

Software Modeling & Analysis

Distributed vending machine

OOPT Stage 1000 - Plan & Elaboration

Title

DVM_Planning_Ver_1

Date

2021-04-01

Team 3

201711354 지준호 - 팀장

201311280 서영일

201611254 김병찬

201914175 선정민

Index

Activity 1001. Define Draft Plan

1. Motivation
2. Objective
3. Functional Requirements
4. Non-Functional Requirements
5. Resource

Activity 1002. Create Preliminary Investigation Report

1. Alternative Solution
2. Project Justification
3. Risk Management
4. Risk Reduction Plan
5. Market Analysis

Activity 1003. Define Requirements

1. Functional Requirements
2. Performance Requirements
3. Operation Environments
4. Develop Environments
5. Interface Requirements
6. Other Requirements

Activity 1004. Record Terms in Glossary

1. Description Specific Term

Activity 1005. Implements Prototype

Activity 1006. Define Business Use Case

1. Define System Boundary
2. Identify and Describe Actors
3. Identify Use Cases
4. Allocate System Functions / Category
5. Use Case Diagram

Activity 1007. Define Business Concept Model

Activity 1008. Define Draft System Architecture

Activity 1009. Define System Test Case

Activity 1001. Define Draft Plan

1. Motivation

일반적인 자판기는 사용자의 필요에도 불구하고 재고가 없으면 구매할 수 없다. 이는 사용자의 요구가 일방적으로 무시되는 부분이다. 해당 문제점 개선하고자 사용자 중심의 새로운 자판기를 설계하고자 한다. ‘선결제’를 통해 음료를 먼저 구입할 수 있으며, 판매하고 있는 자판기의 위치 정보까지 제공해 사용자의 요구를 빠르게 충족시킨다. 또한 사용자의 변심으로 ‘선결제’한 제품을 아직 수령하지 않았다면, 환불 또한 가능하다.

2. Objective

- 사용자가 쉽게 사용할 수 있는 간단한 UI를 제공한다.
- 사용에 불편함이 없는 빠른 속도를 보장한다.
- 허용되지 않은 사용자가 선결제된 음료를 수령하지 않도록 안전한 인증코드를 사용한다.
- 사용자가 시스템을 쉽게 유동적으로 사용할 수 있도록 환불 기능을 제공한다.

3. Functional Requirements

- 재고가 없는 자판기에서 재고가 있는 타 자판기의 위치를 제공한다.
- 카드 잔액이 부족한 경우 결제를 중단한다.
- ‘선결제’ 기능을 제공하며 결제 완료 시 인증코드를 제공한다.
- 인증코드는 서로 중복되지 않는다.
- 선결제 시 제한된 시간 내 수령하지 않은 경우, 자동으로 환불되는 기능을 제공한다.

4. Non-Functional Requirements

- 자판기 사용에 지장없는 빠른 속도
- 사용자 친화적인 인터페이스를 제공
- 정확한 정보 제공

5. Estimate resources

- Human Resources : 4명

- Project Duration : 2개월
- Cost : 인당 1,000만원, 총 4,000만원

Activity 1002. Create Preliminary Investigation Report

1. Alternative Solution

- 외부 업체에 시스템 개발을 의뢰한다.
- 공모전을 개최해 수상작과 개발자를 선정한다.

2. Project Justification

- Cost : 4,000만원
- Duration : 2개월
- Risk :
 - 이미 많은 무인 판매점이 존재하고 있고, 더 많은 종류의 제품들을 판매한다.
 - 네트워크의 통신의 장점을 부각하기 위해 많은 자판기 수가 필요하다.
 - 서로 네트워크 통신이 가능해야 하므로 설치 지역에 제한이 있다.
- Effect
 - 선결제를 통해 사용자가 필요한 음료를 구매할 수 있다.

3. Risk Management

Risk	Probability	Significance	Weight
JAVA 미숙	5	5	25
코로나로 인한 모임 제한	2	2	4
시간 부족	5	5	25
팀원 간 갈등	1	4	4
대형 프로젝트 경험 부족	4	4	16

4. Risk Reduction Plan

Risk	Risk Reduction Plan
JAVA 미숙(25)	구글링 및 서적 참고
코로나로 인한 모임 제한(4)	디스 코드 음성 회의와 Github 자료 공유
시간 부족(25)	휴일에도 시간을 투자하여 개발
팀원 간 갈등(4)	다른 프로젝트 개발 예시를 참고한다
대형 프로젝트 경험 부족(16)	항상 밝은 분위기를 유지하고 문제 발생시 교수님께 도움을 요청하여 해결한다.

5. Market Analysis

- 인건비 상승으로 티켓발매기, 키오스크 시장이 급성장하고 있다.
- 네트워크를 사용하기 때문에 설치 공간 제한이 있다
- 편의점, 마트 등 다양한 음료 구입 경로가 있다.
- 편의점 마트와 비교적 공간의 제약을 덜 받아, 사각지대에도 자판기를 설치하여 수익을 올릴 수 있다.

6. Other Managerial Issues

- 2개월 내 개발을 완료해야 한다.
- SW에 적합한 하드웨어가 필요하다.
- 자판기의 주기적인 업데이트가 필요하다.

Activity 1003. Define Requirements

1. Functional Requirements

1. 상품 확인	<ul style="list-style-type: none"> -투명한 유리창에 20가지 종류의 모든 음료수가 진열되어 있으며, 각 음료수 아래엔 빛나는 버튼과 음료수 이름이 있다. -녹색 빛 버튼은 현재 재고가 있음을, 적색 빛 버튼은 현재 재고가 없음을 의미한다.
2. 상품 선택 및 판매 여부 확인	<ul style="list-style-type: none"> -녹색 빛 버튼을 눌러 결제를 진행할 수 있다. -적색 빛 버튼을 눌러 타 자판기들의 음료의 재고 상태를 파악한다. -적색 빛 버튼을 누른 사용자가 선택한 음료의 상태를 디스플레이에 출력한다. -출력 문구는 판매가능 / 판매 불가능 총 2가지이다.
3. 상품 결제	<ul style="list-style-type: none"> -자판기는 선택한 음료의 가격과 카드 삽입 요청 문구를 디스플레이에 출력한다. -사용자가 카드 리더기에 카드를 투입하면 자판기는 카드 결제를 진행한다. -결제에 성공한 경우, 자판기는 결제 성공 문구를 출력하고 결제한 음료수를 음료 배출구로 배출한다. -결제에 실패한 경우, 자판기는 결제 실패 사유를 출력한다. -사용자는 임의로 현재 결제 진행을 멈출 수 있다.
4. 상품 선결제	<ul style="list-style-type: none"> -자판기는 디스플레이에 사용자가 선택한 음료를 구매할 수 있는 타 자판기들의 위치를 출력한다. -사용자가 출력된 자판기의 위치 중 한 곳을 선택하면, 자판기는 선택한 음료의 가격, 카드 삽입 요청 문구와 선택한 자판기의 위치를 디스플레이에 출력한다 -결제 완료 후, 자판기는 인증코드를 디스플레이에 출력한다. -사용자는 임의로 현재 선결제 진행을 멈출 수 있다.
5. 카드 결제	<ul style="list-style-type: none"> -카드 리더기에서 사용자의 카드를 읽어 들어 결제 및 선결제 성공 여부를 결정한다. -잔액이 부족하거나 카드 읽기 실패, 사용할 수 없는 카드일 경우, 결제 실패 문구를 디스플레이에 출력한다.

	-잔액이 충분할 경우, 카드 잔액에서 결제 금액을 제한 후 결제 성공 문구를 디스플레이에 출력한다.																				
6. 인증코드 확인	-사용자가 발급받은 인증코드를 입력하면, 자판기는 해당 코드를 확인한다. -잘못된 인증코드를 입력한 경우, 재입력 요청 문구를 출력한다. -적합한 인증코드를 입력한 경우, 상품의 이름을 디스플레이에 출력하고 배출구로 선결제 한 음료수를 배출한다. -사용자는 임의로 인증코드 입력을 취소할 수 있다.																				
7. 환불	-선결제 이 후, 1시간 동안 상품을 수령하지 않았을 경우, 자동 환불이 진행된다.																				
8. Msg Protocol	-자판기가 네트워크를 통해 메시지를 보낼때 지켜야할 규칙 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">현재 자판기 ID</th> <th style="width: 15%;">대상 자판기 ID</th> <th style="width: 20%;">Msg 보낸 목적</th> <th style="width: 50%;">Msg 내용</th> </tr> </thead> <tbody> <tr> <td rowspan="7" style="text-align: center; vertical-align: middle;">1 ~ 10</td> <td rowspan="7" style="text-align: center; vertical-align: middle;">-1 ~ 10 -broadcast- 전체 발송은 0</td> <td style="text-align: center;">재고 확인 요청</td> <td style="text-align: center;">음료 이름</td> </tr> <tr> <td style="text-align: center;">재고 확인 요청 응답</td> <td style="text-align: center;">True, False</td> </tr> <tr> <td style="text-align: center;">선결제 확인</td> <td style="text-align: center;">음료 이름 혹은 인증코드</td> </tr> <tr> <td style="text-align: center;">주소 요청</td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;">주소 요청 응답</td> <td style="text-align: center;">자판기 위치 정보</td> </tr> <tr> <td style="text-align: center;">음료 판매 확인</td> <td style="text-align: center;">음료 이름</td> </tr> <tr> <td style="text-align: center;">음료 판매 확인 응답</td> <td style="text-align: center;">True, False</td> </tr> </tbody> </table>	현재 자판기 ID	대상 자판기 ID	Msg 보낸 목적	Msg 내용	1 ~ 10	-1 ~ 10 -broadcast- 전체 발송은 0	재고 확인 요청	음료 이름	재고 확인 요청 응답	True, False	선결제 확인	음료 이름 혹은 인증코드	주소 요청	X	주소 요청 응답	자판기 위치 정보	음료 판매 확인	음료 이름	음료 판매 확인 응답	True, False
현재 자판기 ID	대상 자판기 ID	Msg 보낸 목적	Msg 내용																		
1 ~ 10	-1 ~ 10 -broadcast- 전체 발송은 0	재고 확인 요청	음료 이름																		
		재고 확인 요청 응답	True, False																		
		선결제 확인	음료 이름 혹은 인증코드																		
		주소 요청	X																		
		주소 요청 응답	자판기 위치 정보																		
		음료 판매 확인	음료 이름																		
		음료 판매 확인 응답	True, False																		

Ref.#	Function	Description
1.1	displayLight	상품 재고 현황을 색상으로 표현한다.
2.1	detectProduct	사용자가 선택한 음료를 감지한다.
2.2	showProductStatus	사용자가 선택한 음료의 상태를 출력한다.
3.1	updateProductStatus	자판기 내 음료 정보를 최신화한다.
3.2	showCodeInput	인증코드 입력란을 출력한다.
3.3	Cancel	현재 진행 상태를 종료한다.
4.1	showPayment	음료의 가격과 카드 삽입 요청 문구를 출력한다.
4.2	startPayment	결제를 시작한다.

4.3	endPayment	결제를 종료한다.
5.1	showPrepayment	자판기 위치, 음료의 가격과 카드 삽입 요청 문구를 출력한다.
5.2	startPrepayment	선결제를 시작한다.
5.3	endPrepayment	선결제를 종료한다.
5.4	showLocation	사용자가 선택한 음료의 재고가 있는 타 자판기의 위치와 현재 위치를 출력한다.
5.5	dectectLocation	나열된 위치 중 사용자가 선택한 자판기의 위치를 감지한다.
6.1	sendMsg	자판기에서 네트워크를 이용해 다른 자판기로 msg를 보낸다.
6.2	getMsg	타 자판기가 송신한 msg를 수신 한다.
7.1	checkCardInput	카드 투입 여부를 확인한다.
7.2	checkCardInfo	카드 정보를 확인한다.
7.3	checkCardValid	사용자가 투입한 카드의 결제 가능 여부를 확인한다.
7.4	payCard	카드 결제를 실행한다.
7.5	showPaymentResult	결제 성공 혹은 실패 문구를 출력한다.
8.1	makeCode	인증코드를 생성한다.
8.2	deleteExpiredCode	인증코드 유효 시간이 지난 코드를 확인하여 삭제한다.
8.3	showCode	인증코드를 디스플레이에 출력한다.
8.4	saveCode	수신받은 코드와 음료 이름을 저장한다.
8.5	detectCode	입력한 인증코드를 감지한다.
8.6	checkCodeValid	입력한 인증코드가 적합한지 확인한다.
8.7	showCodeResult	인증코드 입력 결과 문구를 출력한다.
8.8	deleteUsedCode	사용된 인증코드를 소멸시킨다.
9.1	getRefund	지정된 시간내에 상품을 수령하지 않을 경우, 환불을 한다.
10.1	releaseProduct	사용자가 결제한 음료수를 배출한다.

2. Non-Functional Requirements

- 빠른 속도 보장
- 사용자 친화적 인터페이스
- 정확한 정보 표현

3. Operating Environments

- Microsoft Windows 10 이상

- JDK 1.8.0 이상

Activity 1004. Record Terms in Glossary

1.Description Specific Term

Term	Description
Card	상품 결제 시 사용되는 사용자의 카드
Product	DVM에서 판매하는 음료 상품
Code	현재 시각에 해시 함수를 적용해 만들어낸 코드. 숫자로만 이루어져 있으며, 중복되지 않는다.
msg	네트워크 내에서 자판기가 사용하는 네트워크 메세지
Network	자판기가 서로 정보를 공유하기 위한 전산망
Location	자판기의 현재 위치
Refund	지정된 시간내에 음료수를 수령안할 시 환불되는 기능
PrePayment	현재 자판기에 없는 상품을 미리 결제하는 기능
Payment	현재 자판기에 있는 상품을 결제하는 기능
User	자판기 사용자
문구	디스플레이에 출력되는 문장이다. 출력시간은 1초
결제 실패 문구	카드 읽기 실패, 잔액 부족, 사용할 수 없는 카드 중 하나이다.
프로토콜	네트워크 메세지 형식
인증코드	대소문자 영어 및 숫자를 포함하는 6자리 단어
로그	함수의 실행 결과로 결제, 선결제 성공 여부 결제 승인 거부 사유, 인증코드 확인 성공 여부

Activity 1005. Implements Prototype

Activity 1006. Define Business Use Case

1. Define System Boundary

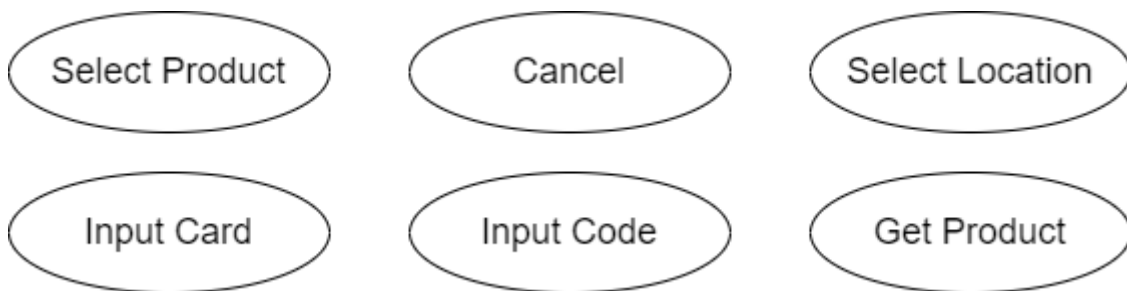


2. Identify and Describe Actors

- User : DVM을 이용하는 사람
- Other DVMs : DVM에 재고 정보를 송수신

3. Identify Use Cases

- User Based Event



- Other DVMs Based Event



- System Based Event

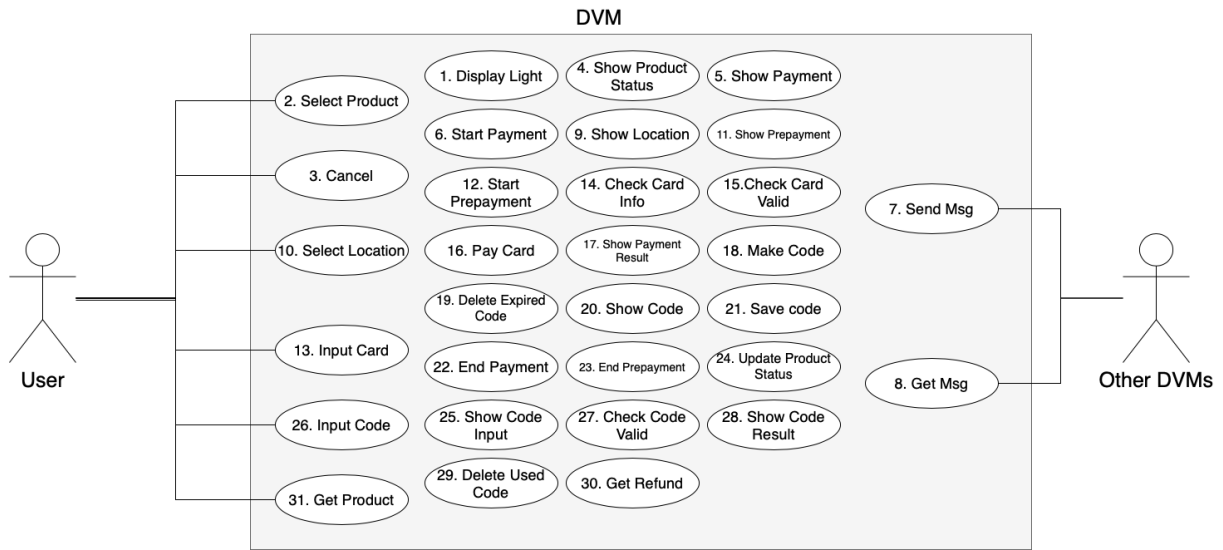


4. Allocate System Functions / Category

Ref	Function	Use Case Number & Name	Actor	Category
1.1	displayLight	1. Display Light	System	Hidden
2.1	detectProduct	2. Select Product	User	Evident
3.3	Cancel	3. Cancel	User	Evident
2.2	showProductStatus	4. Show Product Status	System	Hidden
4.1	showPayment	5. Show Payment	System	Hidden
4.2	startPayment	6. Start Payment	System	Hidden
6.1	sendMsg	7. Send Msg	System	Hidden

6.2	getMsg	8. Get Msg	System	Hidden
5.4	showLocation	9. Show Location	System	Hidden
5.5	detectLocation	10. Select Location	User	Evident
5.1	showPrepayment	11. Show Prepayment	System	Hidden
5.2	startPrepayment	12. Start Prepayment	System	Hidden
7.1	checkCardInput	13. Input Card	User	Evident
7.2	checkCardInfo	14. Check Card Info	System	Hidden
7.3	checkCardValid	15. Check Card Valid	System	Hidden
7.4	payCard	16. Pay Card	System	Hidden
7.5	showPaymentResult	17. Show Payment Result	System	Hidden
8.1	makeCode	18. Make Code	System	Hidden
8.2	deleteExpiredCode	19. Delete Expired Code	System	Hidden
8.3	showCode	20. Show Code	System	Evident
8.4	saveCode	21. Save Code	System	Hidden
4.3	endPayment	22. End Payment	System	Hidden
5.3	endPrepayment	23. End Prepayment	System	Hidden
3.1	updateProductStatus	24. Update Product Status	System	Hidden
3.2	showCodeInput	25. Show Code Input	System	Hidden
8.5	detectCode	26. Input Code	User	Evident
8.6	checkCodeValid	27. Check Code Valid	System	Hidden
8.7	showCodeResult	28. Show Code Result	System	Hidden
8.8	deleteUsedCode	29. Delete Used Code	System	Hidden
9.1	getRefund	30. Get Refund	System	Hidden
10.1	releaseProduct	31. Get Product	User	Evident

5. Use Case Diagram



6. Describe Use Cases

Use Case	1. Display Light
Actors	System
Description	-진열대에 있는 음료 버튼을 LED로 알린다. -현재 재고가 있는 경우 녹색 빛을, 재고가 없거나 판매하지 않는 상품의 경우 붉은 빛을 켜준다.

Use Case	2. Select Product
Actors	User
Description	-사용자가 버튼을 눌러 음료를 선택하는 기능 -진열대에서 20종의 음료를 선택할 수 있다. -녹색 빛의 버튼을 누르면 결제를 진행할 수 있고, 적색 빛의 버튼을 눌렀을 때 사용자가 선택한 음료의 상태를 디스플레이에 출력한다.

Use Case	3. Cancel
Actors	User
Description	-사용자가 현재 진행 상황을 초기 상태로 되돌린다. -진행 중 사용한 데이터를 모두 초기화한다.

Use Case	4. Show Product Status
Actors	System
Description	-사용자가 선택한 음료의 현재 상태를 디스플레이에 출력하는 기능

	<p>-판매 가능 /판매 불가능 중 현재 제품의 상태에 맞는 문구를 1초간 출력한다.</p> <p>-자판기는 SendMsg함수를 실행하여 타 자판기에 판매 정보를 요청한다.</p> <p>getMsg함수를 실행해 정보를 수신한다.</p>
--	---

Use Case	5. Show Payment
Actors	System
Description	-음료의 가격과 카드 투입 문구를 출력한다.

Use Case	6. Start Payment
Actors	System
Description	<p>-녹색 빛의 버튼을 눌렀을 때 수행된다.</p> <p>-결제는 카드 결제만 가능하다.</p> <p>-사용자는 디스플레이에 출력된 상품의 가격과 카드 투입 문구를 확인할 수 있으며, 카드 결제를 수행할 수 있다.</p>

Use Case	7. Send Msg
Actors	System
Description	<p>-자판기 간 네트워크 통신으로 Msg를 보낸다.</p> <p>-Msg Protocol을 준수한다.</p>

Use Case	8. Get Msg
Actors	System
Description	<p>-네트워크 통신으로 타 자판기의 정보를 받는다.</p> <p>-Msg Protocol을 준수한다</p>

Use Case	9. Show Location
Actors	System
Description	<p>-사용자가 선택한 음료의 재고가 있는 타 자판기의 위치를 출력한다.</p> <p>-타 자판기의 위치 정보는 텍스트 리스트 형식으로 표현되며, 자판기의 id 순으로 정렬되어 있다.</p> <p>-이때, 리스트는 자기 자신을 제외한 최대 9개 위치를 표시한다.</p>

Use Case	10. Select Location
Actors	User

Description	-사용자가 Show Location에 의해 출력된 위치 중 하나를 선택한다. -단일 자판기만 선택 가능하다.
-------------	---

Use Case	11. Show Prepayment
Actors	System
Description	-선택한 자판기 위치, 음료의 가격과 카드 삽입 요청 문구를 출력한다.

Use Case	12. Start Prepayment
Actors	System
Description	-Select Location 이후에 선결제를 진행한다.

Use Case	13. Input Card
Actors	User
Description	-사용자가 카드 리더기에 카드를 투입한다.

Use Case	14. Check Card Info
Actors	System
Description	-투입된 카드의 정보를 확인한다.. -카드 사, 카드 번호, 소유자 정보를 확인한다.

Use Case	15. Check Card Valid
Actors	System
Description	-투입된 카드의 자체 유효기간을 확인한다. -투입된 카드를 중간에 빼는 경우를 확인한다. -사용자의 카드 잔액과 선택한 음료수의 가격을 비교하여 결제하기 충분한 금액이 있는지 확인한다. -문제가 생긴 케이스의 로그를 저장한다. -투입된 카드로 결제가 불가능 하다면 Show Payment Result를 실행한다.

Use Case	16. Pay Card
Actors	System
Description	-상품 금액 결제 승인을 요청한다.

Use Case	17. Show Payment Result
----------	-------------------------

Actors	System
Description	-결제에 성공했다면 결제 성공,문구를 출력한다. -결제에 실패했다면, 결제 로그에서 결제 실패 사유를 출력한다.

Use Case	18. Make Code
Actors	System
Description	-현재 날짜 및 시간을 해쉬함수에 적용해 인증코드를 생성한다. -인증코드는 서로 중복되지 않는다.

Use Case	19. Delete Expired Code
Actors	System
Description	-인증코드의 생성 시간을 확인하여 만료된 인증코드는 삭제한다. -유효 시간은 인증코드 발급 이후 1시간이다.

Use Case	20. Show Code
Actors	System
Description	-생성된 인증코드를 사용자가 확인할 수 있도록 디스플레이에 출력한다.

Use Case	21. Save Code
Actors	System
Description	-수신받은 코드와 음료 이름을 저장한다.

Use Case	22. End Payment
Actors	System
Description	-결제가 완료되거나 사용자가 결제를 취소한 경우 결제를 종료한다. -사용자가 Cancel 을 수행한 경우, 결제 로그에 사용자 결제 취소를 저장한 뒤 결제를 종료한다.

Use Case	23. End Prepayment
Actors	System
Description	-선결제가 완료되거나 사용자가 선결제를 취소한 경우 실행된다. -선결제가 완료된 경우, Show Code 를 실행한다. -사용자가 Cancel 을 수행한 경우, 선결제 로그에 사용자 선결제 취소를 저장한 뒤 결제를 종료한다.

Use Case	24. Update Product Status
Actors	System
Description	-자판기가 현재 제품 상태를 최신화한다. -자판기가 초기 상태로 돌아갈 때 수행되며, 제품의 수량 및 판매 여부를 재확인 하여 Display Light에 영향을 준다.

Use Case	25. Show Code Input
Actors	System
Description	-자판기가 디스플레이에 인증코드 입력란을 출력한다. -대문자 영어와 숫자로 이루어진 키와, 그 위에 입력한 인증코드를 확인할 수 있는 확인란이 있다.

Use Case	26. Input Code
Actors	User
Description	-사전에 발급된 인증코드를 사용자가 입력한다.

Use Case	27. Check Code Valid
Actors	System
Description	-사용자로부터 인증코드를 입력받고 올바른 인증코드인지 확인한다. -잘못된 문법 또는 유효하지 않은 인증코드 입력 시 오류 로그를 리스트에 저장한다.

Use Case	28. Show Code Result
Actors	System
Description	-사용자로부터 인증코드를 입력 받은 뒤 이에 대한 입력 결과를 출력한다. -Check Code Valid에서 저장된 로그를 출력한다.

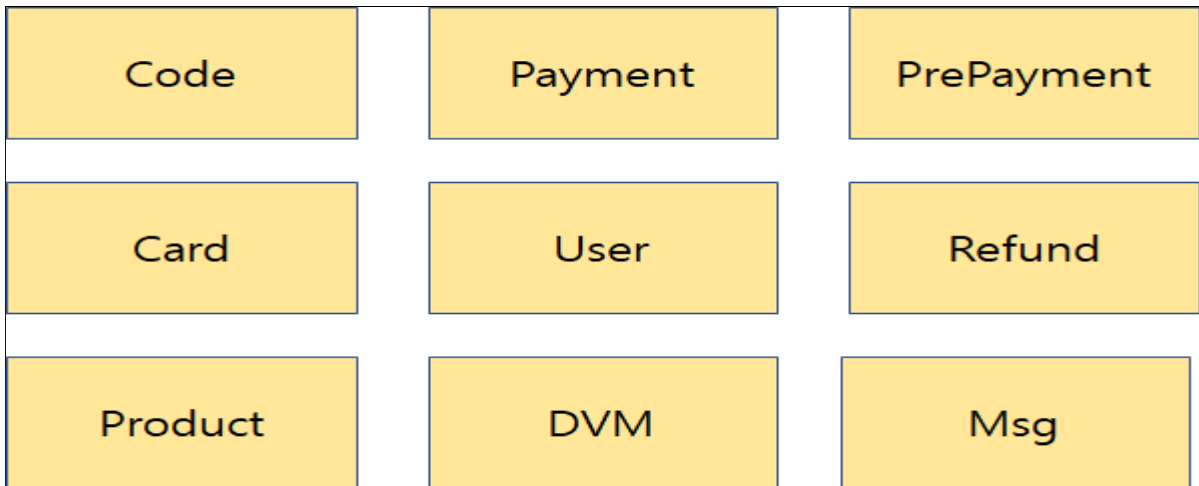
Use Case	29. Delete Used Code
Actors	System
Description	-사용된 인증코드를 소멸시킨다. -리스트에서 해당 인증코드를 제거한다.

Use Case	30. Get Refund
----------	----------------

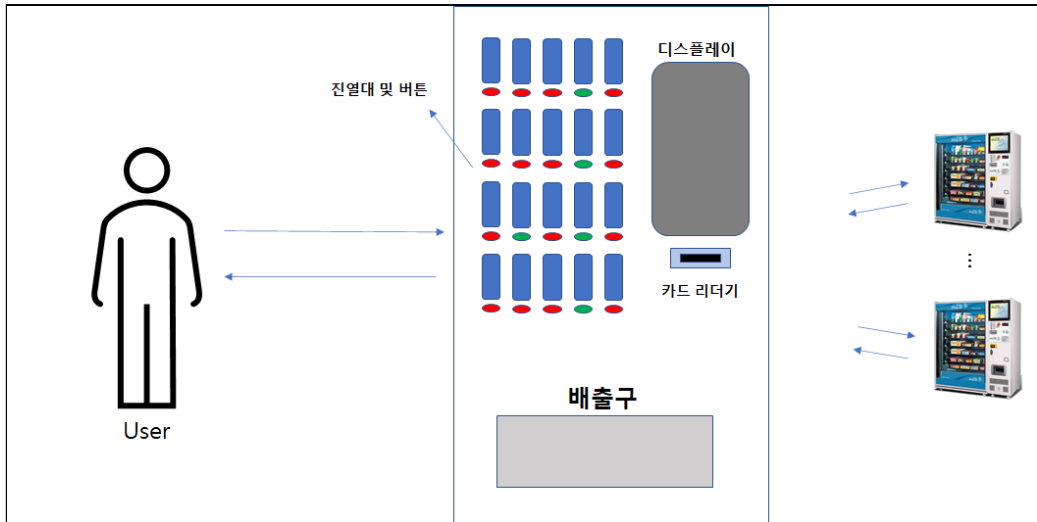
Actors	System
Description	<p>-사용자가 선결제 후 발급된 인증코드를 사용자로부터 1시간내로 입력받지못하면 음료수를 자동으로 환불해준다.</p> <p>-필요한 정보 :</p> <ol style="list-style-type: none"> 1. 카드 정보(카드 결제 계좌 정보, 이름 등) 2. 결제 정보(무슨 음료를 얼마주고 샀는지) 3. 인증코드 정보(어디 인증코드에 해당 결제가 할당되었는지, 몇분 지났는지) <p>-이후, 리스트에서 해당 코드를 제거한다.</p>

Use Case	31. Get Product
Actors	User
Description	<p>-사용자가 결제한 음료수를 배출한다.</p> <p>-배출은 올바른 인증코드 입력 또는 결제 완료 이후 이루어진다.</p> <p>-배출은 1개의 음료만 이루어진다.</p>

Activity 1007. Define Business Concept Model



Activity 1008. Define Draft System Architecture



Activity 1009. Define System Test Case

Test Number	Use Cases	Description
1	Display Light	-음료의 현 상태와 버튼의 불빛이 일치하는지 확인한다.
2	Select Product	-사용자가 선택한 버튼의 불빛에 따른 알맞은 작업이 진행되는지 확인한다.
3	Cancel	-Cancel 아이콘이 잘 출력되어 있는지 확인한다. -터치 가능 여부와 터치 시 해당 상태를 취소 할 수 있는지 확인한다.
4	Show Payment	-사용자가 선택한 음료의 가격과 출력된 음료의 가격이 일치하는지 확인한다. -카드 투입 요청 문구가 누락없이 출력되는지 확인한다.
5	Start Payment	-녹색 버튼을 눌렀을 때 결제를 시작하는지 확인한다.
6	Send Msg	-보내고자 하는 Msg정보와 실제로 보낸 Msg 정보가 일치하는지 확인한다.
7	Get Msg	-타 자판기의 정보를 올바르게 수신하는지 확인한다.
8	Show Location	-사용자가 선택한 음료를 판매 가능한 자판기만 보여주는지 확인한다.
9	Select Location	-자판기 위치를 사용자가 임의로 선택할 수 있는지 확인한다.
10	Show Prepayment	-사용자가 선택한 음료의 가격과 출력된 음료의 가격이 일치하는지 확인한다. -카드 투입 요청 문구가 누락없이 출력되는지 확인한다.
11	Input Card	-카드 리더기가 카드를 인식하는지 확인한다.
12	Check Card Info	-자판기가 인식한 카드 정보와 실제 투입된 카드 정보가 일치하는지 확인한다.

13	Check Card Valid	-투입된 카드의 실제 사용 가능 여부와 자판기가 판단한 사용 가능 여부가 일치하는지 확인한다. -결제로그에 실패 사유가 정상적으로 저장이 되는지 확인한다.
14	Pay Card	-카드사에 결제 요청이 성공적으로 이뤄지는지 확인한다.
15	Show Payment Result	-선결제 성공/실패 시 문구가 올바르게 출력되는지 확인한다. -실패 케이스에 따른 문구가 올바르게 출력되는지 확인한다.
16	Make Code	-선결제 완료 후 실행되는지 확인한다. -인증코드가 영문자 및 숫자로 이루어진 6자리 코드인지 확인한다. -인증코드의 중복여부를 확인한다.
17	Delete Expired Code	-생성된지 1시간 이상이 지난 인증코드에 대해서만 실행되는지 확인한다. -정상적으로 인증코드가 삭제되어 사용할 수 없는지 확인한다.
18	Save Code	-발급한 인증코드와 저장된 인증코드가 일치하는지 확인한다. -인증코드와 함께 쌍을 이루는 음료의 정보가 사용자가 선결제를 한 음료의 정보와 일치하는지 확인한다.
19	End Payment	-취소 버튼을 누르거나, 결제가 완료되었을 때 실행되는지 확인한다.
20	End Prepayment	-취소 버튼을 누르거나, 선결제가 완료되었을 때 실행되는지 확인한다.
21	Update Product Status	-자판기가 파악한 재고 및 판매 상태가 최신화되었는지 확인한다.
22	Check Code Valid	-사용자가 입력한 인증코드에 대하여 실행되는지 확인한다.
23	Delete Used Code	-사용된 인증코드에 대해서만 실행되는지 확인한다. -정상적으로 인증코드가 삭제되어 사용할 수 없는지 확인한다.
24	Get Refund	-사용자의 결제 금액과 환불 금액이 일치하는지 확인한다. -삭제되는 코드에 대해서만 실행되는지 확인한다. -중복 실행 여부를 확인한다.