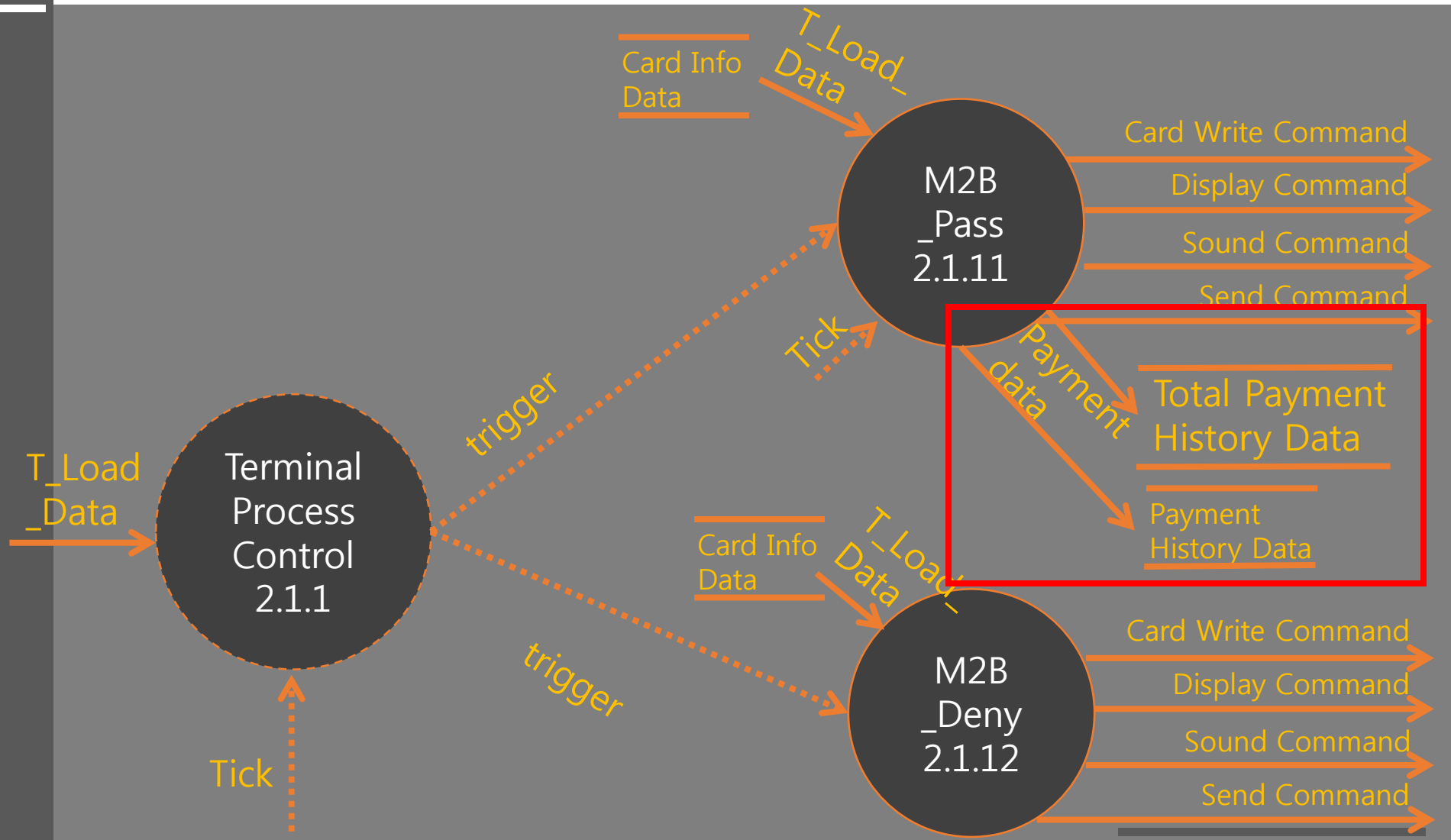
Modification

Added Data

Data Name	Description	Format / Type
Payment data	개별 단말기의 Payment data(Balance : 차감된 금액, Inout : 승/하차, Mot : 탑승 수단, LastTime : 태그 시간, TerminalInfo : 단말기 정보)가 누적 저장 되는 저장소이다.	LastTime : int Mot : Boolean InOut : Boolean Balance : double TerminalInfo : int
Total Payment History Data	모든 단말기의 Payment data가 저장되는 데이터 저장소	Data Store
Payment History Data	개별 단말기의 Payment data가 저장되는 데이터 저장소	Data Store

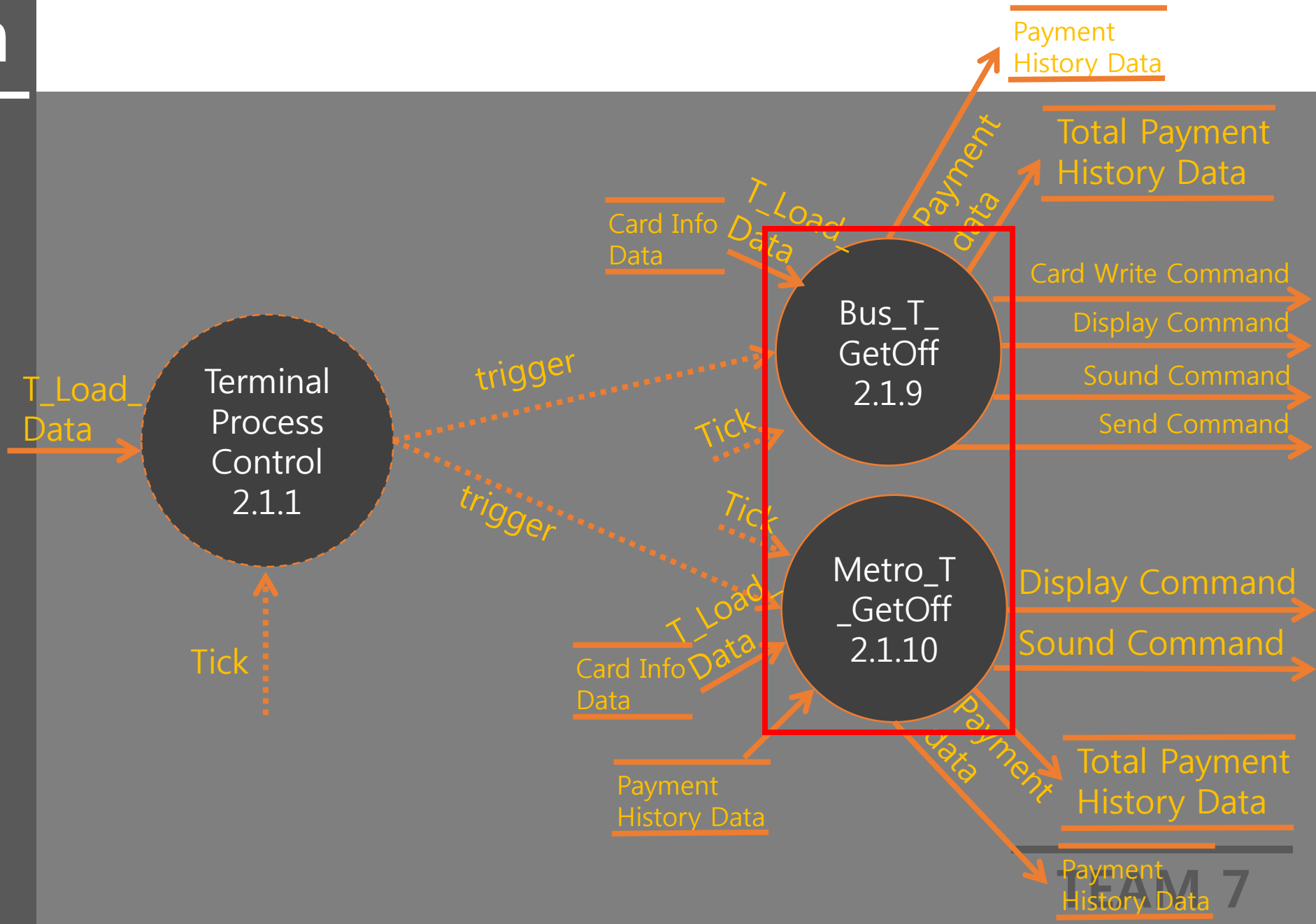
Modification

Added Data



Modification

Added Process



Modification

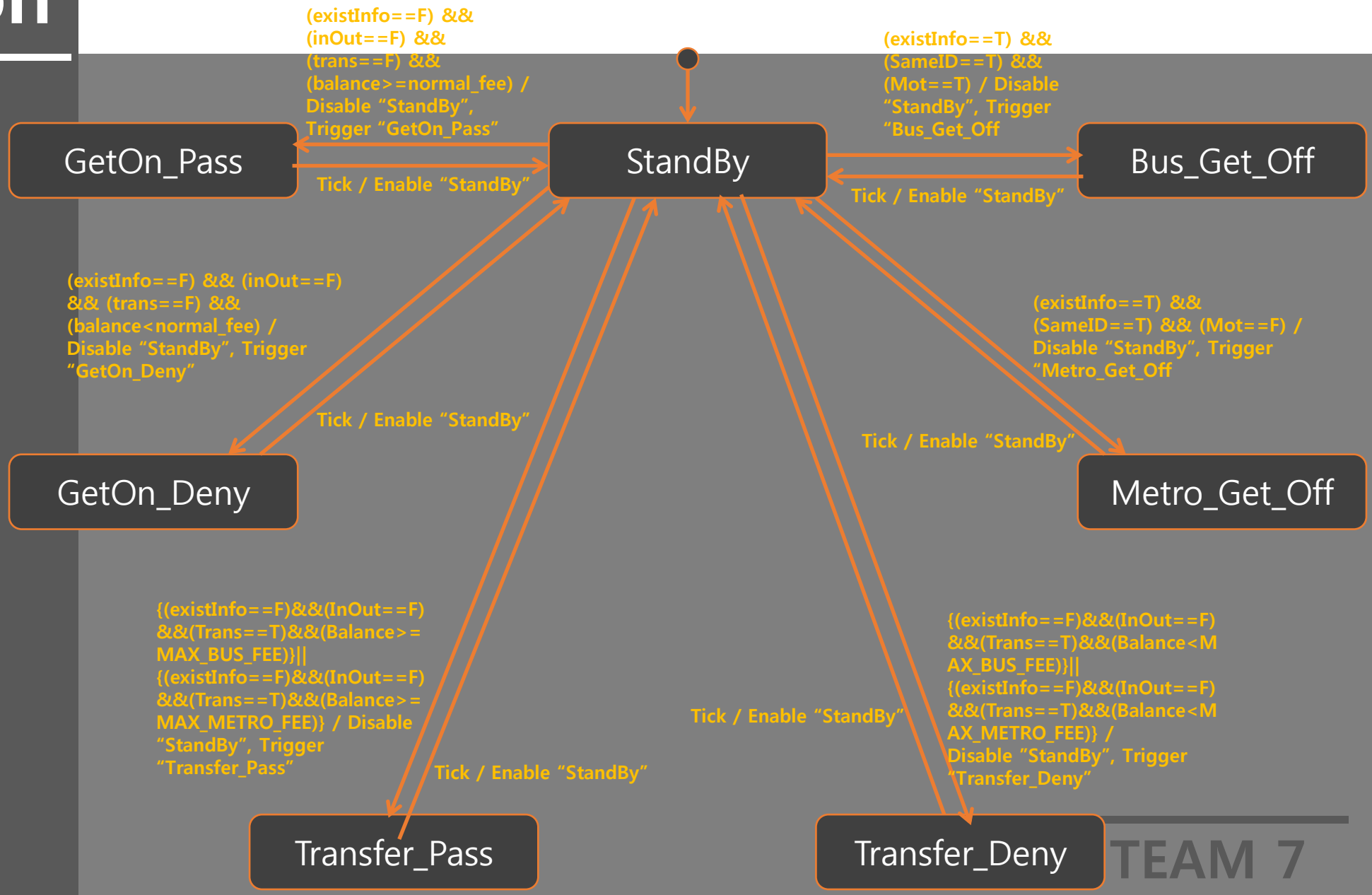
Added Process

Reference No.	2.1.9
Name	Bus_T_GetOff
Input	T_Load_Data, Trigger
Output	Card Write Command, Display Command, Sound Command, Send Command, Payment data
Process Description	(existInfo==T)&&(SameID==F)&&(Mot==T)의 경우에 실행된다. 저장소에서 카드 정보를 받아와 LastTime과 현재시간을 비교해 추가요금(30초당 100원 추가. 최고 700원을 넘지 않는다.)을 차감하고, inOut을 F로 변경한다. 변경된 payment data(balance, inout, mot, LastTime, TerminalInfo)를 payment_history data, Total payment history data에 저장하고 각각의 Interface로 커맨드를 전달한다.

Reference No.	2.1.10
Name	Metro_T_GetOff
Input	T_Load_Data, payment history data, Trigger
Output	Card Write Command, Display Command, Sound Command, Send Command, Payment data
Process Description	(existInfo==T)&&(SameID==F)&&(Mot==F)의 경우에 실행된다. 저장소에서 card info data와 payment history data를 받아와 이동정거장 수를 확인하고 추가요금(1정거장당 300원 추가. 최고 600원을 넘지 않는다.)을 차감하고, inOut을 F로 변경한다. 변경된 payment data(balance, inout, mot, LastTime, TerminalInfo)를 payment_history data, Total payment history data에 저장하고 각각의 Interface로 커맨드를 전달한다.

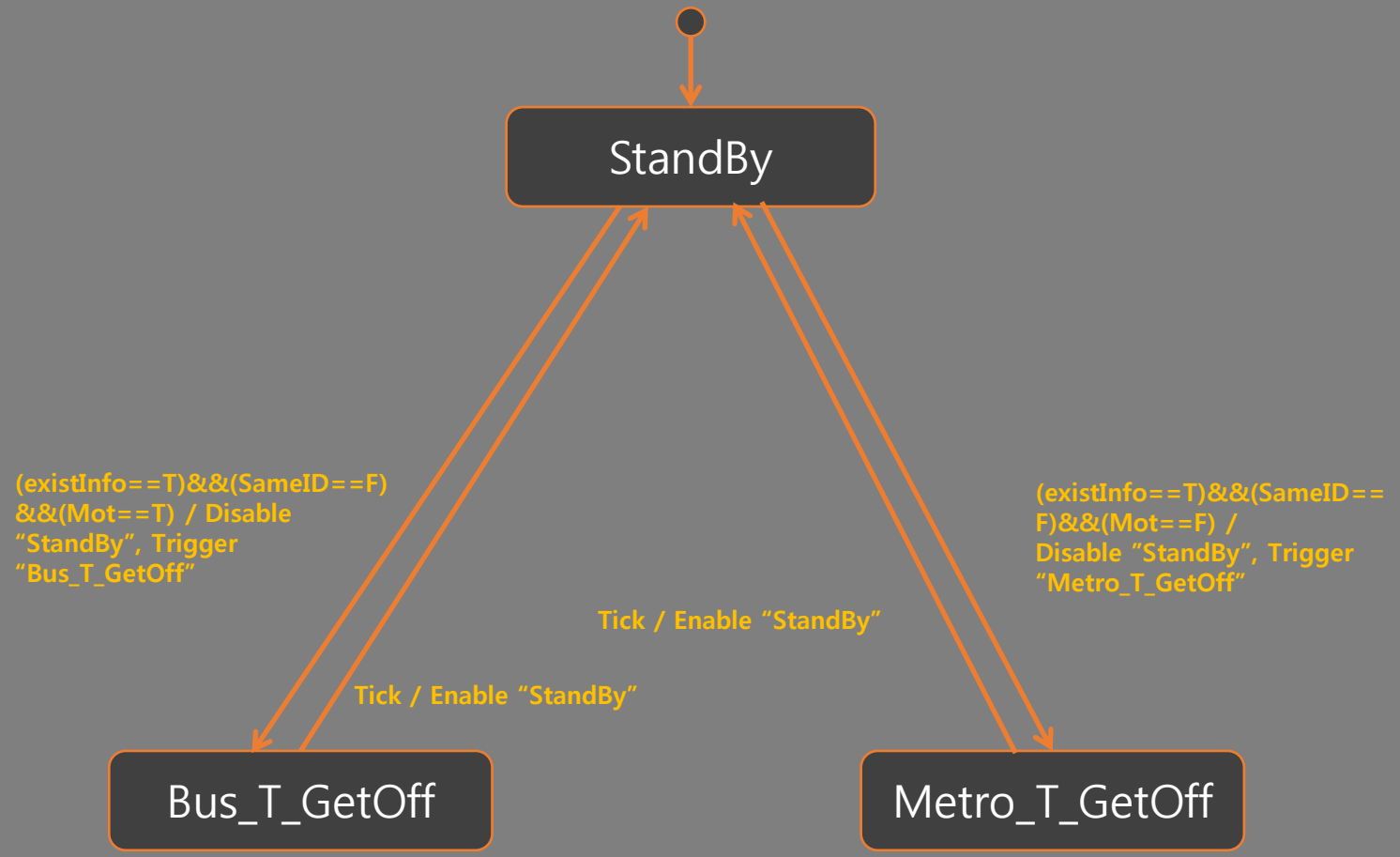
Modification

Modified STD



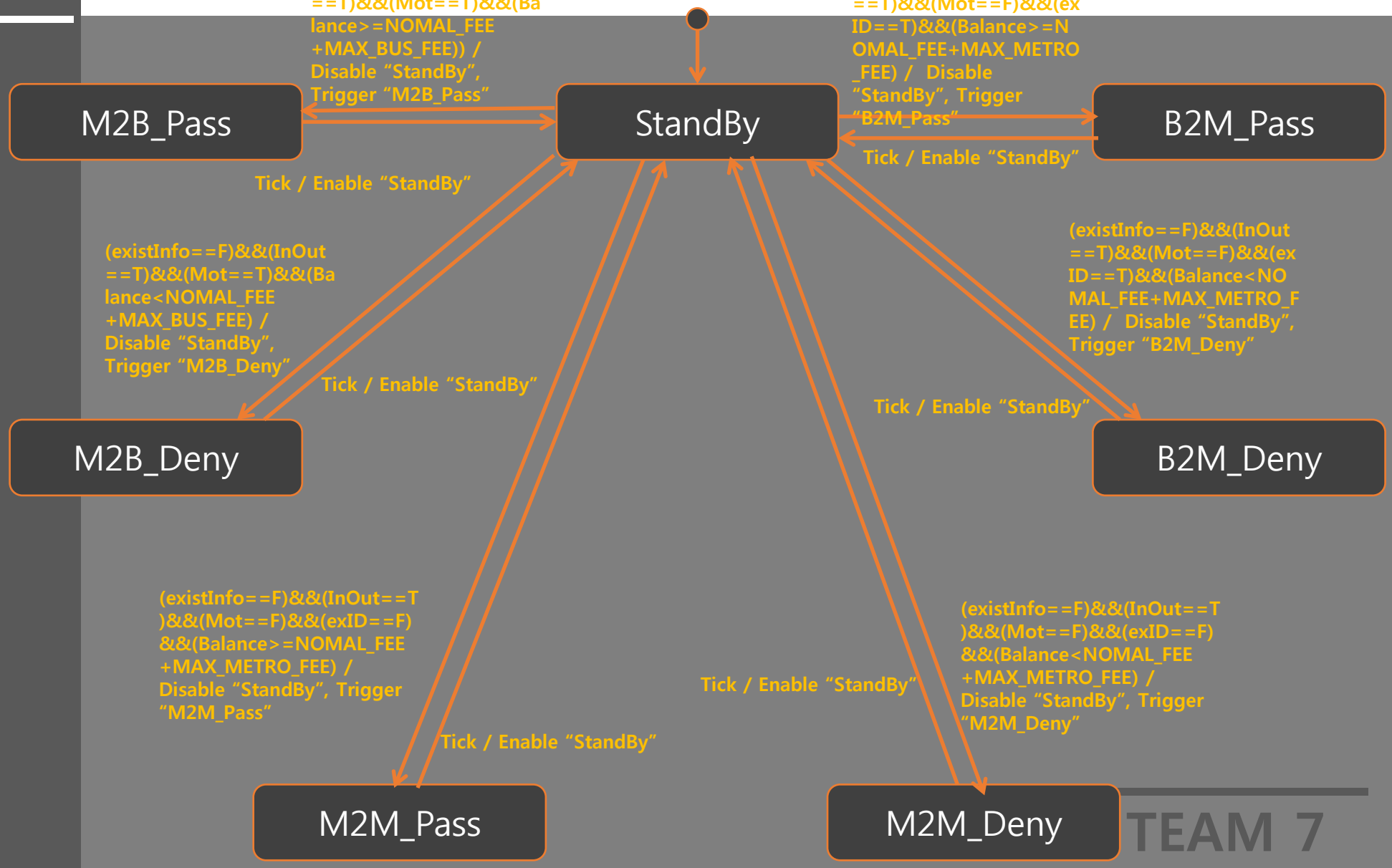
Modification

Modified STD



Modification

Modified STD



Unit Test Plan

Unit Test Plan

Test Items

- PTS를 구성하는 최소 단위 Module들이 Unit Test의 대상
- STD에 직접적으로 연관 되는 프로세스
- 외부 장치 드라이버, 단순 데이터 전달 프로세스 등은 제외
- Library Module(Interface Process) 제외

Unit Test Plan

Lists to be tested

Terminal Part (1/3)

ID	Name	Description
2.1.1	Terminal Process Control	Card Info Data의 T_Load Data와 Digital Clock의 Tick(시간Data)를 받아온 후, 카드의 상태를 비교 판단하여 적절한 하위 프로세스로 Trigger명령을 보낸다.(Trigger시 StandBy의 상태를 Disable로 변경)
2.1.3	GetOn_Pass	정상적인 탑승가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.
2.1.4	GetOn_Deny	정상적인 탑승거부의 경우 실행되는 프로세스 Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다.
2.1.5	Bus_GetOff	버스에서 하차하는 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.
2.1.6	Metro_GetOff	지하철에서 하차하는 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Total Payment History Data와 Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command를 보낸다.

Unit Test Plan

Lists to be tested

Terminal Part (2/3)

ID	Name	Description
2.1.7	Transfer_Pass	환승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.
2.1.8	Transfer_Deny	환승 거부의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다.
2.1.9	Bus_T_GetOff	버스로 환승 후 하차시 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.
2.1.10	Metro_T_GetOff	지하철로 환승 후 하차시 실행되는 프로세스이다. Trigger 명령을 받으면, Total Payment History Data와 Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.
2.1.11	M2B_Pass	미정산(지하철에서 버스 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.

Unit Test Plan

Lists to be tested

Terminal Part (3/3)

ID	Name	Description
2.1.12	M2B_Deny	미정산(지하철에서 버스 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 거부의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다
2.1.13	M2M_Pass	미정산(지하철에서 일반 하차 시 단말기를 태그 하지 않았을 때) 탑승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.
2.1.14	M2M_Deny	미정산(지하철에서 일반 하차 시 단말기를 태그 하지 않았을 때) 탑승 거부의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다
2.1.15	B2M_Pass	미정산(버스에서 지하철 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다.
2.1.16	B2M_Deny	미정산(버스에서 지하철 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 거부의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다.

Unit Test Plan

Lists to be tested

Calculating System Part

ID	Name	Description
2.1.1	Initiation Control	StandBy의 상태를 Enable로 유지하고 있다가, 하루(3분)마다 Tick을 받는다. 이 때 StandBy상태를 Disable로 변경시키고 Start Process에 Trigger 명령을 보낸다.
2.1.3	Start Process	Trigger명령을 받으면 하위프로세스들 에게 Execute Data를 보낸다.

Unit Test Plan

Lists not to be tested

Terminal Part

ID	Name	Description
1.1	Card Reader Interface	Card로부터 들어온 Data를 Card Reader Manager에게 보낸다.
1.2	Card Reader Manager	Card Reader Interface로부터 받아온 Data를 Type별로 나눠서 Card Info Data로 보낸다.
2.1.2	StandBy	카드가 태그 되지 않을 때의 상태로 입력이 들어올 때까지 정지 상태의 화면을 출력하며 상태를 유지한다.
2.2.1	Card Writer Interface	Card Info Data 의 T_Load Data를 받아와 Write Command의 Data와 비교 판단하여 History Output Data로 Save History Data를 보내고 Card로 Write Data를 보낸다.
2.3.1	Display Interface	Display Command를 받으면 단말기 화면에 띄워 줄 Display Data를 보낸다.
2.4.1	Sound Interface	Sound Command를 받으면 단말기로 Sound Data를 보낸다.
2.5.1	Server Send Interface	Send Command를 받으면 History Output Data로부터 Load History Data를 받아와 서버(Calculating System)으로 Send Data를 보낸다.

Unit Test Plan

Lists not to be tested

Calculating System Part (1/3)

ID	Name	Description
1.1	History Data Reader Interface	단말기로부터 받은 Data를 History Data Reader Manager로 보낸다.
1.2	History Data Reader Manager	History Data Reader Interface로부터 받은 Data를 6가지 형태의 Data로 나눈 후 각각에 알맞은 저장소로 저장한다.
2.1.2	StandBy	컨트롤러의 대기상태 프로세스이다
2.1.4	Normal Bus Process	버스의 일반 승차 경우 정산하는 프로세스이다. Normal Bus Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를 보내고, End Process에 Execute EP Data를 보낸다.
2.1.5	B2M Trans Process	버스에서 지하철로 환승 경우 정산하는 프로세스이다. B2M Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다.

Unit Test Plan

Lists not to be tested

Calculating System Part (2/3)

ID	Name	Description
2.1.6	M2B Trans Process	지하철에서 버스로 환승 경우 정산하는 프로세스이다. M2B Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다.
2.1.7	Normal Metro Process	지하철의 일반 승차 경우 정산하는 프로세스이다. Normal Metro Load Data를 받아와 정산 처리 후 Total Metro Data에 Total Metro Sav e Data를 보내고, End Process에 Execute EP Data를 보낸다.
2.1.8	B2M Not Process	버스에서 지하철로 환승 한 후 하차태그를 찍지 않은 미정산 경우 정산하는 프로세스이다. M2B Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다
2.1.9	M2B Not Process	지하철에서 버스로 환승 한 후 하차태그를 찍지 않은 미정산 경우 정산하는 프로세스이다. M2B Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다

Unit Test Plan

Lists not to be tested

Calculating System Part (3/3)

ID	Name	Description
2.1.10	End Process	Execute EP Data를 받으면 Complete Command, Display Command, Bus Payment Command, Metro Payment Command를 보낸다.
2.2.1	Bus Company Interface	Bus Payment Command를 받으면 Total Bus Load Data를 받아와 버스회사로 S_Send Data를 보낸다.
2.3.1	Metro Company Interface	Metro Payment Command를 받으면 Total Metro Load Data를 받아와 지하철회사로 S_Send Data를 보낸다.
2.4.1	Terminal Interface	Complete Command를 받으면 단말기에게 완료 신호(Signal Data)를 보낸다.
2.5.1	Display Interface	Display Command를 받으면 모니터로 S_Display Data를 보낸다.

Unit Test Plan

Unit Test Design Specification - Teminal Part

Tetminal Part (1/5)

Identifier	Feature	Vaild/Invaild value
PTS_UTC_211_000	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, inOut==F, trans==F, balance>=normal_fee인 경우
PTS_UTC_211_001	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, inOut==F, trans==F, balance<normal_fee인 경우
PTS_UTC_211_002	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance>=MAX_BUS_FEE인 경우
PTS_UTC_211_003	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance>=MAX_METRO_FEE인 경우
PTS_UTC_211_004	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance>=MAX_BUS_FEE인 경우
PTS_UTC_211_005	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance>=MAX_METRO_FEE인 경우
PTS_UTC_211_006	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance<MAX_BUS_FEE인 경우
PTS_UTC_211_007	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance<MAX_METRO_FEE인 경우
PTS_UTC_211_008	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance<MAX_BUS_FEE인 경우
PTS_UTC_211_009	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance<MAX_METRO_FEE인 경우

Unit Test Plan

Unit Test Design Specification

Tetminal Part (2/5)

Identifier	Feature	Vaild/Invaild value
PTS_UTC_211_0010	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, SameID==T, Mot==T인 경우
PTS_UTC_211_0011	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, SameID==T, Mot==F인 경우
PTS_UTC_211_0012	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, SameID==F, Mot==T인 경우
PTS_UTC_211_0013	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, SameID==F, Mot==F인 경우
PTS_UTC_211_0014	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE인 경우
PTS_UTC_211_0015	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE인 경우
PTS_UTC_211_0016	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE인 경우
PTS_UTC_211_0017	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE인 경우

Unit Test Plan

Unit Test Design Specification

Tetminal Part (3/5)

Identifier	Feature	Vaild/Invaild value
PTS_UTC_211_0018	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우
PTS_UTC_211_0019	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우
PTS_UTC_211_0020	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우
PTS_UTC_211_0021	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우
PTS_UTC_211_0022	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우
PTS_UTC_211_0023	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우
PTS_UTC_211_0024	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우

Unit Test Plan

Unit Test Design Specification

Tetminal Part (4/5)

Identifier	Feature	Vaild/Invaild value
PTS_UTC_211_0025	2.1.1 Terminal Process Control	StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우
PTS_UTC_213_0000	2.1.3 GetOn_Pass	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_214_0000	2.1.4 GetOn_Deny	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다.
PTS_UTC_215_0000	2.1.5 Bus_GetOff	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_216_0000	2.1.6 Metro_GetOff	Trigger 입력이 들어오면, Payment History Data와 T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_217_0000	2.1.7 Transfer_Pass	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_218_0000	2.1.8 Transfer_Deny	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다.

Unit Test Plan

Unit Test Design Specification

Tetminal Part (5/5)

Identifier	Feature	Vaild/Invaild value
PTS_UTC_219_0000	2.1.9 Bus_T_GetOff	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_2110_0000	2.1.10 Metro_T_GetOff	Trigger 입력이 들어오면, Payment History Data와 T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_2111_0000	2.1.11 M2B_Pass	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_2112_0000	2.1.12 M2B_Deny	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다.
PTS_UTC_2113_0000	2.1.13 M2M_Pass	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_2114_0000	2.1.14 M2M_Deny	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다.
PTS_UTC_2115_0000	2.1.15 B2M_Pass	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다.
PTS_UTC_2116_0000	2.1.16 B2M_Deny	Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다.

Unit Test Plan

Unit Test Design Specification

Calculating Part

Identifier	Feature	Vaild/Invaild value
PTS_UTC_211_000	2.1.1 Initiation Control	StandBy상태에서 Tick을 받는다.
PTS_UTC_211_001	2.1.1 Initiation Control	Start Process상태에서 Tick을 받는다.
PTS_UTC_213_000	2.1.3 Start Process	Trigger입력이 들어오면, 6개의 각 프로세스로 Execute Data를 전송한다.

Unit Test Plan

Unit Test Case Specification

Tetminal Part (1/6)

Test Case Identifier	Input Specification	Output Specification
PTS_UTC_211_000	StandBy==Enable / existInfo==F, inOut==F, trans==F, balance>=normal_fee	StandBy==Disable / Trigger -> GetOn_Pass
PTS_UTC_211_001	StandBy==Enable / existInfo==F, inOut==F, trans==F, balance<normal_fee	StandBy==Disable / Trigger -> GetOn_Deny
PTS_UTC_211_002	StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance>=MAX_BUS_FEE	StandBy==Disable / Trigger -> Transfer_Pass
PTS_UTC_211_003	StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance>=MAX_METRO_FEE	StandBy==Disable / Trigger -> Transfer_Pass
PTS_UTC_211_004	StandBy==Enable / existInfo==F, InOut==T, Trans==T, Balance>=MAX_BUS_FEE	StandBy==Enable
PTS_UTC_211_005	StandBy==Enable / existInfo==F, InOut==T, Trans==T, Balance>=MAX_METRO_FEE	StandBy==Enable
PTS_UTC_211_006	StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance<MAX_BUS_FEE	StandBy==Disable / Trigger -> Transfer_Deny
PTS_UTC_211_007	StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance<MAX_METRO_FEE	StandBy==Disable / Trigger -> Transfer_Deny
PTS_UTC_211_008	StandBy==Enable / existInfo==F, InOut==T, Trans==T, Balance<MAX_BUS_FEE	StandBy==Enable

Unit Test Plan

Unit Test Case Specification

Tetminal Part (2/6)

Test Case Identifier	Input Specification	Output Specification
PTS_UTC_211_009	StandBy==Enable / existInfo==F, InOut==T, Trans==T, Balance<MAX_METRO_FEE	StandBy==Enable
PTS_UTC_211_0010	StandBy==Enable / existInfo==T, SameID==T, Mot==T	StandBy==Disable / Trigger -> Bus_GetOff
PTS_UTC_211_0011	StandBy==Enable / existInfo==T, SameID==T, Mot==F	StandBy==Disable / Trigger -> Metro_GetOff
PTS_UTC_211_0012	StandBy==Enable / existInfo==T, SameID==F, Mot==T	StandBy==Disable / Trigger -> Bus_T_GetOff
PTS_UTC_211_0013	StandBy==Enable / existInfo==T, SameID==F, Mot==F	StandBy==Disable / Trigger -> Metro_T_GetOff
PTS_UTC_211_0014	StandBy==Enable / existInfo==F, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE	StandBy==Disable / Trigger -> M2B_Pass
PTS_UTC_211_0015	StandBy==Enable / existInfo==T, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE	StandBy==Enable
PTS_UTC_211_0016	StandBy==Enable / existInfo==F, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE	StandBy==Disable / Trigger -> M2B_Deny
PTS_UTC_211_0017	StandBy==Enable / existInfo==T, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE	StandBy==Enable

Unit Test Plan

Unit Test Case Specification

Tetminal Part (3/6)

Test Case Identifier	Input Specification	Output Specification
PTS_UTC_211_0018	StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE	StandBy==Disable / Trigger -> M2M_Pass
PTS_UTC_211_0019	StandBy==Enable / existInfo==T, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE	StandBy==Enable
PTS_UTC_211_0020	StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE	StandBy==Disable / Trigger -> M2M_Deny
PTS_UTC_211_0021	StandBy==Enable / existInfo==T, InOut==T, Mot==F, exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE	StandBy==Enable
PTS_UTC_211_0022	StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE	StandBy==Disable / Trigger -> B2M_Pass
PTS_UTC_211_0023	StandBy==Enable / existInfo==T, InOut==T, Mot==F, exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE	StandBy==Enable
PTS_UTC_211_0024	StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE	StandBy==Disable / Trigger -> B2M_Deny
PTS_UTC_211_0025	StandBy==Enable / existInfo==T, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE	StandBy==Enable

Unit Test Plan

Unit Test Case Specification

Tetminal Part (4/6)

Test Case Identifier	Input Specification	Output Specification
PTS_UTC_213_0000	Trigger in / T_Load_Data	Balance-=NORMAL_FEE, Inout==True, Mot==True(or False)(Current값) / Card Write Command, Display Command, Sound Command, Send Command, Payment data
PTS_UTC_214_0000	Trigger in / T_Load_Data	Display Command, Sound Command
PTS_UTC_215_0000	Trigger in / T_Load_Data.	Balance, InOut==False / Card Write Command, Display Command, Sound Command, Send Command, Payment data
PTS_UTC_216_0000	Trigger in / T_Load_Data / Payment History Data	Balance, InOut==False / Card Write Command, Display Command, Sound Command, Send Command, Payment data
PTS_UTC_217_0000	Trigger in / T_Load_Data	Balance-=MAX_BUS_FEE or MAX_METRO_FEE, Inout==True, Mot==True(or False)(Current값) / Card Write Command, Display Command, Sound Command, Send Command, Payment data

Unit Test Plan

Unit Test Case Specification

Tetminal Part (5/6)

Test Case Identifier	Input Specification	Output Specification
PTS_UTC_218_0000	Trigger in / T_Load_Data	Display Command,Sound Command
PTS_UTC_219_0000	Trigger in / T_Load_Data	Balance, InOut==False / Card Write Command, Display Command, Sound Command, Send Command, Payment data
PTS_UTC_2110_0000	Trigger in / T_Load_Data / Payment History Data	Balance, InOut==False / Card Write Command, Display Command, Sound Command, Send Command, Payment data
PTS_UTC_2111_0000	Trigger in / T_Load_Data	Balance-=(NOMAL_FEE+MAX_BUS_FEE), Inout==True, Mot==True(or False)(Current값) / Card Write Command, Display Command, Sound Command, Send Command, Payment data
PTS_UTC_2112_0000	Trigger in / T_Load_Data	Display Command, Sound Command,
PTS_UTC_2113_0000	Trigger in / T_Load_Data	Balance-=(NOMAL_FEE+ADDITIONAL_METRO_FEE), InOut==True, Mot==True(or False)(Current값) / Card Write Command, Display Command, Sound Command, Send Command, Payment data

Unit Test Plan

Unit Test Case Specification

Tetminal Part (6/6)

Test Case Identifier	Input Specification	Output Specification
PTS_UTC_2114_0000	Trigger in / T_Load_Data	Display Command, Sound Command,
PTS_UTC_2115_0000	Trigger in / T_Load_Data	Balance-=(NOMAL_FEE+MAX_METRO_FEE), Inout==True, Mot==True(or False)(Current값) / Card Write Command, Display Command, Sound Command, Send Command, Payment data
PTS_UTC_2116_0000	Trigger in / T_Load_Data	Display Command, Sound Command,

Unit Test Plan

Unit Test Case Specification

Calculating Part

Test Case Identifier	Input Specification	Output Specification
PTS_UTC_211_000	Tick	StandBy==Disable / Trigger -> Start Process
PTS_UTC_211_001	Tick	StandBy==Enable
PTS_UTC_213_002	Trigger	Execute Data

Implementation

inputPart.c | dgdg.c | main.c | card.txt | paymentData... | PTSsys.exe.s... | sub.h | controlPart.h

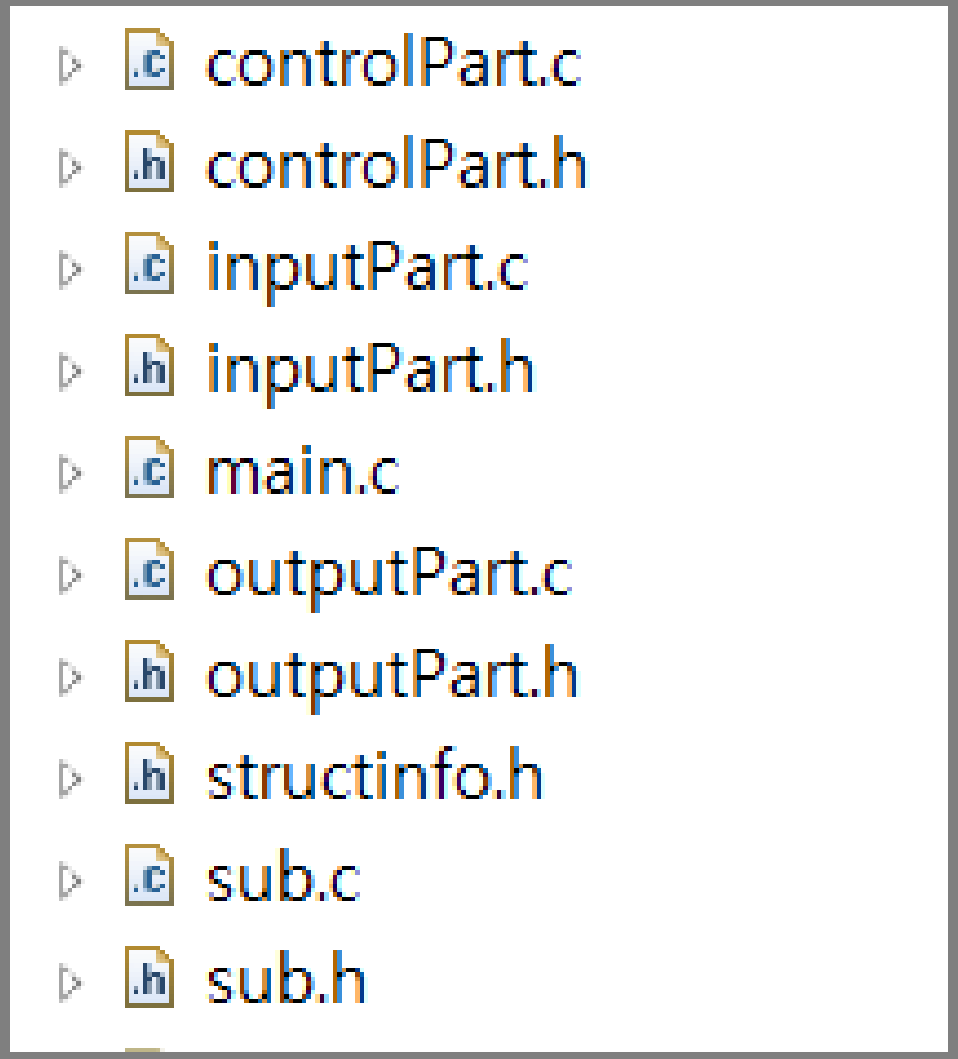
```
1 Exception: STATUS_ACCESS_VIOLATION at rip=0018016D01C
2 rax=000000010040503B rbx=000000000022AAF0 rcx=0000000000000000
3 rdx=000000010040503B rsi=0000000600028380 rdi=000000000022AB3D
4 r8 =000000000022D758 r9 =0000000000000001 r10=0000000000000000
5 r11=0000000000000000 r12=0000000000000001 r13=0000000000000000
6 r14=0000000000000000 r15=000000000022AB3D
7 rbp=000000000022AA40 rsp=000000000022A988
8 program=C:\Users\1\workspace\PTSsys\Debug\PTSsys.exe, pid 10736, thread main
9 cs=0033 ds=002B es=002B fs=0053 gs=002B ss=002B
10 Stack trace:
11 Frame        Function    Args
12 0000022AA40  0018016D01C (34363331332D3132, 00100403B00, 00000000000, 0018004002C)
13 0000022AA40  001004024B8 (00100405050, 00600062AA0, 0000022AAF0, 00600028380)
14 0000022AAA0  001004026BC (00000000020, FF0700010302FF00, 00180048400, 00180047540)
15 0000022AB90  00180048471 (00000000000, 00000000000, 00000000000, 00000000000)
16 00000000000  0018004625B (00000000000, 00000000000, 00000000000, 00000000000)
17 00000000000  001800463B4 (00000000000, 00000000000, 00000000000, 00000000000)
18 00000000000  00100403391 (00000000000, 00000000000, 00000000000, 00000000000)
19 00000000000  00100401010 (00000000000, 00000000000, 00000000000, 00000000000)
20 00000000000  000775559ED (00000000000, 00000000000, 00000000000, 00000000000)
21 00000000000  0007799C541 (00000000000, 00000000000, 00000000000, 00000000000)
22 End of stack trace
23
```

Problems | Tasks | Console | Properties | Problem Details

<terminated> PTSsys.exe [C/C++ Application] C:₩Users₩1₩workspace₩PTSsys₩Debug₩PTSsys.exe (14. 11. 21. 오전 6:29)

```
0 [main] PTSsys 10736 cygwin_exception::open_stackdumpfile: Dumping stack trace to PTSsys.exe.stackdump
```

Implementation



Implementation

```
 * sub.h

#ifndef SUB_H_
#define SUB_H_

#include "structinfo.h"

#define TRUE 1
#define FALSE 0
#define ENABLE 1
#define DISABLE -1
#define MAX_LENGTH 100
#define NORMAL_FEE 1050
#define MAX_BUS_FEE 700
#define MAX_METRO_FEE 600
#define ADDITIONAL_METRO_FEE 200
#define SUCCESS 1
#define TRANS 2
#define OFF 3
#define DENY 4

extern struct cardInfoData cids;
extern struct historyOutputData his;
extern char terminalInfoFromCurrent[10]; //맨 앞자리 숫자 버스:5, 지하철:7
extern int currentState;
extern struct paymentData paymentData_P[MAX_LENGTH];
extern int numOfpaymentData_P;
extern struct paymentData paymentData_U[MAX_LENGTH];
extern int numOfpaymentData_U;

void initInfo(struct cardInfoData * cid,char * str);
int getTime(void);
void getPaymentData_U(void);
void getPaymentData_P(void);
int search_U(char*str);
int search_P(char*str);



#endif /* PTS_HEADERFOLDER_SUB_H_ */
```

Implementation

```
 * sub.c

#include <stdio.h>
#include <time.h>
#include <string.h>
#include "sub.h"

struct cardInfoData cids;//cardInfoDataStorage 비교를 위한 초기저장값
struct historyOutputData his;//historyOutputDataStorage
char terminalInfoFromCurrent[10];//현재 단말기의 단말기정보
int currentState;//단말기의 StandBy상태
struct paymentData paymentData_P[MAX_LENGTH];//누적결제기록정보
int numOfpaymentData_P;//누적결제기록 카운팅
struct paymentData paymentData_U[MAX_LENGTH];//공유누적결제기록정보
int numOfpaymentData_U;//공유누적결제기록 카운팅

//초기화(cardinfodata 및 단말기 내부전역변수)
void initInfo(struct cardInfoData * cid, char * str)
//현재시간을 가져오는 함수
int getTime(void)
//공유누적결제기록에서 Data가져오는 함수
void getPaymentData_U(void)
//일반누적결제기록에서 Data가져오는 함수
void getPaymentData_P(void)
//공유누적결제기록에서 검색
int search_U(char*str)
//일반누적결제기록에서 검색
int search_P(char*str)

int search_P_sub(int id,char*str)
```

Implementation

```
 * structinfo.h

#ifndef STRUCTINFO_H_
#define STRUCTINFO_H_

struct cardInfoData
{
    int lastTimeInfo; //마지막 태그된 시간
    int mot; //탑승수단 정보
    int inOut; //승하차 정보
    int balance; //잔액
    char terminalInfoFromCard[10]; //카드에기록된 단말기정보

    int existInfo; //누적결제기록에서 승차시기록 검색관련변수
    int trans; //환승상태 여부
    int sameId; //단말기의 고유ID앞 숫자와 카드의 단말기정보 ID 앞숫자 비교변수
    int exId; //직전단말기정보 구분변수
};

struct paymentData
{
    int lastTimeInfo_P; //마지막 태그된 시간
    int mot_P; //탑승수단 정보
    int inOut_P; //승하차 정보
    int payment; //결제금액(기본결제금액은 승차시,추가결제금액은 하차시)
    char terminalInfo[10]; //단말기정보

};

struct historyOutputData
{
    int balance; //잔액
    int mot; //교통수단
    int trans; //환승여부
    int unCal; //미정산여부

};

#endif /* PTS_HEADERFOLDER_STRUCTINFO_H_ */
```

Implemen

```c
 * outputpart.c
#include <string.h>
#include <stdio.h>
#include "structinfo.h"
#include "sub.h"

void cardWriterInterface(struct cardInfoData cid,int opUnCal) //opUnCal:미정산인지아닌지구분하는변수
{
    char temp[MAX_LENGTH];
    char temp2[10];

    //historyOutputData 기록

     //cid는 프로세스거치며 변화한값, cids는 처음 읽어온 그대로 초기값
     his.balance = cids.balance - cid.balance; //차액 = 초기값-변화값
     his.trans = cid.trans;
     his.unCal = opUnCal; //TRUE(1):미정산경우, FALSE(0):미정산아닌경우

     if(terminalInfoFromCurrent[0]=='5')
     {
         his.mot=TRUE;
     }
     else
     {
         his.mot=FALSE;
     }

    //카드로 파일전송 부분
     strcpy(temp,cid.lastTimeInfo); //마지막태그시간기록

     //탑승수단기록
     if(cid.mot==TRUE)
     {
         strcat(temp,"BUS");
     }
     else
     {
         strcat(temp,"METRO");
     }

     //승하차기록
     if(cid.inOut==TRUE)
     {
         strcat(temp,"IN");
     }
     else
     {
         strcat(temp,"OUT");
     }

     //잔액기록

     sprintf(temp2,"%d",cid.balance);
     strcat(temp,temp2);

     //단말기정보기록
     strcat(temp,cid.terminalInfoFromCard);

     FILE * p;
     p = fopen("card2.txt","w");

     fprintf(p,temp);

     fclose(p);
}

void displayInterface(int opt, struct cardInfoData cid)

void soundInterface(int opt)

void serverSendInterface()
```

THE END