6th :: final presentation

Digital Door Lock System

Team 1

박소은 김주호 박선민 이유민

6th :: final presentation

Digital Door Lock System

Contents

1. Evolution of DDL System
2. Unit Testing & False Position
3. Demonstration

Evolution of DDL System

1.0 to **1.5** :: 정확히 명시되지 않은 용어 정의
데이터의 우선순위
데이터 저장소 세분화

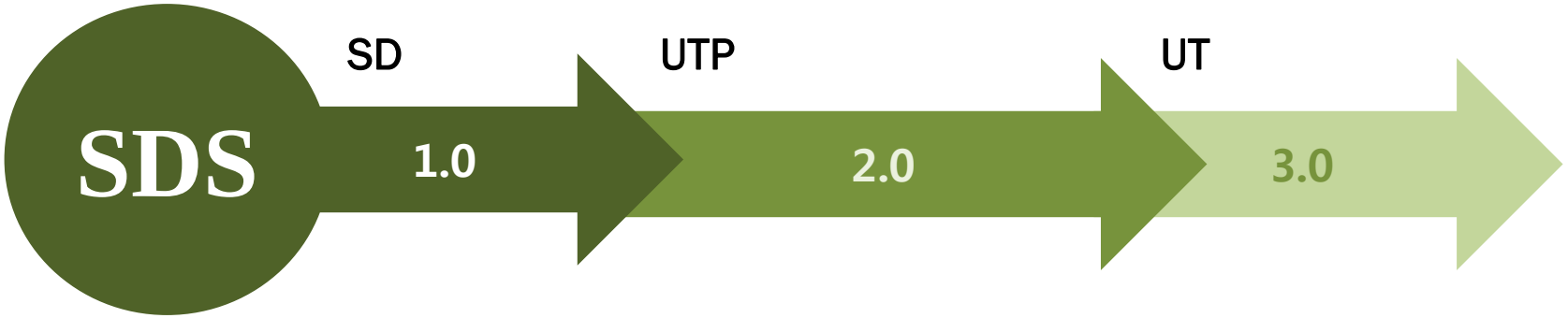
Evolution of DDL System

1.5 to 2.0 :: DFD의 잘못된 표기 수정
Display, Backlight, Password STD 수정

Evolution of DDL System

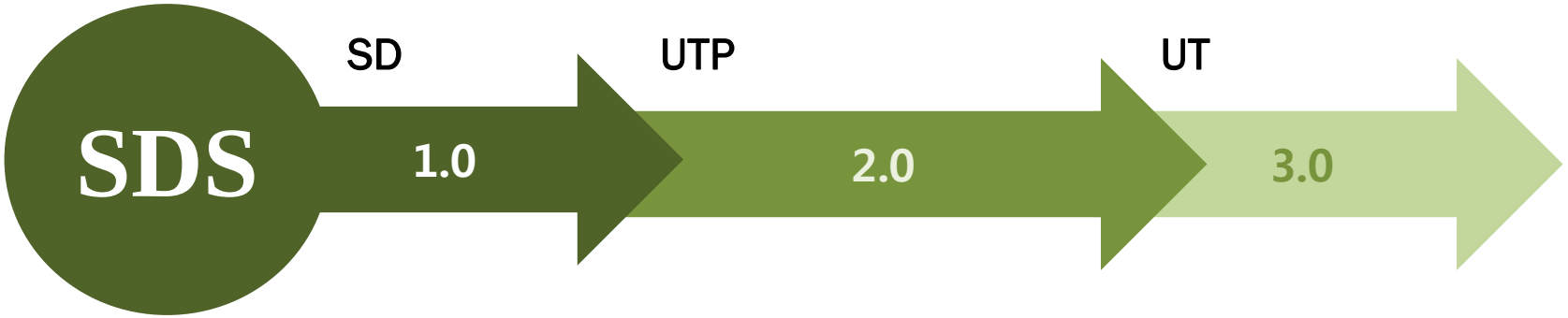
2.0 to 3.0 :: 각 컨트롤러에 들어오는 Input 수정
정확히 명시되지 않은 용어 정의
데이터의 우선순위
데이터 저장소 세분화

Evolution of DDL System



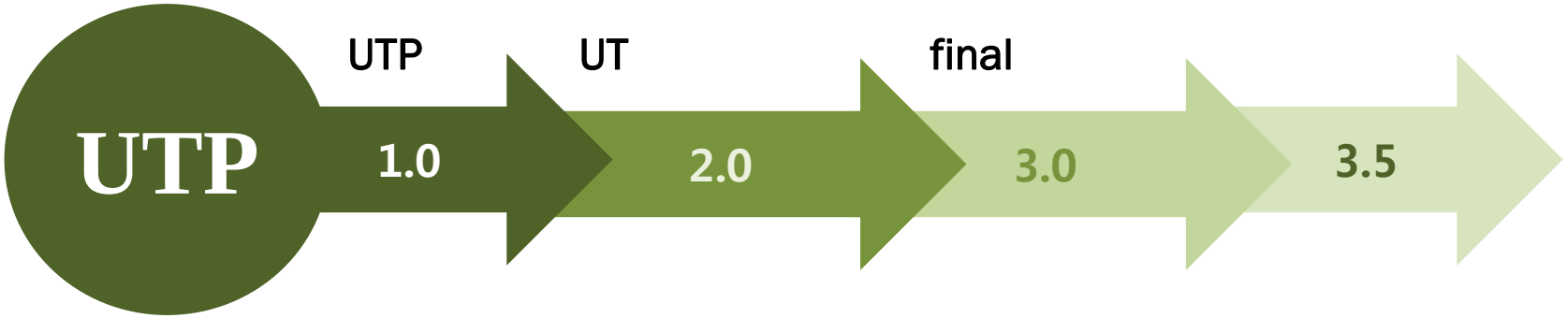
1.0 to **2.0** :: Structured Charts 수정

Evolution of DDL System



2.0 to **3.0** :: Structured Charts 상의 Input Data 명확히 표기

Unit Testing & False Position



UTP 3.0:: Test Case 수 41 ➡ 76

Test Case의 Value를 상세하게!

➡ **UTP 3.5**

UTR 2.0

Unit Testing & False Position

Type	Total	Ran	Passed	Failed	Inactive
suites	15	15	n/a	0	0
tests	76	76	54	22	0
asserts	76	76	54	22	n/a

Unit Testing & False Position

(suite_0 : EDLS_UTC_000_000)
(suite_0 : EDLS_UTC_000_001)
(suite_0 : EDLS_UTC_000_002)
(suite_0 : EDLS_UTC_000_006)
(suite_0 : EDLS_UTC_000_009)
(suite_0 : EDLS_UTC_000_010)
(suite_0 : EDLS_UTC_000_015)
(suite_0 : EDLS_UTC_000_016)
(suite_0 : EDLS_UTC_000_017)
(suite_1 : EDLS_UTC_001_001)
(suite_2 : EDLS_UTC_002_000)
(suite_2 : EDLS_UTC_002_001)
(suite_2 : EDLS_UTC_002_002)
(suite_2 : EDLS_UTC_002_005)
(suite_3 : EDLS_UTC_003_006)
(suite_4 : EDLS_UTC_004_005)
(suite_4 : EDLS_UTC_004_007)
(suite_4 : EDLS_UTC_004_010)
(suite_12 : EDLS_UTC_012_002)
(suite_12 : EDLS_UTC_012_004)
(suite_12 : EDLS_UTC_012_006)
(suite_14 : EDLS_UTC_014_000)))

Total Number of Failures : 22

Unit Testing & False Position

suite_0 Lock Controller

EDLS_UTC_000_000	State=Lock/ [BO]==FALSE	State=Lock
EDLS_UTC_000_001	State=Lock/ [C]==FALSE&&[L]==TRUE &&[K]==TRUE&&[D]==TRUE	Enable/ Locking==1
EDLS_UTC_000_002	State=Unlock/ [L]==FALSE &&[D]==FALSE&& [LB]==TRUE	State=Unlock
EDLS_UTC_000_006	State=Lock/ [L]==TRUE&& [C]==FALSE&&[P]==TRUE&&[D]==TRUE	Disable/ Locking==0
EDLS_UTC_000_009	State=Unlock/ [D]==TRUE&&[L]==FALSE&&[LB]==TRUE	Enable/ Locking==1

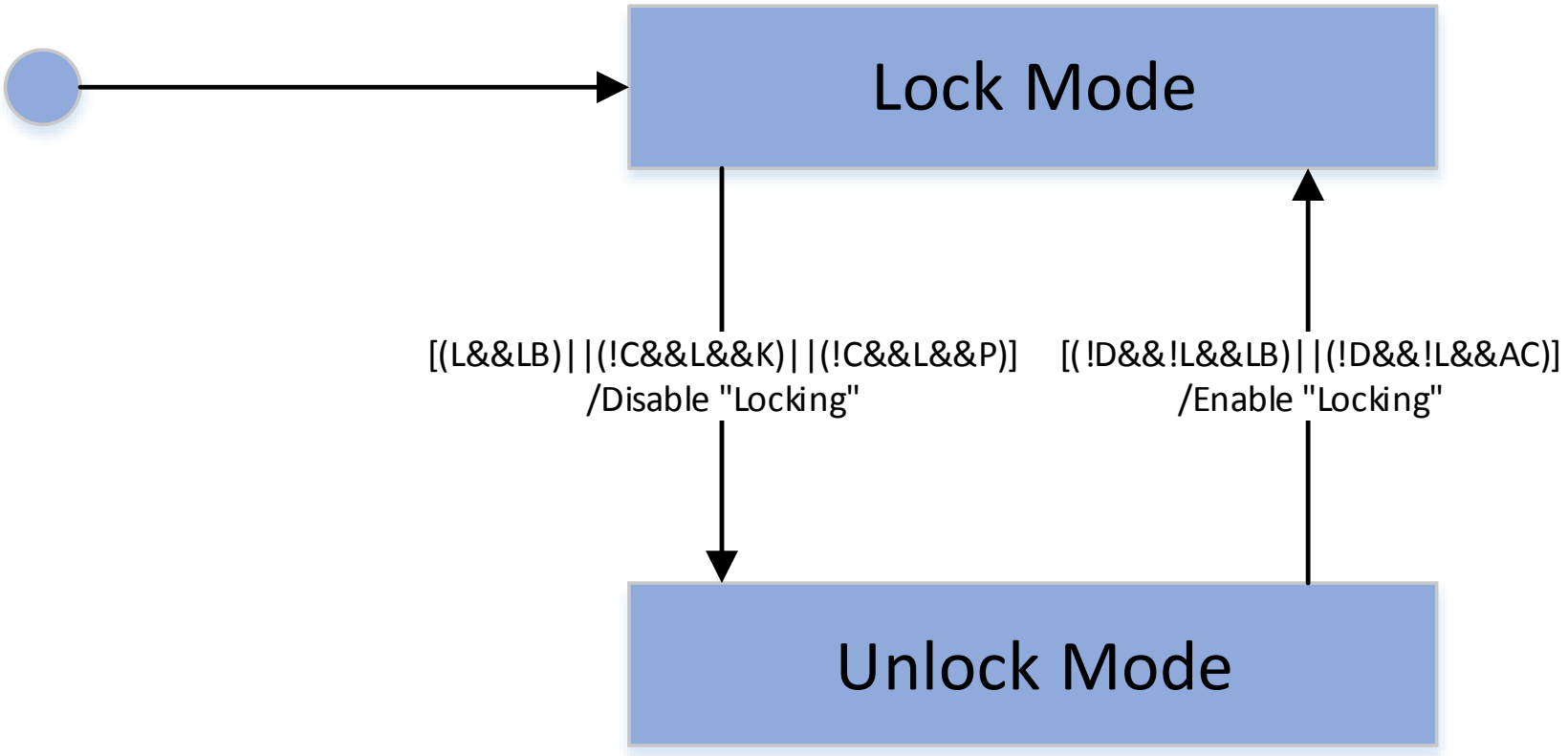
Unit Testing & False Position

suite_0 Lock Controller

EDLS_UTC_000_010	State=Unlock/ [D]==TRUE&&[L]==FALSE&&[AC]==TRUE		Enable/ Locking==1
EDLS_UTC_000_015	Unlock Mode에서 [D]==FALSE&&[L]==FALSE&&[AC]==FALSE Input		State=Unlock
EDLS_UTC_000_016	Unlock Mode에서 [D]==FALSE&&[L]==FALSE&&[LB]==FALSE Input		State=Unlock
EDLS_UTC_000_017	Unlock Mode에서 Tick==3 Input		Enable/ Locking==1

Unit Testing & False Position

suite_0 Lock Controller



Unit Testing & False Position

suite_1 Locking

EDLS_UTC_001_001	Disable	[L]==FALSE

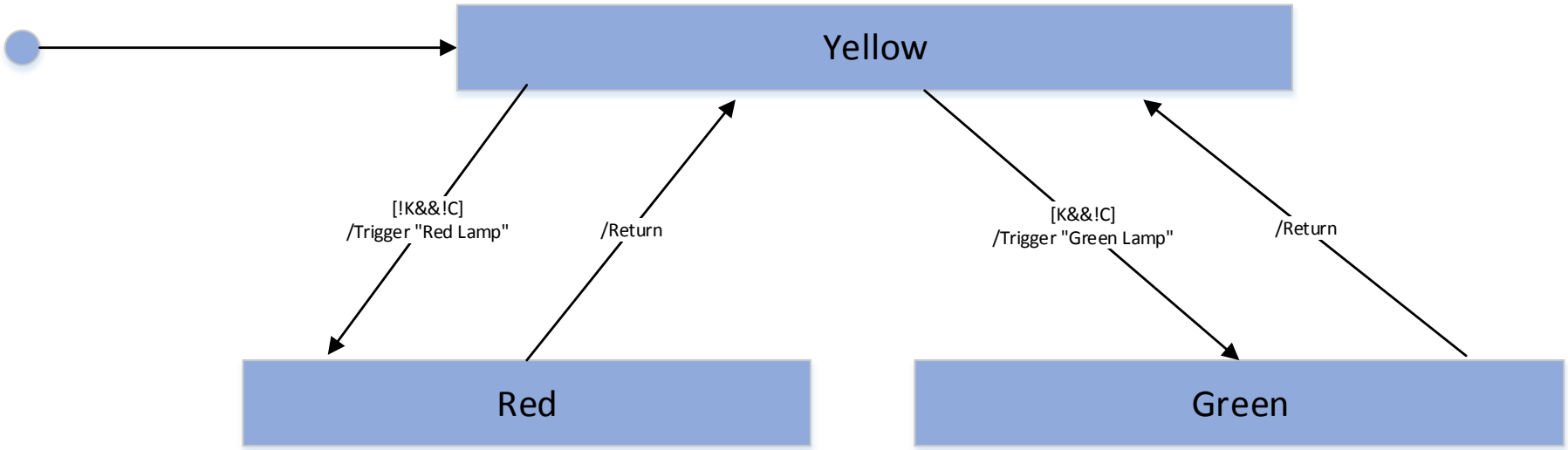
Unit Testing & False Position

suite_2 Display Controller

EDLS_UTC_002_000	State == Yellow / [C]==TRUE && [K]==TRUE	Enable / State == Yellow
EDLS_UTC_002_001	State == Yellow / [C]==TRUE && [K]==WAIT	Enable / State == Yellow
EDLS_UTC_002_002	State == Yellow / [C]==TRUE && [K]==FALSE	Enable / State == Yellow
EDLS_UTC_002_005	State == Yellow / [C]==FALSE && [K]==FALSE	Trigger / State == Red

Unit Testing & False Position

suite_2 Display Controller



Unit Testing & False Position

```
int display_controller(KEY_SENSOR* key, COVER_SENSOR* cover)
{
        if(cover->C==FALSE&&key->K==WRONG){
                yellow_color(DISABLE);
                red_color(TRIGGER);
                key ->K = WAIT ;
                return TRIGGER;
        }
        else if(cover->C==FALSE&&key->K==RIGHT){
                yellow_color(DISABLE);
                green_color(TRIGGER);
                return TRIGGER;
        }
        else if(cover->C == FALSE && key->K == WAIT){
                yellow_color(ENABLE);
                return ENABLE;
        }
    else return NONE;
}
```

Unit Testing & False Position

```
int display_controller(KEY_SENSOR* key, COVER_SENSOR* cover)
{
        if(cover->C==FALSE && key->K==WRONG){
                yellow_color(DISABLE);
                red_color(TRIGGER);
                key ->K = WAIT ;
                return TRIGGER;
        }
        else if(cover->C==FALSE && key->K==RIGHT){
                yellow_color(DISABLE);
                green_color(TRIGGER);
                return TRIGGER;
        }

        key->K = WAIT;
        cover->C=FALSE;
        yellow_color(ENABLE);
        return ENABLE;
}
```

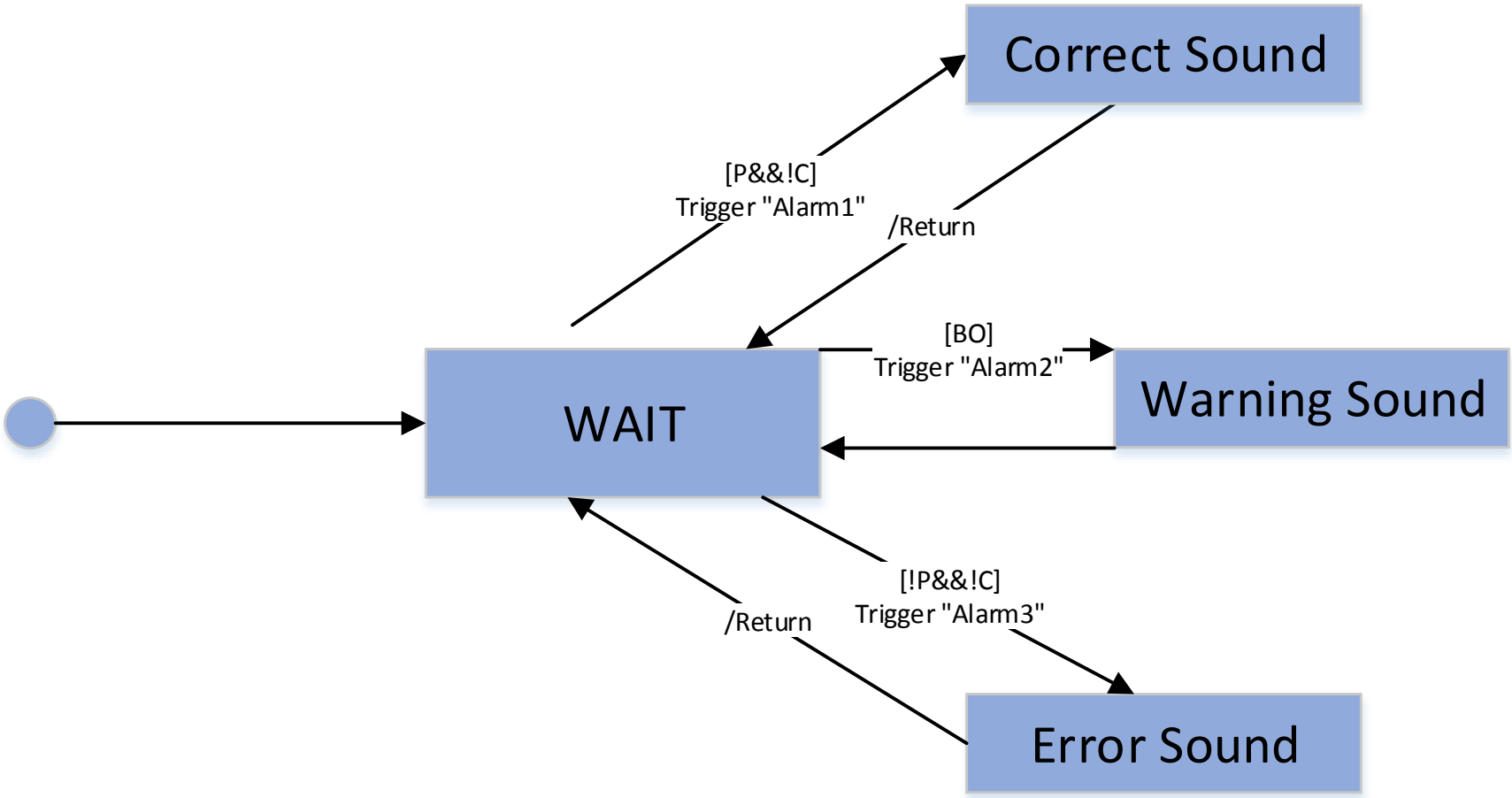
Unit Testing & False Position

suite_3 Alarm Controller

EDLS_UTC_003_006	State ==Wait / [BO] == TRUE && [P] == TRUE && [C] == FALSE	State == Wait

Unit Testing & False Position

suite_3 Alarm Controller



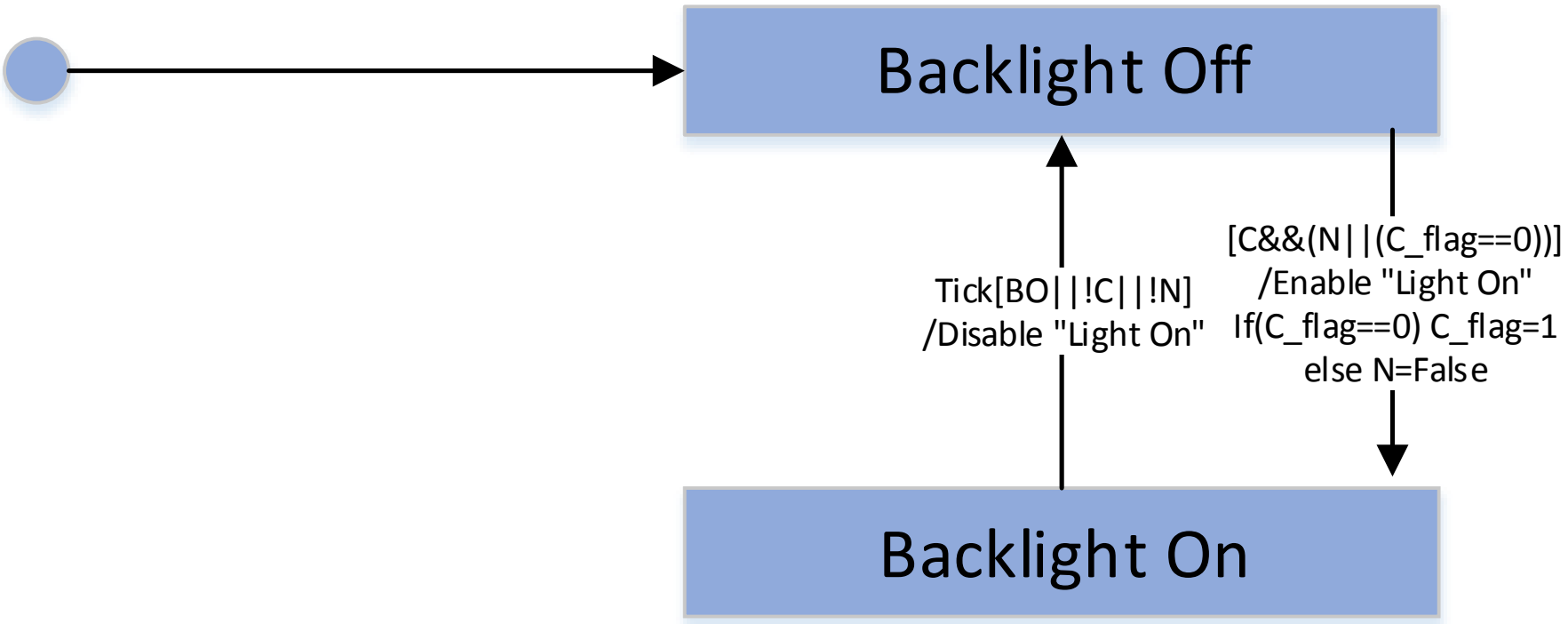
Unit Testing & False Position

suite_4 Backlight Controller

EDLS_UTC_004_005	State=Backlight Off/ [C]==FALSE	State=Backlight Off
EDLS_UTC_004_007	State=Backlight Off&&[C_flag]=1&& [N]==FALSE/ [C]==TRUE	Disable/ State=Backlight Off
EDLS_UTC_004_010	State=Backlight On&&[C]==TRUE / [C_flag]=1	Enable/ State=Backlight On

Unit Testing & False Position

suite_4 Backlight Controller



Unit Testing & False Position

void EDLS_UTC_004_005()

State=Backlight Off/ [C]==FALSE ➡ State=Backlight Off

```
KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

C_flag=NONE;
key->K=NONE;
cover->C=FALSE;
state->AC=NONE;
state->BO=FASLE;
state->INPUT_END=NONE;
state->L=NONE;
password->N=NONE;
password->P=NONE;
password->count=0;
password->input_password=NULL;
password->password=NULL;

CU_ASSERT_EQUAL(backlight_controller(state,password,cover,key,0),TRUE);
```

Unit Testing & False Position

void EDLS_UTC_004_005()

State=Backlight Off/ [C]==FALSE ➡ State=Backlight Off

```
KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

C_flag=NONE;
key->K=NONE;
cover->C=FALSE;
state->AC=NONE;
state->BO=TRUE;
state->INPUT_END=NONE;
state->L=NONE;
password->N=NONE;
password->P=NONE;
password->count=0;
password->input_password=NULL;
password->password=NULL;

CU_ASSERT_EQUAL(backlight_controller(state,password,cover,key,0),DISABLE);
```

Unit Testing & False Position

void EDLS_UTC_004_007()

State=Backlight Off&&[C_flag]=1&&[N]==FALSE/ [C]==TRUE

➡ Disable/State=Backlight Off

```
KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

C_flag=1;
key->K=NONE;
cover->C=TRUE;
state->AC=NONE;
state->BO=FALSE;
state->INPUT_END=NONE;
state->L=NONE;
password->N=FALSE;
password->P=NONE;
password->count=0;
password->input_password=NULL;
password->password=NULL;

CU_ASSERT_EQUAL(backlight_controller(state,password,cover,key,0),DISABLE);
```

Unit Testing & False Position

void EDLS_UTC_004_007()

State=Backlight Off&&[C_flag]=1&&[N]==FALSE/ [C]==TRUE

➡ Disable/State=Backlight Off

```
KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

C_flag=1;
key->K=NONE;
cover->C=TRUE;
state->AC=NONE;
state->BO=TRUE;
state->INPUT_END=NONE;
state->L=NONE;
password->N=FALSE;
password->P=NONE;
password->count=0;
password->input_password=NULL;
password->password=NULL;

CU_ASSERT_EQUAL(backlight_controller(state,password,cover,key,0),DISABLE);
```

Unit Testing & False Position

void EDLS_UTC_004_010()

State=Backlight On&&[C]==TRUE/ [C_flag]=1

➡ Enable/State=Backlight On

```
KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

C_flag=1;
key->K=NONE;
cover->C=TRUE;
state->AC=NONE;
state->BO=TRUE;
state->INPUT_END=NONE;
state->L=NONE;
password->N=NONE;
password->P=NONE;
password->count=0;
password->input_password=NULL;
password->password=NULL;

CU_ASSERT_EQUAL(backlight_controller(state,password,cover,key,0),ENABLE);
```

Unit Testing & False Position

void EDLS_UTC_004_010()

State=Backlight On&&[C]==TRUE/ [C_flag]=1

➡ Enable/State=Backlight On

```
KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

C_flag=1;
key->K=NONE;
cover->C=TRUE;
state->AC=NONE;
state->BO=FALSE;
state->INPUT_END=NONE;
state->L=NONE;
password->N=NONE;
password->P=NONE;
password->count=0;
password->input_password=NULL;
password->password=NULL;

CU_ASSERT_EQUAL(backlight_controller(state,password,cover,key,0),ENABLE);
```

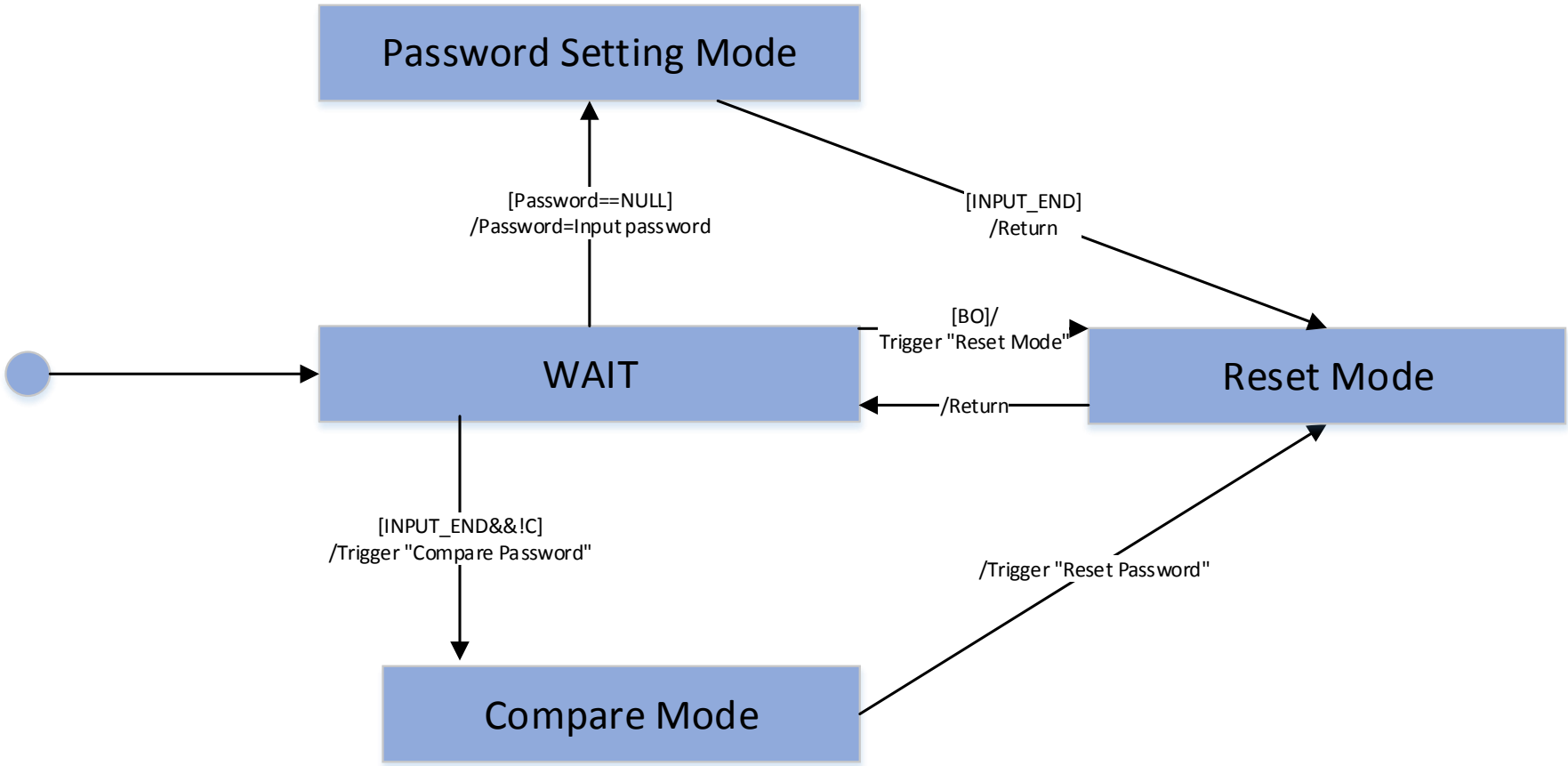
Unit Testing & False Position

suite_4 Password Controller

EDLS_UTC_012_002	State==wait/ [INPUT_END]==TRUE&& [C]==TRUE	Trigger/ State==Wait Mode
EDLS_UTC_012_004	State==Wait/ [INPUT_END]==TRUE&& [C]==TRUE&&[BO]==TRUE	Trigger/ State==Reset Mode
EDLS_UTC_012_006	State==Wait/ [L]==FALSE&& [C]==FALSE&&[INPUT_END]==TRUE	State==Wait Mode

Unit Testing & False Position

suite_4 Password Controller



Unit Testing & False Position

suite_4 Reset Password

EDLS_UTC_014_000	Trigger	INPUT_PASSWORD==NULL

Unit Testing & False Position

Type	Total	Ran	Passed	Failed	Inactive
suites	15	15	n/a	0	0
tests	75	75	75	0	0
asserts	76	76	0	0	n/a

Demonstration

QNA