소프트웨어공학개론 [T3]
Public Transportation System

201111333 / 권태헌
201111375 / 윤지수
201111379 / 이한빈
201111384 / 정국빈

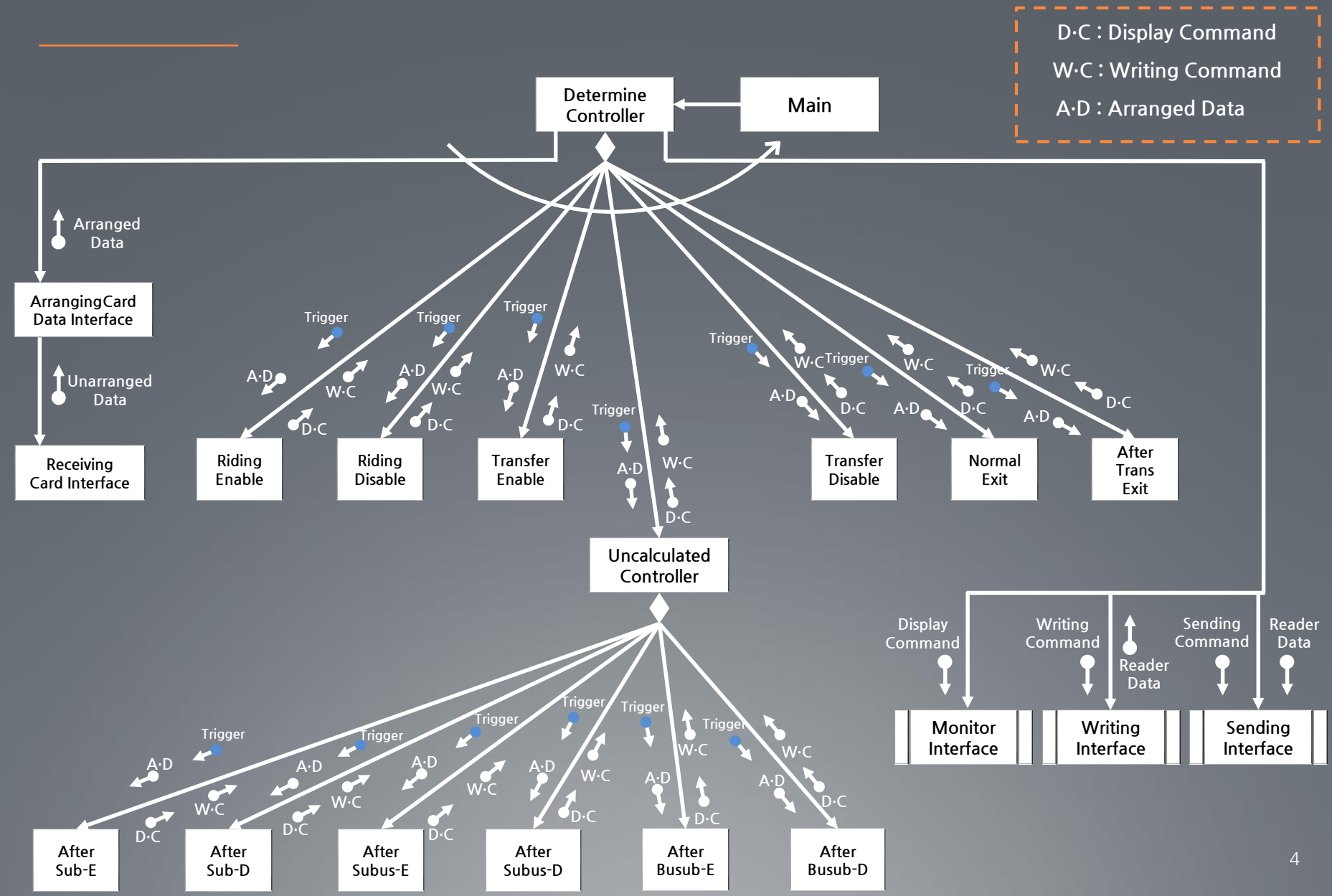
Contents

- Explanation
  - Structured Charts
  - Programing Code
- Unit Test Plan
  - Test items
  - Features to be tested / not to be tested
  - Test Identification & Specification
- Unit Test Result
  - Unit Test Report
  - Demo

Explanation

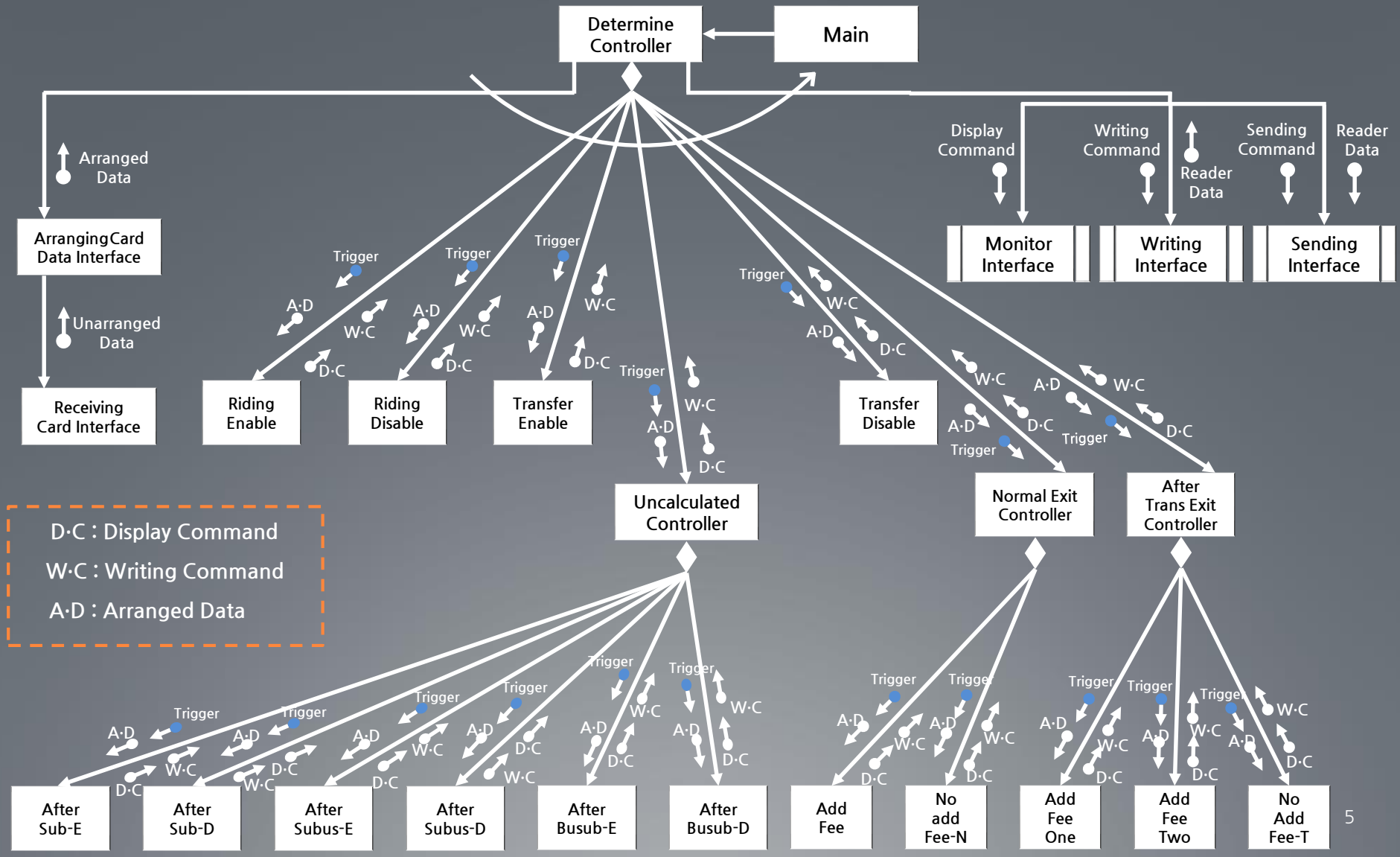
- Structured Charts
- Programing Code

Structured Chart Advanced

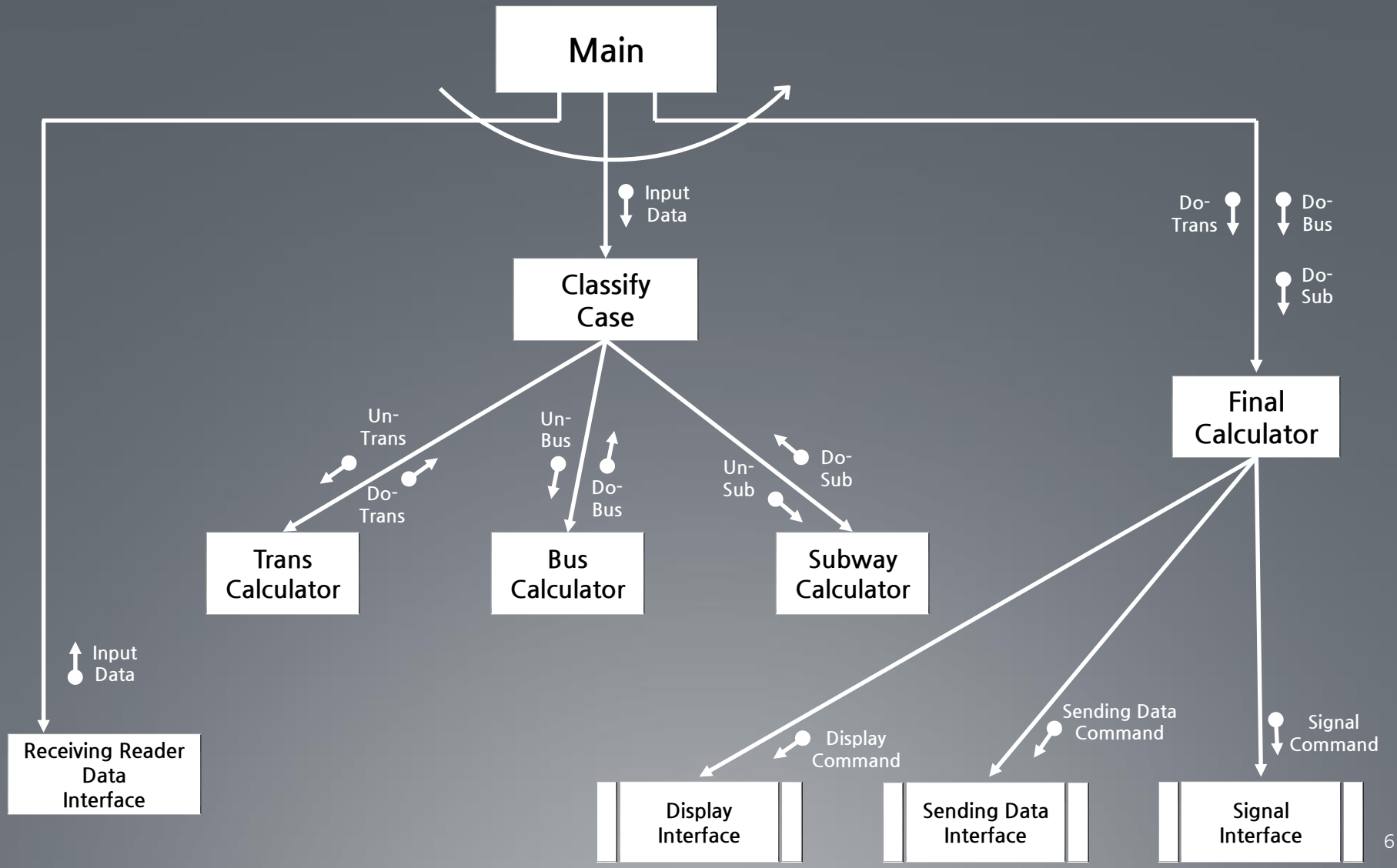


Subway

Structured Chart Advanced



Structured Chart Advanced



Programing Code

Bus & Subway Reader

```
typedef struct
{
    long long time;
    char vechicle[7];
    char inout[4];
    int lo;
    char info[10];
}unrranged_data;

typedef struct
{
    char cname[20];
    long long time;
    char info[10];
    bool uncal_day;
    bool uncal;
    bool trans;
    bool after;
    bool in;
    bool bus;
    bool uncal_s;
    bool uncal_sb;
    bool uncal_bs;
    int lo;
    int sta;
}arranged_data;

typedef struct
{
    long long time;
    char vechicle[7];
    char inout[4];
    int pay;
    char info[10];
}reader_data;
```

```
typedef struct
{
    int fee;
    int lo;
}display_command;

typedef struct
{
    int fee;
    long long time;
    char vechicle[7];
    char inout[4];
    int lo;
    char readerInfo[10];
    char cardInfo[10];
    char name[20];
}writing_command;

typedef struct
{
    display_command dc;
    writing_command wc;
}command;

char readerID[2];
char readerTxt[25];
int count;
time_t init;

void printTime();
long long timetolong(time_t t);
time_t makeTime(long long ud);
long long getTime();
void *checkTime(void *arg);
```

Base.h

Structure	Description
unrragned data	Text 파일에서 읽어온 마지막 두 줄을 저장할 구조체
arranged data	unarranged data를 토대로 탑승 정보를 정리한 구조체
display command	화면에 출력될 거래내역
writing command	카드에 저장될 거래내역
command	Display command와 writing command를 가지고 있는 구조체

Programing Code

Bus & Subway Reader

Input.h

```
unrranged_data* receivingCardInterface();
arranged_data arrangingCardDataInterface(unrranged_data* ud);
```

Receiving Card Interface	카드의 정보를 받아와 unarranged data에 저장한다.
Arranging Card Data Interface	받아온 unarranged data를 arranged data로 정리한다.

DetermineController.h

```
void determineController();
```

Determine Controller	Arranged data를 토대로 경우를 정해 적절한 동작을 실행시킨다.

Programing Code

Bus & Subway Reader

Controller.h

```
command ridingEnable(arranged_data *ad);
command ridingDisable(arranged_data *ad);
command transferEnable(arranged_data *ad);
command transferDisable(arranged_data *ad);
command afterSub_E(arranged_data *ad);
command afterSub_D(arranged_data *ad);
command afterSubus_E(arranged_data *ad);
command afterSubus_D(arranged_data *ad);
command afterBusub_E(arranged_data *ad);
command afterBusub_D(arranged_data *ad);
command addFee(arranged_data *ad);
command noAddFeeN(arranged_data *ad);
command addFeeOne(arranged_data *ad);
command addFeeTwo(arranged_data *ad);
command noAddFeeT(arranged_data *ad);
command uncalculatedController(arranged_data *ad);
command normalExitController(arranged_data *ad);
command afterTransExitController(arranged_data *ad);
```

Output.h

```
void moniterInterface(display_command dc);
void writingInterface(writing_command wc);
void sendingInterface(command c);
```

Function	Description
Riding Enable	일반 승차인 경우
Riding Disable	일반 승차인데 잔액이부족하여 탑승이 불가능한 경우
Transfer Enable	환승 승차인 경우
Transfer Disable	환승 승차인 경우인데 잔액이 부족하여 탑승이 불가능한 경우
Uncalculated Controller	미정산 요금이 남아있는 경우
NormalExit Controller	일반 하차인 경우(Subway)
AfterTransExit Controller	환승 승차 후 하차 시 (Subway)

Moniter interface	콘솔 창에 탑승 정보를 띄운다
Writing Interface	카드와 단말기에 탑승 정보를 씌운다.

Programing Code

Calculator

Base.h

```
typedef enum
{
    false,
    true
}bool;

typedef struct
{
    long long time;
    char vechicle[7];
    char inout[4];
    int lo;
    char info[10];
    bool flag;
}reader_data;
```

```
typedef struct
{
    long long btime;
    int bmoney;
    long long stime;
    int smoney;
}command;

int line;
reader_data *rd;
long long time_bus;
long long time_sub;
int do_trans[2];

void wait();
```

Structure	Description
Reader Data	단말기 내역에서 읽어온 정보를 저장할 구조체
command	정산 완료된 정보를 저장할 구조체

Programing Code

Calculator

Controller.h

```
int* transCalculator();
int busCalculator();
int subwayCalculator();
void classifyCase();
void finalCalculator(int do_trans[], int do_bus, int do_sub);
```

Structure	Description
Classify Case	경우를 분류해 알맞은 계산기로 보낸다
Trans Calculator	환승한 경우의 요금을 계산한다
Bus Calculator	버스만 탑승한 경우의 요금을 계산한다
Subway Calculator	지하철만 탑승한 경우의 요금을 계산한다
Final Calculator	3경우의 금액을 합쳐 command를 내보낸다

Programing Code

Calculator

Output.h

```
void displayInterface(command dc);
void sendingDataInterface(command sc);
void signalInterface();
```

Structure	Description
Display Interface	정산 완료된 정보를 화면에 출력한다
Sending Data	정산 완료된 정보를 각 회사로 전송하고 단말기의 사용내역을 초기화한다

Unit Test Plan

- Test items
- Features to be tested / not to be tested
- Test Identification & Specification

Test Items

Bus & Subway Reader

ID	Name
1.1	Receiving Card Interface
1.2	Arranging Card Data Interface
2.1.1	Determine Controller
2.1.2	Riding Enable
2.1.3	Riding Disable
2.1.4.1	Uncalculated Controller
2.1.4.2	After Sub-E
2.1.4.3	After Sub-D
2.1.4.4	After Subus-E
2.1.4.5	After Subus-D
2.1.4.6	After Busub-E
2.1.4.7	After Busub-D

ID	Name (Subway)
2.1.5	Transafter Enable
2.1.6	Transafter Disable
2.1.7	Normal Exit
2.1.7.2	Normal Exit Controller
2.1.7.3	Add Fee
2.1.7.4	No Add Fee-N
2.1.8.2	After Trans Exit Controller
2.1.8.3	Add Fee One
2.1.8.4	Add Fee Two
2.1.8.5	No Add Fee-T
2.2	Monitor Interface
2.3	Writing Interface

Test Items

Calculator

ID	Name
1.1	Reciving Reader Data Interface
2.1.1	Classify Case
2.1.2	Trans Calculator
2.1.3	Bus Calculator
2.1.4	Subway Calculator
2.1.5	Final Calculator

ID	Name
2.2	Display Interface
2.3	Sending Data Interface
2.4	Signal Interface

Test Items

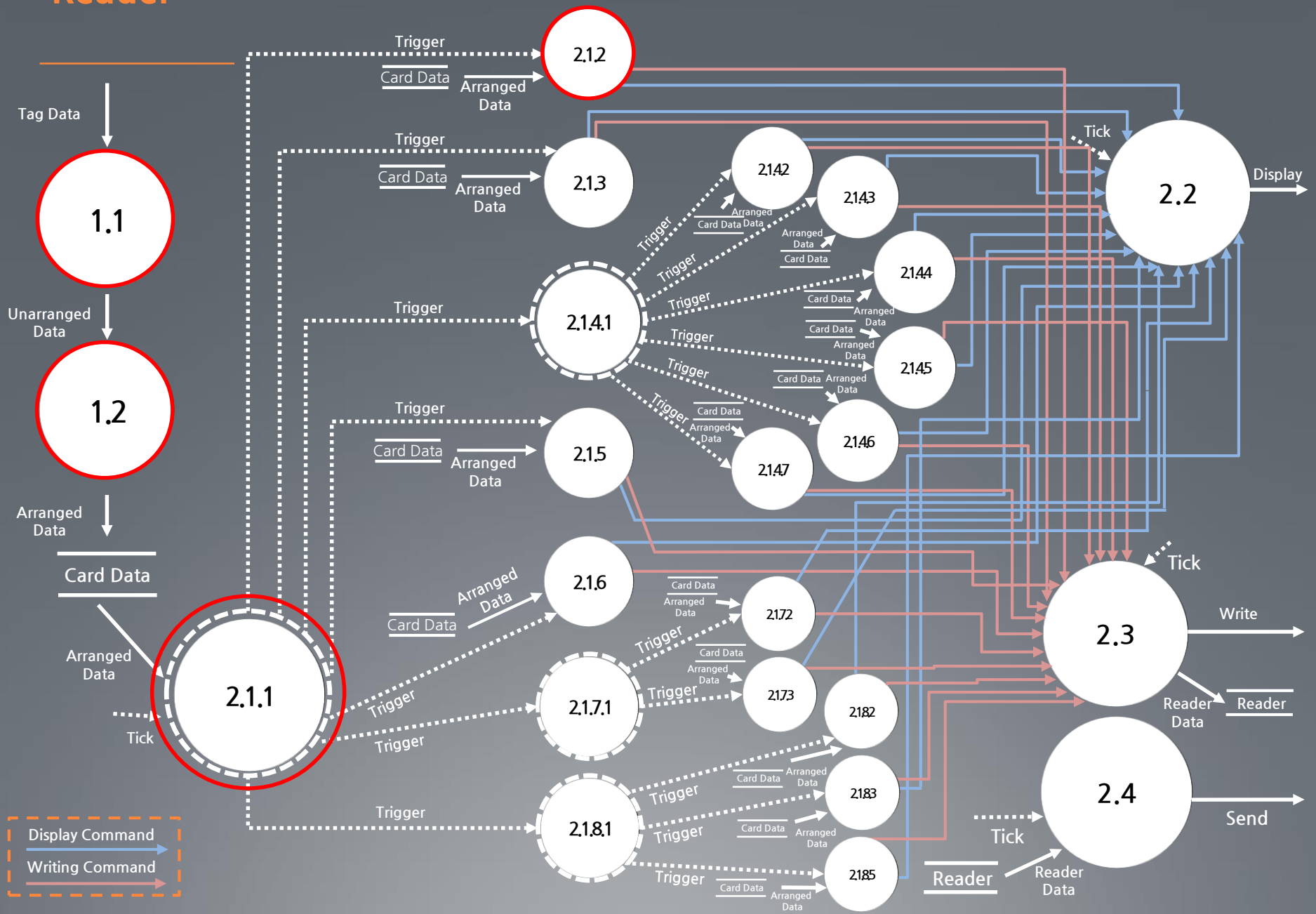
Calculator

ID	Name
1.1	Reciving Reader Data Interface
2.1.1	Classify Case
2.1.2	Trans Calculator
2.1.3	Bus Calculator
2.1.4	Subway Calculator
2.1.5	Final Calculator

ID	Name
2.2	Display Interface
2.3	Sending Data Interface
2.4	Signal Interface

Reader

Features to be tested



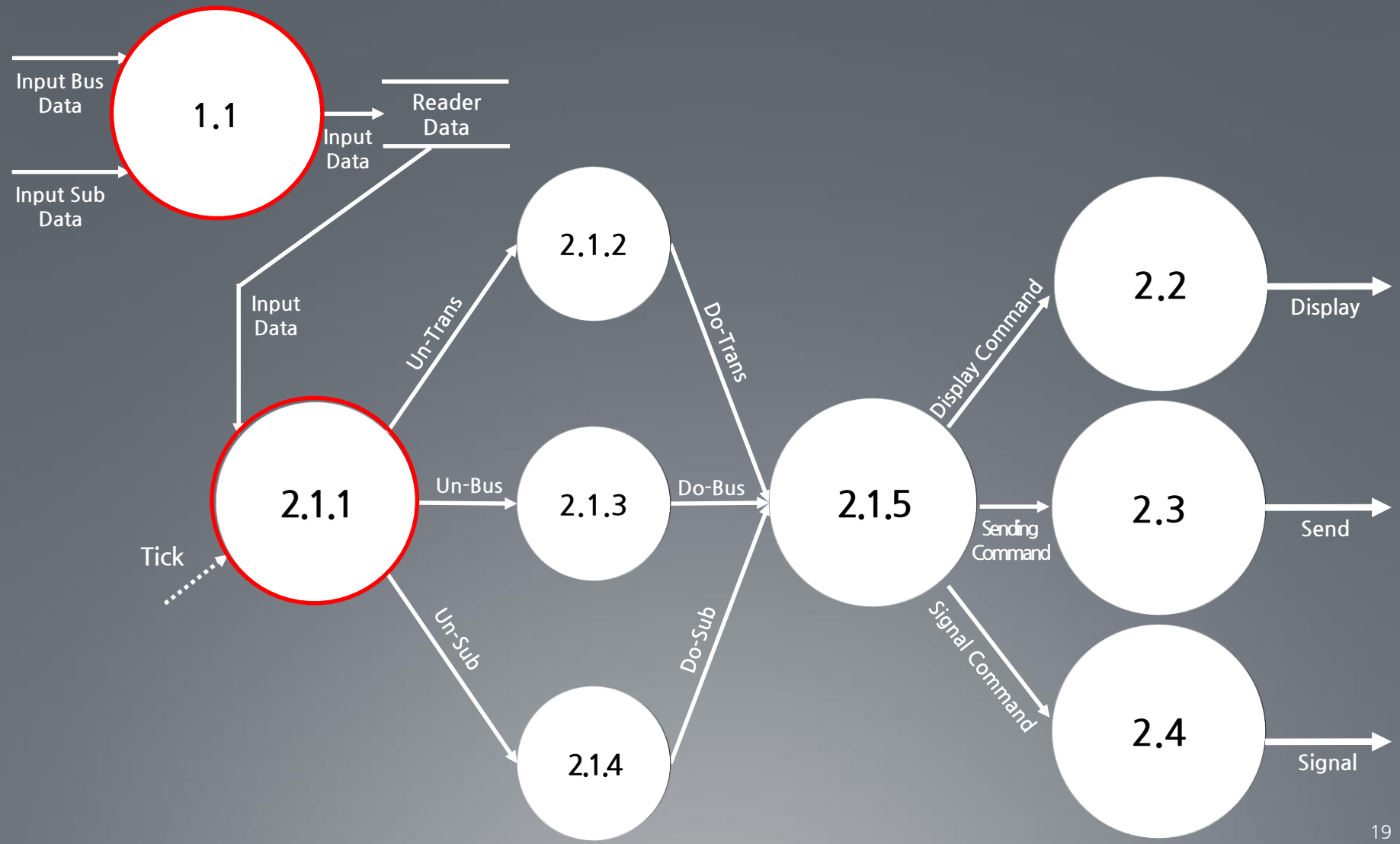
Reader

Features to be tested

ID	Name	Description
1.1	Receiving Card Interface	카드가 태그 되었을 시 해당 카드의 텍스트파일을 읽어와서 가장 최근 거래내역 2개(Unarranged Data)를 내보낸다
1.2	Arranging Card Data Interface	Unarranged Data를 받아서 UNAL, TRANS, STA 등 Arranged Data 로 정리 후 내보낸다
2.1.1	Determine Controller	Arranged Data 를 받아 어떠한 경우인지 판단하여 해당하는 경우의 프로세스를 동작시킨다
2.1.2	Riding Enable	일반 승차이고 탑승이 가능할 때 실행되며 요금을 계산하여 Command를 내보낸다

Calculator

Features to be tested



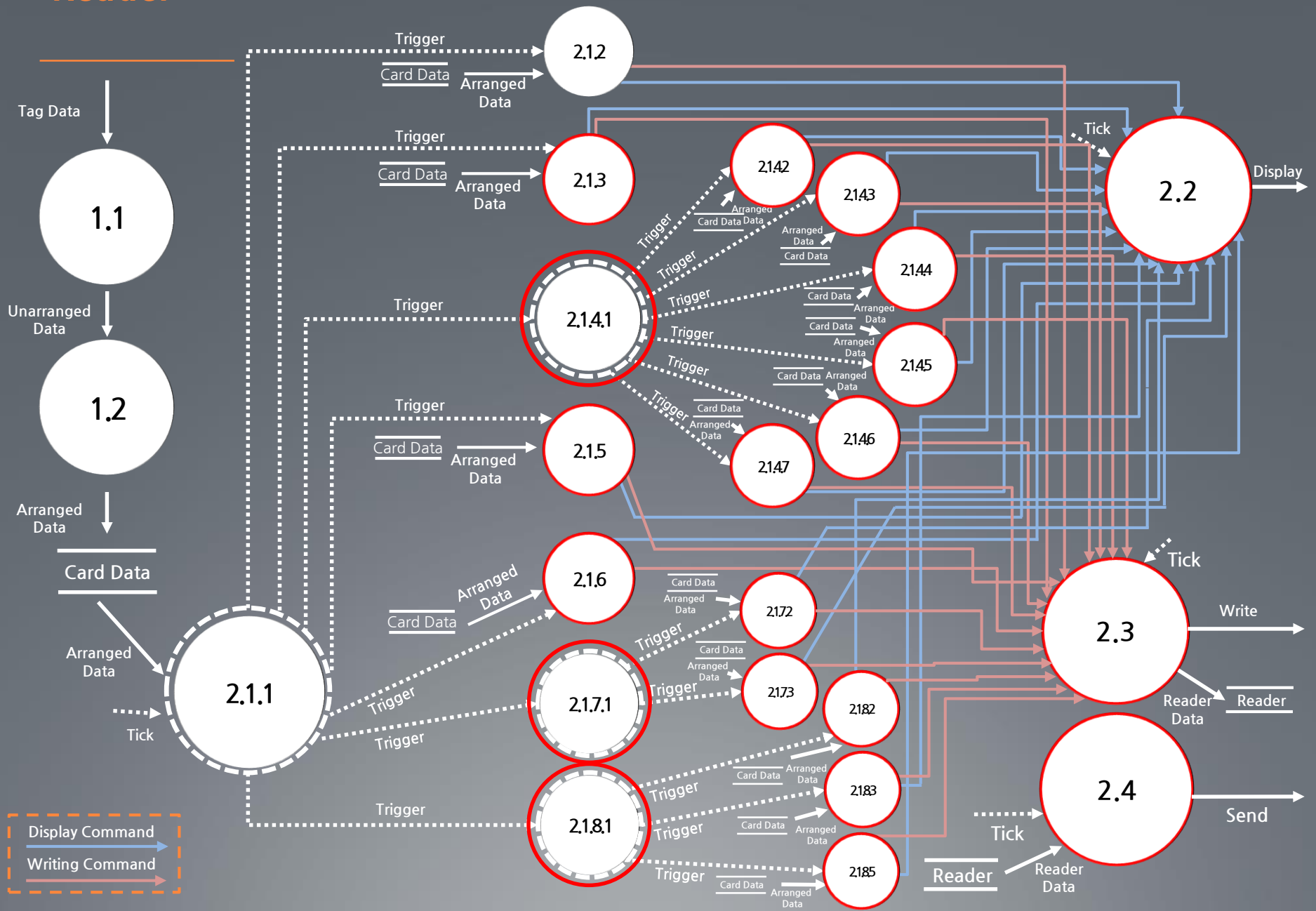
Calculator

Features to be tested

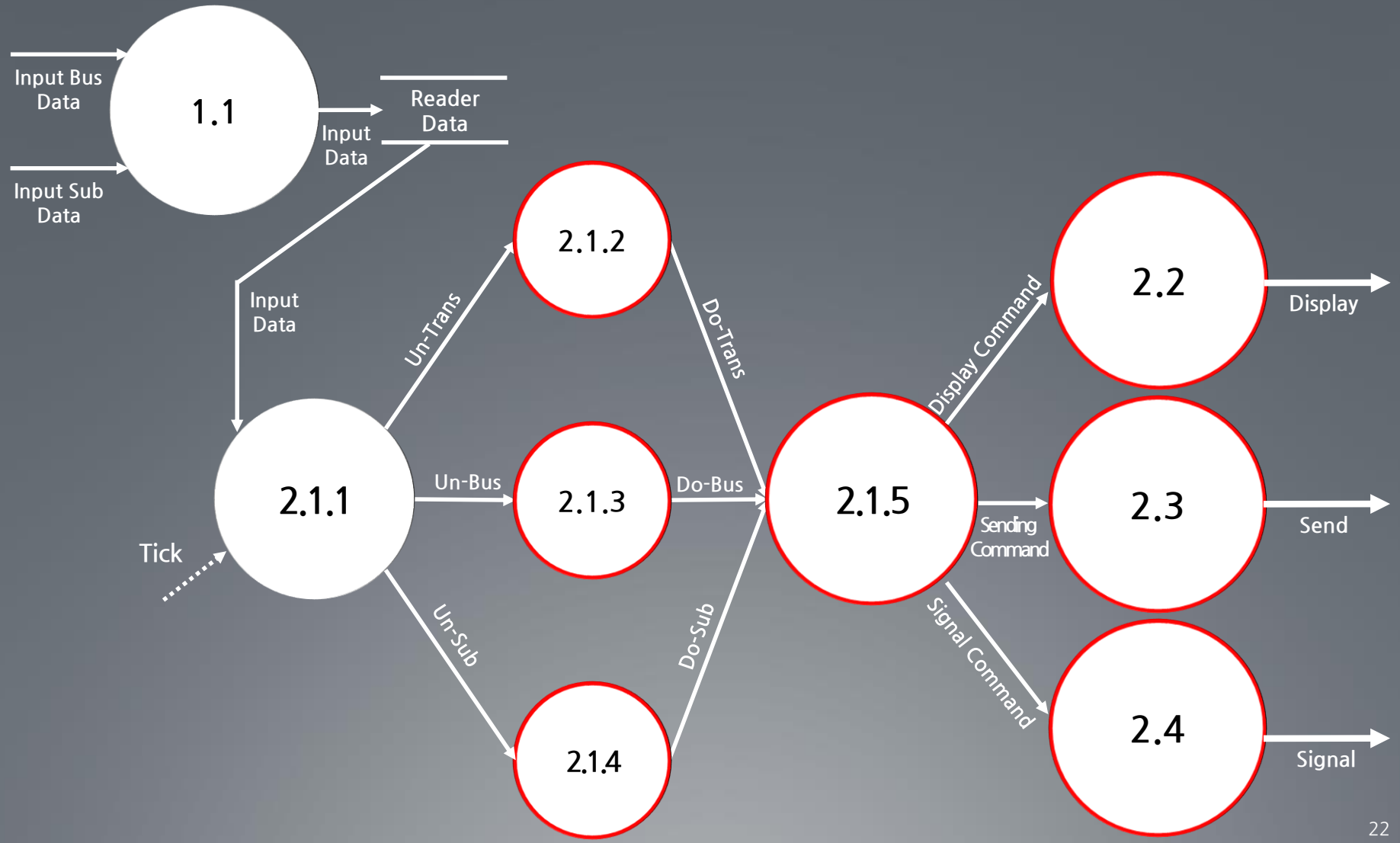
ID	Name	Description
1.1	Receiving Card Interface	버스와 지하철로부터 단말기 거래 내역을 받아 종합하여 전송한다
2.1.1	Classify Case	종합된 기록을 버스만 이용한 경우, 지하철만 이용한 경우, 버스와 지하철을 환승하여 이용한 경우의 3가지 경우로 분류해서 각각의 프로세스를 전송한다

Reader

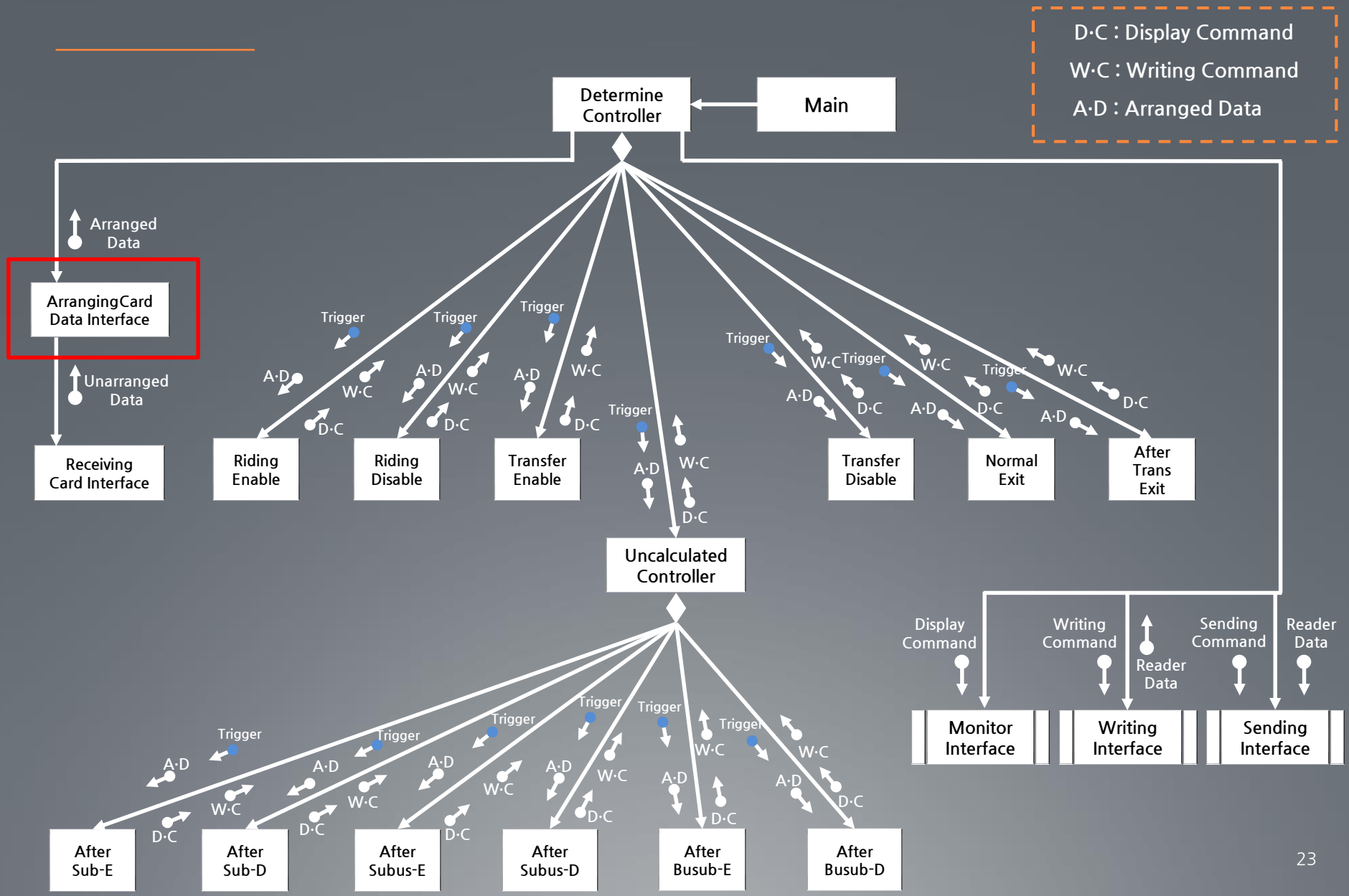
Features not to be tested



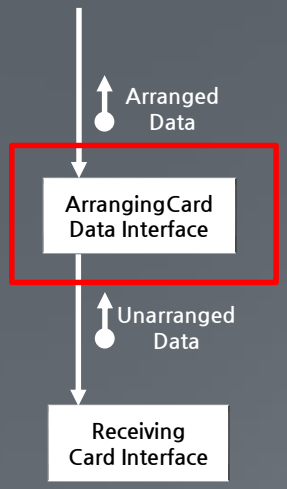
Features not to be tested



Test Identification & Specification



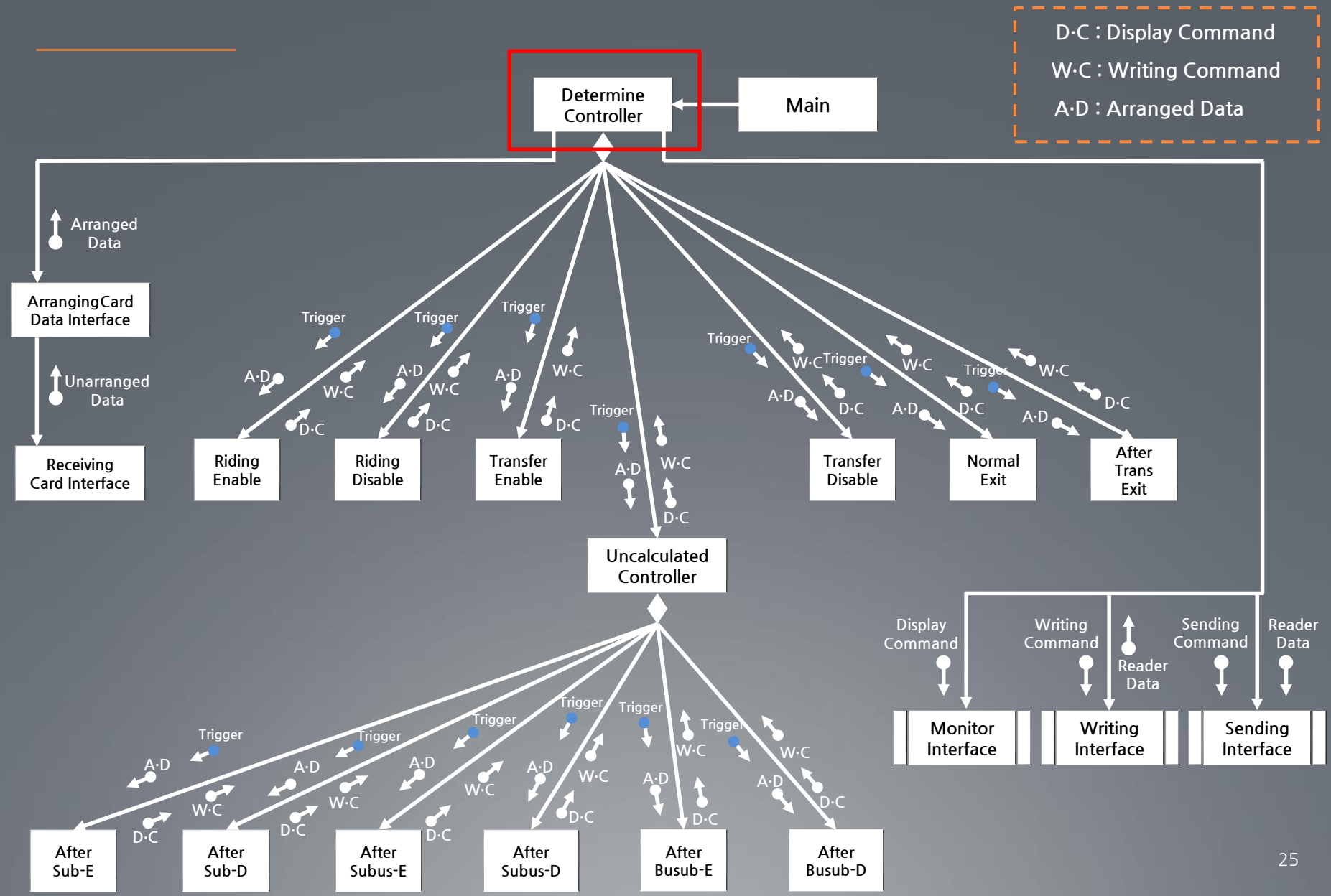
Reader



Test Identification & Specification

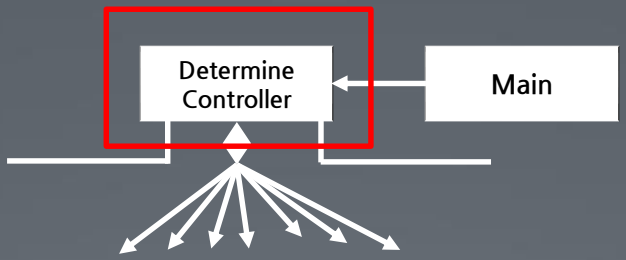
Identifier	Input Specification	Output Specification
PTS UTC 000 000	Unarranged data[2] = {20141117195032, "BUS", "OUT", 5950, "B_1"}, {20141117195035, "BUS", "IN", 5950, "B_2"}	Arranged data의 In == True
PTS UTC 000 001	Unarranged data[2] = {20141117195032, "BUS", "OUT", 5950, "B_1"}, {20141117195035, "BUS", "IN", 5950, "B_2"}	Arranged data의 Bus == True
PTS UTC 000 002	Unarranged data[2] = {20141117195032, "SUBWAY", "IN", 5950, "S1_1"}, {20141117195035, "SUBWAY", "OUT", 5950, "S2_2"} (and 마지막 태그시간이 현재시간과 15초이내)	Arranged data의 Trans == True
PTS UTC 000 003	Unarranged data[2] = {20141117195032, "SUBWAY", "OUT", 5950, "S1_1"}, {20141117195035, "BUS", "IN", 5950, "B_1"}	Arranged data의 After == True
PTS UTC 000 004	Unarranged data[2] = {20141117195032, "SUBWAY", "OUT", 5950, "S1_1"}, {20141117195035, "BUS", "IN", 5950, "B_1"} 이고 하루(3분)가 지났을 때	Arranged data의 Uncal day == True
PTS UTC 000 005	Unarranged data[2] = {20141117195032, "SUBWAY", "OUT", 5950, "S1_1"}, {20141117195035, "SUBWAY, "IN", 5950, "S1_2"}	Arranged data의 Uncal s == True
PTS UTC 000 006	Unarranged data[2] = {20141117195032, "BUS", "OUT", 5950, "B_1"}, {20141117195035, "SUBWAY, "IN", 5950, "S1_2"}	Arranged data의 Uncal bs == True
PTS UTC 000 007	Unarranged data[2] = {20141117195032, "SUBWAY", "OUT", 5950, "B_1"}, {20141117195035, "BUS, "IN", 5950, "S1_2"	Arranged data의 Uncal sb == True
PTS UTC 000 008	Unarranged data[2] = {20141117195032, "BUS", "OUT", 5950, "B_1"}, {20141117195035, "SUBWAY, "IN", 5950, "S1_2"}	Arranged data의 Uncal == True
PTS UTC 000 009	Unarranged data[2] = {20141117195032, "BUS", "OUT", 5950, "B_1"}, {20141117195035, "SUBWAY, "IN", 5950, "S1_2"}	Arranged data의 Time == 20141117195035
PTS UTC 000 010	Unarranged data[2] = {20141117195032, "BUS", "OUT", 5950, "B_1"}, {20141117195035, "SUBWAY", "IN", 5950, "S1_2"}	Arranged data의 Lo == 5950
PTS UTC 000 011	Unarranged data[2] = {20141117195032, "BUS", "OUT", 5950, "B_1"}, {20141117195035, "SUBWAY", "IN", 5950, "S1_2"}	Arranged data의 Info == S1 2

Test Identification & Specification



Reader

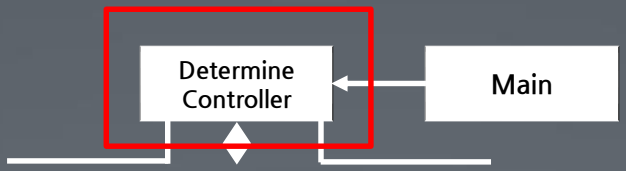
Test Identification & Specification



Identifier	Input Specification	Output Specification
PTS UTC 001 000	Unarranged_data[2] = {20141119133000, ",SUBWAY", "IN", 10000, "S1_1"}, {20141119133010, "SUBWAY, "OUT", 10000, "S2_1"}	RidingEnable함수 호출
PTS UTC 001 001	Unarranged_data[2] = {20141119133000, ",SUBWAY", "IN", 10000, "S1_1"}, {20141119133010, "SUBWAY", "OUT", 200, "S2_1"}	ridingDisable함수 호출
PTS UTC 001 002	Unarranged_data[2] = {20141119133000, ",SUBWAY", "IN", 10000, "S1_1"}, {20141119133010, "SUBWAY", "OUT", 10000, "S2_1"} (and 마지막 태그시간과 현재시간이 15초 이내)	transferEnable함수 호출
PTS UTC 001 003	Unarranged_data[2] = { 20141119133000, ",SUBWAY", "IN", 10000, "S1_1"}, { 20141119133010, "SUBWAY", "OUT", 100, "S2_1"}	transferDisable함수 호출
PTS UTC 001 004	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133010, "BUS," "IN", 10000, "B_2"}	normalExit함수 호출
PTS UTC 001 005	Unarranged_data[2] = {20141119133000, "SUBWAY", "OUT", 10000, "S1_1"}, {20141119133010, "BUS," "IN", 10000, "B_2"}	afterTransExit함수 호출
PTS UTC 001 006	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133110, "SUBWAY" "IN", 10000, "S1_2"}	afterSub_E함수 호출
PTS UTC 001 007	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133110, "SUBWAY" "IN", 100, "S1_2"}	afterSub_D함수 호출

Reader

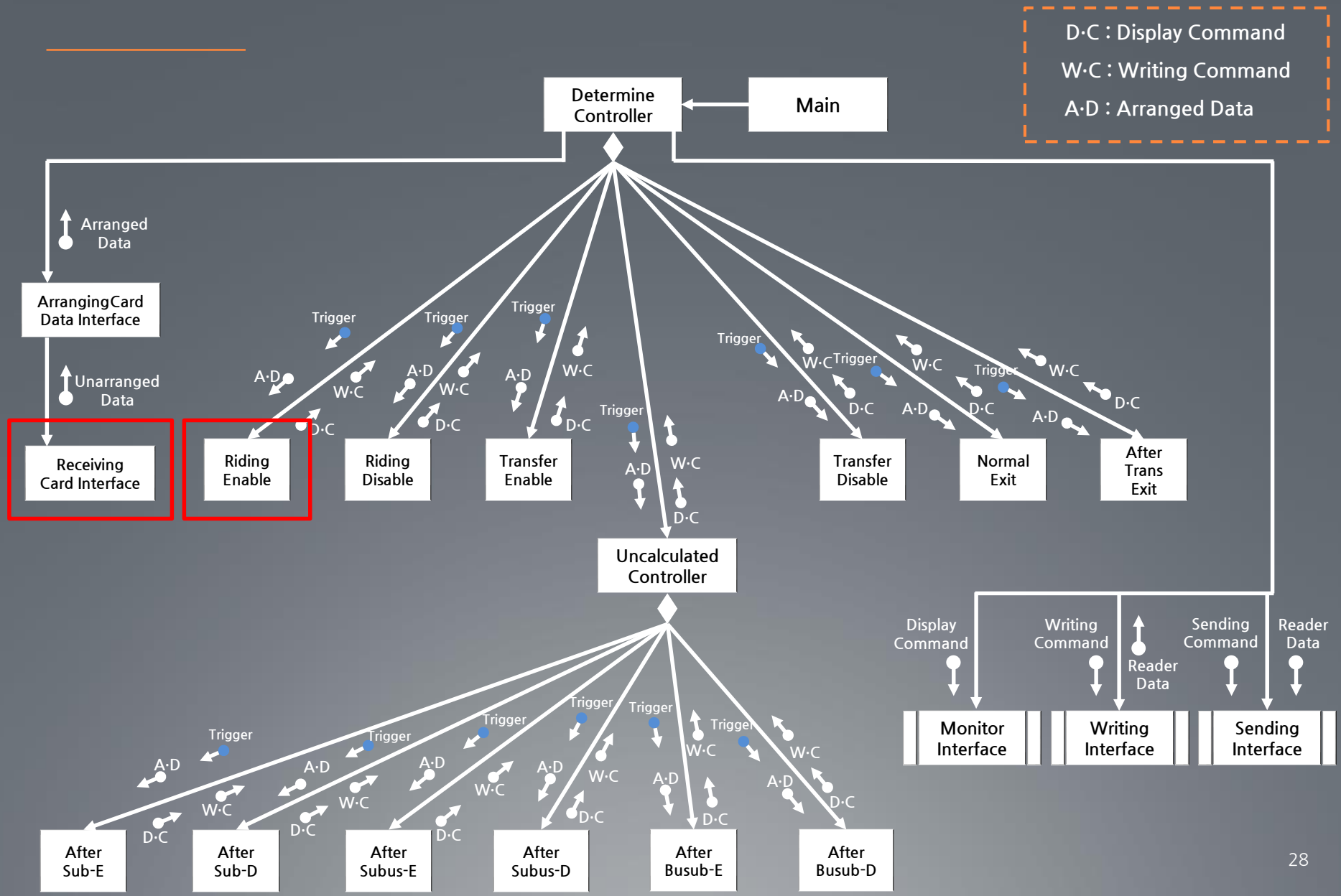
Test Identification & Specification



Identifier	Input Specification (Subway)	Output Specification (Subway)
PTS UTC 001 008	Unarranged_data[2] = {20141119133000, "SUBWAY", "OUT", 10000, "S1_1"}, {20141119133010, "BUS" "IN", 10000, "B_2"}	afterSubus_E함수 호출
PTS UTC 001 009	Unarranged_data[2] = {20141119133000, "SUBWAY", "OUT", 10000, "S1_1"}, {20141119133010, "BUS" "IN", 100, "B_2"}	afterSubus_D함수 호출
PTS UTC 001 010	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133010, "SUBWAY" "IN", 10000, "S1_2"}	afterBusub_E함수 호출
PTS UTC 001 011	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133010, "SUBWAY" "IN", 100, "S1_2"}	afterBusub_D함수 호출
PTS UTC 001 012	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133110, "SUBWAY" "IN", 10000, "S3_2"}	addFee함수 호출
PTS UTC 001 013	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133110, "SUBWAY" "IN", 10000, "S4_2"}	noAddFee_N함수 호출
PTS UTC 001 014	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133010, "SUBWAY" "IN", 10000, "S4_2"}	addFeeOne함수 호출
PTS UTC 001 015	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133010, "SUBWAY" "IN", 10000, "S3_2"}	addFeeTwo함수 호출
PTS UTC 001 006	Unarranged_data[2] = {20141119133000, "BUS", "OUT", 10000, "B_1"}, {20141119133010, "SUBWAY" "IN", 10000, "S5_2"}	noAddFee_T함수 호출

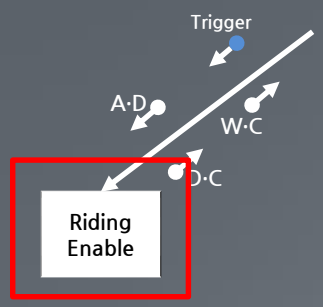
Reader

Test Identification & Specification



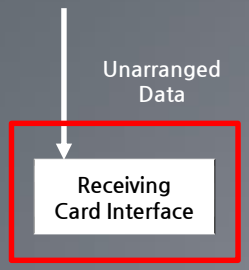
Reader

Test Identification & Specification



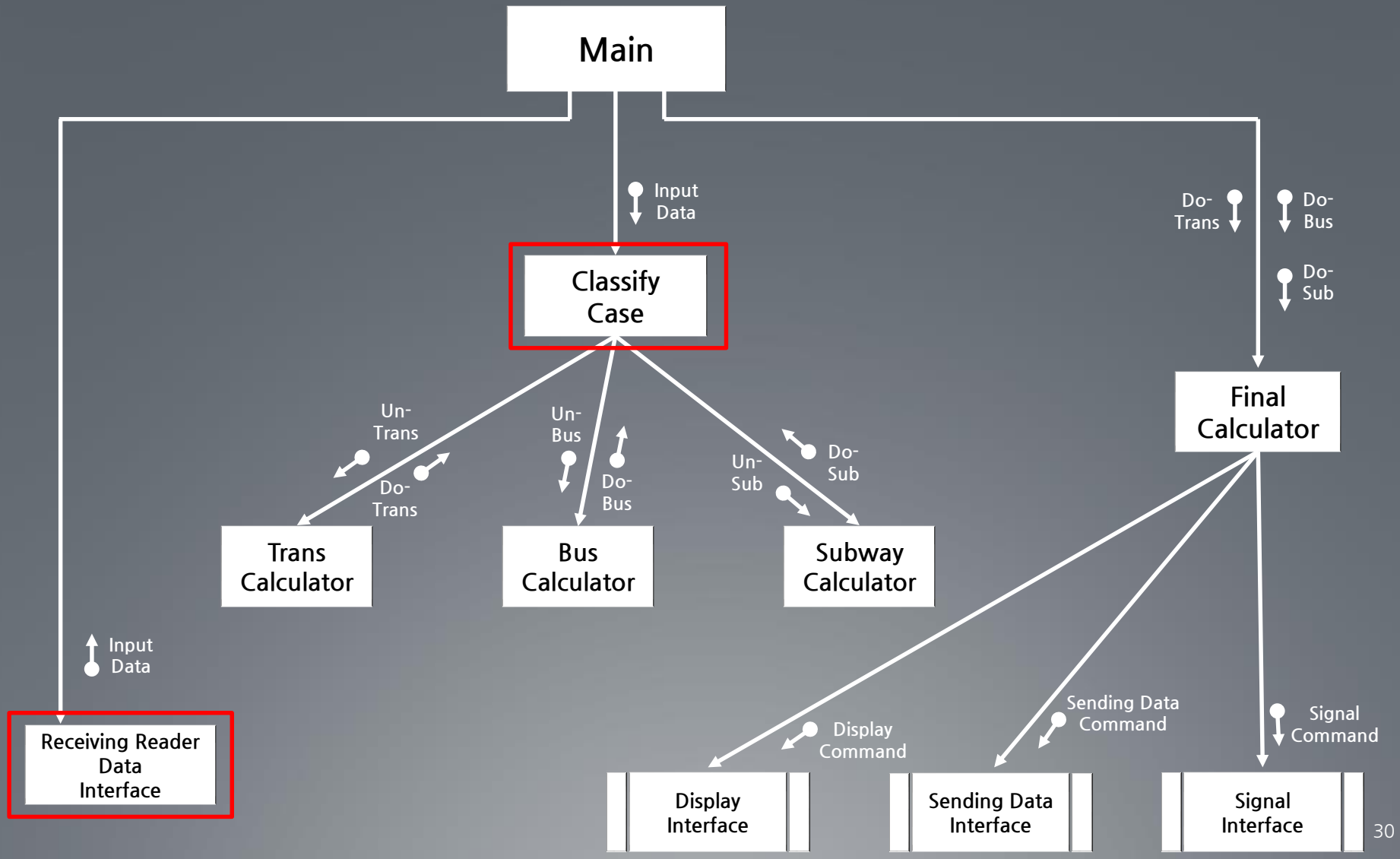
{20141117195032, "BUS", "IN", 5950, "B_1"},
{20141117195035, "BUS, "OUT", 5950, "B_2"}

Identifier	Input Specification	Output Specification
PTS UTC 002 000	Arranged_data {cname[20]="input.txt", time=20141117195035, info[10]=B_2, lo=5950, in=0, bus=1, trans=0, after=0, uncal_day=0, uncal_s=0 uncal_bs=0 uncal_sb=0 uncal=0, sta=0}	Display_command (fee=1050, lo=4900) Writing command {fee=1050, time=20141117195035, vechicle="BUS", inout="IN", lo=4900, readerInfo="B_1", cardInfo="B_1", name="input.txt"}



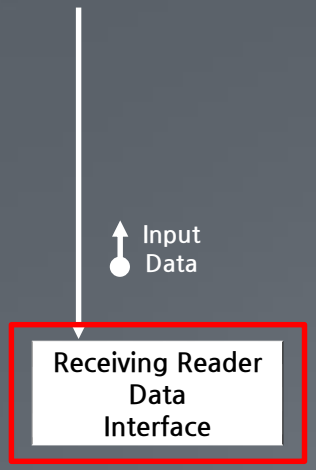
Identifier	Input Specification	Output Specification
PTS UTC 003 000	임의의 카드정보가 담긴 텍스트파일	Unarranged data (텍스트파일의 가장 밑의 두줄)

Test Identification & Specification

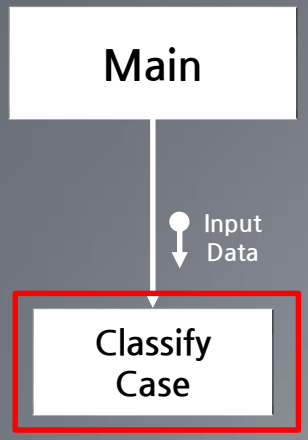


Calculator

Test Identification & Specification



Identifier	Input Specification	Output Specification
PTS UTC 004 000	Bus.txt, subway1.txt, subway2.txt, subway3.txt, subway4.txt, subway5.txt	Inputdata.txt



Identifier	Input Specification	Output Specification
PTS UTC 005 000	Inputdata.txt	un-trans.txt, un-bus.txt, un-sub.txt

Unit Test Result

- Unit Test Report
- Demo

UTR

Unit Test Report

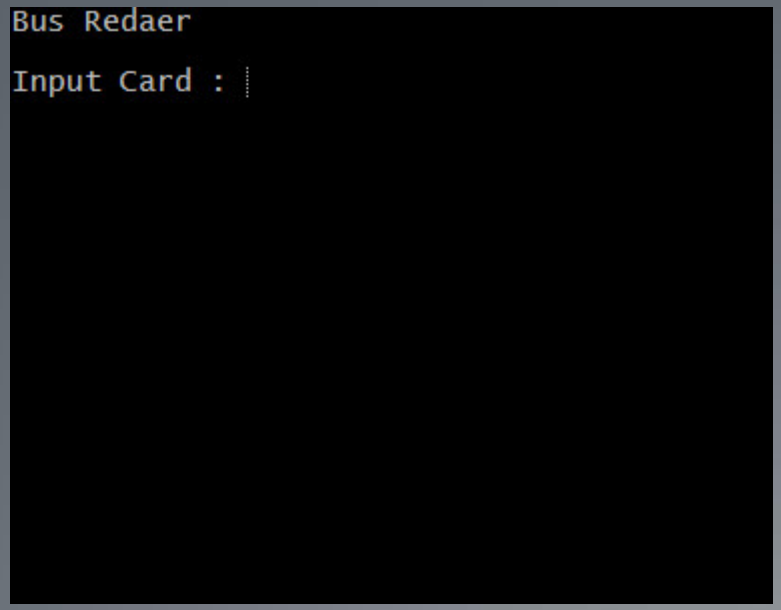
Identifier	Result
PTS UTC 000 000	Passed
PTS UTC 000 001	Passed
PTS UTC 000 002	Passed
PTS UTC 000 003	Passed
PTS UTC 000 004	Passed
PTS UTC 000 005	Passed
PTS UTC 000 006	Passed
PTS UTC 000 007	Passed
PTS UTC 000 008	Passed
PTS UTC 000 009	Passed
PTS UTC 000 010	Passed
PTS UTC 000 011	Passed

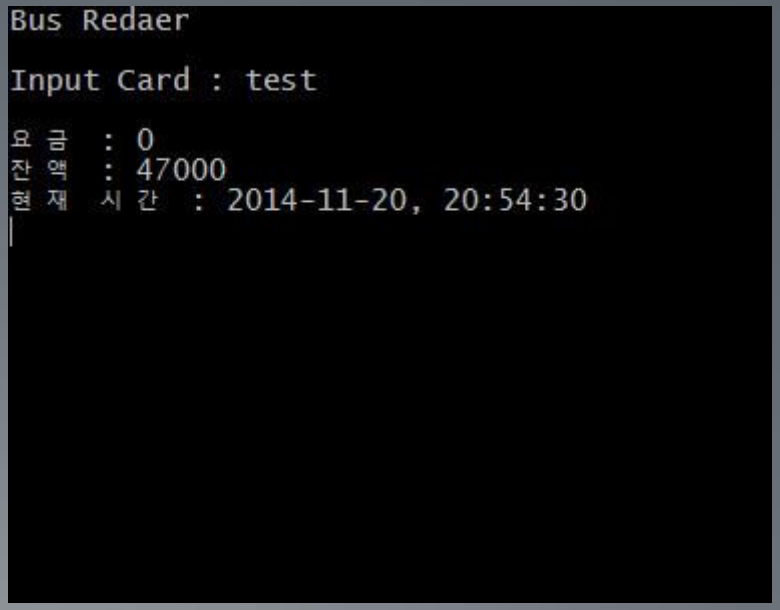
Identifier	Result
PTS UTC 001 000	Passed
PTS UTC 001 001	Passed
PTS UTC 001 002	Passed
PTS UTC 001 003	Passed
PTS UTC 001 004	Passed
PTS UTC 001 005	Passed
PTS UTC 001 006	Passed
PTS UTC 001 007	Passed
PTS UTC 001 008	Passed
PTS UTC 001 009	Passed
PTS UTC 001 010	Passed
PTS UTC 001 011	Passed

Identifier (subway)	Result
PTS UTC 001 010	Passed
PTS UTC 001 011	Passed
PTS UTC 001 012	Passed
PTS UTC 001 013	Passed
PTS UTC 001 014	Passed
PTS UTC 001 015	Passed
PTS UTC 001 016	Passed
PTS UTC 002 000	Passed
PTS UTC 003 000	Passed
PTS UTC 004 000	Passed
PTS UTC 005 000	Passed

UTR

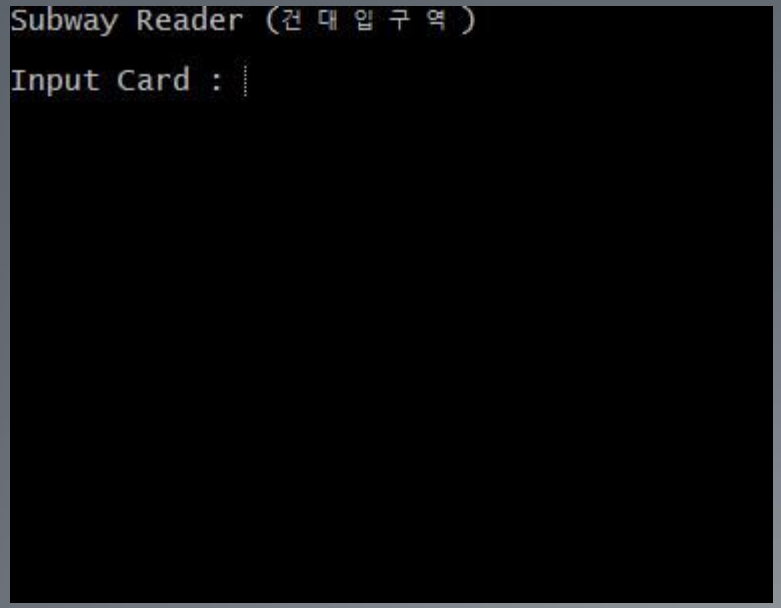
Bus Reader Demo

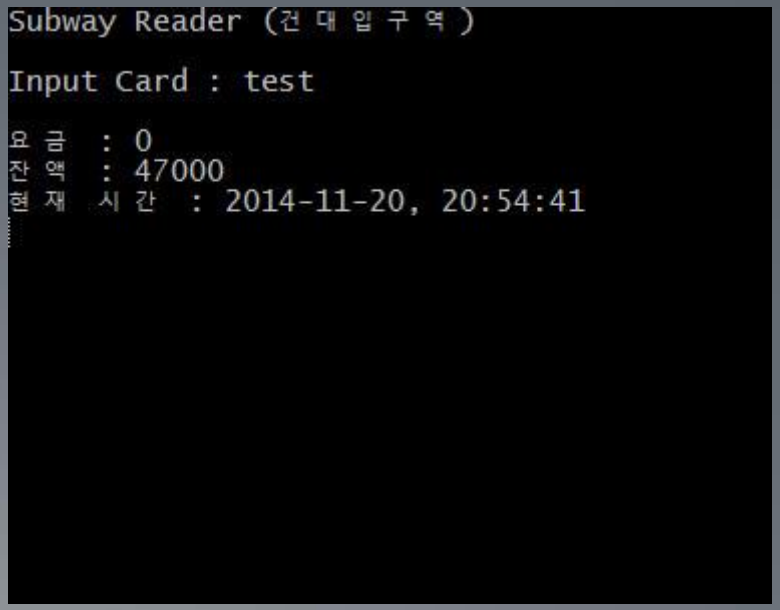




UTR

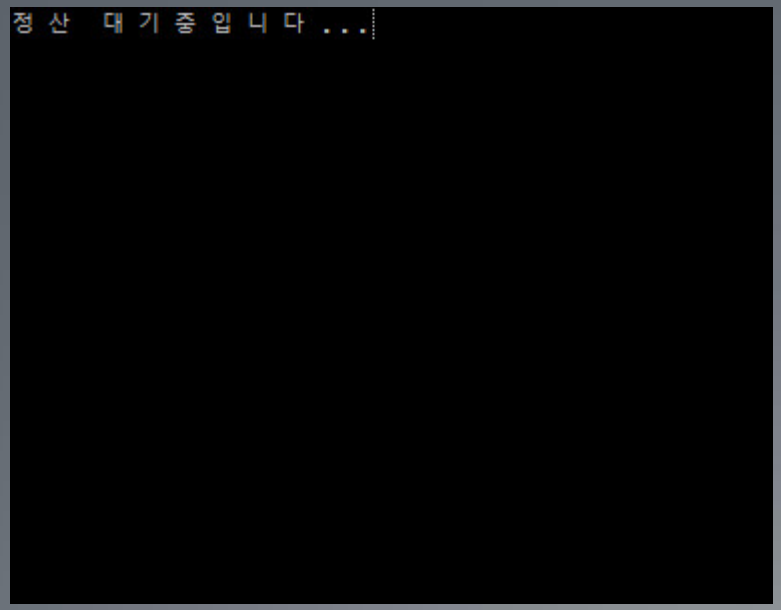
Subway Reader Demo

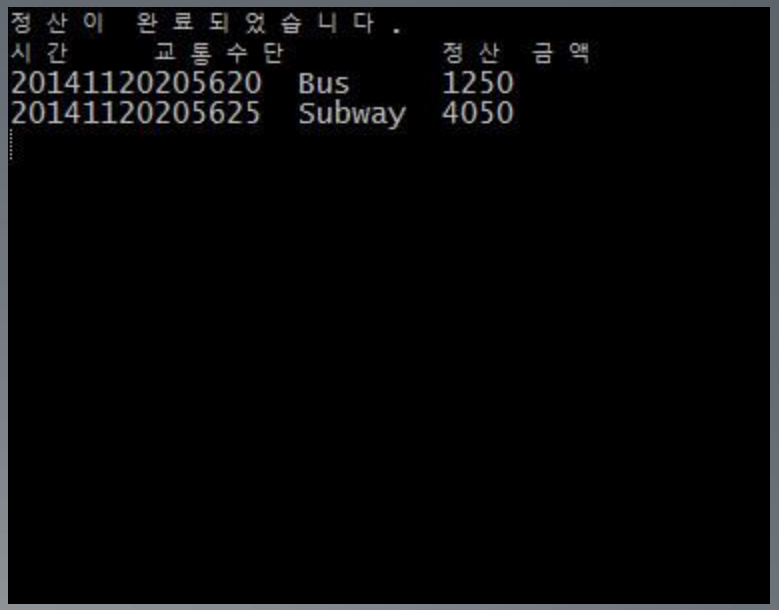




UTR

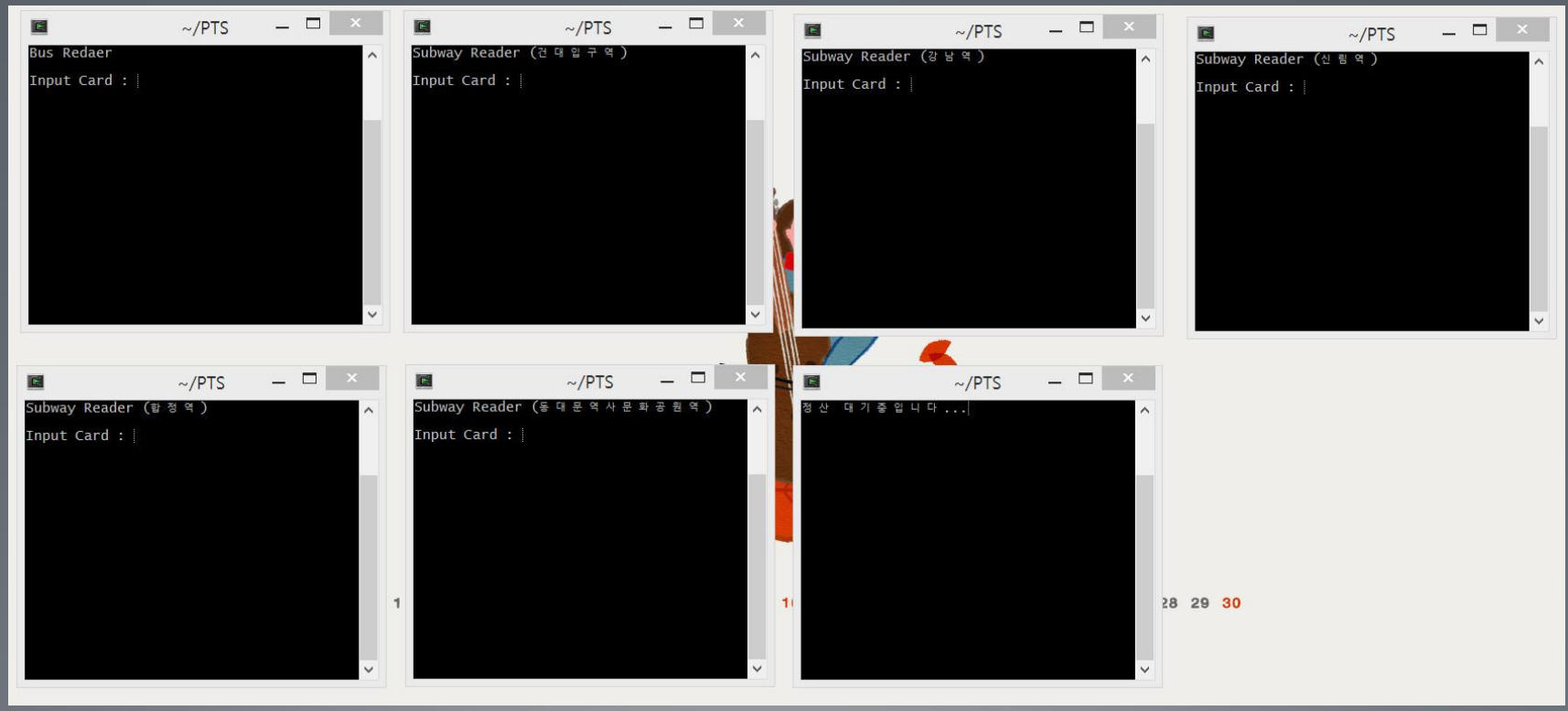
Calculator Demo





UTR

PTS Demo


~/PTS
Bus Redaer
Input Card :
~/PTS
Subway Reader (건대입구역)
Input Card :
~/PTS
Subway Reader (강남역)
Input Card :
~/PTS
Subway Reader (신림역)
Input Card :
~/PTS
Subway Reader (합정역)
Input Card :
~/PTS
Subway Reader (동대문역사문화공원역)
Input Card :
~/PTS
정산 대기중입니다...

Q & A