

Unit Testing Plan

for Coffee Machine System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

Class A Team 7

Date

2016-11-03

Team Information

201414134 오세욱

201414136 임현유

201211938 황준익

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

8.3 Input specifications 4

8.4 Output specifications 4

9 Testing tasks 5

10 Environmental needs 5

11 Unit Test deliverables 5

12 Schedules 5

1 Introduction

1.1 Objectives

본 문서는 2016년 건국대학교의 소프트웨어공학 개론 강의의 실습과제를 설명한다. 실습 과제는 Coffee Machine을 소프트웨어로 구성된 가상의 시스템으로 구현하는 것이다.

1.2 Background

Coffee Machine System(이하 CMS)은 실시간으로 사용자의 명령을 받아서 Machine의 상태를 체크한 후 커피를 추출하는 시스템이다.

Unit Test는 시스템을 구성하는 단위 별로 나누어서 테스트하는 것이며, 요구사항을 기준으로 입력과 출력을 정의하여 그 의도대로 올바른 결과가 나오는지 확인하는 것이다.

1.3 Scope

Coffee machine은 사용자의 요청에 따라 Machine의 상태를 체크한 후 커피를 추출한다. 2016SE 수업을 통해 진행하게 될 프로젝트는 Coffee Machine을 가상 시스템으로 구현하는 것으로, Coffee Machine은 자신의 상태를 관리하여 사용자에게 알려주고, 사용자로부터 명령 및 입력 내용을 받아 출력을 만들어내는 역할을 한다. 모든 시스템 SW만으로 구현한다. HW가 필요한 부분은 SW모듈을 만들어 가상의 HW를 구현한다.

1.4 Project plan

CMS는 SRA, SDS를 기반으로 Program source code 및 Unit Test를 위한 Test Code는 Cygwin 환경에서 이루어진다.

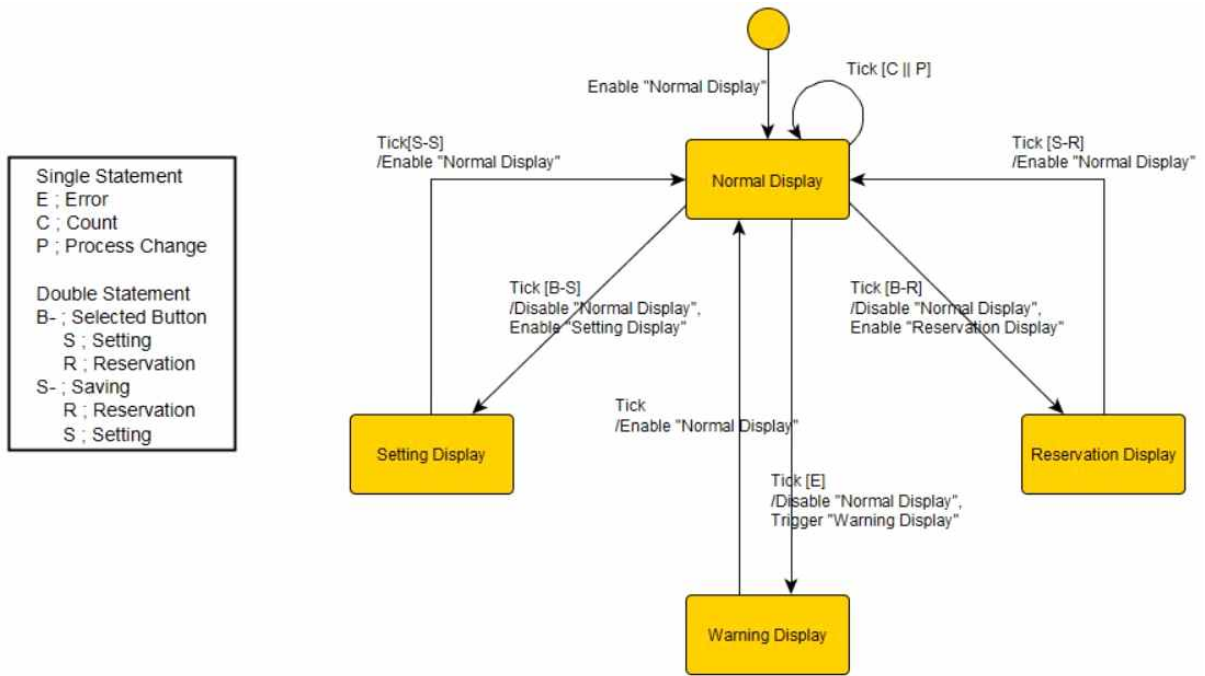
1.5 Configuration management plan

Coffee Machine System(CMS)은 Program Source Code 및 Unit Test를 위한 Test Code는 Cygwin 환경에서 이루어지며, program Source Code 및 Test Code의 변경 및 수정 사항은 지속적으로 통합되고 테스트된다.

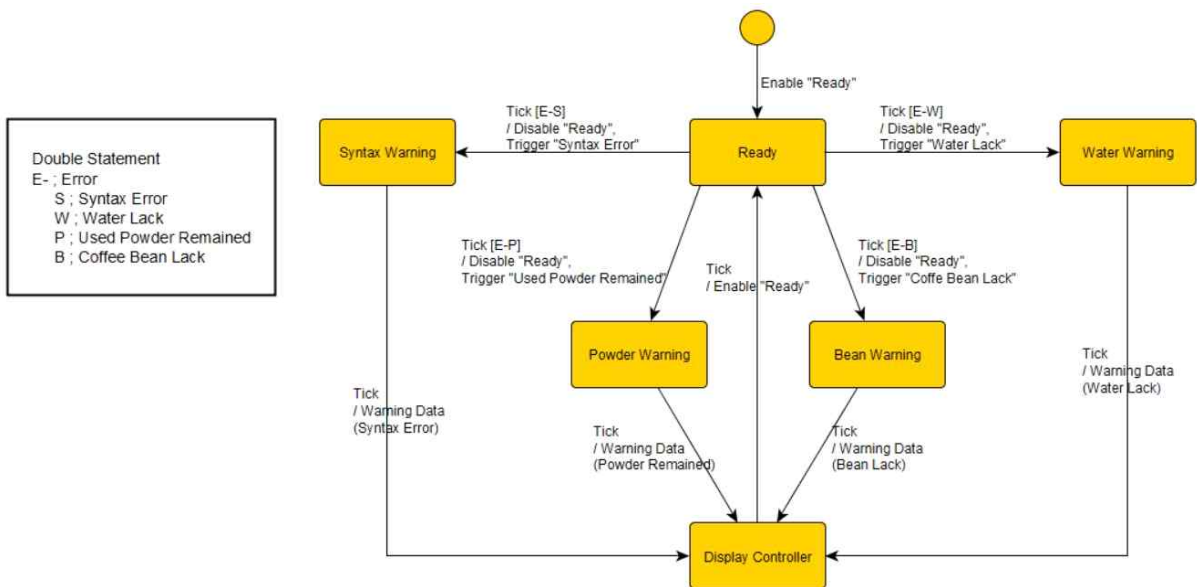
1.6 References

SRA_Team7, SDS_Team7

(3) Display STD



(4) Warning STD



3 Features to be tested

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

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

<Table 1. Testing Process(DFD) List>

ID	Name	Description
1.1	Button Sensor Interface	User 의 버튼 입력을 받아 Controller 에 전달한다.
1.2	Setting / Reservation Sensor Interface	User 의 Setting / Reservation 입력을 받아 Controller 에 전달한다.
1.3	User Input Detection	1.1 과 1.2 를 호출해 User 입력을 받는다.
1.4	Ingredient Sensor Interface	저장된 재료 정보를 읽어 Controller 에 전달한다.
1.5	User Interface	User 의 재료 보충 입력을 받아 Controller 에 전달한다.
1.6	Ingredients Detection	1.4 와 1.5 를 호출해 재료 정보를 받는다.
1.7	Setting Interface	저장된 설정 정보를 읽어 Controller 에 전달한다.
1.8	Current State Interface	저장된 현재 상태 정보를 읽어 Controller 에 전달한다.
1.9	Reservation Interface	저장된 예약 정보를 읽어 Controller 에 전달한다.
1.10	Storage Data Detection	1.7 과 1.8 과 1.9 를 호출해 받아온 정보를 Controller 에 전달한다.
2.1.2	Display Controller	화면 출력될 정보들을 받아 Warning Message를 띄우거나 기본 화면을 띄운다.
2.1.3	Warning Controller	남은 재료 량을 항시 검사하고 구문오류에 대해서 Warning Flag를 조정한다.
2.1.4	Clean	커피 머신의 내부를 물 청소한다.
2.1.5	Make Coffee	커피의 농도 값을 전달받아서 커피를 추출한다.
2.1.6	Grind	커피 추출 요청이 들어왔을 때 커피 추출을 위한 가루가 남아있지 않다면 원두를 분쇄한다.
2.1.7	Boil	따뜻한 커피 추출 요청이 들어온 경우 물을 가열한다.
2.1.10	Save Reservation	예약 정보 (Work, Time)을 저장하는 프로세스
2.1.2.1	Normal Display	현재 커피 머신에서 가지고 있는 정보들을 전달 받아 화면에 출력한다.

4 Features not to be tested

(1) Process in SRA : 외부 장치 드라이버, 단순 데이터 전달 & 저장 프로세스 등은 Test에서 제외한다.

(2) Modules in SDS

<Table 2. 테스트하지 않을 Process(DFD) List>

ID	Name	Description
2.1.8	Save Setting	설정 값(Temperature , Concentration)을 받아 txt파일에 저장하는 Process.
2.1.9	Save Current State	현재 상태(Work, Button)정보를 받아 txt파일에 저장하는 Process.
2.1.10	Save Reservation	예약 정보(Work, Time)들을 리스트형태로 저장하는 Process.
2.1.11	Terminate	CMS의 전원이 꺼진 상태 Process.
2.1.3.1	Water Lack	warning_flag값에 따라 적절히 처리되는 Process
2.1.3.2	Coffee Bean Lack	
2.1.3.3	Used Powder Remained	
2.1.3.4	Syntax Error	

5 Approach

Coffee Machine System의 Unit Test는 Cygwin 환경에서 이루어지며 Unit Test를 위하여 simplectest라는 도구를 사용해서 진행하였다. Program Source Code의 변경 및 수정 사항은 지속적으로 통합되고 테스트된다.

6 Item pass/fail criteria

각 모듈은 요구사항을 모두 만족하여야 한다. 입력 값에 따른 출력 값의 결과가 같아야 한다.

7 Unit test design specification

7.1 Test design specification identifier

CMS_UTP_000

7.2 Features to be tested

<Table 1. Testing Process(DFD) List> 참조

7.3 Approach refinements

White Box Testing : Coffee Machine System의 각 모듈이 요구사항을 만족하는지 확인하기 위하여, 요구사항에 정의된 내용에 기반하여 Test code를 작성하고, 코드의 흐름에서 모든 진행을 테스트한다. 예를 들어 조건문이 있을 때 조건문을 모두 거쳐갈 수 있도록 테스트 가능한 여러 경우의 수를 전달하여 함수 내 모든 코드가 정상 작동 되는지 테스트한다.

7.4 Test identification

Identifier	Feature (Process ID in DFD)	Valid / Invalid Value
CMS_UTP_000	1.1 Button Sensor Interface	User 로부터 0 ~ 5 의 값 (버튼 입력) 을 받아서 반환, 범위 내의 값이 입력될 경우 -1 로 변환 후 반환
CMS_UTP_001	1.2 Setting / Reservation Interface	Flag 값을 받아서 3 일 경우와 4일 경우로 나눠 프로세스 진행, 정상 입력일 경우 입력 값을 반환하고, 정상 입력이 아닌 경우 값을 변환 후 반환
CMS_UTP_002	1.3 User Input Detection	Flag 값을 받아서 3이나 4인 경우 1.2 프로세스를 호출하고 그 외 1.1 프로세스를 호출한다.
CMS_UTP_003	1.4 Ingredient Sensor Interface	Ingredient.txt 파일을 열어 저장된 값 읽어서 반환
CMS_UTP_004	1.5 User Interface	Powder 값을 받아서 powder 값에 따라 프로세스 진행, 유저 입력을 받아 정상 입력일 경우 입력 값 반환, 정상 입력이 아닌 경우 값을 변환 후 반환
CMS_UTP_005	1.6 Ingredient Detection	Flag 값을 받아서 1인 경우 1.4 호출 후 1.5 를 호출하고, 0인 경우 1.4만 호출. 프로세스 호출 후 반환된 값을 ingredient.txt 에 저장하고 반환
CMS_UTP_006	1.7 Setting Interface	Setting.txt 파일을 열어 저장된 값 읽어서 반환
CMS_UTP_007	1.8 Current State	State.txt 파일 열어서 저장된 값 저장,

	Interface	저장된 값과 진행 대기중인 값을 읽어 값에 따라 프로세스 진행. 프로세스 종료 후 저장된 값 반환
CMS_UTP_008	1.9 Reservation Interface	예약된 작업 정보와 Digital Clock Tick을 받아서 Time 이 일치하는 경우 작업 반환
CMS_UTP_009	1.10 Storage Data Detection	1.7 과 1.8 프로세스를 호출하고 저장된 값에 따라 1.9 호출 여부 결정 저장된 값을 반환
CMS_UTP_010	2.1.2 Display Controller	Warning_flag.txt에서 flag값을 읽고 0이 아닌 값에 대하여 적절한 Warning Message를 화면에 알린다. flag값이 0이라면 현재 커피 머신 정보를 출력한다.
CMS_UTP_011	2.1.4 Clean	청소 요청이 들어온 경우 수행되는 Process. 현재 재료 값에서 물 500ml를 소비하고, 청소를 진행하는 동안 현재 작업정보 -> 4(청소), 버튼명령 -> 2(청소)로 바꿔준다.
CMS_UTP_012	2.1.5 Make Coffee	커피 추출 요청이 들어온 경우 수행되는 Process. 농도 설정 값을 입력 받아 현재 재료 값에서 농도만큼의 물을 소비하고, Coffee가 추출되는 동안 현재 작업정보 ->3(추출), 버튼명령 ->1(추출)로 바꿔준다.
CMS_UTP_013	2.1.6 Grind	커피 추출 요청이 들어온 경우 사용할 커피가루가 없을 때 수행되는 Process. 원두 10g을 소비하고, Grind가 진행되는 동안 현재 작업정보 ->1(분쇄), 버튼명령 ->1(추출)로 바꿔준다.
CMS_UTP_014	2.1.7 Boil	따뜻한 커피 추출 요청이 들어온 경우 수행되는 Process. Boil이 진행되는 동안 현재 작업정보 ->2(가열), 버튼명령 ->1(추출)로 바꿔준다.
CMS_UTP_015	2.1.2.1 Normal Display	txt파일로부터 현재 Display할 정보들을 읽어와 화면에 출력하는 Process.
CMS_UTP_016	2.1.3 Warning Controller	ingredient.txt에서 재료 량을 state.txt에서 현재 작업상태를 setting.txt에서 커

		피 종류를 각각 받아와 warning 조건에 대해 검사하여 warning_flag.txt에 오류 종류 대로 0(정상), 1(물부족), 2(원두부족), 3(찌꺼기잔류), 4(구문에러)를 출력한다.
--	--	--

7.5 Feature pass/fail criteria

CMS의 각 모듈은 SRA에 정의되어 있는 요구사항(입출력, 동작)을 모두 만족해야 한다. 각 프로세스의 입출력 및 동작은 SRA의 Process description 항목 및 State Transition Diagram을 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

Identifier	Input Specification	Output Specification
CMS_UTP_000_000	1.1 Button Sensor Interface	정상 범위 내 (0 ~ 5) 입력 - 입력값 반환 (입력 ; 3)
CMS_UTP_000_001	1.1 Button Sensor Interface	정상 범위 외 !(0 ~ 5) 입력 - 입력값 변환 (-1) 후 반환 (입력 ; 8)
CMS_UTP_001_000	1.2 Setting / Reservation Sensor Interface	flag == 3 첫 번째 입력 (1 ~ 2) 두 번째 입력 (1 ~ 3) - 입력값 배열로 반환 (입력 ; 1, 2)
CMS_UTP_001_001	1.2 Setting / Reservation Sensor Interface	flag == 3 첫 번째 입력 !(1 ~ 2) 두 번째 입력 (1 ~ 3) - 반환 배열의 첫 번째 원소를 0으로 변환 후 반환, warning_flag.txt 파일 생성해 4 입력 (입력 ; 5, 3)
CMS_UTP_001_002	1.2 Setting / Reservation Sensor Interface	flag == 3 첫 번째 입력 (1 ~ 2) 두 번째 입력 !(1 ~ 3) - 반환 배열의 두 번째 원소를 0으로 변환 후 반환, warning_flag.txt 파일 생성해 4 입력 (입력 ; 2, 4)

CMS_UTP_001_003	1.2 Setting / Reservation Sensor Interface	flag == 3 첫 번째 입력 !(1 ~ 2) 두 번째 입력 !(1 ~ 3) - 반환 배열의 첫 번째, 두 번째 원소를 0 으로 변환 후 반환, warning_flag.txt 파일 생성해 4 입력 (입력 ; 8, 20)
CMS_UTP_001_004	1.2 Setting / Reservation Sensor Interface	flag == 4 첫 번째 입력 (1 ~ 2) 두 번째 입력 (0000 ~ 2359) - 입력값 배열로 반환 (입력 ; 1, 1823)
CMS_UTP_001_005	1.2 Setting / Reservation Sensor Interface	flag == 4 첫 번째 입력 !(1 ~ 2) 두 번째 입력 (0000 ~ 2359) - 반환 배열의 첫 번째 원소를 0 으로 변환 후 반환, warning_flag.txt 파일 생성해 4 입력 (입력 ; -1 0824)
CMS_UTP_001_006	1.2 Setting / Reservation Sensor Interface	flag == 4 첫 번째 입력 (1 ~ 2) 두 번째 입력 !(0000 ~ 2359) - 반환 배열의 두 번째 원소를 -1 으로 변환 후 반환, warning_flag.txt 파일 생성해 4 입력 (입력 ; 2, 2580)
CMS_UTP_001_007	1.2 Setting / Reservation Sensor Interface	flag == 4 첫 번째 입력 !(1 ~ 2) 두 번째 입력 !(1 ~ 3) - 반환 배열의 첫 번째 원소를 0 으로, 두 번째 원소를 -1 으로 변환 후 반환, warning_flag.txt 파일 생성해 4 입력 (입력 ; 5, 8888)
CMS_UTP_002_000	1.3 User Input Detection	iflag == 3 flag == 4 1.2 프로세스 호출 후 해당 반환값 반환 (입력 ; 2, 1)
CMS_UTP_002_001	1.3 User Input Detection	flag != 3 && flag != 4 1.1 프로세스 호출 후 해당 반환값 반

		환 (입력 ; 3)
CMS_UTP_003_000	1.4 Ingredient Sensor Interface	Ingredient.txt 파일을 열어서 저장된 3개의 정수값 반환 (txt ; 200 30 1)
CMS_UTP_004_000	1.5 User Interface	powder == 0 첫 번째 입력 (1 ~ 10) 두 번째 입력 (1 ~ 10) 세 번째 입력 'y' 'Y' - 입력값 배열로 반환, powder 값 1로 변경 (입력 ; 3, 2)
CMS_UTP_004_001	1.5 User Interface	powder == 0 첫 번째 입력 (1 ~ 10) 두 번째 입력 (1 ~ 10) 세 번째 입력 'n' 'N' - 입력값 배열로 반환, powder 값 유지 (입력 ; 9, 1)
CMS_UTP_004_002	1.5 User Interface	powder == 0 첫 번째 입력 !(1 ~ 10) 두 번째 입력 (1 ~ 10) 세 번째 입력 'y' 'Y' - 반환 배열의 첫 번째 원소 0으로 변환 후 배열 반환, powder 값 1로 변경, warning_flag.txt 파일 생성해 4 입력 (입력 ; -3, 6)
CMS_UTP_004_003	1.5 User Interface	Powder == 0 첫 번째 입력 (1 ~ 10) 두 번째 입력 !(1 ~ 10) 세 번째 입력 'y' 'Y' - 반환 배열의 두 번째 원소 0으로 변환 후 배열 반환, powder 값 1로 변경, warning_flag.txt 파일 생성해 4 입력 (입력 ; 8 14)
CMS_UTP_004_004	1.5 User Interface	powder == 0 첫 번째 입력 !(1 ~ 10) 두 번째 입력 !(1 ~ 10) 세 번째 입력 'n' 'N'

		- 반환 배열의 첫 번째, 두 번째 원소 0으로 변환 후 배열 반환, powder 값 유지, warning_flag.txt 파일 생성해 4 입력 (입력 ; 21, -88)
CMS_UTP_004_005	1.5 User Interface	powder == 1 첫 번째 입력 (1 ~ 10) 두 번째 입력 (1 ~ 10) 세 번째 입력 'y' 'Y' - 입력값 배열로 반환, powder 값 0으로 변경 (입력 ; 9 1)
CMS_UTP_004_006	1.5 User Interface	powder == 1 첫 번째 입력 (1 ~ 10) 두 번째 입력 (1 ~ 10) 세 번째 입력 'n' 'N' - 입력값 배열로 반환, powder 값 유지 (입력 ; 2, 2)
CMS_UTP_004_007	1.5 User Interface	powder == 1 첫 번째 입력 !(1 ~ 10) 두 번째 입력 (1 ~ 10) 세 번째 입력 'y' 'Y' - 반환 배열의 첫 번째 원소 0으로 변환 후 배열 반환, powder 값 0으로 변경, warning_flag.txt 파일 생성해 4 입력 (입력 ; 89, 10)
CMS_UTP_004_008	1.5 User Interface	powder == 1 첫 번째 입력 (1 ~ 10) 두 번째 입력 !(1 ~ 10) 세 번째 입력 'y' 'Y' - 반환 배열의 두 번째 원소 0으로 변환 후 배열 반환, powder 값 0으로 변경, warning_flag.txt 파일 생성해 4 입력 (입력 ; 3, 22)
CMS_UTP_004_009	1.5 User Interface	powder == 1 첫 번째 입력 !(1 ~ 10) 두 번째 입력 !(1 ~ 10)

		<p>세 번째 입력 'n' 'N'</p> <p>- 반환 배열의 첫 번째, 두 번째 원소 0으로 변환 후 배열 반환, powder 값 유지,</p> <p>warning_flag.txt 파일 생성해</p> <p>4 입력 (입력 ; 100, -100)</p>
CMS_UTP_004_010	1.5 User Interface	<p>powder == 2</p> <p>첫 번째 입력 (1 ~ 10)</p> <p>두 번째 입력 (1 ~ 10)</p> <p>세 번째 입력 'y' 'Y'</p> <p>- 입력값 배열로 반환, powder 값 0으로 변경 (입력 ; 1 1)</p>
CMS_UTP_004_011	1.5 User Interface	<p>powder == 2</p> <p>첫 번째 입력 (1 ~ 10)</p> <p>두 번째 입력 (1 ~ 10)</p> <p>세 번째 입력 'n' 'N'</p> <p>- 입력값 배열로 반환, powder 값 유지 (입력 ; 3 3)</p>
CMS_UTP_004_012	1.5 User Interface	<p>powder == 2</p> <p>첫 번째 입력 !(1 ~ 10)</p> <p>두 번째 입력 (1 ~ 10)</p> <p>세 번째 입력 'y' 'Y'</p> <p>- 반환 배열의 첫 번째 원소 0으로 변환 후 배열 반환, powder 값 0으로 변경,</p> <p>warning_flag.txt 파일 생성해</p> <p>4 입력 (입력 ; 99 10)</p>
CMS_UTP_004_013	1.5 User Interface	<p>powder == 2</p> <p>첫 번째 입력 (1 ~ 10)</p> <p>두 번째 입력 !(1 ~ 10)</p> <p>세 번째 입력 'y' 'Y'</p> <p>- 반환 배열의 두 번째 원소 0으로 변환 후 배열 반환, powder 값 0으로 변경,</p> <p>warning_flag.txt 파일 생성해</p> <p>4 입력 (입력 ; 8 14)</p>
CMS_UTP_004_014	1.5 User Interface	<p>powder == 2</p> <p>첫 번째 입력 !(1 ~ 10)</p>

		<p>두 번째 입력 !(1 ~ 10) 세 번째 입력 'n' 'N' - 반환 배열의 첫 번째, 두 번째 원소 0으로 변환 후 배열 반환, powder 값 유지, warning_flag.txt 파일 생성해 4 입력 (입력 ; 15 15)</p>
CMS_UTP_005_000	1.6 Ingredients Detection	<p>flag == 1 1.4 호출 후 반환 배열의 세 번째 원소를 인자로 1.5 호출, ret[0] <= 1000 && ret[1] <= 100 - ingredient.txt 파일 생성해 배열 값 저장, 배열 반환 (txt ; 300 20 1 / 입력 ; 2 3 'n')</p>
CMS_UTP_005_001	1.6 Ingredients Detection	<p>flag == 1 1.4 호출 후 반환 배열의 세 번째 원소를 인자로 1.5 호출, ret[0] <= 1000 && ret[1] > 100 - ret[1] 값 100으로 변환 후 ingredient.txt 파일 생성해 배열 값 저장, 배열 반환, warning_plag.txt 파일 생성해 4 저장 (txt ; 300 20 1 / 입력 ; 3 9 'n')</p>
CMS_UTP_005_002	1.6 Ingredients ;Detection	<p>flag == 1 1.4 호출 후 반환 배열의 세 번째 원소를 인자로 1.5 호출, ret[0] > 1000 && ret[1] <= 100 - ret[0] 값 1000 으로 변환 후 ingredient.txt 파일 생성해 배열 값 저장, 배열 반환, warning_plag.txt 파일 생성해 4 저장 (txt ; 300 20 1 / 입력 ; 8 4 'n')</p>

CMS_UTP_005_003	1.6 Ingredients Detection	flag == 0 1.4 호출 - 배열 반환 (txt ; 300 20 1)
CMS_UTP_006_000	1.7 Setting Interface	setting.txt 파일 열어서 저장된 정수 2개 반환 (txt ; 1 2)
CMS_UTP_007_000	1.8 Current State Interface	state.txt 파일 열어서 저장된 정수 2개 저장 ret[0] == 0! && ret[1] == 0 && progress != empty Queue 의 첫 번째 값 ret[2] 에 저장 후 배열 반환 (txt ; 0 0 / Node ; 1)
CMS_UTP_007_001	1.8 Current State Interface	state.txt 파일 열어서 저장된 정수 2개 저장 ret[0] ==; 0 && ret[1] == 0 && progress == empty ret[2] 에 0 저장 후 배열 반환 (txt ; 0 0 / Node ; empty)
CMS_UTP_007_002	1.8 Current State Interface	state.txt 파일 열어서 저장된 정수 2개 저장 ret[0] != 0 && ret[1] == 0 ret[2] 에 0 저장 후 배열 반환 (txt ; 3 0)
CMS_UTP_007_003	1.8 Current State Interface	state.txt 파일 열어서 저장된 정수 2개 저장 ret[0] == 0 && ret[1] != 0 ret[2] 에 0 저장 후 배열 반환 (txt ; 0 2)
CMS_UTP_007_004	1.8 Current State Interface	state.txt 파일 열어서 저장된 정수 2개 저장 ret[0] != 0 && ret[1] != 0 ret[2] 에 0 저장 후 배열 반환 (txt ; 5 1)
CMS_UTP_008_000	1.9 Reservation Interface	reservation != empty && reservation time

		예약 작업 반환 (Node ; 1-1820 / time ; 1820)
CMS_UTP_008_001	1.9 Reservation Interface	reservation != empty && ! reservation time 0 반환 (Node ; 2-1740 / time ; 0800)
CMS_UTP_008_002	1.9 Reservation Interface	reservation == empty 0 반환 (Node ; empty / time ; 0720)
CMS_UTP_009_000	1.10 Storage Data Detection	1.7 프로세스 호출 후 2개의 정수 배열에 저장 1.8 프로세스 호출 후 3개의 정수 배열에 저장 ret[2] != 0 ret[4] != 0 && reservation != 0 - 1.9 프로세스 호출 후 반환 값 progress queue 에 삽입, ret[5] 에 0 저장 후 배열 반환 (setting.txt ; 1 2 / state.txt ; 1 1 / reservation ; 2)
CMS_UTP_009_001	1.10 Storage Data Detection	1.7 프로세스 호출 후 2개의 정수 배열에 저장 1.8 프로세스 호출 후 3개의 정수 배열에 저장 ret[2] != 0 ret[4] != 0 && reservation != 0 - ret[5] 에 0 저장 후 배열 반환 (setting.txt ; 1 2 / state.txt ; 1 1 / reservation ; 1)
CMS_UTP_009_002	1.10 Storage Data Detection	1.7 프로세스 호출 후 2개의 정수 배열에 저장 1.8 프로세스 호출 후 3개의 정수 배열에 저장 ret[2] == 0 && ret[4] == 0 && reservation == 0 - 1.9 프로세스 호출 후 반환값 ret[5] 에 저장, 배열 반환 (setting.txt ; 2 3 / state.txt ; 0 0 /

		Reservation ; 0)
CMS_UTP_009_003	1.10 Storage Data Detection	1.7 프로세스 호출 후 2개의 정수 배열에 저장 1.8 프로세스 호출 후 3개의 정수 배열에 저장 ret[2] == 0 && ret[4] == 0 && reservation != 0 - 1.9 프로세스 호출 후 반환 값 ret[5] 에 저장, 배열 반환 (setting.txt ; 2 3 / state.txt ; 0 0 / Reservation ; 2)
CMS_UTP_010_000	w_flag = 1	"Water Lack" Message 출력
CMS_UTP_010_001	w_flag = 2	"Bean Lack" Message 출력
CMS_UTP_010_002	w_flag = 3	"Coffee Powder Remained" Message 출력
CMS_UTP_010_003	w_flag = 4	"Syntax Error" Message 출력
CMS_UTP_011_000	Call Clean (2.1.4)	Ingredient.txt -> (water bean powder) wait : 10 sec & state.txt -> (4 2) ingredient.txt -> (w b-500 p) & state.txt -> (0 0)
CMS_UTP_012_000	Coffee(1)	Ingredient.txt -> (w b p) wait : 10 sec & state.txt -> (3 1) ingredient.txt -> (w b-300 p) & State.txt -> (0 1)
CMS_UTP_012_001	Coffee(2)	Ingredient.txt -> (w b p) wait : 10 sec & state.txt -> (3 1) ingredient.txt -> (w b-200 p) & State.txt -> (0 1)
CMS_UTP_012_002	Coffee(3)	Ingredient.txt -> (w b p) wait : 10 sec & state.txt -> (3 1) ingredient.txt -> (w b-100 p) & State.txt -> (0 1)
CMS_UTP_013_000	Call Grind	Ingredient.txt -> (w b p) wait : 5 sec & state.txt -> (1 1) ingredient.txt -> (w b 1) & state.txt -> (0 1)
CMS_UTP_014_000	Call Boil	wait : 10 sec & state.txt -> (2 1)

		state.txt -> (0 1)
CMS_UTP_015_000	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 1, 1, 0)	Current State : Ready Selected Concentration : Light Selected Temperature : Cold
CMS_UTP_015_001	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 2, 2, 1)	Current State : Grinding Selected Concentration : Normal Selected Temperature : Hot
CMS_UTP_015_002	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 3, 2, 2)	Current State : Boiling Selected Concentration : Strong Selected Temperature : Cold
CMS_UTP_015_003	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 3, 2, 3)	Current State : Extracting Selected Concentration : Strong Selected Temperature : Cold
CMS_UTP_015_004	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 3, 2, 4)	Current State : Cleaning Selected Concentration : Strong Selected Temperature : Cold
CMS_UTP_015_005	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 3, 2, 5)	Current State : Setting Selected Concentration : Strong Selected Temperature : Cold
CMS_UTP_015_006	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 3, 2, 6)	Current State : Reservating Selected Concentration : Strong Selected Temperature : Cold
CMS_UTP_015_007	normalDisplay(t, water, bean, powder, cf_rtime, cl_rtime, 3, 2, 7)	Current State : Refilling Selected Concentration : Strong Selected Temperature : Cold
CMS_UTP_016_000	ingredient.txt -> 500 10 0 setting.txt-> 1 3 state.txt-> 3 1	warning_flag.txt->0
CMS_UTP_016_001	ingredient.txt -> 500 10 0 setting.txt-> 1 3 state.txt-> 3 1 warning_flag.txt->4	warning_flag.txt->4->(2seconds)->0
CMS_UTP_016_002	ingredient.txt -> 299 10 0 setting.txt-> 1 1	warning_flag.txt->1

	state.txt-> 3 1	
CMS_UTP_016_003	ingredient.txt -> 199 10 0 setting.txt-> 1 2 state.txt-> 3 1	warning_flag.txt->1
CMS_UTP_016_004	ingredient.txt -> 99 10 0 setting.txt-> 1 3 state.txt-> 3 1	warning_flag.txt->1
CMS_UTP_016_005	ingredient.txt -> 500 5 0 setting.txt-> 1 3 state.txt-> 3 1	warning_flag.txt->2
CMS_UTP_016_006	ingredient.txt -> 500 5 1 setting.txt-> 1 3 state.txt-> 3 1	warning_flag.txt->0
CMS_UTP_016_007	ingredient.txt -> 500 50 2 setting.txt-> 1 3 state.txt-> 3 1	warning_flag.txt->3
CMS_UTP_016_008	ingredient.txt -> 499 50 2 setting.txt-> 1 3 state.txt-> 3 2	warning_flag.txt->1

8.2 Test items

8.1 Test case identification 참조

8.3 Input specifications

8.1 Test case identification 참조

8.4 Output specifications

8.1 Test case identification 참조

9 Testing tasks

<Table5 Testing tasks & Schedule>

Task	Predecessor	Special Skills	Effort	Finish Date
1. Unit Test Plan 작성	SRS 작성 SRA 작성 SDS 작성 CMS 구현			
2. Test design specification	Task1	CMS에 대한 이해		
3. Test case specification	Task2	CMS에 대한 이해		
4. Test Execution	Task3	Test Code 작성 Test Tool에 대한 이해		
5. Test Result Reoprt	Task4			

10 Environmental needs

CMS의 Unit Test를 위한 환경적 요구사항은 다음과 같다.

- (1) Hardware & Platform : GCC compiler/linker
- (2) CTIP Environment : Cygwin
- (3) Unit Test Tool : Simplectest

11 Unit Test deliverables

12 Schedules

<Table5 Testing tasks & Schedule> 참조