

# Unit Test for Coffee Machine

Team Presentation #ver 3.0

Team [T3]

201211335 김윤식

201311281 송종원

201411317 조민규

201511282 이재승

2016/11/21 월

# INDEX

- References
- Features to be tested
- Features not to be tested
- System test design specification
- Test Identification
- Test case specification
- SRA & Code Structure

# 01 References

SRS for coffee machine system

[2016SE\_B][T3]SRA ver 5.0

[2016SE\_B][T3]SDA ver 4.0

# 02 Features to be tested

## SRS for Coffee Machine.pdf

### 2.2 Product functions (Statements of purpose)

Coffee machine은 커피 추출이 가능하다.

Coffee machine의 커피 추출은 조건에 따라 불가능 할 수 있다.

Coffee machine의 커피 추출은 사용자의 기호에 따라 세팅 할 수 있다.

Coffee machine은 예약이 가능하다.

Coffee machine은 커피가루가 없을 경우 원두를 자동으로 분쇄 후 커피를 추출 한다.

Coffee machine은 물과 커피의 잔량을 지속적으로 사용자에게 알려 준다.

Coffee machine의 물과 커피는 사용자에게 의해 충전된다.

Coffee machine은 machine의 내부를 청소 할 수 있다.

## 02 Features to be tested

1. Coffee machine은 커피 추출이 가능하다.
2. Coffee machine의 커피 추출은 조건에 따라 불가능 할 수 있다.
3. Coffee machine은 커피의 농도를 조절할 수 있다.
4. Coffee machine은 커피의 온도를 조절할 수 있다.
5. Coffee machine의 물과 커피는 사용자에게 의해 충전된다.
6. Coffee machine은 machine의 내부를 물 청소 할 수 있다.
7. Coffee machine은 예약이 가능하다.
8. Coffee machine은 예약 취소가 가능하다.
9. Coffee machine은 커피 추출 후 남아있는 찌꺼기를 청소할 수 있다.

# 03 Features not to be tested

SRS에서 요구한 사항 이외에 추가된 사항에 대해서는 X

# 04 System test design specification

< Table1 Testing 리스트 >

ID	Description
T3_CMS_STP_000	Coffee machine은 커피 추출이 가능하다.
T3_CMS_STP_001	Coffee machine의 커피 추출은 조건에 따라 불가능 할 수 있다.
T3_CMS_STP_002	Coffee machine은 커피의 농도를 조절할 수 있다.
T3_CMS_STP_003	Coffee machine은 커피의 온도를 조절할 수 있다.
T3_CMS_STP_004	Coffee machine의 물과 커피는 사용자에게 의해 충전된다.
T3_CMS_STP_005	Coffee machine은 machine의 내부를 물 청소 할 수 있다.
T3_CMS_STP_006	Coffee machine은 예약이 가능하다.
T3_CMS_STP_007	Coffee machine은 예약 취소가 가능하다.
T3_CMS_STP_008	Coffee machine은 커피 추출 후에 남아있는 커피 찌꺼기를 청소 할 수 있다.
T3_CMS_STP_009	Coffee machine은 사용자가 원할 때 전원을 끌 수 있다.

# 05 Test Identification

< Table2 Test Design Identification >

ID	Feature	Valid value
T3_CMS_STP_000	Coffee machine은 커피 추출이 가능하다.	커피를 추출할 경우 커피 생성
T3_CMS_STP_001	Coffee machine의 커피 추출은 조건에 따라 불가능 할 수 있다.	추출이 불가능할 경우 Beep음 발생
T3_CMS_STP_002	Coffee machine은 커피의 농도를 조절 할 수 있다.	추출 시에 필요한 물의 양 을 선택한 농도로 변경
T3_CMS_STP_003	Coffee machine은 커피의 온도를 조절 할 수 있다.	커피의 온도가 "냉"일 경우 "온"으로, "온"일 경우 "냉"으로 변경
T3_CMS_STP_004	Coffee machine의 물과 커피는 사용자에게 의해 충전된다.	입력 받은 양 만큼의 물, 원두, 커피 가루 충전 여부 확인
T3_CMS_STP_005	Coffee machine은 machine의 내부를 물 청소 할 수 있다.	요청 시 500ml의 물을 소모해 청소함
T3_CMS_STP_006	Coffee machine은 예약이 가능하다.	원하는 시간에 추출 또는 물 청소 가능
T3_CMS_STP_007	Coffee machine은 예약 취소가 가능하다.	예약 상태일 경우 예약 취소
T3_CMS_STP_008	Coffee machine은 커피 추출 후에 남아있는 커피 찌꺼기를 청소할 수 있다.	요청 시 남아있는 커피 찌꺼기 제거
T3_CMS_STP_009	Coffee machine은 사용자가 원할 때 전원을 끌 수 있다.	요청 시 CMS 종료



# 06 Test Case Specification

ID	Input specification	Output specification
T3_CMS_STP_000_000	커피 추출 가능한 조건에서 추출 요청	농도 만큼의 물과 원두 10g 소모되고 커피가 추출됨을 확인
T3_CMS_STP_001_000	설정해 놓은 농도보다 현재 물의 양이 적을 때 추출 요청	경고와 함께 Beep음 한번 발생을 확인
T3_CMS_STP_001_001	원두 잔량이 10g 미만일 때 추출 요청	경고와 함께 Beep음 세번 발생을 확인
T3_CMS_STP_001_002	남아있는 커피 찌꺼기가 있을 때 추출 요청	경고와 함께 Beep음 두번 발생을 확인
T3_CMS_STP_001_003	Waiting 상태가 아닐 때 추출 요청	추출 불가능
T3_CMS_STP_002_000	원하는 농도로 변경 요청	농도 값 변경과 디스플레이에 표시됨을 확인
T3_CMS_STP_002_001	표시된 값 이외의 값 입력	변경 사항 없음
T3_CMS_STP_003_000	"Ice" 상태일 때 변경 요청	"Hot"으로 변경됨을 확인
T3_CMS_STP_003_001	"Hot" 상태일 때 변경 요청	"Ice"로 변경됨을 확인
T3_CMS_STP_004_000	1000ml를 초과하지 않는 물 충전 요청	현재 물 양에 충전 요청한 물 양을 더한 값으로 변경됨을 확인
T3_CMS_STP_004_001	100g을 초과하지 않는 원두 충전 요청	현재 원두 양에 충전 요청한 원두 양을 더한 값으로 변경됨을 확인
T3_CMS_STP_004_002	커피 가루 유무가 X일 때 충전 요청	커피 가루 유무가 X에서 O로 변경됨을 확인
T3_CMS_STP_004_003	1000ml를 초과하는 물 충전 요청	변경 사항 없음

# 06 Test Case Specification

T3_CMS_STP_004_004	100g을 초과하는 원두 충전 요청	변경 사항 없음
T3_CMS_STP_004_005	커피 가루 유무가 0일 때 충전 요청	변경 사항 없음
T3_CMS_STP_005_000	물의 양이 500ml 이상일 때 물 청소 요청	물의 양이 500ml 줄고, 현재 상태가 Cleaning으로 변경됨을 확인
T3_CMS_STP_005_001	물의 양이 500ml 미만일 때 물 청소 요청	경고와 함께 Beep음 한 번 발생을 확인
T3_CMS_STP_006_000	커피 추출이 가능한 조건에서 커피 추출 예약 요청	입력한 시간이 될 때 까지 상태는 Reserved, 예약 취소 버튼 외에는 다른 입력 받지 못하고, 입력한 시간이 되면 커피 추출
T3_CMS_STP_006_001	물 청소가 가능한 조건에서 물 청소 예약 요청	입력한 시간이 될 때 까지 상태는 Reserved, 예약 취소 버튼 외에는 다른 입력 받지 못하고, 입력한 시간이 되면 물 청소 실행
T3_CMS_STP_006_002	커피 추출이 불가능한 조건에서 커피 추출 예약 요청	예약 불가능
T3_CMS_STP_006_003	물 청소가 불가능한 조건에서 물 청소 예약 요청	예약 불가능

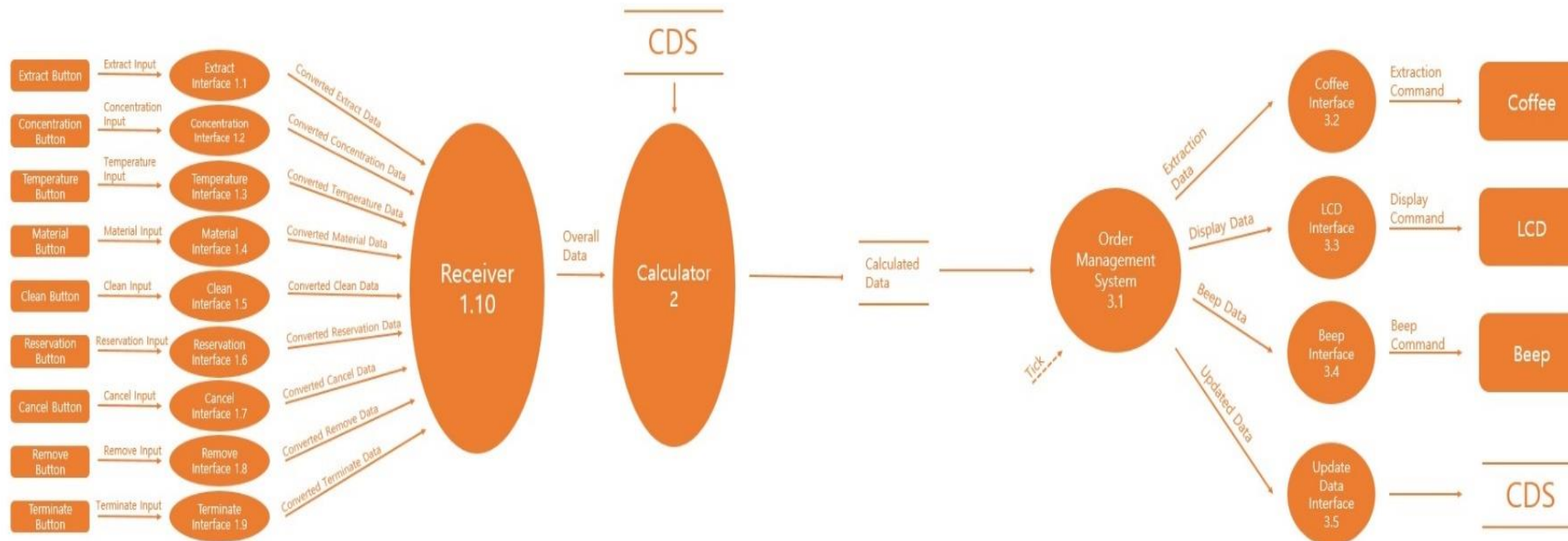
# 06 Test Case Specification

T3_CMS_STP_007_000+	현재 상태가 Reserved일 때 예약 취소 요청	예약이 취소됨을 확인
T3_CMS_STP_007_001+	현재 상태가 Reserved가 아닐 때 예약 취소 요청	변경 사항 없음
T3_CMS_STP_008_000+	가루 청소 필요 유무가 O일 때 가루 청소 요청	가루 청소 필요 유무가 X로 변경됨을 확인
T3_CMS_STP_008_001+	가루 청소 필요 유무가 X일 때 가루 청소 요청	변경 사항 없음
T3_CMS_STP_009_000+	Waiting 상태 일 때 전원 종료 요청	전원이 종료됨을 확인
T3_CMS_STP_009_001+	Waiting 상태가 아닐 때 전원 종료 요청	변경 사항 없음

**총 27개의 Test Case**

**결과는 모두 Ok**

# 07 SRA & Code Structure

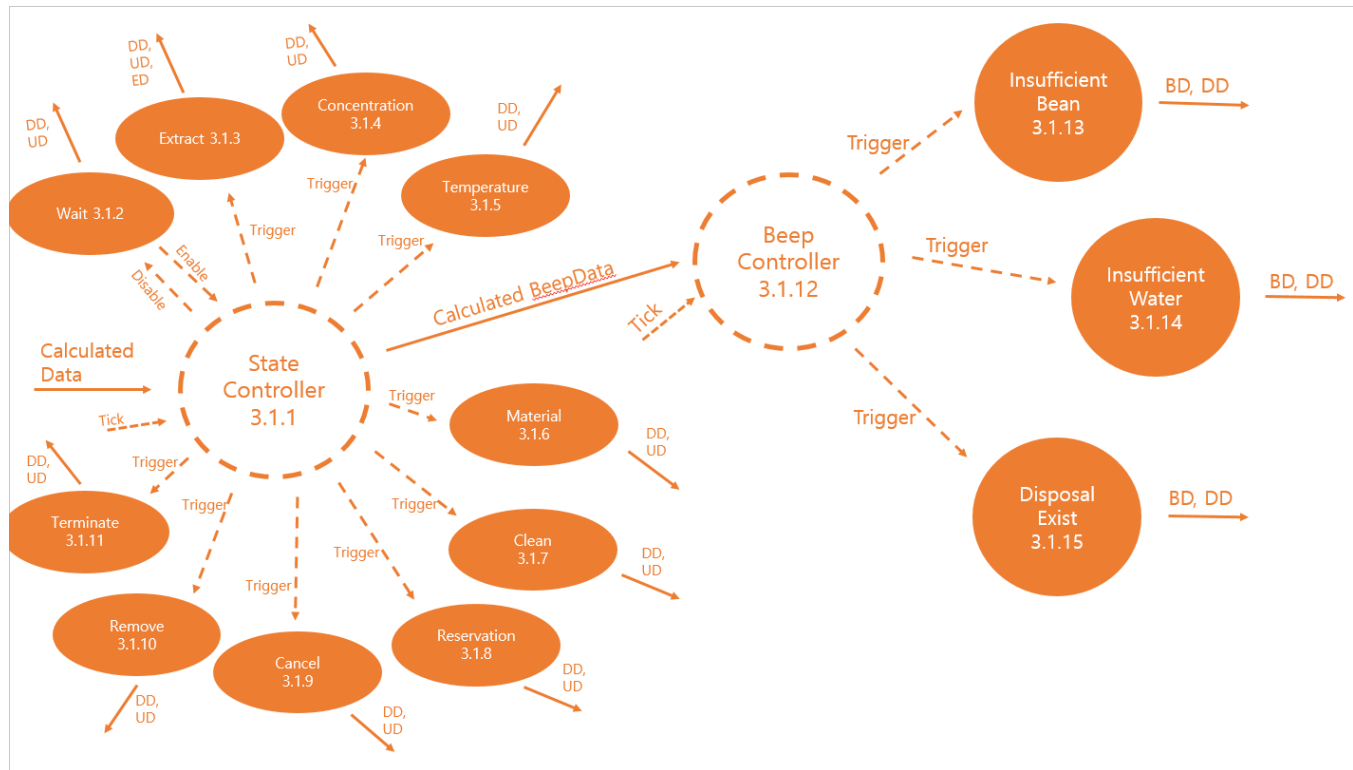


**Button Interface**

**Receiver Calculator**

**Controller**

# 07 SRA & Code Structure



**State Controller**

**Beep Controller**

# 07 SRA & Code Structure

## Main 함수

```
int choice = 0;
if (!kbhit()) {
    overallData = (overall_data *)malloc(sizeof(overall_data));
    syncOverallData(overallData, currentData);
}
else {
    choice = 0;
    scanf("%d", &choice);
    system("clear");
    if (choice == 1){
        BOOLEAN CED = extractInterface();
        overallData = receiver(CED, 0, FALSE, NULL, FALSE, NULL, FALSE, FALSE, FALSE);
    }
    else{
        .....생략
    }
}
```

버튼입력 ->

인터페이스->

Receiver ->

Calculator->

StateController->

BeepController->

```
calculatedData = calculator(overallData, currentData);
calculatedBeepData = stateController(calculatedData, currentData);
beepController(calculatedBeepData);
```

# 07 SRA & Code Structure

## Receiver

```
overall_data* receiver(BOOLEAN CED, int CCD, BOOLEAN CTD, material_data* CMD, BOOLEAN CLD, reservation_data* CRD, BOOLEAN CAD, BOOLEAN COD, BOOLEAN CND) {  
  
    overall_data *overallData = (overall_data *)malloc(sizeof(overall_data));  
    initOverallData(overallData);  
  
    if (!CED==FALSE) {  
        overallData->convertedExtractData = CED;  
        overallData->select = 1;  
    }  
    else {  
        .....생략  
    }  
    fflush(stdin);  
    return overallData;  
}
```

**9개의 파라미터를 받아서 OverallData 생성**

# 07 SRA & Code Structure

## Calculator

```
data_storage* calculator(overall_data *overallData, data_storage *currentData) {  
  
    data_storage *calculatedData = (data_storage *)malloc(sizeof(data_storage));  
    initData(calculatedData);  
    int over_select = overallData->select;  
    BOOLEAN cur_wait = currentData->wait;  
  
    if (over_select == -1 && (cur_wait)) {  
        calculatedData->wait = currentData->wait;  
    }  
    else {  
        .....나머지 생략  
    }  
  
    calculatedData->mode = overallData->select;  
    fflush(stdin);  
    return calculatedData;  
}
```

**2개의 구조체를 받아와서 계산한 후 calculatedData 반환**



# 07 SRA & Code Structure

## StateController

```
beep_data* stateController(data_storage *calculatedData, data_storage *currentData) {
    BOOLEAN cur_wait = currentData->wait;
    BOOLEAN cal_wait = calculatedData->wait;
    int cal_mode = calculatedData->mode;
    int cal_bmode = calculatedData->b_mode;
    beep_data *calculatedBeepData = (beep_data*)malloc(sizeof(beep_data));
    initBeepData(calculatedBeepData);

    if ((cal_mode == -1 || cal_mode == 0) && (cur_wait)) {
        currentData->mode = calculatedData->mode;
        calculatedData->wait = currentData->wait;
        wait(calculatedData, currentData);
    }
    else{
        .....나머지 생략
    }
    fflush(stdin);
    return calculatedBeepData;
}
```

**받은 데이터로 특정 프로세스를 실행시킴**

# 07 SRA & Code Structure

## BeepController

```
void beepController(beep_data* calculatedBeepData) {  
    int beepData = calculatedBeepData->b_mode;  
  
    if (beepData == 1) {  
        insufficientWater(calculatedBeepData);  
    }  
    else if (beepData == 2) {  
        disposalExist(calculatedBeepData);  
    }  
    else if (beepData == 3) {  
        insufficientBean(calculatedBeepData);  
    }  
}
```

**StateController에서 나온 beepData를 이용한 BeepController**

**Thank you**