

Object-Oriented Software Development

OOPT 1000

Team 2

201810286 박혜린

201811264 변장훈

전성우

201311312 정인욱

Activity 1001. Define A Draft Plan	3
Motivation	3
Project Objectives	3
Functional Requirements	3
Non-Functional Requirements	3
Resource Estimation	4
Activity 1002. Create Preliminary Investigation Report	4
Alternative Solutions	4
Project Justification (Business Demands)	4
Risk Management	4
Risk Reduction Plan	5
Market Analysis	5
Activity 1003. Define Requirements	6
Functional Requirements (Categorized Table)	6
Performance Requirements	7
Operating Environment	7
Activity 1004. Record Terms in Glossary	7
Activity 1005. Implement Prototype	8
Prototype for DVM system	8
Activity 1006. Define Business Use Case	9
Step 1. Define system boundary	9
Step 2. Identify the actors related to a system/organization	9
Step 3. Identify user goals for each actor	9
Step 4. Record the primary actors and their goals in an actor-goal list	9
Step 6. Categorize use cases	10
Step 9. Draw a use case diagram	11
Step 10. Describe use cases	12
Activity 1007. Define Business Concept Model	19
Identify 'Concepts' in the target domain.	19
Activity 1008. Define Draft System Architecture	20
Define system architecture	20
Activity 1009. Develop System Test Cases	20
Develop system test cases	20

Activity 1001. Define A Draft Plan

● Motivation

- 사용자들의 선호 식품이 다양화됨에 따라 기존의 자판기로는 다양한 수요를 만족시키기 어렵게 되었다.
- 판매자들은 소비자의 다양해진 기호를 만족시키기 위해 최대한 많은 종류의 음료를 판매하고자 한다.
- 네트워크로 연결된 자판기들이 자신이 판매하는 음료 이외에도 가까운 다른 곳에서 판매하는 음료의 정보를 제공한다면 다양한 소비자의 수요를 만족시킬 수 있다.

● Project Objectives

- 하나의 자판기가 수용할 수 있는 음료 재고에 제한되지 않고 더 높은 판매 효과를 낼 수 있는 자판기 시스템을 개발한다. 다른 자판기와 네트워크로 더 많은 음료 선택지를 제공하며, 선결제 시스템으로 바로 판매까지 연결될 수 있도록 한다.
- 소비자가 간편하게 음료를 구매할 수 있는 자판기 시스템을 개발한다. 신속한 결제과정과 타 자판기에 대한 위치 정보를 제공한다.
- 누구나 쉽고 간단하게 이용할 수 있는 인터페이스
- 가능한 한 빠른 결제 절차
- 자판기 특성상 공간을 차지하기 때문에 작고 효율적인 부피

● Functional Requirements

1. 카드 결제 시스템
2. 자판기 위치 시스템
3. 재고 관리 시스템
4. 네트워크 시스템
5. 선결제 시스템(인증코드 중복관리, 생성, 삭제)
6. GUI시스템
7. 자판기 목록 관리 시스템

● Non-Functional Requirements

- 자판기의 본래 목적에 충실하게 네트워크 상의 지연이 없어야 한다.

- 사용자와 시스템간 소통의 방식을 최소화하여 사용자의 잘못된 조작이 시스템 동작을 방해하지 않도록 한다.

● Resource Estimation

Human Efforts(Man-Month) : 2 M/M

Human Resource : 4인

Project Duration : 약 2달

Cost : 4,000,000(원) / 4인

Activity 1002. Create Preliminary Investigation Report

● Alternative Solutions

- 기존 자판기를 여러 대 구매한다.
- 외부 업체에 의뢰한다.

● Project Justification (Business Demands)

- 기존 자판기를 여러 대 구매하거나 외부 업체에 의뢰하는 것에 비해 저렴하다.
- 기존 자판기가 판매하는 음료의 개수는 정해져있다. 음료의 종류를 늘리기 위해서 기존 자판기는 공간의 제약을 받을 수 밖에 없지만, DVM은 공간의 제약을 최소화한다.
- 기존 자판기의 경우 재고가 없을 시 소비자가 구매를 포기하거나 일일이 다른 자판기를 찾아다녀야 하는 불편함이 있었다. 이를 해결하기 위해 다른 자판기의 정보까지 보여줘 소비자의 선택폭을 넓힐 수 있는 기능을 제공한다.
- 다양한 무인 판매 시스템이 등장하면서 소비자들이 이에 익숙해져 보다 복잡한 자판기 사용에 무리가 없으며, 기존의 무인 판매 시스템보다 다양한 정보를 제공한다는 강점이 있다.

● Risk Management

- Java 숙련도가 낮은 팀원이 있다.
- 직장을 다니는 팀원이 있어 일정 조율이 힘들다.
- 코로나로 인해 오프라인 모임이 어렵다.

- 팀원 간의 불화
- 다른 과목의 공부와 시험

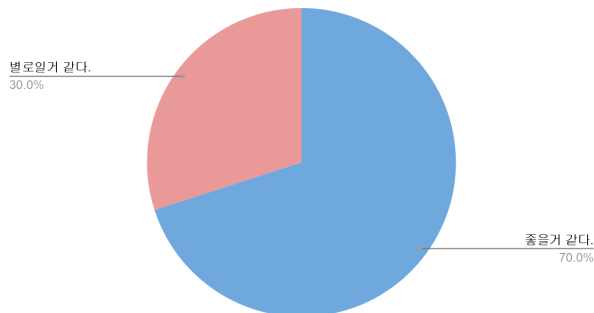
Risk	Probability	Significance	Weight(P*S)
Java 숙련도	4	9	36
팀원 간 일정 조율	8	10	80
코로나	4	5	20
팀원 간 불화	1	5	5
다른 과목의 부담	2	10	20

● Risk Reduction Plan

Risk	Reduction Plan
Java 숙련도	Discord,Slack을 이용한 온라인 스터디 모임 추가
팀원 간 일정 조율	저녁 ~ 새벽은 최소한의 협업이 가능하도록 스케줄 조정
코로나	화상회의 + 협업 tool
팀원 간 불화	발생 시 의사 결정 권한을 다른 팀장에게 전권 양도
다른 과목의 부담	타이트하게 planning하기

● Market Analysis

판매하는 다른 자판기의 위치를 알려주면 좋을것같다.



2조 지인들 10명을 대상으로 진행된 설문

1. 해당 자판기에 음료가 없을 경우 해당 음료를 판매하고 있는 다른 자판기의 위치를 알려주면 좋을 것 같다고 응답한 사람이 70%
2. 위치를 안내해줘도 귀찮아서 안가고 아무거나 마실 것 같다고 응답한 사람이 30%

Activity 1003. Define Requirements

- Functional Requirements (Categorized Table)

Ref. #	Function	Category
R 1.1	Card payment	Evident
R 1.2	InvalidCard	Hidden
R 1.3	balanceCard	Hidden
R 1.4	refund	Hidden
R 2.1	setAddress	Evident
R 2.2	getAddress	Hidden
R 3.1	setBeverage	Evident
R 3.2	getBeverage	Hidden
R 3.3	calculateNum	Hidden
R 3.4	checkBeverage	Hidden
R 4.1	stockCheck	Evident
R 4.2	respondStock	Evident
R 4.3	requestVerificationcode	Evident
R 4.4	respondVerificationcode	Evident
R 4.5	respondError	Evident
R 5.1	makeVerificationcode	Hidden
R 5.2	checkVerificationcode	Hidden
R 5.3	removeVerificationcode	Hidden
R 5.4	setCollection	Hidden
R 6.1	관리자 화면(Address)	Evident
R 6.2	관리자 화면(재고 관리)	Evident
R 6.3	초기화면	Evident

R 6.4	인증코드 입력화면	Evident
R 6.5	구매 확인	Evident
R 6.6	카드 결제 대기화면	Evident
R 6.7	선결제 확인화면	Evident
R 6.8	인증코드 화면	Evident
R 6.9	에러 화면(잔액부족)	Evident
R 6.10	에러 화면(도난)	Evident
R 6.11	에러 화면(인증코드)	Evident
R 6.12	에러 화면(타 자판기 재고가 없어 환불 요청하는 화면 / 확인 버튼)	Evident
R 7.1	getID	Hidden
R 7.2	vmOn	Evident
R 7.3	vmOff	Evident
R 7.4	updateVMlist	Evident
R 7.5	removeVM	Evident

● Performance Requirements

- 각 자판기의 네트워크 전송 시간은 1ms 이하여야 한다.
- 카드 입력 후 오류 혹은 결제 완료까지 6초 이내여야 한다.

● Operating Environment

- Windows 10

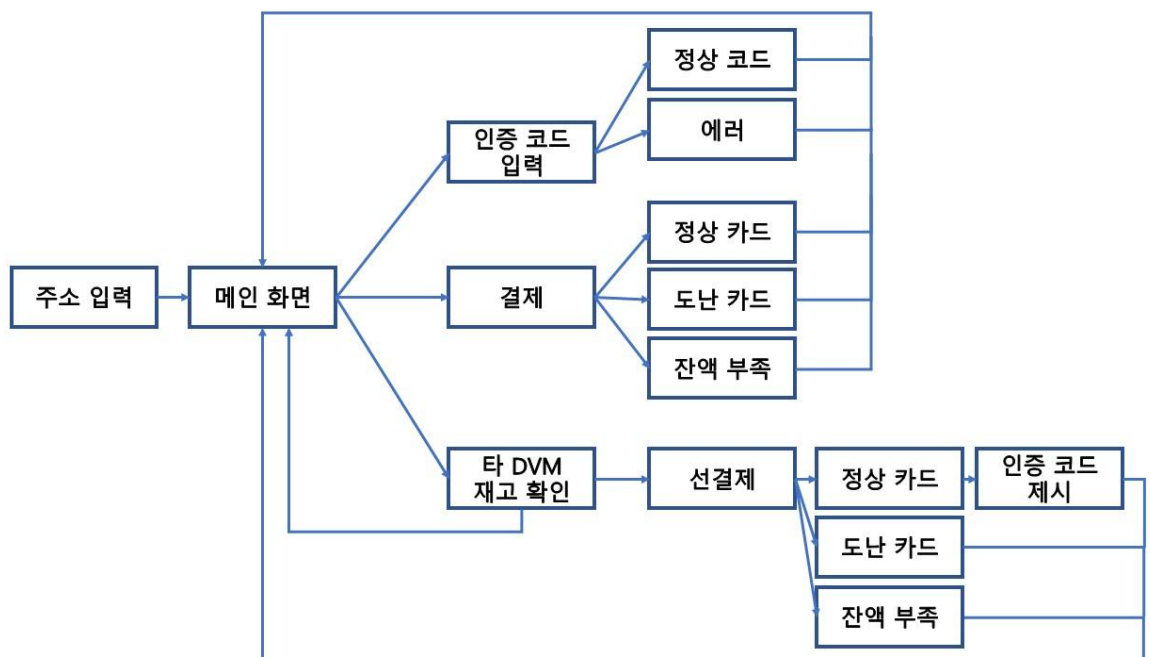
Activity 1004. Record Terms in Glossary

Num	Term	Description
1	DVM	Distributed Vending Machine의 약자로 네트워크로 연결된 분산 자판기 중 하나를 지칭한다.
2	선결제	사용자가 이용하고 있는 DVM에서 판매 불가능한 메뉴를 다른 DVM에서 구매하고자 할 때 이용하는 결제 시스템이다. 사용자는 음료를 미리 구매하고 인증코드를 받아 해당 자판기에서 권리를 행사할 수 있다.

3	Verification Code	사용자가 선결제를 한 경우 음료 구매권을 행사하기 위한 인증 코드
4	Collection	유효한 Verification Code를 저장한 배열
5	Address	DVM이 설치된 주소
6	Beverage	DVM이 판매하는 음료
7	Beveragenum	각 Beverage의 재고 수 배열
8	broadcast	네트워크로 연결된 모든 DVM에게 보내는 메시지
9	ID	각 DVM의 고유값
10	VMlist	네트워크로 연결된 모든 DVM의 ID를 저장한 리스트
11	관리자 모드 코드	인증코드 입력화면에서 관리자 화면으로 진입하기 위한 코드
12	on	DVM의 On/Off 여부를 의미
13	Card List	카드 결제 함수의 테스트를 위하여 카드 번호와 잔액, 유효성을 저장해 놓은 표

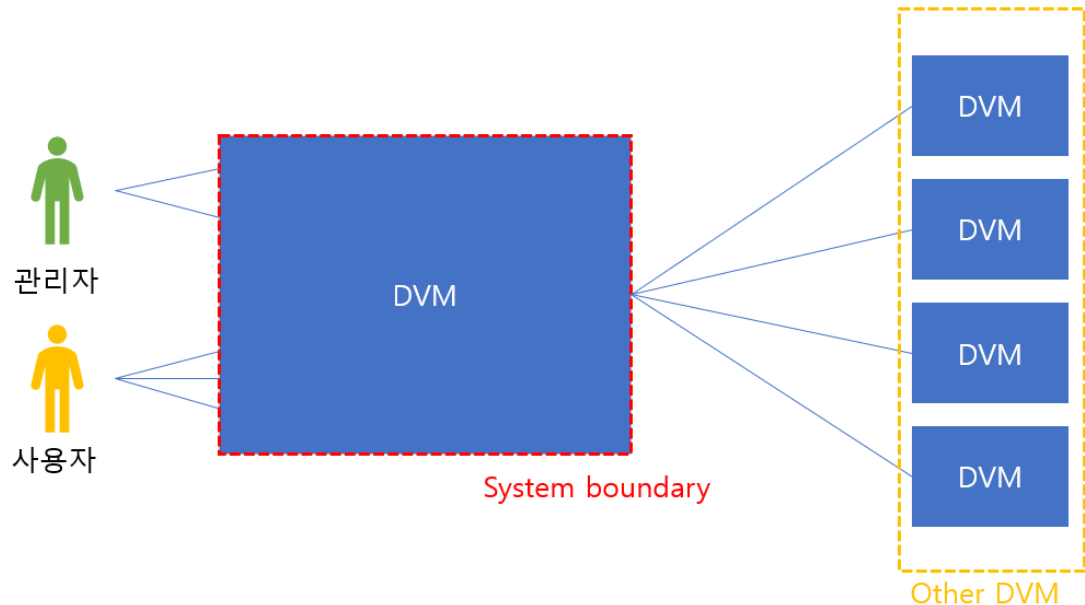
Activity 1005. Implement Prototype

- Prototype for DVM system



Activity 1006. Define Business Use Case

- Step 1. Define system boundary



- Step 2. Identify the actors related to a system/organization

- **Primary Actors**

사용자 : DVM을 통해 음료를 구매하는 구매자

관리자 : DVM의 주소를 입력하고 전원을 켜고 끄거나 재고를 보충하는 판매자

- **Supporting Actors**

Other DVM : 네트워크를 통해 연결된 다른 모든 DVM

- Step 3. Identify user goals for each actor

- 사용자 : 음료를 구매한다. 구매가 불가능할 경우 타 자판기의 재고 여부 정보를 얻거나 선결제를 하여 음료를 구매한다.
- 관리자 : 자판기의 주소를 입력하여 전원을 켜다. 음료를 보충하여 재고를 업데이트 한다.
- **Other DVM** : 다른 DVM과 통신하여 재고 여부를 파악한다. 선결제가 이루어진 경우 인증 코드를 전송하여 관리한다.

- Step 4. Record the primary actors and their goals in an actor-goal list

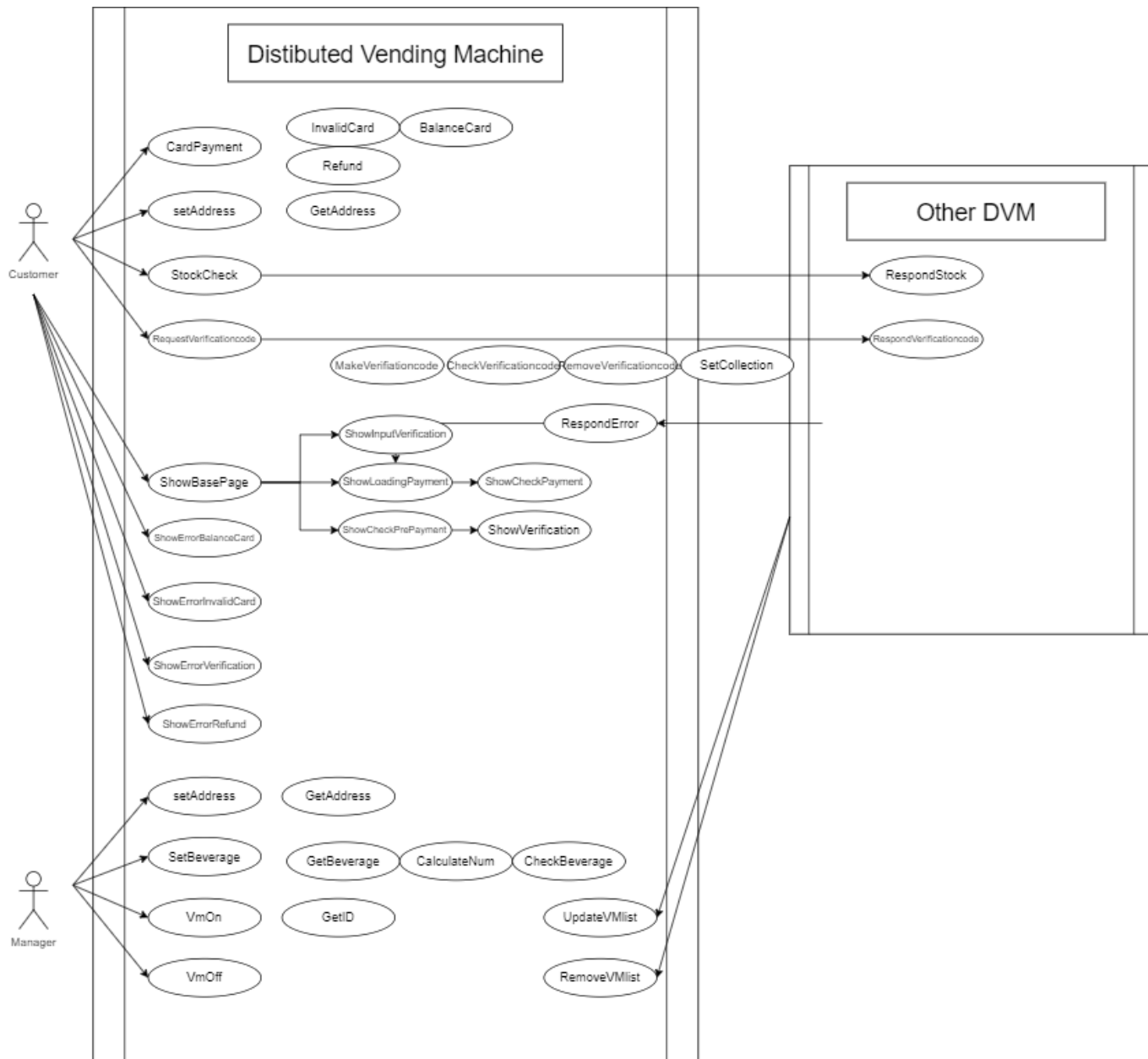
Actor	Goal
사용자	Select Beverage Pay by card Purchase by Verification Code Prepay
관리자	Set DVM address DVM turn on Stock update DVM turn off

● Step 6. Categorize use cases

Ref. #	Function	Use Case Name,Num	Category
R 1.1	cardPayment	1.CardPayment	Evident
R 1.2	invalidCard	2.InvalidCard	Hidden
R 1.3	balanceCard	3.BalanceCard	Hidden
R 1.4	refund	4.Refund	Hidden
R 2.1	setAddress	5.SetAddress	Evident
R 2.2	getAddress	6.GetAddress	Hidden
R 3.1	setBeverage	7.SetBeverage	Evident
R 3.2	getBeverage	8.GetBeverage	Hidden
R 3.3	calculateNum	9.CalculateNum	Hidden
R 3.4	checkBeverage	10.CheckBeverage	Hidden
R 4.1	stockCheck	11.StockCheck	Evident
R 4.2	respondStock	12.RespondStock	Evident
R 4.3	requestVerificationcode	13.RequestVerificationcode	Evident
R 4.4	respondVerificationcode	14.RespondVerificationcode	Evident
R 4.5	respondError	15.RespondError	Evident
R 5.1	makeVerificationcode	16.MakeVerificationcode	Hidden
R 5.2	checkVerificationcode	17.CheckVerificationcode	Hidden

R 5.3	removeVerificationcode	18.RemoveVerificationcode	Hidden
R 5.4	setCollection	19.SetCollection	Hidden
R 6.1	관리자 화면(Address)	20.ShowAdminPage_1	Evident
R 6.2	관리자 화면(재고 관리)	21.ShowAdminPage_3	Evident
R 6.3	초기화면	22.ShowBasePage	Evident
R 6.4	인증코드 입력화면	23.ShowInputVerification	Evident
R 6.5	구매 확인	24.ShowCheckPayment	Evident
R 6.6	카드 결제 대기화면	25.ShowLoadingPayment	Evident
R 6.7	선결제 확인화면	26.ShowCheckPrePayment	Evident
R 6.8	인증코드 화면	27.ShowVerification	Evident
R 6.9	에러 화면(잔액부족)	28.ShowErrorBalanceCard	Evident
R 6.10	에러 화면(도난)	29.ShowErrorInvalidCard	Evident
R 6.11	에러 화면(인증코드)	30.ShowErrorVerification	Evident
R 6.12	에러 화면(타 자판기 재고가 없어 환불 요청하는 화면 / 확인 버튼)	31.ShowErrorRefund	Evident
R 7.1	getID	32.GetID	Hidden
R 7.2	vmOn	33.VmOn	Evident
R 7.3	vmOff	34.VmOff	Evident
R 7.4	updateVMlist	35.UpdateVMlist	Evident
R 7.5	removeVM	36.RemoveVM	Evident

- Step 9. Draw a use case diagram



● Step 10. Describe use cases

Use Case	CardPayment
Actors	User
Description	사용자가 카드 단말기에 카드를 입력 하면 결제를 시행한다.

Use Case	InvalidCard
----------	-------------

Actors	None
Description	사용자가 입력한 카드가 유효하지 않을 때 시행한다.

Use Case	BalanceCard
Actors	None
Description	사용자가 입력한 카드의 잔액이 부족할 때 시행한다.

Use Case	Refund
Actors	None
Description	선결제 시스템에서 재고 오류가 발생할 시 환불을 시행한다.

Use Case	SetAddress
Actors	Admin
Description	관리자가 DVM 실행시 위치값(String) 를 저장한다.

Use Case	GetAddress
Actors	None
Description	VmOn이 실행되면 호출되며 외부에서 위치값(String)을 호출 가능하게 한다.

Use Case	SetBeverage
Actors	Vm Sensor
Description	<p>다음 두가지 경우에 실행된다.</p> <ul style="list-style-type: none"> • ShowAdminPage_1에서 시작버튼을 선택할 경우 실행된다. • ShowAdminPage_2에서 재고관리 버튼을 선택할 경우 실행된다. <p>실행될 때 DVM에 음료의 재고를 채울경우 센서가 재고값을 인식하고 저장한다. 재고값을 저장하되 재고가 0이 아닌 음료를 7개까지로 한정하게 한다.</p>

Use Case	GetBeverage
Actors	None
Description	객체 외부에서 모든 음료의 재고량(길이가 20인 배열)을 호출 가능하게 한다.

Use Case	CalculateNum
Actors	None
Description	파라미터로 받아온 인덱스에 해당하는 음료의 재고 값을 1 낮춘다.

Use Case	CheckBeverage
Actors	None
Description	파라미터로 받아온 인덱스에 해당하는 음료의 재고가 0인지 아닌지를 boolean 값으로 리턴한다.

Use Case	StockCheck
Actors	User
Description	broadcast 방식으로 VMlist상의 DVM들에 특정 음료의 재고의 유무를 확인하도록 request 한다.

Use Case	RespondStock
Actors	OtherDVM
Description	VMlist상의 DVM들로부터 특정 음료의 재고의 유무를 respond 한다.

Use Case	RequestVerificationcode
Actors	User
Description	사용자가 재고가 특정 음료의 선결제를 요구한 경우 재고가 있는 자판기에 Verification Code 를 생성할 것을 요청한다.

Use Case	RespondVerificationcode
Actors	OtherDVM
Description	선결제 를 통해 생성된 Verificationcode 를 respond 한다.

Use Case	RespondError
Actors	OtherDVM
Description	선결제 과정중 음료가 팔려 Verification Code 를 제공하지 못할 경우 호출된다.

Use Case	MakeVerificationcode
Actors	None
Description	RequestVerificationcode 를 respond 할 경우 Verification Code 를 생성한다.

Use Case	CheckVerificationcode
Actors	None
Description	MakeVerificationcode 를 통해 생성된 Verification Code 의 중복을 검사한다.

Use Case	RemoveVerificationcode
Actors	None
Description	Verification Code 를 통해 결제될 경우 Verification Code 들의 배열에서 해당 Verification Code 를 제거한다.

Use Case	SetCollection
Actors	None
Description	CheckVerificationcode 을 거친 Verification Code 를 Verification Code 들의 배열에 저장한다.

Use Case	ShowAdminPage_1
----------	-----------------

Actors	Admin
Description	시작하고 바로 시작되는 페이지로 SetAddress 를 호출하여 위치값(String)을 저장한다. 화면에는 입력란, 키보드, 시작 버튼이 있다.

Use Case	ShowAdminPage_2
Actors	Admin
Description	ShowInputVerification 창에서 관리자 모드 코드를 입력하면 출력하는 창이다. 화면에는 재고 관리버튼, VmOff 버튼 그리고 취소 버튼이 있다.

Use Case	ShowBasePage
Actors	User
Description	화면에는 음료 20개를 선택할 수 있는 버튼과 선결제 인증코드를 입력할 수 있는 버튼이 있다.

Use Case	ShowInputVerification
Actors	User
Description	ShowBasePage 에서 선결제 인증코드 버튼을 선택할 경우 호출되는 창이다. 화면에는 입력란, 0부터 9까지의 숫자버튼 10개와 확인, 취소 버튼이 있다.

Use Case	ShowCheckPayment
Actors	User
Description	ShowLoadingPayment 에서 결제 버튼을 누를 경우 호출되는 창이다. 화면에는 취소와 구매 버튼이 있다.

Use Case	ShowLoadingPayment
Actors	User
Description	ShowBasePage 에서 재고가 0이 아닌 음료 선택할 경우 호출되는 창으로 화면에는 음료사진과 음료의 이름, 가격이 표기되어

	있으며 결제와 취소 버튼이 있다.
--	--------------------

Use Case	ShowCheckPrePayment
Actors	User
Description	ShowBasePage에서 재고가 0인 음료를 선택할 경우 호출되는 창으로 화면에는 음료사진과 음료 이름, 가격이 표기되어 있다. 또한, 해당 음료가 판매되고 있는 다른 자판기의 위치가 번호와 매칭되어 버튼으로 있으며 추가로 취소 버튼과 선결제 버튼이 있다.

Use Case	ShowVerification
Actors	User
Description	ShowCheckPrePayment 에서 선결제 버튼을 선택하여 인증코드가 성공적으로 생성될 경우 사용자에게 인증코드를 보여주는 창이다.

Use Case	ShowErrorBalanceCard
Actors	User
Description	ShowCheckPayment에서 카드 잔액이 부족할 경우 호출되는 창이다. 화면에는 확인 버튼이 있다.

Use Case	ShowErrorInvalidCard
Actors	User
Description	ShowCheckPayment에서 도난카드일 경우 호출되는 창이다. 화면에는 확인 버튼이 있다.

Use Case	ShowErrorVerification
Actors	User
Description	ShowInputVerification에서 사용자가 입력한 인증코드가 유효하지 않을 경우 호출되는 창으로 RemoveVerificationcode의 리턴값이 false면 호출된다.

Use Case	ShowErrorRefund
Actors	User
Description	ShowCheckPrePayment에서 선결제 버튼을 눌렀으나 해당 자판기에서 선택된 음료의 재고가 0일 경우 호출되는 창이다. 확인 버튼이 있으며 즉시 Refund를 호출한다.

Use Case	GetID
Actors	None
Description	VmOn이 호출될 경우 호출되며 자판기 고유의 식별코드를 외부에서 호출 가능하게 한다.

Use Case	VmOn
Actors	Admin
Description	ShowAdminPage_1에서 시작버튼을 누르면 호출되며 broadcast로 GetID와 GetAddress를 호출하여 자판기 고유값과 주소값을 메시지로 보낸다.

Use Case	VmOff
Actors	Admin
Description	ShowAdminPage_2에서 VmOff버튼을 선택할 경우 실행되며 broadcast로 해당 자판기의 종료 메시지를 보낸다.

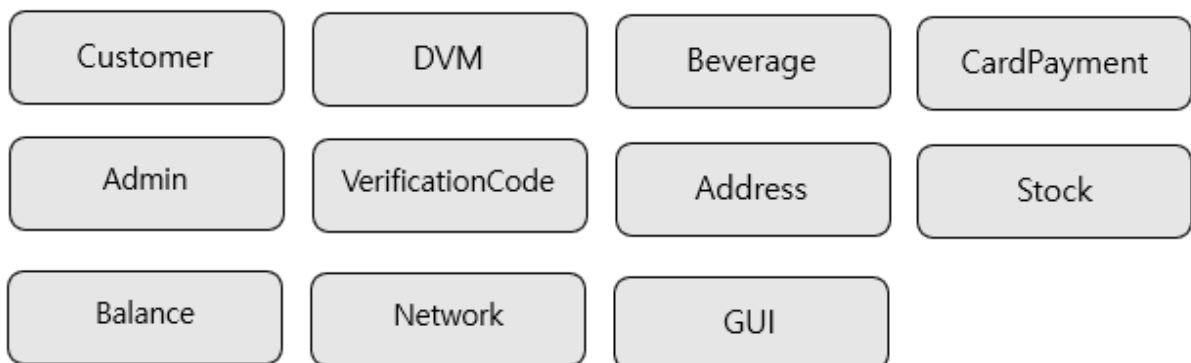
Use Case	UpdateVMlist
Actors	OtherDVM
Description	다른 DVM으로부터 VmOn 메시지를 받은 경우 해당 자판기 고유값을 VMlist에 추가한다.

Use Case	RemoveVM
----------	----------

Actors	OtherDVM
Description	다른 DVM으로부터 VmOff 메시지를 받은 경우 해당 자판기 고유값을 VMlist에서 제거한다.

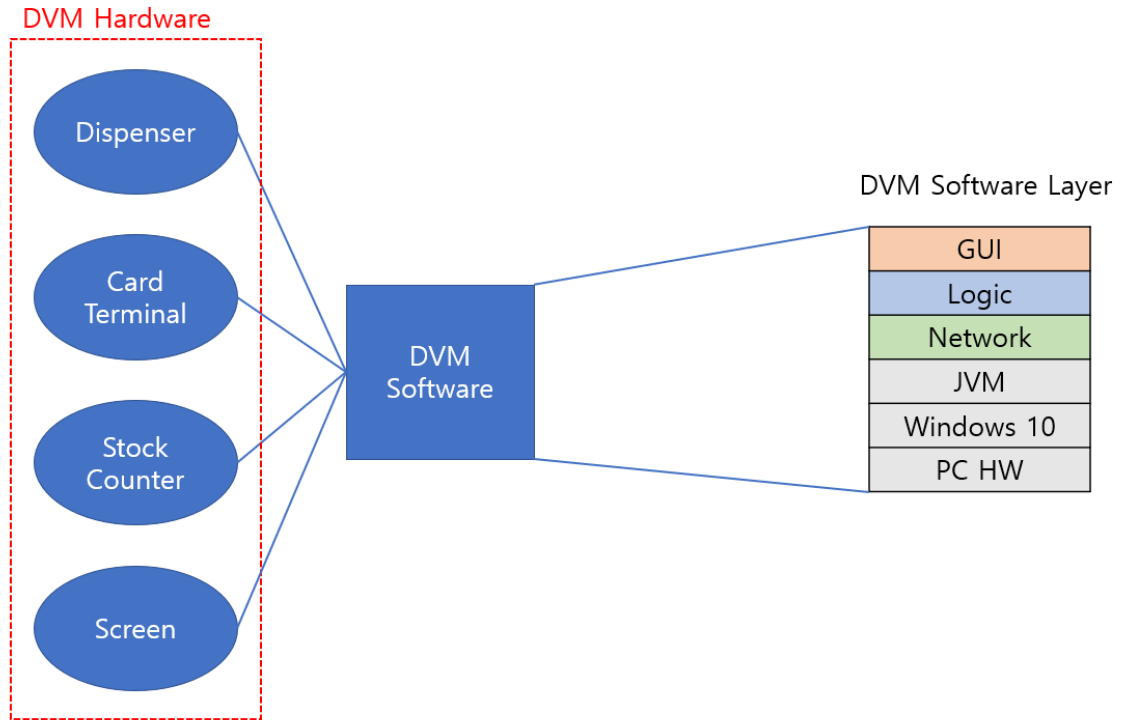
Activity 1007. Define Business Concept Model

- Identify 'Concepts' in the target domain.
- Drink
- OtherDVM



Activity 1008. Define Draft System Architecture

- Define system architecture



Activity 1009. Develop System Test Cases

- Develop system test cases

num	TEST	description
1	CardPayment	사용자가 입력한 카드번호를 Card List에 있는 값인지를 확인 후 있으면 true를 리턴하는지 확인한다.
2	CardPayment	사용자가 입력한 카드번호가 Null이면 어떻게 되는지 확인한다.
3	InvalidCard	사용자가 입력한 카드를 Card List에서 유효성을 검사한 후 유효한 카드이면 true를 리턴하는지 확인한다.
4	invalidCard	Card List의 값이 Null이면 어떻게 되는지 확인한다.
5	BalanceCard	사용자가 입력한 카드가 Card List에서 대조한 결과 유효한 카드이지만, 잔액이 선택한 상품의 금액 미만일 경우 false를 리턴하는지 확인한다.

6	BalanceCard	Card List에서의 잔액값이 정수값이 아닐경우 어떻게 되는지 확인한다.
7	Refund	해당 음료의 가격만큼 카드값에 매칭되는 Balance에 더해주는지 확인한다.
8	Refund	해당 Balance값이 int가 표시할 수 있는 값을 넘어가면 어떻게 되는지 확인한다.
9	SetAddress	ShowAdminPage_1에서 호출되는지 확인한다.
10	SetAddress	null을 입력 가능한지 확인한다.
11	GetAddress	Address값이 문자열 값인지 확인한다.
12	GetAddress	Address값이 null이면 어떻게 되는지 확인한다.
13	SetBeverage	ShowAdminPage_1에서 시작버튼을 선택할 경우와 ShowAdminPage_2에서 재고관리 버튼을 선택할 경우 호출되는지 확인한다.
14	SetBeverage	Vm Sensor가 꺼져있는 경우 어떻게 되는지 확인한다.
15	GetBeverage	Beverage 에서 0이 아닌 정수값이 7개 이하인지 확인한다.
16	GetBeverage	Beverage값이 null이면 어떻게 되는지 확인한다.
17	GetBeverage	Beverage 에서 0이 아닌 정수값이 8개 이상이면 어떻게 되는지 확인한다.
18	GetBeverage	Beverage에서 임의의 인덱스에 해당되는 값이 양의 정수가 아니면 어떻게 되는지 확인한다.
19	CalculateNum	음료가 판매되면 GetBeverage를 호출하여 해당 제품의 재고를 1씩 감소하여 다시 저장하는지 확인한다.
20	CalculateNum	GetBeverage에서 해당 제품의 재고가 0이라 언더플로우가 일어나는지 확인한다.
21	CheckBeverage	Beverage에서 인덱스에 해당하는 값이 0이면 false 0이 아니면 true를 리턴하는지 확인한다.
22	CheckBeverage	Beverage에서 0이 아니면서 양의정수 값이 아닐경우 true를 리턴하는지 확인한다.
23	StockCheck	선택한 메뉴의 재고가 0개일 때 실행되는지 확인한다.
24	StockCheck	선택한 메뉴의 재고가 1개 이상일 때 실행되지 않는지 확인한다.
25	respondStock	다른 DVM으로부터 stockCheck 메시지를 받았을 때 정상적으로 호출되는지 확인한다.
26	requestVerificationcode	사용자로부터 선결제 요청을 받은 경우 정상

		실행되는지 확인한다.
27	respondVerificationcode	다른 DVM으로부터 인증코드 요청 메시지를 받은 경우이며, 해당 음료의 재고가 1개 이상일 때 정상 실행되는지 확인한다.
28	respondVerificationcode	다른 DVM으로부터 인증코드 요청 메시지를 받은 경우이며, 해당 음료의 재고가 0개 일 때 실행 되지 않는지 확인한다.
29	respondError	다른 DVM으로부터 인증코드 요청 메시지를 받은 경우이며, 해당 음료의 재고가 0개일 때 정상 실행되는지 확인한다.
30	respondError	다른 DVM으로부터 인증코드 요청 메시지를 받은 경우이며, 해당 음료의 재고가 1개 이상일 때 실행 되지 않는지 확인한다.
31	makeVerificationcode	RequestVerificationcode를 respond하는 경우에 Verification Code를 생성하는지 확인한다.
32	makeVerificationcode	RequestVerificationcode를 respond하지 않는 경우에 Verification Code를 생성하지 않음을 확인한다.
33	checkVerificationcode	중복되는 Verification Code 입력 시 중복검사가 되는지 확인한다.
34	checkVerificationcode	Collection의 길이가 0일 경우 어떻게 되는지 확인한다.
35	removeVerificationcode	Verification Code를 통해 결제될 경우 Collection에서 해당 Verification Code가 제거되는지 확인한다.
36	removeVerificationcode	결제가 완료되지 않은 Verification Code를 제거하지 않음을 확인한다.
37	setCollection	CheckVerificationcode를 거친 Verification Code를 Collection에 저장하는지 확인한다.
38	setCollection	CheckVerificationcode에서 중복검사에 통과한 값은 저장되고 통과하지 못한 값은 저장되지 않음을 확인한다.
39	setCollection	checkVerificationcode를 통과한 값이 null일 경우 어떻게 되는지 확인한다.
40	ShowAdminPage_1	인증코드 입력 화면에서 관리자 코드를 입력하였을 때 관리자 화면이 잘 나오는지 확인한다.
41	ShowAdminPage_2	DVM을 실행 시켰을 때 주소 입력 화면이 나오는지 확인한다.
42	ShowBasePage	관리자가 주소를 입력한 후 확인 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
43	ShowBasePage	인증코드 입력 화면에서 취소 버튼을 누르면 메인

		화면이 잘 나오는지 확인한다.
44	ShowBasePage	카드 입력 대기 화면(CheckPayment)에서 취소 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
45	ShowBasePage	결제 정보 화면>LoadingPayment)에서 취소 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
46	ShowBasePage	선결제 화면(PrePayment)에서 취소 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
47	ShowBasePage	인증코드 확인 화면>ShowVerification)에서 돌아가기 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
48	ShowBasePage	에러-잔액 부족 화면>ShowErrorBalanceCard)에서 돌아가기 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
49	ShowBasePage	에러-유효하지 않은 카드 화면>ShowErrorInvalidCard)에서 돌아가기 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
50	ShowBasePage	에러-유효하지 않은 인증코드 화면>ShowErrorVerification)에서 돌아가기 버튼을 누르면 메인 화면이 잘 나오는지 확인한다.
51	ShowInputVerification	메인 화면에서 인증코드 버튼을 누르면 인증코드 입력 화면이 잘 나오는지 확인한다.
52	ShowCheckPayment	결제 확인 화면에서 결제 버튼을 누를 경우 카드 입력 대기 화면이 잘 나오는지 확인한다.
53	ShowLoadingPayment	메뉴를 선택하고 해당 메뉴의 재고가 1 이상일 경우 결제 확인 화면이 잘 나오는지 확인한다.
54	ShowCheckPrePayment	메뉴를 선택하고 해당 메뉴의 재고가 0일 경우 선결제 화면이 잘 나오는지 확인한다.
55	ShowVerification	선결제가 완료된 경우 인증코드를 정상적으로 출력하는지 확인한다.
56	ShowErrorBalanceCard	입력된 카드가 잔액부족일 경우 에러-잔액부족 화면이 잘 나오는지 확인한다.
57	ShowErrorInvalidCard	입력된 카드가 유효하지 않은 카드일 경우 에러-유효하지 않은 카드 화면이 잘 나오는지 확인한다.
58	ShowErrorVerification	입력된 인증코드가 유효하지 않을 경우 에러-유효하지 않은 인증코드화면이 잘 나오는지 확인한다.
59	ShowErrorRefund	다른 DVM으로부터 인증코드 발송 에러 메시지를 받은 경우 에러-환불 화면이 잘 나오는지 확인한다.
60	GetID	broadcast로 request되는 ID와 자판기 고유의

		식별코드가 같은지 확인한다.
61	GetID	GetID가 리턴하는 변수값이 null이면 어떻게 되는지 확인한다.
62	VmOn	ShowAdminPage_1에서 시작버튼을 선택하면 broadcast로 GetID 와 GetAddress 를 호출하여 그 값들을 request하는지 확인한다.
63	VmOn	VMlist의 길이가 0이면 어떻게 되는지 확인한다.
64	VmOff	broadcast로 자판기의 종료 메세지가 request되는지 확인한다.
65	VmOff	VMlist의 길이가 0이면 어떻게 되는지 확인한다.
66	UpdateVMlist	다른 DVM으로부터 VmOn 메세지를 respond할 경우 VMlist의 길이가 1 증가하는지 확인한다.
67	UpdateVMlist	동시에 두개의 DVM으로부터 VmOn 메세지를 respond할 경우 VMlist의 길이 변화를 확인한다.
68	RemoveVM	다른 DVM으로부터 VmOn 메세지를 respond할 경우 VMlist의 길이가 1 감소하는지 확인한다.
69	RemoveVM	동시에 두개의 DVM으로부터 VmOff 메세지를 respond할 경우 VMlist의 길이 변화를 확인한다.