Unit Testing Plan

for Coffee Machine System

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

Project Team
Team 5

Date
2016-11-07

Team Information

201111389 조정원

201210194 김정환

201211352 백대현

201211390 허윤제

Table of Contents

1. Introduction 4

1.1. Objectives 4

1.2. Background 4

1.3. Scope 4

1.4. Project plan 4

1.5. Configuration management plan 4

1.6. References 4

2. Test items 5

3. Features to be tested 5

4. Features not to be tested 10

5. Approach 12

6. Item pass/fail criteria 13

7. Unit test design specification 13

7.1. Test design specification identifier 13

7.2. Features to be tested 13

1. Process in SRA 13

7.3. Approach refinements 13

7.4. Test identification 13

7.5. Feature pass/fail criteria 25

8. Unit test case specification 25

8.1. Test case specification identifier 25

8.2. Test items 51

8.3. Input specifications 51

8.4. Output specifications 51

9. Testing tasks 51

10. Environmental needs 51

11. Unit Test deliverables 51

12. Schedules 51

1. Introduction

1.1. Objectives

본 문서는 Team 5 에서 작성한 Software Requirement Analysis for Coffee Machine System(SRA)에 의거하여 개발된 Coffee Machine System 에 대하여 Unit test 를 수행하기 위한 계획을 제시한다. 테스트 케이스는 SRA 로부터 도출한다.

1.2. Background

Coffee Machne System 은 사용자의 기호에 따라 농도와 온도 설정(냉/온)이 다른 커피를 추출한다. 또한 이 시스템은 작업 수행 중 비정상 종료 시 이전 상태를 기억한다. 시스템 재가동 시 이전 상태를 복원하여 커피 추출을 재개할 수 있다.

1.3. Scope

Team 5 에서 개발한 CMS 를 대상으로 한다.

1.4. Project plan

1.5. Configuration management plan

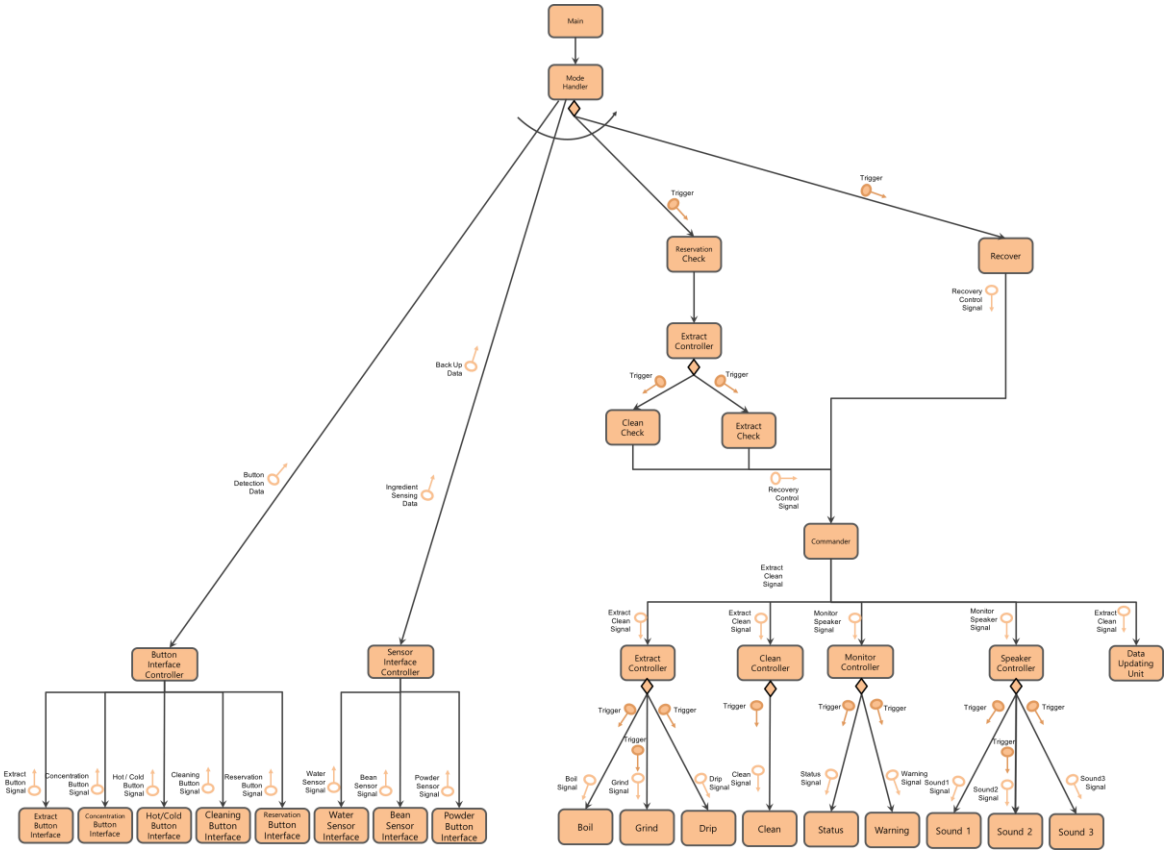
1.6. References

DS-2016SE-CM-SRS-1.0

DS-2016SE-CM-SRA-2.3

DS-2016SE-CM-SDS-1.5

2. Test items



테스트 대상은 SRA 에 명세된 CMS 에서 선별한 18 개의 모듈이다.

3. Features to be tested

(1) Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는지를 test 한다.

(2) Modules in SDS : 각 모듈이 가지고 있는 데이터 인터페이스를

test 한다. <Table 1 테스트할 Process(DFD) 리스트> 의 Process name 참조

<Table 1 테스트할 Process(DFD) 리스트>

ID	Name	Description
1.1	Extract Button Interface	추출버튼 입력을 확인한 후 Button Interface Controller 가 사용할 수 있는 디지털 신호로 바꾼다.
1.2	Concentration Button Interface	농도버튼 입력을 확인한 후 Button Interface Controller 가 사용할 수 있는 디지털 신호로 바꾼다.
1.3	Hot/Cold Button Interface	입력받은 Hot/Cold Button Input 을 Hot/Cold Button Signal 로 변환하여 Button Interface Controller 에 보낸다.
1.4	Cleaning Button Interface	청소버튼 입력을 확인한 후 Button Interface Controller 가 사용할 수 있는 디지털 신호로 바꾼다.
1.6	Button Interface Controller	Button Interface Controller 가 받은 Button Signal 을 CM System 이 사용할 수 있는 Button Data 로 바꾼다. 커피 머신이 대기상태가 아닐 때는 Input 을 받지 않는다.
2.4	Sensor Interface Controller	Sensor Interface Controller 가 받은 Sensor Signal 을 CM System 이 사용할 수 있는 Sensor Data 로 바꾼다.
3.1.1	Mode Handler	Button Detection Data, Ingredient Sensing Data, Back Up Data 을 받아와서 복원 필요 유무에 따라 Recover 혹은 Reservation Check 를 트리거한다.

ID	Name	Description
3.1.2	Reservation Check	트리거를 받으면 Button Detection Data & Ingredient Sensing Data 을 읽고 현재 시간이 예약 시간 인지 확인한 후 Execute Info 를 Execute Controller 에 보낸다.
3.1.3	Recover	트리거를 받으면 Button Detection Data & Ingredient Sensing Data 를, Back Up Data 를 읽고 Recovery Control Data 를 내보낸다.
3.1.4	Execute Controller	Execute Info 의 명령정보를 받아서 청소명령일 경우 Clean Check 에 물의 양을 전송하고 Trigger 한다. 추출명령일 경우 Extract Check 에 각각 재료의 양을 전송하고 Trigger 한다.
3.1.5	Clean Check	청소 동작을 수행할 수 있는지 커피 머신이 보유한 물의 잔량을 검사한다. 만약 청소 명령이 존재한다면 Button Detection Data & Ingredient Sensing Data 에서 커피 머신이 보유한 물 보유량을 읽고 청소 가능 여부와 경보음 발생 여부를 Normal Control & Warning Data 에 저장한다.
3.1.6	Extract Check	커피 추출 동작을 수행할 수 있는지 커피 머신이 보유한 재료의 잔량을 검사한다. 만약 커피 추출 명령이 존재한다면 Button Detection Data & Ingredient Sensing Data 에서 커피 머신이 보유한 재료의 양을 읽고 추출 가능 여부와 경보음 발생 여부를 Normal Control & Warning Data 에 저장한다. 만약 커피 가루가 존재하는지를 확인하여

ID	Name	Description
		존재할 시에 추출이 불가능하고, 경보음 발생 여부를 저장한다.
3.1.7	Commander	만약 복원모드 라면 Recovery Control Data 를 가지고 Extract & Clean Command(추출이라면 어느 과정에서 어디까지 진행이 되었는지, 청소라면 어느 정도 청소가 진행되었는지)를 Motor Controller 에게 전송하고, Monitor & Sound Command 를 Display Controller 에게 전송한다. 만약 복원모드가 아니고 Normal Control & Warning Data 에 저장된 명령 정보(커피 추출/청소)를 읽고 추출 명령 정보 혹은 청소 명령 정보를 Motor Controller 에게 전송한다. Normal Control & Warning Data, Ingredient Sensing Data 에 저장된 경보음 발생 여부와 화면에 출력할 정보를 Display Controller 에게 전송한다. 화면에 출력할 정보는 다음과 같다. [현재 시간, 현재 상태, 선택 농도, 선택 온도, 물 잔량, 커피 잔량(원두), 가루 커피 유무, 커피 가루 정소 필요, 커피 예약 시간, 청소 예약 시간]

ID	Name	Description
3.2.1.3	Data Updating Unit	Extract & Clean Command 의 명령 정보를 받으면 매초마다 Ingredient Sensing Data 에서 해당 상황에 맞는 재료의 양(물, 원두)을 차감한다. Extract & Clean Command 의 명령 정보를 가지고 청소, 추출 시작할 때 Ingredient Sensing Data 에서 현재 재료 잔량(물, 원두, 커피가루)을 Back Up Data 에 저장한다. 또한 청소와 추출 단계(가열, 분쇄, 드립)가 시작될 때와 종료될 때 Back Up Data 의 시작 플래그(추출 시작, 가열 시작, 분쇄 시작, 드립 시작, 청소 시작)를 1 과 0 으로 변경 저장한다.
3.3.1.1	Monitor Controller	Monitor & Speaker Command 를 읽고 Status 에 Status Info 를 보내며 재료 부족 또는 커피 가루 제거가 필요할 때 Warning 을 트리거한다.
3.3.1.2	Speaker Controller	Monitor & Speaker Command 에 담긴 경보음 발 생 여부 값과 화면에 출력할 정보를 받는다.
3.3.2.1	Status	Monitor Controller 로부터 화면에 출력할 정보를 받는다. 받은 정보를 Status Signal 을 보내 화면에 출력하도록 한다. 화면에 출력할 정보는 다음과 같다. [현재 시간, 현재 상태, 선택 농도, 선택 온도, 물 잔량, 커피 잔량(원두), 가루 커피 유무, 커피 가 루 정소 필요,

ID	Name	Description
		커피 예약 시간, 청소 예약 시간]
3.3.2.2	Warning	Monitor Controller 부터 화면에 출력할 경고를 받 아 Warning Signal 을 내보내 화면에 출력하도록 한다. 오직 다음의 경우에만 알람을 받는다. [물부족, 원두부족, 커피가루제거필요]

4. Features not to be tested

(1) Process in SRA : 외부 장치 드라이버, 단순 데이터 전달 프로세스 등은 test 에서 제외한다.

(2) Modules in SDS : <Table 2 테스트 하지 않을 Process(DFD) 리스트>의 Process name 참조

<Table 2 테스트하지 않을 Process(DFD) 리스트>

ID	Name	Description
1.5	Reservation Button Interface	입력받은 Reservation Button Input 을 Reservation Button Signal 로 변환하여 Button Interface Controller 에 보낸다.
2.1	Water Sensor Interface	Sensor 가 감지한 물의 양을 Sensor Interface Controller 가 사용할 수 있는 Signal 로 변환한다.
2.2	Bean Sensor Interface	Sensor 가 감지한 원두의 양을 Sensor Interface Controller 가 사용할 수 있는 Signal 로 변환한다.
2.3	Powder Sensor Interface	Powder Sensor Input 을 받아서 Powder Sensor Signal 로 변환하여 Sensor Interface Controller 에 보낸다.
3.2.1.2	Clean Controller	Extract & Clean Command 의 청소 명령 여부를 판단하여 Trigger Clean 을 내보내 청소하도록 한다.
3.2.2	Extract Controller	Extract Command 를 받으면 Extract Signal 을 내보낸다. Extract 가 수행되는 동안 Back Up Data 에 커피 추출 단계(커피 가루 추출 여부/가열 완료 여부/커피 추출 완료 여부)를 갱신한다.
3.2.2.1	Boil	Trigger 를 받으면 Boil Signal 을 내보내 물을 끓이도록 한다.
3.2.2.2	Grind	Trigger 를 받으면 Grind Signal 을 내보내 원두를 분쇄하도록 한다.
3.2.2.3	Drip	Trigger 를 받으면 Drip Signal 을 내보내 커피를 추출하도록 한다.

ID	Name	Description
3.2.3	Clean	Clean Controller 의 trig-ger 를 받아 Clean Signal 을 내보낸다.
3.3.3.1	Sound1	Sound1 에 해당하는 소리를 내는 하드웨어가 인 식할수 있는 Signal 을 보내 Sound1 을 출력하도록 한다. (짧게 한번 소리를 낸다.)
3.3.3.2	Sound2	Sound2 에 해당하는 소리를 내는 하드웨어가 인 식할수 있는 Signal 을 보내 Sound2 을 출력하도록 한다. (길게 3 번 소리를 낸다.)
3.3.3.3	Sound3	Sound3 에 해당하는 소리를 내는 하드웨어가 인 식할수 있는 Signal 을 보내 Sound3 을 출력하도록 한다. (짧게 2 번 소리를 낸다.)

5. Approach

각 모듈이 요구사항을 만족하는지를 확인할 수 있는 test case 를 작성한다. 그 밖의 예외사항은 test 하지 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

CMS.UTC.0000.000 (시스템이름.테스트 케이스.프로세스 참조 번호.케이스넘버)

7.2. Features to be tested

1. Process in SRA

<Table 1 테스트할 Process(DFD) 리스트> 참조

7.3. Approach refinements

7.4. Test identification

Identifier	Feature	Valid/Invalid value
CMS.UTC.1100.001	1.1 Extract Button Interface	추출버튼 입력
CMS.UTC.1200.001	1.2 Concentration Button Interface	Volume 이 300 일 때 농도버튼 입력이 들어온다.
CMS.UTC.1200.002	1.2 Concentration Button Interface	Volume 이 100 일 때 농도버튼 입력이 들어온다.
CMS.UTC.1200.003	1.2 Concentration Button Interface	Volume 이 200 일 때 농도버튼 입력이 들어온다.
CMS.UTC.1300.001	1.3 Hot/Cold Button Interface	temperature == 0 일 때 Hot/Cold Button Input 이 들어온다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.1300.002	1.3 Hot/Cold Button Interface	temperature == 1 일 때 Hot/Cold Button Input 이 들어온다.
CMS.UTC.1400.001	1.4 Cleaning Button Interface	청소버튼 입력
CMS.UTC.1600.001	1.6 Button Interface Controller	ButtonDetectionData.txt 파일이 없고 forcedTermination_flag=0 이다.
CMS.UTC.1600.002	1.6 Button Interface Controller	ButtonDetectionData.txt 파일이 없고 forcedTermination_flag=1 이다.
CMS.UTC.1600.003	1.6 Button Interface Controller	ButtonDetectionData.txt 파일이 있고 forcedTermina-tion_flag=0 이다.
CMS.UTC.1600.004	1.6 Button Interface Controller	ButtonDetectionData.txt 파일이 있고 forcedTermina-tion_flag=1 이다.
CMS.UTC.2400.001	2.4 Sensor Interface Controller	IngredientSensingData.txt 가 있다.
CMS.UTC.2400.002	2.4 Sensor Interface Controller	IngredientSensingData.txt 가 없다. 물,원두,가루의 양을 각각 100, 10, 10 으로 초기화해준다.
CMS.UTC.3110.001	3.1.1 Mode Handler	CMS 을 부팅한 상태에서 termi-nated == 1, extract_start_flag == 1 이고 clean_flag ==0 이다.
CMS.UTC.3110.002	3.1.1 Mode Handler	CMS 이 작동 중인 상태에서 ter-minated == 1, ex-tract_start_flag ==0 이고 clean_flag ==0 이다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3110.003	3.1.1 Mode Handler	CMS 이 작동 중인 상태에서 terminated == 0, extract_start_flag ==1 이고 clean_flag ==0 이다.
CMS.UTC.3110.004	3.1.1 Mode Handler	CMS 이 작동 중인 상태에서 terminated == 0, extract_start_flag == 0 이고 clean_flag ==1 이다.
CMS.UTC.3110.005	3.1.1 Mode Handler	CMS 이 작동 중인 상태에서 terminated == 0, extract_start_flag == 1 이고 clean_flag == 1 이다.
CMS.UTC.3110.006	3.1.1 Mode Handler	CMS 이 작동 중인 상태에서 terminated == 0, extract_start_flag ==0 이고 clean_flag == 0 이다.
CMS.UTC.3120.001	3.1.2 Reservation Check	예약 상태가 NONE 이고 current_time 이 11:11 일 때 Trigger 가 들어온다.
CMS.UTC.3120.002	3.1.2 Reservation Check	예약 상태가 NONE 이지만 reserved_clean_complete 값이 1 이고 current_time 이 11:11 일 때 Trigger 가 들어온다.
CMS.UTC.3120.003	3.1.2 Reservation Check	예약 상태가 NONE 이지만 reserved_extract_complete 값이 1 이고 current_time 이 11:11 일 때 Trigger 가 들어온다.
CMS.UTC.3120.004	3.1.2 Reservation Check	예약 상태가 CLEAN 이고 reserved_time 이 12:12 이며 current_time 이 11:11 일 때 Trigger 가 들어온다.
CMS.UTC.3120.005	3.1.2 Reservation Check	예약 상태가 EXTRACT 이고 reserved_time 이 12:12 이며 current_time 이 11:11 일 때 Trigger 가 들어온다.
CMS.UTC.3120.006	3.1.2 Reservation Check	예약 상태가 BOTH 이고 reserved_clean_time 과 reserved_extract_time 이

Identifier	Feature	Valid/Invalid value
		4:12 이며 current_time 이 11:11 일 때 Trigger 가 들어온다.
CMS.UTC.3120.007	3.1.2 Reservation Check	예약 상태가 EXTRACT 이고 reserved_time 과 current_time 이 6:20 일 때 Trigger 가 들어온다.
CMS.UTC.3120.008	3.1.2 Reservation Check	예약 상태가 EXTRACT 이지만 reserved_extract_complete 값이 1 이고 reserved_time 과 current_time 이 4:46 일 때 Trigger 가 들어온다.
CMS.UTC.3120.009	3.1.2 Reservation Check	예약 상태가 CLEAN 이고 reserved_time 과 current_time 이 12:11 일 때 Trigger 가 들어온다.
CMS.UTC.3120.010	3.1.2 Reservation Check	예약 상태가 CLEAN 이고 reserved_clean_complete 값이 1 이고 reserved_time 과 current_time 이 2:1 일 때 Trigger 가 들어온다.
CMS.UTC.3120.011	3.1.2 Reservation Check	예약 상태가 BOTH 이고 reserved_clean_time, reserved_extract_time 그리고 current_time 이 10:5 일 때 Trigger 가 들어온다.
CMS.UTC.3120.012	3.1.2 Reservation Check	예약 상태가 BOTH 이지만 reserved_clean_complete 값이 1 이며 reserved_clean_time, reserved_extract_time 그리고 current_time 이 9:4 일 때 Trigger 가 들어온다.
CMS.UTC.3130.001	3.1.3 Recover	command_type == -1, temperature == -1, volume == -1 일 때 Trigger 가 들어온다.
CMS.UTC.3130.002	3.1.3 Recover	command_type == -1, temperature == COLD, volume == 100 일 때 Trigger 가 들어온다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3130.003	3.1.3 Recover	command_type == COMMAND_CLEAN, temperature == -1, volume == -1 일 때 Trigger 가 들어온다.
CMS.UTC.3130.004	3.1.3 Recover	command_type == COMMAND_CLEAN, temperature == -1, volume == -1 일 때 Trigger 가 들어온다.
CMS.UTC.3130.005	3.1.3 Recover	command_type == COMMAND_RESERVATION_CLEAN, temperature == -1, volume == -1 일 때 Trigger 가 들어온다.
CMS.UTC.3130.006	3.1.3 Recover	command_type == COMMAND_RESERVATION_CLEAN, temperature == HOT, volume == 200 일 때 Trigger 가 들어온다.
CMS.UTC.3130.007	3.1.3 Recover	command_type == COMMAND_EXTRACT, temperature == HOT, volume == 100 일 때 Trigger 가 들어온다.
CMS.UTC.3130.008	3.1.3 Recover	command_type == COMMAND_EXTRACT, temperature == HOT, volume == 200 일 때 Trigger 가 들어온다.
CMS.UTC.3130.009	3.1.3 Recover	command_type == COMMAND_EXTRACT, temperature == HOT, volume == 300 일 때 Trigger 가 들어온다.
CMS.UTC.3130.010	3.1.3 Recover	command_type == COMMAND_EXTRACT, temperature == COLD, volume == 100 일 때 Trigger 가 들어온다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3130.011	3.1.3 Recover	command_type == COMMAND_EXTRACT, temperature == COLD, volume == 200 일 때 Trigger 가 들어온다.
CMS.UTC.3130.012	3.1.3 Recover	command_type == COMMAND_EXTRACT, temperature == COLD, volume == 300 일 때 Trigger 가 들어온다.
CMS.UTC.3130.013	3.1.3 Recover	command_type == COMMAND_RESERVATION_EXTRACT, temperature == HOT, volume == 100 일 때 Trigger 가 들어온다.
CMS.UTC.3130.014	3.1.3 Recover	command_type == COMMAND_RESERVATION_EXTRACT, temperature == HOT, volume == 200 일 때 Trigger 가 들어온다.
CMS.UTC.3130.015	3.1.3 Recover	command_type == COMMAND_RESERVATION_EXTRACT, temperature == HOT, volume == 300 일 때 Trigger 가 들어온다.
CMS.UTC.3130.016	3.1.3 Recover	command_type == COMMAND_RESERVATION_EXTRACT, temperature == COLD, volume == 100 일 때 Trigger 가 들어온다.
CMS.UTC.3130.017	3.1.3 Recover	command_type == COMMAND_RESERVATION_EXTRACT, temperature == COLD, volume == 200 일 때 Trigger 가 들어온다.
CMS.UTC.3130.018	3.1.3 Recover	command_type == COMMAND_RESERVATION_EXTRACT, temperature == COLD, volume == 300 일 때 Trigger 가 들어온다.
CMS.UTC.3140.001	3.1.4 Execute Controller	Command 가 COMMAND_CLEAN 이다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3140.002	3.1.4 Execute Controller	Command 가 COMMAND_RESERVATION_CLEAN 이다.
CMS.UTC.3140.003	3.1.4 Execute Controller	Command 가 COMMAND_EXTRACT 이다.
CMS.UTC.3140.004	3.1.4 Execute Controller	Command 가 COMMAND_RESERVATION_EXTRACT 이다.
CMS.UTC.3150.001	3.1.5 Clean Check	물의 양이 500 미만이다.
CMS.UTC.3150.002	3.1.5 Clean Check	물의 양이 500 이상이다.
CMS.UTC.3160.001	3.1.6 Extract Check	Ingredient Sensing Data 의 커피가루가 사용한 커피가루이다.
CMS.UTC.3160.002	3.1.6 Extract Check	Ingredient Sensing Data 의 물의 양이 설정한 농도값보다 작다.
CMS.UTC.3160.003	3.1.6 Extract Check	Ingredient Sensing Data 의 원두의 양이 10g 보다 작고 커피가루가 없다.
CMS.UTC.3160.004	3.1.6 Extract Check	Ingredient Sensing Data 의 물의 양이 설정한 농도보다 많고 원두의 양이 10g 보다 많거나 커피가루가 새것일 경우 설정한 온도가 HOT 이다.
CMS.UTC.3170.001	3.1.7 Commander	Warning 이 있다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3170.002	3.1.7 Commander	Warning 이 없다.
CMS.UTC.3213.001	3.2.1.3 Data Updating Unit	Command 가 COMMAND_CLEAN 이거나 COMMAND_RESERVATION_CLEAN 이고 Control 이 CLEAN 이다.
CMS.UTC.3213.002	3.2.1.3 Data Updating Unit	Command 가 COMMAND_EXTRACT 이거나 COMMAND_RESERVATION_EXTRACT 이고 Control 이 HOT_100 이거나 HOT_200 이거나 HOT_300 이고 Back Up Data 의 extract_start_flag 가 1 이고 boil_flag 가 1 이고 boiling 이 10 이 아니다.
CMS.UTC.3213.003	3.2.1.3 Data Updating Unit	Command 가 COMMAND_EXTRACT 이거나 COMMAND_RESERVATION_EXTRACT 이고 Back Up Data 의 extract_start_flag 가 1 이고 boil_flag 가 2 이다.
CMS.UTC.3213.004	3.2.1.3 Data Updating Unit	Command 가 COMMAND_EXTRACT 이거나 COMMAND_RESERVATION_EXTRACT 이고 Control 이 HOT_100 이거나 HOT_200 이거나 HOT_300 이고 Back Up Data 의 extract_start_flag 가 1 이고 boil_flag 가 2 이고 grind_flag 가 0 이고 drip_flag 가 1 이다.
CMS.UTC.3311.001	3.3.3.1 Monitor Controller	WARNING 상태일 때 need_water == 1, need_bean == 1, must_clean_powder == 1 이 들어온다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3311.002	3.3.3.1 Monitor Controller	NORMAL 상태가 들어온다.
CMS.UTC.3311.003	3.3.3.1 Monitor Controller	WARNING 상태일 때 need_water == 0, need_bean == 0, must_clean_powder == 1 이 들어온다.
CMS.UTC.3311.004	3.3.3.1 Monitor Controller	WARNING 상태일 때 need_water == 1, need_bean == 0, must_clean_powder == 1 이 들어온다.
CMS.UTC.3311.005	3.3.3.1 Monitor Controller	WARNING 상태일 때 need_water == 0, need_bean == 1, must_clean_powder == 1 이 들어온다.
CMS.UTC.3311.006	3.3.3.1 Monitor Controller	WARNING 상태일 때 need_water == 1, need_bean == 1, must_clean_powder == 1 이 들어온다.
CMS.UTC.3312.001	3.3.1.2 Speaker Controller	NormalControlWarningData 의 Warning 이 1 이다.
CMS.UTC.3312.002	3.3.1.2 Speaker Controller	NormalControlWarningData 의 Warning 이 2 이다.
CMS.UTC.3312.003	3.3.1.2 Speaker Controller	NormalControlWarningData 의 Warning 이 3 이다.
CMS.UTC.3321.001	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3321.002	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.003	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.004	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.005	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.006	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.007	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3321.008	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.009	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.010	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.011	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.012	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.013	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3321.014	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.015	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.016	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.017	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3321.018	3.3.2.1 Status	ButtonDetectionData, IngredientSensingData, BackUpData 데이터를 가지고 상태를 출력한다.
CMS.UTC.3322.001	3.3.2.2 warning	NormalControlWarningData 의 warning 이 1 이다.
CMS.UTC.3322.002	3.3.2.2 warning	NormalControlWarningData 의 warning 이 2 이다.

Identifier	Feature	Valid/Invalid value
CMS.UTC.3322.003	3.3.2.2 warning	NormalControlWarningData 의 warning 이 2 이다.

7.5.

7.5. Feature pass/fail criteria

8. Unit test case specification

8.1. Test case specification identifier

Identifier	Input Specification	Output Specification
CMS.UTC.1100.001	Input 없음	리턴값 == COM-MAND_EXRACT
CMS.UTC.1200.001	Volume == 300	Volume == 100
CMS.UTC.1200.002	Volume == 100	Volume == 200
CMS.UTC.1200.003	Volume == 200	Volume == 300
CMS.UTC.1300.001	temperature == 0	is_cold == 1
CMS.UTC.1300.002	temperature == 1	is_cold == 0

Identifier	Input Specification	Output Specification
CMS.UTC.1400.001	Input 없음	반환값==COMMAND_CLEAN
CMS.UTC.1600.001	struct ButtonDetectionData code, struct ButtonDetectionData file, forcedTermination_flag==0	ButtonDetectionData_generate = 1 file.command==0, file.temerature==HOT, file.volume==200, file.reservationCleanTime_hour==-1, file.reservationCleanTime_min==-1, file.reservationExtractTime_hour==-1, file.reservationExtractTime_min==-1, file.reservedTemperature==-1, file.reservedVolume==-1
CMS.UTC.1600.002	struct ButtonDetectionData file, forcedTermination_flag==1	file.command == 0, file.temperature == 0(HOT), file.volume == 200, file.reservedCleanTime_hour == -1, file.reservedCleanTime_min == -1, file.reservedExtractTime_hour == -1, file.reservedExtractTime_min == -1, file.reservedTemperature == -1, file.reservedVolume == -1

Identifier	Input Specification	Output Specification
CMS.UTC.1600.003	struct ButtonDetectionData code ={2,COLD,400,12,34,23,59,HOT,100} , struct ButtonDetectionData file, forcedTermination_flag == 0, “ButtonDetectionData.txt”={-1,-1,-1,-1,-1,-1,-1,-1,-1}	code.command == file.command, code.temperature==file.temperature, code.volume==file.volume, code.reservedCleanTime_hour==file.reservedCleanTime_hour, code.reservedCleanTime_min==file.reservedCleanTime_min, code.reservedExtractTime_hour==file.reservedExtractTime_hour, code.reservedExtractTime_min==file.reservedExtractTime_min, code.reservedTemperature==file.reservedTemperature, code.reservedVolume==file.reservedVolume
CMS.UTC.1600.004	struct ButtonDetectionData code ={2,COLD,400,12,34,23,59,HOT,100} , struct ButtonDetectionData file, forcedTermination_flag == 1, “ButtonDetectionData.txt”={-1,-1,-1,-1,-1,-1,-1,-1,-1}	file.command==-1, file.temperature==-1, file.volume==-1, file.reservedCleanTime_hour==-1, file.reservedCleanTime_min==-1, file.reservedExtractTime_hour==-1, file.reservedExtractTime_min==-1, file.reservedTemperature==-1, file.reservedVolume==-1

CMS.UTC.2400.001	struct IngredientSensingData test1;
	
	test1.water = 0;
	
	test1.bean = 0;
	
	test1.powder = 0;
	
	FILE *f_ingredientSensingData;
	
	int water = 1;
		ingredientSensingData->water = 100;
		
		ingredientSensingData->bean = 10;
		
		ingredientSensingData->powder = 10;

Identifier	Input Specification	Output Specification

	int bean = 1;
	
	int powder = 1;	

CMS.UTC.2400.002	struct IngredientSensingData test;
	
	test.water = 100;
	
	test.bean = 10;
	
	test.powder = 10;	

CMS.UTC.3110.001	terminated == 1, extract_start_flag == 1, clean_flag == 0	is_recovery == 1, is_ongoing == 0, is_waiting == 0
CMS.UTC.3110.002	terminated == 1, extract_start_flag == 0, clean_flag == 0	is_recovery == 1, is_ongoing == 0, is_waiting == 0
CMS.UTC.3110.003	terminated == 0, extract_start_flag == 1, clean_flag == 0	is_recovery == 0, is_ongoing == 1, is_waiting == 0

Identifier	Input Specification	Output Specification
CMS.UTC.3110.004	terminated == 0, extract_start_flag == 0, clean_flag == 1	is_recovery == 0, is_ongoing == 1, is_waiting == 0
CMS.UTC.3110.005	terminated == 0, extract_start_flag == 1, clean_flag == 1	is_recovery == 0, is_ongoing == 1, is_waiting == 0
CMS.UTC.3110.006	terminated == 0, extract_start_flag == 0, clean_flag == 0	is_recovery == 0, is_ongoing == 0, is_waiting == 1
CMS.UTC.3120.001	current_hour == 11, current_min == 11, reserved_clean_hour == -1, reserved_clean_min== -1, reserved_extract_hour == -1, reserved_extract_min == -1, reserved_clean_complete == 0, reserved_extract_complete == 0	result_command == 3

Identifier	Input Specification	Output Specification
CMS.UTC.3120.002	current_hour == 11, current_min == 11, reserved_clean_hour == -1, reserved_clean_min== -1, reserved_extract_hour == -1, reserved_extract_min == -1, reserved_clean_complete == 1, reserved_extract_complete == 0	result_command == 3
CMS.UTC.3120.003	current_hour == 11, current_min == 11, reserved_clean_hour == -1, reserved_clean_min== -1, reserved_extract_hour == -1, reserved_extract_min == -1, reserved_clean_complete == 0, reserved_extract_complete == 1	result_command == 3
CMS.UTC.3120.004	current_hour == 11, current_min == 11, reserved_clean_hour == 12, reserved_clean_min== 12, reserved_extract_hour == -1, reserved_extract_min == -1, reserved_clean_complete == 0, reserved_extract_complete == 0	result_command == 3
CMS.UTC.3120.005	current_hour == 11, current_min == 11, reserved_clean_hour == -1, reserved_clean_min== -1, reserved_extract_hour == 2, reserved_extract_min == 5, reserved_clean_complete == 0, reserved_extract_complete == 0	result_command == 3
CMS.UTC.3120.006	current_hour == 11, current_min == 11, reserved_clean_hour == 4, reserved_clean_min== 12, reserved_extract_hour == 4, reserved_extract_min == 12, reserved_clean_complete == 0, reserved_extract_complete == 0	result_command == 3

Identifier	Input Specification	Output Specification
CMS.UTC.3120.007	current_hour == 6, current_min == 20, reserved_clean_hour == -1, reserved_clean_min== -1, reserved_extract_hour == 6, reserved_extract_min == 20, reserved_clean_complete == 0, reserved_extract_complete == 0	result_command == 2
CMS.UTC.3120.008	current_hour == 4, current_min == 46, reserved_clean_hour == -1, reserved_clean_min== -1, reserved_extract_hour == 4, reserved_extract_min == 46, reserved_clean_complete == 0, reserved_extract_complete == 1	result_command == 3
CMS.UTC.3120.009	current_hour == 12, current_min == 11, reserved_clean_hour == 12, reserved_clean_min== 11, reserved_extract_hour == -1, reserved_extract_min == -1, reserved_clean_complete == 0, reserved_extract_complete == 0	result_command == 1
CMS.UTC.3120.010	current_hour == 2, current_min == 1, reserved_clean_hour == 2, reserved_clean_min== 1, reserved_extract_hour == -1, reserved_extract_min == -1, reserved_clean_complete == 1, reserved_extract_complete == 0	result_command == 3
CMS.UTC.3120.011	current_hour == 10, current_min == 5, reserved_clean_hour == 10, reserved_clean_min== 5, reserved_extract_hour == 10, reserved_extract_min == 5, reserved_clean_complete == 0, reserved_extract_complete == 0	result_command == 1

Identifier	Input Specification	Output Specification
CMS.UTC.3120.012	current_hour == 9, current_min == 4, reserved_clean_hour == 9, reserved_clean_min== 4, reserved_extract_hour == 9, reserved_extract_min == 4, reserved_clean_complete == 1, reserved_extract_complete == 0	result_command == 2
CMS.UTC.3130.001	command_type == -1, temperature == -1, volume == -1	result_control_command = 0
CMS.UTC.3130.002	command_type == -1, temperature == COLD, volume == 100	result_control_command = 0
CMS.UTC.3130.003	command_type == COMMAND_CLEAN, temperature == -1, volume == -1	result_control_command = CLEAN

Identifier	Input Specification	Output Specification
CMS.UTC.3130.004	command_type == COM-MAND_CLEAN, temperature == -1, volume == -1	result_control_command = CLEAN
CMS.UTC.3130.005	command_type == COM-MAND_RESERVA-TION_CLEAN, temperature == -1, volume == -1	result_control_command = CLEAN;
CMS.UTC.3130.006	command_type == COM-MAND_RESERVA-TION_CLEAN, temperature == HOT, volume == 200	result_control_command = CLEAN;
CMS.UTC.3130.007	command_type == COM-MAND_EXTRACT, tempera-ture == HOT, volume == 100	result_control_command = HOT_100

Identifier	Input Specification	Output Specification
CMS.UTC.3130.008	command_type == COMMAND_EXTRACT, temperature == HOT, volume == 200	result_control_command = HOT_200
CMS.UTC.3130.009	command_type == COMMAND_EXTRACT, temperature == HOT, volume == 300	result_control_command = HOT_300
CMS.UTC.3130.010	command_type == COMMAND_EXTRACT, temperature == COLD, volume == 100	result_control_command = COLD_100
CMS.UTC.3130.011	command_type == COMMAND_EXTRACT, temperature == COLD, volume == 200	result_control_command = COLD_200
CMS.UTC.3130.012	command_type == COMMAND_EXTRACT, temperature == COLD, volume == 300	result_control_command = COLD_300
CMS.UTC.3130.013	command_type == COMMAND_RESERVATION_EXTRACT, temperature == HOT, volume == 100	result_control_command = HOT_100

Identifier	Input Specification	Output Specification
CMS.UTC.3130.014	command_type == COMMAND_RESERVATION_EXTRACT, temperature == HOT, volume == 200	result_control_command = HOT_200
CMS.UTC.3130.015	command_type == COMMAND_RESERVATION_EXTRACT, temperature == HOT, volume == 300	result_control_command = HOT_300
CMS.UTC.3130.016	command_type == COMMAND_RESERVATION_EXTRACT, temperature == COLD, volume == 100	result_control_command = COLD_100
CMS.UTC.3130.017	command_type == COMMAND_RESERVATION_EXTRACT, temperature == COLD, volume == 200	result_control_command = COLD_200
CMS.UTC.3130.018	command_type == COMMAND_RESERVATION_EXTRACT, temperature == COLD, volume == 300	result_control_command = COLD_300
CMS.UTC.3140.001	Command == COMMAND_CLEAN, Temperature == HOT, Volume == 100, Ingredientsensingdata.water == 300, Ingredientsensingdata.bean == 30, Ingredientsensingdata.powder == POWDER_NEW, Normalcontrolwarningdata.control == 0, Normalcontrolwarningdata.warning == 0	Call CleanCheck(ingredientsensingdata->water, normalcontrolwarningdata)

Identifier	Input Specification	Output Specification
CMS.UTC.3140.002	Command == COMMAND_RESERVATION_CLEAN, Temperature == HOT, Volume == 100, Ingredientsensingdata.water == 300, Ingredientsensingdata.bean == 30, Ingredientsensingdata.powder == POWDER_NEW, Normalcontrolwarningdata.control == 0, Normalcontrolwarningdata.warning == 0	Call CleanCheck(ingredientsensingdata->water, normalcontrolwarningdata)
CMS.UTC.3140.003	Command == COMMAND_EXTRACT, Temperature == HOT, Volume == 100, Ingredientsensingdata.water == 300, Ingredientsensingdata.bean == 30, Ingredientsensingdata.powder == POWDER_NEW, Normalcontrolwarningdata.control == 0, Normalcontrolwarningdata.warning == 0	Call ExtractCheck(ingredientsensingdata, temperature, volume, normalcontrolwarningdata)
CMS.UTC.3140.004	Command == COMMAND_RESERVATION_EXTRACT, Temperature == HOT, Volume == 100, Ingredientsensingdata.water == 300, Ingredientsensingdata.bean == 30, Ingredientsensingdata.powder == POWDER_NEW, Normalcontrolwarningdata.control == 0, Normalcontrolwarningdata.warning == 0	Call ExtractCheck(ingredientsensingdata, temperature, volume, normalcontrolwarningdata)
CMS.UTC.3150.001	water==499, struct NormalControlWarningData test = {-1, -1}	test.control == -1, test.warning == WARNING_WATER_SHORT

Identifier	Input Specification	Output Specification
CMS.UTC.3150.002	Water==500, Struct NormalControlWarning-Data test = {-1, -1}	test.control == CLEAN, test.warning== -1
CMS.UTC.3160.001	Temperature == HOT, Volume == 100, Ingredientsensingdata.water == 300, Ingredientsensingdata.bean == 30, Ingredientsensingdata.powder == POWDER_USED, Normalcontrolwarningdata.con-trol == 0, Normalcontrolwarning-data.warning == 0	
CMS.UTC.3160.002	Temperature == HOT, Volume == 200, Ingredientsensingdata.water == 100, Ingredientsensingdata.bean == 30, Ingredientsensingdata.powder == POWDER_NEW, Normalcontrolwarningdata.con-trol == 0, Normalcontrolwarning-data.warning == 0	Normalcontrolwarningdata.con-trol == 0 Normalcontrolwarning-data.warning == WARN-ING_WATER_SHORT
CMS.UTC.3160.003	Temperature == HOT, Volume == 100, Ingredientsensingdata.water == 300, Ingredientsensingdata.bean == 0, Ingredientsensingdata.powder == POWDER_NO, Normalcontrolwarningdata.con-trol == 0, Normalcontrolwarning-data.warning == 0	Normalcontrolwarningdata.con-trol == 0 Normalcontrolwarning-data.warning == WARN-ING_BEAN_SHORT

Identifier	Input Specification	Output Specification
CMS.UTC.3160.004	Temperature == HOT, COLD Volume == 100, 200, 300 Ingredientsensingdata.water == 300, Ingredientsensingdata.bean == 30, Ingredientsensingdata.powder == POWDER_NEW, Normalcontrolwarningdata.control == 0, Normalcontrolwarningdata.warning == 0	Normalcontrolwarningdata.control == HOT_100 || HOT_200 || HOT_300 || COLD_100 || COLD_200 || COLD_300 Normalcontrolwarningdata.warning == 0
CMS.UTC.3170.001	warning==0, command==-1, extractController==0, cleanController==0, dataUpdatingUnit==0, monitorController==0, speakerController==0 (extractController, cleanController, dataUpdatingUnit, monitorController, speakerController는 각 함수가 호출되었음을 표시하는 값)	command == -1, extractController == 1 cleanController==1, dataUpdatingUnit==1, monitorController==1, speakerController==1

Identifier	Input Specification	Output Specification
CMS.UTC.3170.002	warning==1, command==-1, extractController==0, cleanController==0, dataUpdatingUnit==0, monitorController==0, speakerController==0 (extractController, cleanController, dataUpdatingUnit, monitorController, speakerController 는 각 함수가 호출되었음을 표시하는 값)	command == 0, extractController == 1 cleanController==1, dataUpdatingUnit==1, monitorController==1, speakerController==1

Identifier	Input Specification	Output Specification
CMS.UTC.3213.001	Command == COM- MAND_CLEAN, Control == CLEAN, ingredientsensingdata.water == 500, ingredientsensingdata.bean == 30, ingredientsensingdata.powder == POWDER_NEW, backupdata.extract_start_flag == 0, backupdata.boil_flag == 0, backupdata.grind_flag == 0, backupdata.drip_flag == 0, backupdata.clean_flag == 1, backupdata.backupWater == 500, backupdata.backupBean == 30, boiling == 0, reservedCleanComplete == 0, reservedExtractComplete == 0	IngredientSensingData.txt Water == 500 Bean == 30 Powder == POWDER_NEW BackUpData.txt Extract_start_flag == 0, Boil_flag == 0, Grind_flag == 0, Drip_flag == 0, Clean_flag == 1 BackupWater == 500, backupBean == 30 boiling == 0, reservedCleanComplete == 0, reservedExtractComplete == 0

Identifier	Input Specification	Output Specification
CMS.UTC.3213.002	Command == COMMAND_EXTRACT, Control == HOT_100, ingredientsensingdata.water == 500, ingredientsensingdata.bean == 30, ingredientsensingdata.powder == POWDER_NEW, backupdata.extract_start_flag == 1, backupdata.boil_flag == 1, backupdata.grind_flag == 0, backupdata.drip_flag == 0, backupdata.clean_flag == 0, backupdata.backupWater == 500, backupdata.backupBean == 30, boiling == 1, reservedCleanComplete == 0, reservedExtractComplete == 0	IngredientSensingData.txt Water == 500 Bean == 30 Powder == POWDER_NEW BackUpData.txt Extract_start_flag == 1, Boil_flag == 1, Grind_flag == 0, Drip_flag == 0, Clean_flag == 0 BackupWater == 500, backupBean == 30 boiling == 2, reservedCleanComplete == 0, reservedExtractComplete == 0
CMS.UTC.3213.003	Command == COMMAND_EXTRACT, Control == HOT_100, ingredientsensingdata.water == 500, ingredientsensingdata.bean == 30, ingredientsensingdata.powder == POWDER_NO, backupdata.extract_start_flag == 1, backupdata.boil_flag == 2, backupdata.grind_flag == 1, backupdata.drip_flag == 0, backupdata.clean_flag == 0, backupdata.backupWater == 500, backupdata.backupBean == 30, boiling == 0, reservedCleanComplete == 0, reservedExtractComplete == 0	IngredientSensingData.txt Water == 500 Bean == 28 Powder == POWDER_NO BackUpData.txt Extract_start_flag == 1, Boil_flag == 2, Grind_flag == 1, Drip_flag == 0, Clean_flag == 0 BackupWater == 500, backupBean == 30 boiling == 0, reservedCleanComplete == 0, reservedExtractComplete == 0

Identifier	Input Specification	Output Specification
CMS.UTC.3213.004	Command == COMMAND_EXTRACT, Control == HOT_100, ingredientsensingdata.water == 500, ingredientsensingdata.bean == 30, ingredientsensingdata.powder == POWDER_NEW, backupdata.extract_start_flag == 1, backupdata.boil_flag == 2, backupdata.grind_flag == 0, backupdata.drip_flag == 1, backupdata.clean_flag == 0, backupdata.backupWater == 500, backupdata.backupBean == 30, boiling == 0, reservedCleanComplete == 0, reservedExtractComplete == 0	IngredientSensingData.txt Water == 490 Bean == 30 Powder == POWDER_NEW BackUpData.txt Extract_start_flag == 1, Boil_flag == 2, Grind_flag == 0, Drip_flag == 1, Clean_flag == 0 BackupWater == 500, backupBean == 30 boiling == 0, reservedCleanComplete == 0, reservedExtractComplete == 0
CMS.UTC.3311.001	need_water == 1, need_bean ==1, must_clean_powder == 1, warning == 1	output == 1, is_warning ==0
CMS.UTC.3311.002	need_water == 0, need_bean == 0, must_clean_powder == 0, warning == 0	output == 0, is_warning ==0
CMS.UTC.3311.003	need_water == 1, need_bean ==0, must_clean_powder == 0, warning == 1	output == 1, is_warning ==0
CMS.UTC.3311.004	need_water == 0, need_bean ==1, must_clean_powder == 0, warning == 1	output == 1, is_warning ==0
CMS.UTC.3311.005	need_water == 0, need_bean ==0, must_clean_powder == 1, warning == 1	output == 1, is_warning ==0
CMS.UTC.3311.006	need_water == 1, need_bean ==1, must_clean_powder == 1, warning == 1	output == 1, is_warning ==0

Identifier	Input Specification	Output Specification

CMS.UTC.3312.001	int wsw = 0;
	struct NormalControlWarning-Data ncwd1;
	ncwd1.control = 0;
	ncwd1.warning = 1;
		NormalControlWarning-Data.warning = 1

CMS.UTC.3312.002	int bsw = 0;
	struct NormalControlWarning-Data ncwd2;
	ncwd2.control = 0;
	ncwd2.warning = 2;
		NormalControlWarning-Data.warning = 2

CMS.UTC.3312.003	int psw = 0;
	struct NormalControlWarning-Data ncwd3;
	ncwd3.control = 0;
	ncwd3.warning = 3;
		NormalControlWarning-Data.warning = 3

CMS.UTC.3321.001	struct ButtonDetectionData bdd = {100,0,3,24,60,24,60,3,0}; struct IngredientSensingData isd={3,3,1}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 =	

Identifier	Input Specification	Output Specification
	{0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.002	struct ButtonDetectionData bdd = {100,0,3,24,60,12,13,3,0}; struct IngredientSensingData isd={3,3,1}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.003	struct ButtonDetectionData bdd = {100,0,3,24,60,-1,60,3,0}; struct IngredientSensingData isd={3,3,1}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	

Identifier	Input Specification	Output Specification
CMS.UTC.3321.004	struct ButtonDetectionData bdd = {100,0,3,13,26,-1,60,3,0}; struct IngredientSensingData isd={3,3,1}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.005	struct ButtonDetectionData bdd = {100,0,3,24,60,24,-1,3,0}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.006	struct ButtonDetectionData bdd = {100,0,3,24,60,24,-1,3,0}; struct IngredientSensingData isd={3,3,0}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	

Identifier	Input Specification	Output Specification
CMS.UTC.3321.007	struct ButtonDetectionData bdd = {100,0,3,24,60,12,13,3,0}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.008	struct ButtonDetectionData bdd = {100,0,3,12,13,25,-1,3,0}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.009	struct ButtonDetectionData bdd = {100,0,3,-1,61,7,13,0,100}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 =	

Identifier	Input Specification	Output Specification
	{0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.010	struct ButtonDetectionData bdd = {100,0,3,-1,61,7,13,1,300}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.011	struct ButtonDetectionData bdd = {100,0,3,24,60,24,-1,3,0}; struct IngredientSensingData isd={3,3,0}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	

Identifier	Input Specification	Output Specification
CMS.UTC.3321.012	struct ButtonDetectionData bdd = {100,0,3,24,60,12,13,0,100}; struct IngredientSensingData isd={3,3,0}; struct BackUpdata bud={1,0,0,0,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.013	struct ButtonDetectionData bdd = {100,0,3,-1,61,7,13,0,400}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={0,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.014	struct ButtonDetectionData bdd = {100,0,3,-1,61,7,13,3,100}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={0,1,1,1,0,1,1};	

Identifier	Input Specification	Output Specification
	struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.015	struct ButtonDetectionData bdd = {1,0,3,-1,61,25,-1,3,3}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={1,1,1,1,0,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.016	struct ButtonDetectionData bdd = {1,0,3,-1,61,25,-1,3,3}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={1,1,1,1,1,1,1}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	

Identifier	Input Specification	Output Specification
CMS.UTC.3321.017	struct ButtonDetectionData bdd = {1,0,3,-1,61,25,-1,3,3}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={1,1,0,0,1,0,0}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	
CMS.UTC.3321.018	struct ButtonDetectionData bdd = {1,0,3,-1,61,25,-1,3,3}; struct IngredientSensingData isd={3,3,2}; struct BackUpdata bud={1,1,1,1,0,0,0}; struct index case1 = {0,0,0,0,0,0,0,0,0,0};	

CMS.UTC.3322.001	int warningsignal = 0;
	
	warningsignal = warning(1);
		Call Sound1()

CMS.UTC.3322.002	int warningsignal = 0;
	
	warningsignal = warning(2);
		Call Sound2()

CMS.UTC.3322.003	int warningsignal = 0;
	
	warningsignal = warning(3);
		Call Sound3()

8.2.

8.3. Test items

<Table 3 Test Design Identification> 참조

8.4. Input specifications

<Table 4 Test Case Identification> 참조

8.5. Output specifications

<Table 4 Test Case Identification> 참조

9. Testing tasks

10. Environmental needs

11. Unit Test deliverables

12. Schedules