

Unit Testing Plan

for Coffee Machine System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

T6 Team

Date

2016-11-07

Team Information

201311279 백승대

201311289 우용하

201311314 정재욱

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.....	6
4	Features not to be tested	7
5	Approach.....	7
6	Item pass/fail criteria	7
7	Unit test design specification.....	7
7.1	Test design specification identifier	7
7.2	Features to be tested	7
7.3	Approach refinements.....	7
7.4	Test identification	8
7.5	Feature pass/fail criteria	12
8	Unit test case specification.....	12
8.1	Test case specification identifier	12
8.2	Test items	15

8.3 Input specifications..... 15

8.4 Output specifications..... 15

9 Testing tasks 15

10 Environmental needs 15

11 Unit Test deliverables..... 16

12 Schedules 16

1 Introduction

1.1 Objectives

본 문서는 2016년 2학기 소프트웨어공학개론 수업의 Team T6가 개발한 Coffee Machine System(CMS)을 Unit Testing 하기 위한 계획을 다루고 있다. Team T6가 정의한 Unit Testing을 수행하기 위하여 Testing Pass/Fail Criteria를 정의하고 이를 수행하기 위한 Test Design & Test Case 를 정의한다.

1.2 Background

2016년 2학기 소프트웨어공학개론 수업에서 개발하는 모든 CMS의 요소들은 SASD 기법을 이용하여 개발한다. 기능별로 나누어진 Unit 은 SRA와 SDS 문서에서 모두 정의되어 있다. CMS는 커피 머신 시스템으로서, 사용자가 커맨드를 통해 커피를 가상으로 추출하는 시스템이다. 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

이 계획 문서는 CMS의 unit test를 수행하기 위한 모든 것을 포함한다. CMS의 unit test를 수행하기 위한 지원과 절차, test approach와 technique과 필요로 하는 환경 및 도구들을 정의한다. CMS의 unit test는 시스템을 구성하는 최소 단위의 모듈들을 대상으로 하며, 구현된 모듈이 요구사항을 만족하는지를 test 한다.

1.4 Project plan

1.5 Configuration management plan

1.6 References

[2016SE_A][1][T6]SRA_2_3

[2016SE_A][1][T6]SDA_1_2

2 Test items

Team T6가 SASD 기법을 이용하여 개발한 CMS를 Testing한다. SA와 SD에서 분류한 각 Process/Module 별로 Testing을 수행한다. 각 그림을 참조하여 Unit을 지정하고, 지정한 Unit을 SRA에 명세된 내용과 일치하는 동작을 수행하는지 확인한다. Test item은 다음 자

3 Features to be tested

Process/Module 별로 입력이 들어오면 그에 맞는 출력이 나오는지 Testing 한다.

(1) Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는 지를 test한다.

- CoffeeMachine System

(2) Modules in SDS : 각 모듈이 가지고 있는 기능을 test한다.

- CoffeeMachine System

<Table 1 : 테스트 할 Process List>

Name	Description
Power Button interface	Poewr Button에서 받은 데이터를 Button Manager에 전달한다.
Clean Button interface	Clean Button에서 받은 데이터를 Button Manager에 전달 한다.
Extract Button interface	Extract Button에서 받은 데이터를 Button Manager에 전달 한다.
Reserve Button interface	Reserve Button에서 받은 데이터를 Button Manager에 전달 한다.
Concentrate Button interface	Concentrate Button에서 받은 데이터를 Button Manager에 전달 한다.
Temperature Button interface	Temperature Button에서 받은 데이터를 Button Manager에 전달 한다.
Water Sensor interface	Water Sensor에서 받은 데이터를 Ingredient Manager에 전달한다.
Bean Sensor interface	Bean Sensor에서 받은 데이터를 Ingredient Manager에 전달한다.
CoffeePowder Sensor interface	CoffeePowder Sensor에서 받은 데이터를 Ingredient Manager에 전달한다.
Button Manager	각 버튼 interface에서 받은 데이터를 Button Data에 저장한다.
Ingredient Manager	각 센서 interface에서 받은 데이터를 Ingredient Data에 저장한다.
Reserve Controller	Reserve Data를 받아 시간 갱신과 count down을 한다.
Clean Machine	Coffee Machine Controller로부터 명령을 전달받아 물이 충분할 때 Cleaner interface로 명령을 보낸다

Grind Bean	Make Coffee Controller로부터 명령을 받아 커피를 가는 명령을 Coffee Machine interface로 보낸다. 재료의 양을 저장한다.
Make Coffee	Make Coffee Controller로부터 명령을 받아 커피를 만드는 명령을 Coffee Machine interface로 보낸다. 재료의 양을 저장한다.
Coffee Machine interface	Controller로부터 명령을 받아 커피를 추출한다.

4 Features not to be tested

이번 테스트는 Unit별로 진행하므로 단순히 데이터를 전달하고 예외사항이 존재하지 않는 프로세스들은 진행하지 않는다.

5 Approach

CMS의 Test Code 는 Cygwin + gcc 환경에서 Team T6가 직접 작성한 Unit Test Framework 로 작성되어 실행된다. Program Source Code 와 Test Code 의 변경 및 수정 사항은 지속적으로 통합하고 Test 한다. 각 프로세스 별로 (process_name)_unit_test 내에 유닛 테스트를 위한 코드들을 Test Case에 따라 작성한다.

6 Item pass/fail criteria

Functional Test Pass/Fail Criteria : 각 Process/Module 은 요구사항을 만족해야 한다.

7 Unit test design specification

7.1 Test design specification identifier

CMS_UTC.unit

7.2 Features to be tested

<Table 1 : 테스트할 Process List> 참조

7.3 Approach refinements

각 모듈이 요구사항을 만족하는지 확인하기 위해, 요구사항에 정의된 내용에 기반하여 test case를 작성한다. 그 이외의 사항에 대해서는 test case를 작성하지 않는다.

7.4 Test identification

<Table 2 : Test Design Identification>

Test Case Identifier	Feature	Valid value
CMS.UTC.unit1.1	Power Button interface	Power버튼을 입력 받지 않았을 때 0을 Button Manager로 전달한다.
CMS.UTC.unit1.2	Power Button interface	Power버튼을 입력 받았을 때 Power버튼을 0으로 초기화 해준 후 1을 Button Manager로 전달한다.
CMS.UTC.unit2.1	Clean Button interface	Clean버튼을 입력 받지 않았을 때 0을 Button Manager로 전달한다.
CMS.UTC.unit2.2	Clean Button interface	Clean버튼을 입력 받았을 때 Clean버튼을 0으로 초기화 해준 후 1을 Button Manager로 전달한다.
CMS.UTC.unit3.1	Extract Button interface	Extract 버튼을 입력 받지 않았을 때 0을 Button Manager로 전달한다.
CMS.UTC.unit3.2	Extract Button interface	Extract버튼을 입력 받았을 때 Extract버튼을 0으로 초기화 해준 후 Button Manager로 전달한다.
CMS.UTC.unit4.1	Reserve Button interface	Reserve버튼중 3번 버튼이 입력 받았을 때 모든 버튼 값을 0으로 초기화하고 {0,0,1}을 Button Manager로 전달한다.
CMS.UTC.unit4.2	Reserve Button interface	Reserve버튼 중 4번 버튼이 입력 받았을 때 모든 버튼 값을 0으로 초기화하고 {0,0,-1}을 Button Manager로 전달한다.
CMS.UTC.unit4.3	Reserve Button interface	Reserve버튼 중 1번 버튼이 입력 받았을 때 1번 버튼 값을 0으로 초기화하고 {1,0,0}을 Button Manager로 전달한다.

CMS.UTC.unit4.4	Reserve Button interface	Reserve버튼 중 2번 버튼이 입력 받았을 때 2번 버튼 값을 0으로 초기화하고 {0,1,0}을 Button Manager로 전달한다.
CMS.UTC.unit4.5	Reserve Button interface	Reserve버튼이 입력 받지 않았을 때 NULL을 Button Manager로 전달한다.
CMS.UTC.unit5.1	Concentrate Button interface	Concentrate버튼을 입력 받지 않았을 때 0을 Button Manager로 전달한다.
CMS.UTC.unit5.2	Concentrate Button interface	Concentrate버튼을 입력 받았을 때 1을 Button Manager로 전달한다.
CMS.UTC.unit6.1	Temperature Button interface	Temperature버튼을 입력 받지 않았을 때 0을 Button Manager로 전달한다.
CMS.UTC.unit6.2	Temperature Button interface	Temperature버튼을 입력 받았을 때 1을 Button Manager로 전달한다.
CMS.UTC.unit7.1	Button Manager	Power버튼이 입력 받은 경우에 Button Data에 1을 저장한다.
CMS.UTC.unit7.2	Button Manager	Clean버튼이 입력 받은 경우에 Button Data에 1을 저장한다.
CMS.UTC.unit7.3	Button Manager	Extract버튼이 입력 받은 경우에 Button Data에 1을 저장한다.
CMS.UTC.unit7.4	Button Manager	Concentrate버튼이 입력 받은 경우 Button Data에 100이 저장 되어 있으면 Button Data에 200을 저장 한다.
CMS.UTC.unit7.5	Button Manager	Concentrate버튼이 입력 받은 경우 Button Data에 200이 저장 되어 있으면 Button Data에 300을 저장 한다.
CMS.UTC.unit7.6	Button Manager	Concentrate버튼이 입력 받은 경우 Button Data에 300이 저

		장 되어 있으면 Button Data에 100을 저장 한다.
CMS.UTC.unit7.7	Button Manager	Temperature버튼이 입력 받은 경우 Button Data에 0이 저장 되어 있으면 Button Data에 1을 저장 한다.
CMS.UTC.unit7.8	Button Manager	Temperature버튼이 입력 받은 경우 Button Data에 1이 저장 되어 있으면 Button Data에 0을 저장 한다.
CMS.UTC.unit7.9	Button Manager	Reserve버튼 값이 {1,x,0,0}이고 Button Data의 첫번째 값이 0일 때 Button Data의 첫번째 값에 1을 저장한다.
CMS.UTC.unit7.10	Button Manager	Reserve버튼 값이 {1,x,0,0}이고 Button Data의 첫번째 값이 1일 때 Button Data의 첫번째 값에 0을 저장한다.
CMS.UTC.unit7.11	Button Manager	Reserve버튼 값이 {0,1,0,0}이고 Button Data의 두번째 값이 0일 때 Button Data의 두번째 값에 15를 저장한다.
CMS.UTC.unit7.12	Button Manager	Reserve버튼 값이 {0,1,0,0}이고 Button Data의 두번째 값이 120일 때 Button Data의 두번째 값에 135를 저장한다.
CMS.UTC.unit7.13	Button Manager	Reserve버튼 값이 {0,1,0,0}이고 Button Data의 두번째 값이 24*60+15일 때 Button Data의 두번째 값에 0을 저장한다.
CMS.UTC.unit7.14	Button Manager	Reserve버튼 값이 {x,y,1,z}인 경우 Button Data의 세번째 값에 1을 저장한다.
CMS.UTC.unit7.15	Button Manager	Reserve버튼 값이 {x,y,0,1}인 경우 Button Data의 세번째 값에 -1을 저장한다.
CMS.UTC.unit8.1	Reserve Controller	Button Data의 값에 {0,y,1}인

		경우 Reserve Data의 mode에 0, remainTime에 y, concentrate에 Button Data의 concentrate 값, temperature에 Button Data의 temperature의 값을 저장하고 Button Data의 reserveinfo 세번째 값을 0으로 초기화 한다.
CMS.UTC.unit8.2	Reserve Controller	Button Data의 값에 {0,y,1}인 경우 Reserve Data의 mode에 0, remainTime에 y를 저장하고 Button Data의 reserveinfo 세번째 값을 0으로 초기화 한다.
CMS.UTC.unit8.3	Reserve Controller	Button Data의 값에 {x,y,-1}인 경우 Reserve Data의 mode에 -1을 저장하고 Button Data의 reserveinfo의 값을 0으로 초기화 한다.
CMS.UTC.unit8.4	Reserve Controller	Button Data의 값이 {x,y,0}이고 Reserve Data의 모드가 -1이 아니고 remainTime이 0보다 클 때 remainTime에 remainTime-1한 값을 저장한다.
CMS.UTC.unit9.1	Clean Machine	Ingredient Data의 water값이 500미만이면 "Lack of Water"을 출력한다.
CMS.UTC.unit9.2	Clean Machine	Ingredient Data의 water값이 500이상이면 water값에 water-500한 값을 저장한다.
CMS.UTC.unit10.1	Grind Bean	Ingredient Data의 bean값에 bean-10한 값을 저장하고 coffeePowder값에 coffeePowder+10한 값을 저장한다.
CMS.UTC.unit11.1	Make Coffee	Ingredient Data의 water값에 water-coffeeInfoData.

		Concentrate한 값을 저장하고 coffeePowder값에 coffeePowder-10한 값을 저장한다.
CMS_UTC.unit12.1	Coffee Machine interface	Command가 0일 때 0을 반환한다.
CMS_UTC.unit12.2	Coffee Machine interface	Command가 1일 때 0을 반환한다.
CMS_UTC.unit12.3	Coffee Machine interface	Command가 2일 때 0을 반환한다.
CMS_UTC.unit12.4	Coffee Machine interface	Command가 0,1,2 모두 아닐 경우 -10을 반환한다.

7.5 Feature pass/fail criteria

각 모듈은 SRA에 정의되어 있는 요구사항(입력/출력 및 동작)을 모두 만족해야 한다.

SRA의 process description 항목을 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

<Table 3 : Test Case Identification>

Identifier	Input Specification	Output Specification
CMS_UTC.unit1.1	buttonInput.powerBtn == 0	return 0
CMS_UTC.unit1.2	buttonInput.powerBtn == 1	buttonInput.powerBtn = 0 return 1
CMS_UTC.unit2.1	buttonInput.cleanBtn == 0	return 0
CMS_UTC.unit2.2	buttonInput.cleanBtn == 1	buttonInput.cleanBtn = 0 return 1
CMS_UTC.unit3.1	buttonInput.extractBtn == 0	return 0
CMS_UTC.unit3.2	buttonInput.extractBtn == 1	buttonInput.extractBtn = 0 return 1
CMS_UTC.unit4.1	buttonInput.reserveBtn == {x,y,1,z}	buttonInput.reserveBtn = {0,0,0,0} return {0,0,1}
CMS_UTC.unit4.2	buttonInput.reserveBtn == {x,y,0,1}	buttonInput.reserveBtn = {0,0,0,0} return {0,0,-1}

CMS_UTC.unit4.3	buttonInput.reserveBtn == {1,x,0,0}	buttonInput.reserveBtn[0] = 0 return {1,0,0}
CMS_UTC.unit4.4	buttonInput.reserveBtn == {0,1,0,0}	buttonInput.reserveBtn[1] = 0 return {0,1,0}
CMS_UTC.unit4.5	buttonInput.reserveBtn == {0,0,0,0}	return NULL
CMS_UTC.unit5.1	buttonInput.concentrate == 0	return 0
CMS_UTC.unit5.2	buttonInput.concentrate == 1	buttonInput.concentrate = 0 return 1
CMS_UTC.unit6.1	buttonInput.temperature == 0	return 0
CMS_UTC.unit6.2	buttonInput.temperature == 1	buttonInput.temperature = 0 return 1
CMS_UTC.unit7.1	buttonInput.powerBtn == 1	buttonData.powerBtn = 1
CMS_UTC.unit7.2	buttonInput.cleanBtn == 1	buttonData.cleanBtn = 1
CMS_UTC.unit7.3	buttonInput.extractBtn == 1	buttonData.extractBtn = 1
CMS_UTC.unit7.4	buttonInput.concentrate == 1 && buttonData.concentrate == 100	buttonData.concentrate = 200
CMS_UTC.unit7.5	buttonInput.concentrate == 1 && buttonData.concentrate == 200	buttonData.concentrate = 300
CMS_UTC.unit7.6	buttonInput.concentrate == 1 && buttonData.concentrate == 300	buttonData.concentrate = 100
CMS_UTC.unit7.7	buttonInput.temperature == 1 && buttonData.temperature == 1	buttonData.temperature = 0
CMS_UTC.unit7.8	buttonInput.temperature == 1 && buttonData.temperature == 0	buttonData.temperature = 1
CMS_UTC.unit7.9	buttonInput.reserveBtn == {1,x,0,0} && buttonData.reserveInfo[0] == 0	buttonData.reserveInfo[0] = 1
CMS_UTC.unit7.10	buttonInput.reserveBtn == {1,x,0,0} && buttonData.reserveInfo[0] == 1	buttonData.reserveInfo[0] = 0

CMS_UTC.unit7.11	buttonInput.reserveBtn == {0,1,0,0} && buttonData.reserveInfo[1] == 0	buttonData.reserveInfo[1] =15
CMS_UTC.unit7.12	buttonInput.reserveBtn == {0,1,0,0} && buttonData.reserveInfo[1] == 120	buttonData.reserveInfo[1] =135
CMS_UTC.unit7.13	buttonInput.reserveBtn == {0,1,0,0} && buttonData.reserveInfo[1] == 24*60+15	buttonData.reserveInfo[1] = 0
CMS_UTC.unit7.14	buttonInput.reserveBtn == {x,y,1,z}	buttonData.reserveInfo[2] = 1
CMS_UTC.unit7.15	buttonInput.reserveBtn == {x,y,0,1}	buttonData.reserveInfo[2] = -1
CMS_UTC.unit8.1	buttonData.reserveInfo = {0,y,1}	reserveData.mode = 0 reserveData.remainTime = y reserveData.concentrate = buttonData.concentrate reserveData.temperature = buttonData.temperature buttonData.reserveInfo[2] = 0
CMS_UTC.unit8.2	buttonData.reserveInfo = {1,y,1}	reserveData.mode = 1 reserveData.remainTime = y buttonData.reserveInfo[2] = 0
CMS_UTC.unit8.3	buttonData.reserveInfo = {x,y,-1}	reserveData.mode = -1 buttonData.reserveInfo = {0,0,0}
CMS_UTC.unit8.4	buttonData.reserveInfo == {x,y,0} && reserveData.mode != -1 && reserveData.remainTime > 0	reserveData.remainTime -= 1
CMS_UTC.unit9.1	ingredientData.water < 500	"Lack of Water"출력
CMS_UTC.unit9.2	ingredientData.water >= 500	ingredientData.water -= 500
CMS_UTC.unit10.1	None	ingredientData.bean -= 10 ingredientData.coffeePowder += 10
CMS_UTC.unit11.1	None	ingredientData.water -=

		coffeeInfoData.concentrate ingredientData.coffeePowder - = 10
CMS.UTC.unit12.1	command == 0	return 0
CMS.UTC.unit12.2	command == 1	return 0
CMS.UTC.unit12.3	command == 2	return 0
CMS.UTC.unit12.4	command != 0 && command != 1 && command != 2	return -10

8.2 Test items

<Table 2 : Test Design Identification> 참조

8.3 Input specifications

<Table 3 : Test Case Identification> 참조

8.4 Output specifications

<Table 3 : Test Case Identification> 참조

9 Testing tasks

<Table 4 : Testing tasks & Schedule>

Task	Predecessor tasks	Skills	Effort
Unit Test Plan	SRA, SDS 작성	SASD의 이해	
Test Design Specification	Task 1	CMS 이해	4
Test Case Specification	Task 2	CMS 이해	4
Test Execution	Task 3	테스트 코드작성	3
Test Result Report	Task 4		2

10 Environmental needs

Coffee Machine System의 Unit Test를 위한 환경적 요구사항은 다음과 같다.

(1) Hardware & Platform

GCC compiler/linker

(2) CTIP(Continuous Testing & Integrated Platform) Environment

Cygwin

11 Unit Test deliverables

12 Schedules

<Table 4 : Testing tasks & Schedule> 참조