

# Distributed Vending Machine

**Project Team**

**T4**

**Team Date**

**2021-03-31**

-----  
**Team Information**

201711300 류창현  
201711314 송주한  
201911231 강현수  
201915005 남승협

# ACTIVITY 1001 - DEFINE DRAFT PLAN

## 1. Motivation

VM(Vending Machine)은 사람 없이도 상품을 자동적으로 판매하는 기계로 금액을 결제하고 원하는 물품을 선택하면 사려는 물품을 제공하는 자동 판매기이다. 일반적인 VM의 경우, 원하는 상품이 없거나 재고가 모두 소진된 경우 소비자가 원하는 서비스를 제공하기 힘들다는 문제점이 있다. DVM은 이러한 VM의 기존 단점을 개선하기 위해 여러 VM 간의 네트워크를 구성한 SW를 개발하려 한다.

## 2. Project Objectives

- 기존 자판기의 문제점을 개선하기 위해서 각 자판기 모두 네트워크로 연결되어 **Description** 있는 자판기 시스템을 구축한다.
- 네트워크를 통해 다른 자판기와 정보를 주고 받아 상품이나 재고가 없는 경우 다른 자판기의 위치를 안내한다.

## 3. Functional Requirements

- 사용자가 음료를 선택 후 결제하면 음료가 제공된다.
- 재고가 부족하거나 판매하지 않는 제품의 경우 다른 자판기에 재고를 확인한 후 위치를 안내한다.
- 다른 자판기의 음료 구매를 하는 경우 선결제가 가능하며 해당 자판기에서 인증코드 통하여 음료를 제공받을 수 있다.
- 각 자판기는 모두 네트워크에 연결되어 있고 메시지를 통해 서로의 정보를 주고 받을 수 있다.

## 4. Non-Functional Requirements

- SW 와 네트워크가 안정적으로 동작해야 한다.
- 다른 팀프로젝트에 자료가 유출되면 안 된다.

## 5. Resource Estimation

- **Human Efforts (Man-Month) : 3MM**
- **Human Resources: 4 명**
- **Duration : 10 weeks**
- **Budget :  $5000 \times 200 \times 3 = 3000000$**

# ACTIVITY 1002 - CREATE PRELIMINARY INVESTIGATION REPORT

## 1. Alternative Solutions

- 현존하는 자판기 SW 를 구매한다.
- SW 개발 업체에 외주로 의뢰한다.

## 2. Project Justification

- **Cost** : 학생들의 열정페이로 시스템 구매 혹은 외주보다 값이 저렴하다.
- **Duration** : 2021/06/14 까지 개발이 완료되어야 한다.
- **Risk** :
  - a. 문서 작성 능력 부족
  - b. OOAD 및 OOPT 에 대한
  - c. 타 과목 프로젝트, 시험으로 인한 시간 부족
  - d. 개발 능력 부족
  - e. 코로나로 인한 소통 부재

## 3. Risk Management

Risk	probability	significance	Weight
문서 작성 능력 부족	3	4	12
OOAD 및 OOPT 에 대한 개념 부족	4	4	16
타 과목 프로젝트, 시험으로 인한 시간 부족	5	5	25
개발 능력 부족	5	4	20
코로나로 인한 소통 부재	3	3	9

#### 4. Risk Reduction Plan

Risk	Way of Reduction
문서 작성 능력 부족	팀원끼리 서로 문서를 검수하며 문장력을 올린다.
OOAD 및 OOPT 에 대한 개념 부족	예전 자료를 참조하거나 검색 혹은 교수님에게 자문을 구한다.
타 과목 프로젝트, 시험으로 인한 시간 부족	계획적으로 팀프로젝트를 진행하는 등 시간을 효율적으로 사용한다.
개발 능력 부족	구글링 및 팀원들과 협조를 통해 해결해 나간다.
코로나로 인한 소통 부재	Zoom 이나 디스코드 등을 사용하여 팀원과 토의한다.

#### 5. Market Analysis

- 현존하는 다른 재판기 SW 는 재판기간의 네트워크 연결을 지원하지 않는다.
- 음료 재판기만이 아닌 여타 다른 재판기에도 이 SW 를 제공할 수 있다.

#### 6. Other Managerial Issues

- 해당 프로젝트를 2021 년 6 월 14 일까지 마무리 지어야 한다.

## ACTIVITY 1003 - DEFINE REQUIREMENTS

- 총 음료의 개수는 20 종류이다.
- 한 자판기는 7 종류의 음료를 판매한다.
- 사용자가 음료를 선택한 후 결제하면 음료가 제공된다. 판매하지 않는 음료도 메뉴는 제공한다. 결제는 카드로 한다.
- 잔액이 부족한 경우 결제되지 않는다.
- 구매하려는 음료의 재고가 부족한 경우 다른 자판기에 재고를 확인한 후 위치를 안내한다.
- 현재 자판기에서 판매하지 않는 음료를 구매하려는 경우 판매하는 다른 자판기를 확인 후 위치를 안내한다.
- 네트워크 상의 자판기에 broadcast msg 를 통해 재고 확인을 요청하여 확인하다.
- 네트워크 상의 자판기에 broadcast msg 를 통해 판매 음료 확인을 요청하여 확인한다.
- 네트워크 msg 를 통해 대상 자판기의 위치를 확인하여 안내한다.
- 다른 자판기의 음료 구매에 대해서 결제를 할 수 있다.
- 선결제를 요청하는 경우 결제 후 인증코드를 발급한다.
- 인증코드를 입력하여 음료를 제공받을 수 있다.
- 관리자모드를 통하여 상품의 종류와 가격을 변경할 수 있다.
- 관리자모드를 통하여 자동판매기의 개수, 위치를 변경할 수 있다.

### 1. Functional Requirements(Categorized Table)

Ref.#	Function	Category
R1.1	Show Menu	Hidden
R1.2.1	Select Product	Evident
R1.2.2	Code Input	Evident
R1.3	Check Stock	Hidden
R1.4	Update Stock	Hidden
R1.5	Show VM	Hidden
R1.6	Select VM	Evident
R1.7	Cancel Process	Evident
R1.8	Card Payment	Hidden
R1.9.1	Give Product	Hidden
R1.9.2	Give Code	Hidden
R2.1	Check Info	Hidden

R2.2	Receive Request	Hidden
R2.3	Notify VM's Info	Hidden
R2.4	Confirm Prepay	Hidden
R2.5	Receive Request	Hidden
R3.1	Enter Admin mode	Hidden
R3.2	Edit Product Information	Hidden
R.3.3	Edit VM	Evident
R3.4	Edit Contact	Evident

## 2. Performance Requirements

- 버튼을 눌렀을 때 1 초 안에 동작이 완료되어 다음 동작으로 넘어가야 한다.
- 다른 자판기에 응답을 요청받았을 경우 5 초안에 회답해야 한다.

## 3. Operating Environment

- Microsoft Windows 10 이상

## 4. Interface Requirements

## 5. Other Requirements

- 직관적이고 깔끔한 UI 를 지원한다..

## ACTIVITY 1004 - RECORD TERMS IN GLOSSARY

Term	Description
User	자판기 사용자
Code	선결제 인증코드
Card	결제에 사용하는 카드
Balance	카드 잔액
Price	음료수 가격
Name	음료수 이름
Stock	음료수 재고
Menu	음료수 종류
VM	자판기
Admin	자판기 관리자
ID	자판기 ID
otherVM	다른 자판기

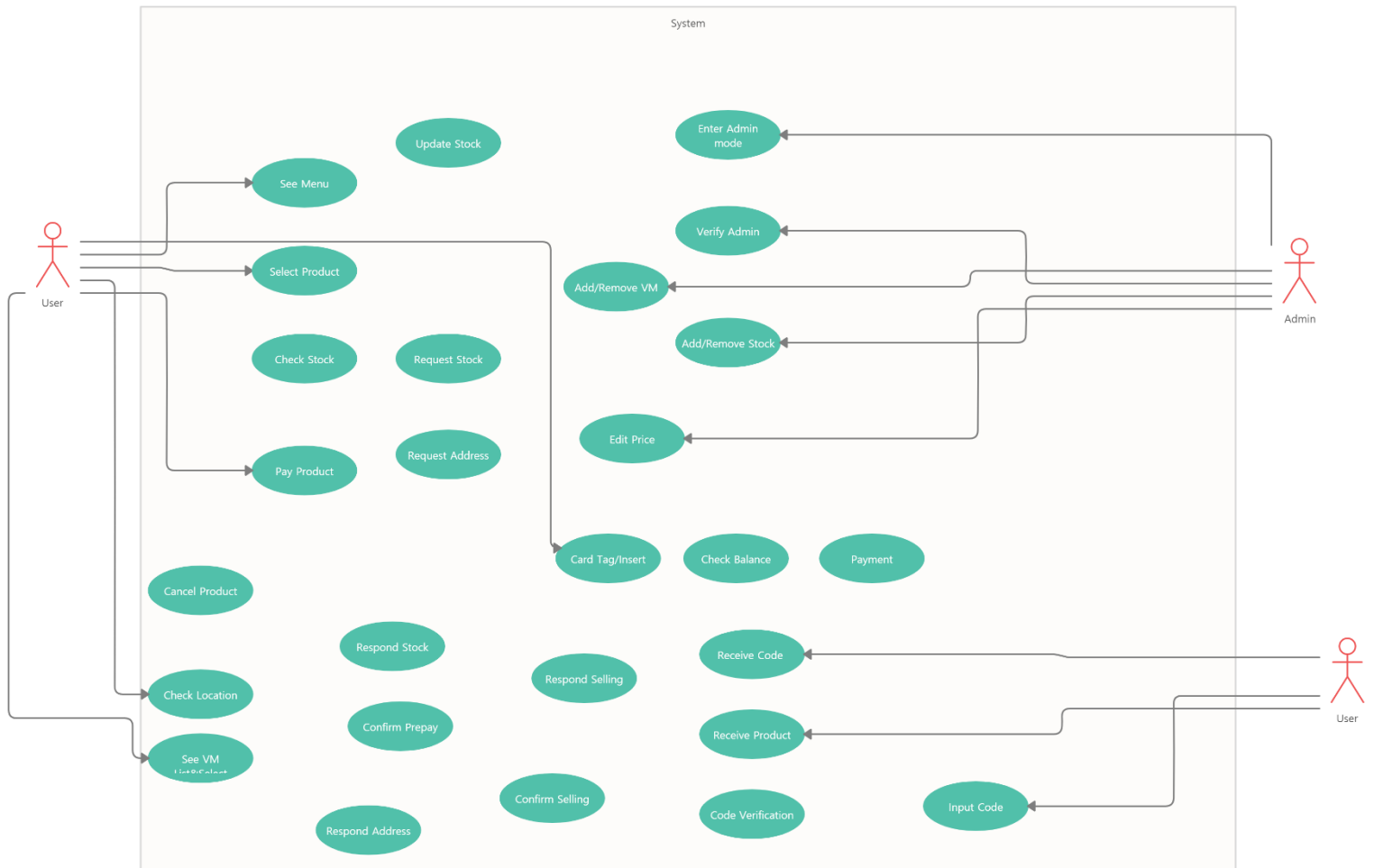
# ACTIVITY 1006 - DEFINE BUSINESS USE CASE

## 1. Categorize Use Cases

Ref.#	Function	Use Case Num & Name	Category
R1.1	Show Menu	1. Show Menu	Hidden
R1.2.1	Select Product	2. Select Product	Evident
R1.2.2	Code Input	3. Input Code	Evident
R1.3	Check Stock	4. Check Stock	Hidden
R1.4	Update Stock	5. Update Stock	Hidden
R1.5	Show VM	6. Show VM List	Hidden
R1.6	Select VM	7. Select VM	Evident
R1.7	Cancel Process	8. Cancel Process	Evident
R1.8	Card Payment	9. Payment	Hidden
R1.9.1	Give Product	10. Give Product	Hidden
R1.9.2	Give Code	11. Give Code	Hidden
R2.1	Check Info	12. Check otherVM's Info	Hidden
R2.2	Receive Request	13. Receive Request	Hidden
R2.3	Notify VM's Info	14. Notify VM's Info	Hidden
R2.4	Confirm Prepay	15. Confirm Prepay	Hidden
R2.5	Receive Request	16. Receive Request	Hidden
R3.1	Enter Admin mode	17. Enter Admin Mode	Evident
R3.2	Edit Product Information	18. Edit Product Information	Evident
R.3.3	Edit VM	19. Add/Remove VM	Evident
R3.4	Edit Contact	20. Edit Contact	Evident



## 2. Use-Case Diagram



## 3. Describe Use Cases

<b>Use Case</b>	1. Show Menu
<b>Actors</b>	User
<b>Description</b>	- 메뉴 목록을 보여준다.

<b>Use Case</b>	2. Select Product
<b>Actors</b>	User
<b>Description</b>	<ul style="list-style-type: none"> <li>- 메뉴 목록에서 구매하고자 하는 상품을 선택한다.</li> <li>- 재고가 있다면, 상품정보를 출력하고, 결제 의사를 묻는다.</li> <li>- 재고가 없다면, Check other vm's info 를 호출해서 응답을 받은 뒤, See VM List 로 넘어가는지 확인한다.</li> </ul>

<b>Use Case</b>	3. Input Code
<b>Actors</b>	User
<b>Description</b>	<ul style="list-style-type: none"> <li>- 사용자는 이미 다른 VM 으로부터 받은 인증코드가 있다면 인증코드를 입력한다.</li> <li>- 인증코드가 등록되어 있다면 상품을 지급받는다.</li> <li>- 인증코드가 등록되어 있지 않다면, 구매가 취소된다.</li> </ul>

<b>Use Case</b>	4. Check Stock
<b>Actors</b>	None
<b>Description</b>	<ul style="list-style-type: none"> <li>- Use Case 2 번으로부터 선택한 상품에 대하여 시스템내에 재고가 있는지를 확인하여 넘겨준다.</li> </ul>

<b>Use Case</b>	5. Update Stock
<b>Actors</b>	None
<b>Description</b>	<ul style="list-style-type: none"> <li>- 상품 판매 시 재고를 갱신한다.</li> </ul>

<b>Use Case</b>	6. Show VM List
<b>Actors</b>	None
<b>Description</b>	<ul style="list-style-type: none"> <li>- 재고가 부족하거나 해당 제품이 판매하지 않는 제품인 경우 Check otherVM's Info 로부터 받은 Available 한 Vm 의 위치정보를 거리순으로 리스트하여 보여준다.</li> </ul>

<b>Use Case</b>	7. Select VM
<b>Actors</b>	User
<b>Description</b>	<ul style="list-style-type: none"> <li>- 리스트된 자판기 중 하나를 선택하여 위치를 확인하고 결제 의사를 확인한다.</li> </ul>

<b>Use Case</b>	8. Cancel Product
<b>Actors</b>	User
<b>Description</b>	<ul style="list-style-type: none"> <li>- 취소 의사를 표현하면 프로세스를 취소한다.</li> </ul>

<b>Use Case</b>	9. Payment
<b>Actors</b>	User
<b>Description</b>	<ul style="list-style-type: none"> <li>- 결제를 결정 하면 시작된다.</li> <li>- 결제하려는 카드를 태그 하거나 리더기에 삽입한다.</li> <li>- 잔액이 부족한 경우, 구매가 취소된다.</li> <li>- 잔액이 충분한 경우, (선)결제가 진행된다.</li> </ul>

<b>Use Case</b>	10. Give Product
<b>Actors</b>	None
<b>Description</b>	- 사용자가 결제를 정상적으로 진행할 경우, 상품을 지급한다.

<b>Use Case</b>	11. Give Code
<b>Actors</b>	None
<b>Description</b>	- 사용자가 선결제를 정상적으로 진행할 경우, 인증코드를 지급한다.

<b>Use Case</b>	12. Check otherVM's Info
<b>Actors</b>	None
<b>Description</b>	- 재고가 부족하거나 해당 제품이 판매하지 않는 제품인 경우, 다른 자판기의 정보를 확인한다. - 확인한 정보를 5. Show VM List 로 전달한다.

<b>Use Case</b>	13. Request Prepay
<b>Actors</b>	None
<b>Description</b>	- 선결제를 진행한 상품에 대해 해당 상품의 이름과 인증코드를 보낸다.

<b>Use Case</b>	14. Notify VM's Info
<b>Actors</b>	None
<b>Description</b>	- 다른 VM 에 현 기계의 정보를 알린다.

<b>Use Case</b>	15. Confirm Prepay
<b>Actors</b>	None
<b>Description</b>	- 다른 VM 에서 선결제를 진행한 상품에 대해 해당 상품의 재고와 인증코드를 저장해둔다. - 저장 성공 여부를 전송한다.

<b>Use Case</b>	16. Receive Request
<b>Actors</b>	otherDVM
<b>Description</b>	- 다른 VM 들로부터의 요청을 받아 분석하여 Use Case 14, 15 을 처리한다.

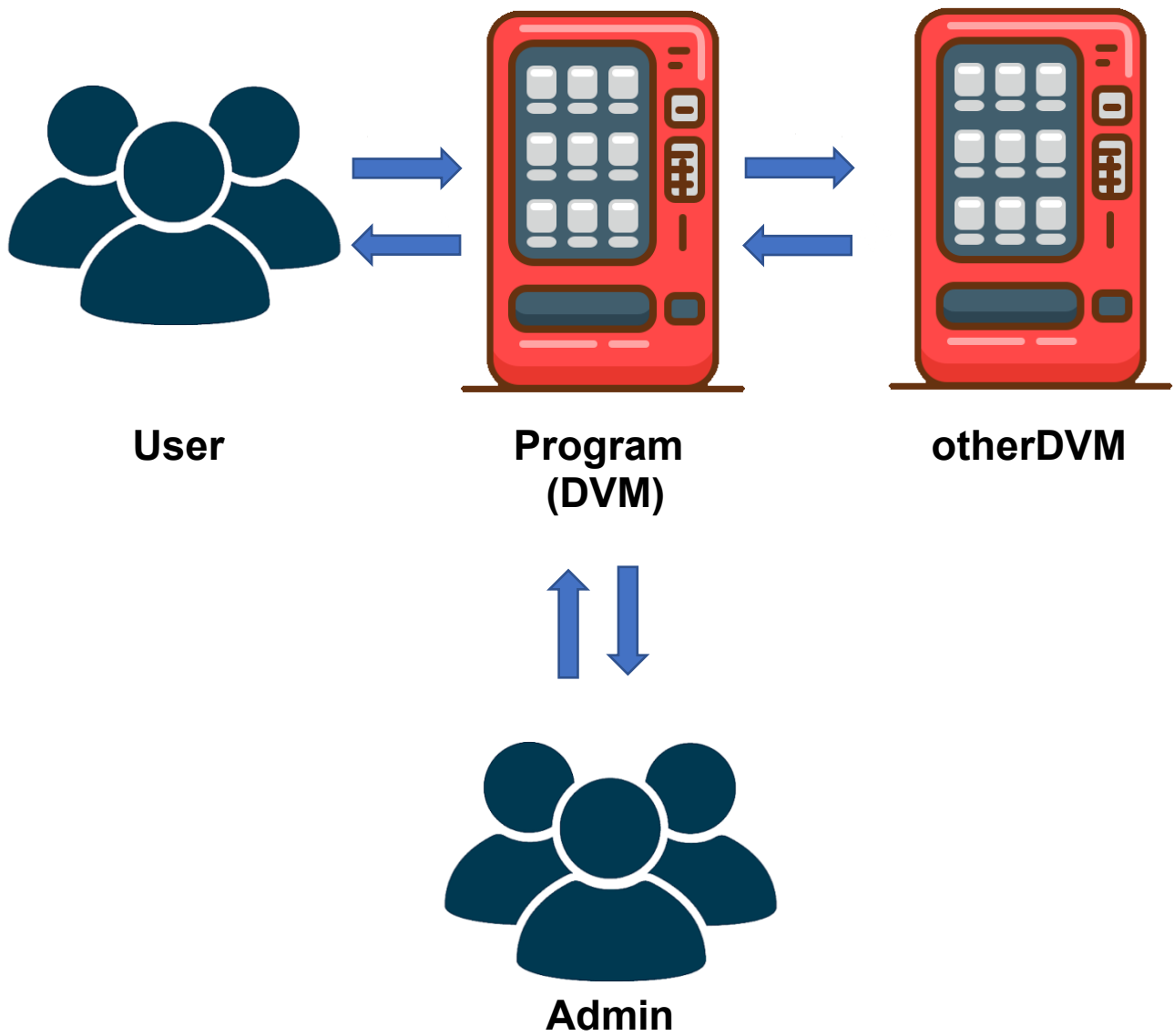
<b>Use Case</b>	17. Enter Admin Mode
<b>Actors</b>	Admin
<b>Description</b>	- 관리자 모드로 들어간다. - 관리자 인증수단을 사용한다. - 관리자 인증수단이 올바른 경우, 관리자 모드에 진입한다. - 틀릴 경우, 관리자 모드를 취소한다.

<b>Use Case</b>	18. Edit Product Information
<b>Actors</b>	None
<b>Description</b>	- 관리자 모드에서 상품 정보를 변경한다.

<b>Use Case</b>	19. Add/Remove VM
<b>Actors</b>	None
<b>Description</b>	- 관리자 모드에서 네트워크 안의 자판기를 추가하거나 제거한다.

<b>Use Case</b>	20. Edit Contact
<b>Actors</b>	None
<b>Description</b>	- 관리자 모드에서 네트워크 안의 자판 연락처 정보를 변경한다.

## ACTIVITY 1008 Define Draft System Architecture



## ACTIVITY 1009 DEFINE SYSTEM TEST CASE

Test #	Function	Description	Use Case Num & Name	System Function
1	Show Menu	- 메뉴의 리스트가 제대로 출력되는지 확인한다.	1. Show Menu	R1.1
2	Select Product	- 사용자가 상품을 입력하면 정상적으로 use case 의 check stock 으로 넘어가는지 확인한다.	2. Select Product	R1.2.1
3	Code Input	- 선결제를 통해 발급받은 인증코드로 상품 구매를 진행한다. - 인증코드가 등록되어 있다면 Give Product 로 넘어가는지 확인한다. - 인증코드가 저장되어 있지 않은 경우, Select Product 으로 넘어가는지 확인한다.	3. Input Code	R1.2.1
4	Check Stock	- 재고가 존재하면 Update Stock 을 호출한 뒤, 상품정보를 출력하고, 아닌 경우, 해당 음료에 대해 Check Other VM's info 를 호출해 응답을 받은 후 - Show VM List 로 넘어가는지 확인한다.	4. Check Stock	R1.3
5	Update Stock	- 재고의 상태가 제대로 업데이트 되는지 확인한다.	5. Update Stock	R1.4
6	Show VM List	- 재고를 가진 자판기들을 출력하는지 확인한다. 재고를 가진 자판기가 없다면 안내 메시지를 출력한 후, Show Menu 로 넘어가는지 확인한다.	6. Show VM List	R1.5
7	Select VM	- VM 을 선택했을 때, 위치정보와 결제 확인 및 취소 버튼이 제대로 출력되는지 확인한다. - 결제 확인 버튼을 선택 시 Payment 로 넘어가지는지 확인한다.	7. Select VM	R1.6
8	Cancel Process	- 결제 프로세스를 취소하고 이전의 값들을 초기화한 후 정상적으로 Select Product 가 호출되는지 확인한다.	8. Cancel Process	R1.7
9	Card Payment	- 잔고가 부족한 카드로 결제 시도 시 결제가 정상적으로 취소되는지 확인한다. - 결제가 완료된 후 Give Product 또는 Give Code 로 넘어가는지 확인한다.	9. Payment	R1.8

10	Give Product	- 구매한 음료를 정상적으로 지급하는지 확인한다.	10. Give Product	R1.9.1
11	Give Code	- 선결제시 구매한 다른 자판기의 음료수의 인증 코드를 사용자에게 출력하여 준다.	11. Give Code	R1.9.2
12	Check Info	- 다른 자판기에 신호를 제대로 보내는지 확인한다. - 다른 자판기의 정보를 가져와서 제대로 전달하는지 확인한다.	12. Check other VM's Info	R2.1
13	Receive Request	- 선결제된 상품의 이름과 인증코드를 제대로 전달하는지 확인한다.	13. Receive Request	R2.2
14	Notify VM's Info	- 전달받은 정보를 제대로 보내는지 확인한다.	14. Notify VM's Info	R2.3
15	Confirm Prepay	- 선결제를 진행한 상품에 대한 상품의 재고와 인증코드를 제대로 저장하는지 확인한다.	15. Confirm Prepay	R2.4
16	Request Prepay	- 요청 받은 Request 를 14,15 중 맞는 함수에 제대로 전달하는지 확인한다.	16. Request Prepay	R2.5
17	Enter Admin mode	- 정상적으로 관리자 모드 진입 방식을 사용했을 때, 관리자 모드에 진입 되는지 확인한다.	17. Enter Admin Mode	R3.1
18	Edit Product Information	- 입력 받은 상품의 정보 변경이 제대로 반영되었는지 확인한다.	18. Edit Product Information	R3.2
19	Edit VM	- 자판기가 10 개까지만 추가되고, 0 개까지만 제거되는지 확인한다.	19. Add/Remove VM	R.3.3
20	Edit Contact	- 입력 받은 연락처의 정보 변경이 제대로 반영되었는지 확인한다.	20. Add/Remove VM	R.3.4

# ACTIVITY 1010 REFINE PLAN

## 1. Project Scope

일반적인 VM의 경우, 원하는 상품이 없거나 재고가 모두 소진된 경우 그 이상의 서비스를 사용자에게 제공하기 어렵다. 이러한 기존 자판기의 단점을 해소하기 위해 자판기 사이에 정보를 교환할 수 있게 하여 위의 경우에서 다른 자판기의 위치를 안내하고 선결제가 가능한 것이 좋다고 생각하여 진행하게 되었다.

## 2. Project Objectives

- 기존 자판기의 문제점을 개선하기 위해서 각 자판기 모두 네트워크로 연결되어 있는 자판기 시스템을 구축한다.
- 네트워크를 통해 다른 자판기와 정보를 주고 받아 상품이나 재고가 없는 경우 다른 자판기의 위치를 안내한다.

## 3. Functional Requirements

Ref.#	Function	Category
R1.1	Show Menu	Hidden
R1.2.1	Select Product	Evident
R1.2.2	Code Input	Evident
R1.3	Check Stock	Hidden
R1.4	Update Stock	Hidden
R1.5	Show VM	Hidden
R1.6	Select VM	Evident
R1.7	Cancel Process	Evident
R1.8	Card Payment	Hidden
R1.9.1	Give Product	Hidden
R1.9.2	Give Code	Hidden
R2.1	Check Info	Hidden
R2.2	Receive Request	Hidden
R2.3	Notify VM's Info	Hidden
R2.4	Confirm Prepay	Hidden
R2.5	Receive Request	Hidden
R3.1	Enter Admin mode	Hidden
R3.2	Edit Product Information	Hidden



R.3.3	Edit VM	Evident
R3.4	Edit Contact	Evident

#### 4. Performance Requirement

- 버튼을 눌렀을 때 1 초 안에 동작이 완료되어 다음 동작으로 넘어가야 한다.
- 다른 자판기에 응답을 요청 받았을 경우 5 초안에 회답해야 한다.

#### 5. Operating Environment

- Microsoft Windows 10 이상

#### 6. Other Requirements

- 직관적이고 깔끔한 UI 를 지원한다.

#### 7. Resources

- **Human Efforts (Man-Month) :** 3MM
- **Human Resources:** 4 명
- **Duration :** 10 weeks
- **Budget :**  $5000 \times 10 \times 10 \times 4 = 2000000$  (200000/person in month)