

Unit Test Report for Coffee Machine System

- Test Cases Specification
- Test Summary Report

Project Team

Team1

Latest update on:

2016-11-06

Team Information

201411259 고수창

201511243 김동연

201511263 박종엽

201511280 이선엽

Table of Contents

1	Introduction	3
1.1	Objectives.....	3
1.2	References.....	3
2	Unit test case specification.....	3
2.1	Test case specification identifier.....	3
2.2	Test items	5
2.3	Input specifications.....	5
2.4	Output specifications.....	5
3	Environmental needs.....	5
4	Unit test summary report	5
4.1	Test summary report identifier.....	5
4.2	Evaluation.....	8

1 Introduction

1.1 Objectives

본 문서는 2016학년도 2학기 소프트웨어 공학 개론 수업의 T1 Team이 개발한 Coffee Machine System(이하 CMS)을 Unit Testing을 수행한 결과에 대한 Report 문서이다. Test 요소들에 대한 Test Case와 Test 수행 결과에 대한 내용을 담고 있다.

1.2 References

T1-2016-CMS-SRA-1.2

T1-2016-CMS-SDA-1.1

T1-2016-CMS-UTP-1.0

2 Unit test case specification

2.1 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);	Now_State = STATE_EXTRACT

	<pre>sensor_update(&sensor_coffee_pow der_weight, 10); new_state(STATE_EXTRACT); extract_tick(state);</pre>	
CMS.UTC.3120.00	<pre>sensor_update(&sensor_hot_weight, 500); sensor_update(&sensor_cup_existen ce, 0); sensor_update(&sensor_coffee_pow der_weight, 10); new_state(STATE_EXTRACT); extract_tick(state);</pre>	Now_State = STATE_WAIT
CMS.UTC.3130	청소 상태인 경우 커피 머신을 청소한다.	
CMS.UTC.3130.00	<pre>sensor_update(&sensor_hot_weight, 500); new_state(STATE_CLEAN); extract_tick(state);</pre>	Now_State = STATE_CLEAN
CMS.UTC.3130.01	<pre>sensor_update(&sensor_hot_weight, 400); new_state(STATE_CLEAN); extract_tick(state);</pre>	error_msg = "물이 부족합니다."
CMS.UTC.3140	공급 상태인 경우 커피 머신에 재료를 공급한다.	
CMS.UTC.3140.00	<pre>sensor_update(&sensor_hot_weight, 300) supply_type = 3; supply_amount = 100; supply_tick(STATE_SUPPLY);</pre>	sensor_get(&sensor_hot_weight) = 400
CMS.UTC.3150	예약 상태인 경우 커피 머신의 예약을 설정한다.	
CMS.UTC.3150.00	단순한 프로세스이므로 제외함.	
CMS.UTC.3160	Grind 상태인 경우 커피 콩을 갈아낸다.	
CMS.UTC.3160.00	<pre>sensor_update(&sensor_coffee_bean _weight, 10); new_state(STATE_GRIND); grind_tick(state);</pre>	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	Wait_tick(STATE_WAIT)	Now_State = STATE_WAIT
CMS.UTC.3110	예약된 예약이 있을 때 예약을 감지한다.	
CMS.UTC.3110.00	예약 시간이 흘러야 하는 특수한 상황에서만 작동하는 프로세스이므로 제외함.	

<Table 1 Test Case Identification>

2.2 Test items

<Table 1 Test Case Identification> 참조

2.3 Input specifications

<Table 1 Test Case Identification> 참조

2.4 Output specifications

<Table 1 Test Case Identification> 참조

3 Environmental needs

T1-2016-CMS-UTP-1.0 Environmental needs 항목 참조

4 Unit test summary report

4.1 Test summary report identifier

Identifier	Feature	Pass / Fail Criteria	Test Result
------------	---------	----------------------	-------------

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	PASSED
CMS.UTC.2000.01	sensor_update(&sensor_coffee_bean_weight, 20); // 20으로 설정한 후에 실행	sensor_get(&sensor_coffee_bean_weight) = 500	PASSED
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	PASSED
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	PASSED
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	PASSED
CMS.UTC.3130.01	sensor_update(&sensor_hot_weight, 400);	error_msg = "물이 부족합니다."	PASSED

	new_state(STATE_CLEAN); extract_tick(state);		
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	PASSED
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"	PASSED
CMS.UTC.3150.01	reserve_change(MK_COFFEE,"11:11"); char *s = reserve_get(MK_COFFEE);	S = "11:11"	PASSED
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	PASSED
CMS.UTC.3160.01	sensor_update(&sensor_coffee_bean_weight, 5); new_state(STATE_GRIND); grind_tick(state);	error_msg = "원두가 부족합니다.";	PASSED
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	PASSED
CMS.UTC.3110	예약된 예약이 있을 때 예약을 감지한다.		
CMS.UTC.3110.00	time_t rawtime;	State=STATE_CLEAN	PASSED

	<pre>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>		
--	--	--	--

4.2 Evaluation

Total Test Case : 14개

Passed : 14개

Failed : 0개