

Unit Testing Plan

for Coffee Machine System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

Team1

Date

2016-11-06

Team Information

201411259 고수창

201511243 김동연

201511263 박종엽

201511280 이선엽

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	4
3	Features to be tested	5
4	Features not to be tested	5
5	Approach.....	5
6	Item pass/fail criteria	6
7	Unit test design specification.....	6
7.1	Test design specification identifier	6
7.2	Features to be tested	6
7.3	Approach refinements.....	6
7.4	Test identification	6
7.5	Feature pass/fail criteria	6
8	Unit test case specification.....	6
8.1	Test case specification identifier	7
8.2	Test items	9

8.3	Input specifications.....	9
8.4	Output specifications.....	9
9	Testing tasks	9
10	Environmental needs	10
11	Unit Test deliverables.....	10
12	Schedules	10

1 Introduction

1.1 Objectives

본 문서는 2016학년도 2학기 소프트웨어 공학 개론 수업의 T1 Team이 개발한 Coffee Machine System(이하 CMS)을 Unit Testing하기 위한 계획 문서이다. T1 Team이 정의한 Unit Testing을 수행하기 위하여 Testing Pass/Fail Criteria를 정의하고 이를 수행하기 위한 Test Design & Test Cases를 제작한다.

1.2 Background

Coffee Machine은 사용자의 요청에 의해 커피를 추출한다.

Unit Test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 Test이며, 시스템의 성능을 좌우하는 요소들이 요구사항을 만족하는지를 확인할 수 있는 기본적인 Test approach이다

1.3 Scope

CMS에 대한 Unit Test를 수행하기 위한 자원과 절차, Test approach와 Technique와 필요로 하는 환경 및 도구 등을 정의한다. Unit Test는 시스템 핵심 기능 관련 프로세스에 중점을 두며 전달 역할 등 단순 프로세스는 Test에서 제외한다.

1.4 Project plan

1.5 Configuration management plan

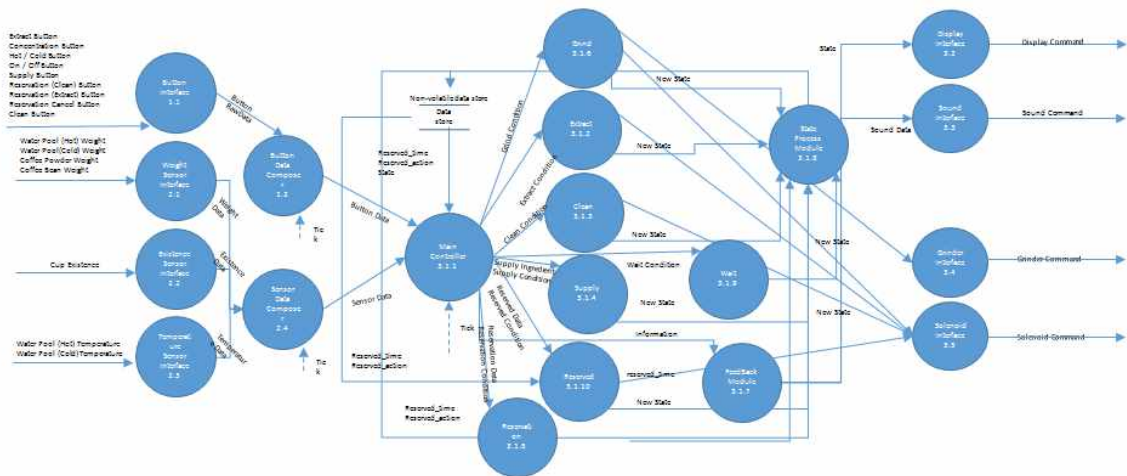
1.6 References

T1-2016.CMS.SRA-1.2

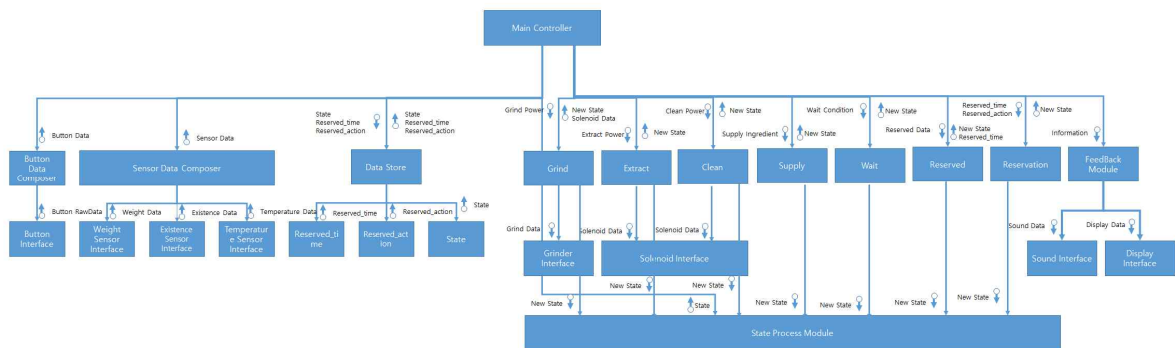
T1-2016.CMS.SRS-1.1

2 Test items

2.1 T1 Team이 SASD 기법을 이용하여 개발한 CMS를 Testing한다. SA에서 최소단위의 각 Process별로 요구사항을 만족하는지, 정상적인 결과가 나오는지, 잘못된 값 입력 시 예외 처리가 동작하는지 Testing을 수행한다. <Figure 1 Overall DFD>는 SA를 이용하여 요구사항을 분석 한 결과를 DFD를 이용해 나타낸 그림이고, <Figure 2 Structural Chart>는 SD의 Basic Structural Chart를 나타낸 그림이다. 각 그림을 참조하여 Unit을 지정하고, 지정한 Unit을 SRA에 명세 된 내용과 같은 동작을 하는지 확인한다.



<Figure 1 Overall DFD>



<Figure 2 Structural Chart>

3 Features to be tested

시스템 핵심 기능 관련 모듈 중 입력 담당 모듈 부분과 처리 담당 모듈 부분을 거쳐 요구사항 만족, 정상적인 동작과 잘못된 값 입력시 동작에 중점을 두어 테스트 한다.

1) Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는지 테스트한다.

2) Modules in SDS : 각 모듈이 가지고 있는 데이터 인터페이스를 테스트 한다.

4 Features not to be tested

전달 역할, 단순한 프로세스, UI 구성 프로세스, 특수한 상황에서만 작동하는 프로세스는 테스트에서 제외한다.

5 Approach

CMS의 Program source code와 Unit Tests는 Mac Terminal 환경에서 이루어지며, Program code의 변경 및 수정사항은 지속적으로 통합되고 테스트 된다.

6 Item pass/fail criteria

6.1 각 Unit별 Pass/Fail Criteria는 <Test Case Identification>을 참조한다.

7 Unit test design specification

7.1 Test design specification identifier

7.2 Features to be tested

7.3 Approach refinements

각 Process Specification에 명시된 내용을 기반으로 Test Design 및 Test Cases를 생성해 낸다.

7.4 Test identification

Identifier	Feature
CMS.UTC.1000	버튼 입력에 대해 버튼이 눌렸는지 확인한다.
CMS.UTC.2000	센서로부터 데이터를 입력 받는다.
CMS.UTC.3110	입력된 버튼에 대해 각 모듈들에게 데이터를 전달한다.
CMS.UTC.3120	추출 상태인 경우 커피를 추출한다.
CMS.UTC.3130	청소 상태인 경우 커피 머신을 청소한다.
CMS.UTC.3140	공급 상태인 경우 커피 머신에 재료를 공급한다.
CMS.UTC.3150	예약 상태인 경우 커피 머신의 예약을 설정한다.
CMS.UTC.3160	Grind 상태인 경우 커피 콩을 갈아낸다.
CMS.UTC.3170	사용자에게 현재 CM의 상태를 보여준다.
CMS.UTC.3180	CM의 Next State를 입력받아 해당 상태로 설정한다.
CMS.UTC.3190	대기 상태인 경우 작업을 일시적으로 멈춘다.
CMS.UTC.3110	예약된 예약이 있을 때 예약을 감지한다.

<Table 1 Test Design Identification>

7.5 Feature pass/fail criteria

CMS의 각 모듈은 SRA에 정의되어 있는 요구사항을 만족해야 한다. 각 모듈의 입/출력 및 동작은 SRA의 Process description 항목 및 STD를 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

Identifier	Feature	Pass / Fail Criteria
CMS.UTC.1000	버튼 입력에 대해 버튼이 눌렸는지 확인한다.	
CMS.UTC.1000.00	btn_pressed(&btn_temperature);	btn_is_pressed(&btn_temperature) = 1
CMS.UTC.2000	센서로부터 데이터를 입력 받는다.	
CMS.UTC.2000.00	sensor_update(&sensor_hot_weight, 500); // 500으로 설정한 후에 실행	sensor_get(&sensor_hot_weight) = 500
CMS.UTC.2000.01	sensor_update(&sensor_coffee_bean_weight, 20); // 20으로 설정한 후에 실행	sensor_get(&sensor_coffee_bean_weight) = 500
CMS.UTC.3110	입력된 버튼에 대해 각 모듈들에게 데이터를 전달한다.	
CMS.UTC.3110.00	데이터를 전달하는 모듈이므로 테스트 하지 않음	
CMS.UTC.3120	추출 상태인 경우 커피를 추출한다.	
CMS.UTC.3120.00	sensor_update(&sensor_hot_weight, 500); sensor_update(&sensor_cup_existence, 1); sensor_update(&sensor_coffee_powder_weight, 10); new_state(STATE_EXTRACT); extract_tick(state);	Now_State = STATE_EXTRACT
CMS.UTC.3120.01	sensor_update(&sensor_hot_weight, 500); sensor_update(&sensor_cup_existence, 0); sensor_update(&sensor_coffee_powder_weight, 10); new_state(STATE_EXTRACT); extract_tick(state);	Now_State = STATE_WAIT
CMS.UTC.3130	청소 상태인 경우 커피 머신을 청소한다.	
CMS.UTC.3130.00	sensor_update(&sensor_hot_weight, 500); new_state(STATE_CLEAN); extract_tick(state);	Now_State = STATE_CLEAN

CMS.UTC.3130.01	sensor_update(&sensor_hot_weight, 400); new_state(STATE_CLEAN); extract_tick(state);	error_msg = "물이 부족합니다."
CMS.UTC.3140	공급 상태인 경우 커피 머신에 재료를 공급한다.	
CMS.UTC.3140.00	sensor_update(&sensor_hot_weight, 300) supply_type = 3; supply_amount = 100; supply_tick(STATE_SUPPLY);	sensor_get(&sensor_hot_weight) = 400
CMS.UTC.3150	예약 상태인 경우 커피 머신의 예약을 설정한다.	
CMS.UTC.3150.00	reserve_change(CLEAN,"11:11"); char *s = reserve_get(CLEAN);	S = "11:11"
CMS.UTC.3150.01	reserve_change(MK_COFFEE,"11:11"); char *s = reserve_get(MK_COFFEE);	S = "11:11"
CMS.UTC.3160	Grind 상태인 경우 커피 콩을 갈아낸다.	
CMS.UTC.3160.00	sensor_update(&sensor_coffee_bean_weight, 10); new_state(STATE_GRIND); grind_tick(state);	Now_State = STATE_GRIND
CMS.UTC.3160.01	sensor_update(&sensor_coffee_bean_weight, 5); new_state(STATE_GRIND); grind_tick(state);	error_msg = "원두가 부족합니다.";
CMS.UTC.3170	사용자에게 현재 CM의 상태를 보여준다.	
CMS.UTC.3170.00	UI 구성 프로세스이므로 제외함.	
CMS.UTC.3180	CM의 Next State를 입력받아 해당 상태로 설정한다.	
CMS.UTC.3180.00	단순한 프로세스이므로 제외함	
CMS.UTC.3190	대기 상태인 경우 작업을 일시적으로 멈춘다.	
CMS.UTC.3190.00	New_state(STATE_WAIT) Wait_tick(state)	Now_State = STATE_WAIT
CMS.UTC.3110	예약된 예약이 있을 때 예약을 감지	

	한다.	
CMS.UTC.3110.00	<pre>time_t rawtime; struct tm * timeinfo; time (&rawtime); timeinfo = localtime (&rawtime); char buf[256]; strftime(buf, sizeof(buf), "%H:%M", timeinfo); reserve_change(CLEAN,buf); new_state(STATE_RESERVED) reserved_tick(state)</pre>	State=STATE_CLEAN

<Table 2 Test Case Identification>

8.2 Test items

<Table 2 Test Case Identification> 참고

8.3 Input specifications

<Table 2 Test Case Identification> 참고

8.4 Output specifications

<Table 2 Test Case Identification> 참고

9 Testing tasks

Task	Predecessor tasks	Special Skills	Effort	Finish Date
(1) Unit Test Plan 작성	SRA, SDS, 작성 CMS 구현		3	2016.11.05
(2) Test Design Specification	Task 1	CMS에 대한 이해		
(3) Test Case Specification	Task 2	CMS에 대한 이해		
(4) Test Execution	Task 3	Test code 작성	2	
(5) Test Result Report	Task 4		1	2016.11.06
(6) Report	Task 5		1	2016.11.06

전달				
----	--	--	--	--

10 Environmental needs

Mac OS

C Language Compiler/Linker ex)GCC

11 Unit Test deliverables

12 Schedules