

OOAD – Planning

Team 2 강상민 김재윤 이찬양 한승현

Activity 1001. Define Draft Plan

Motivation

자판기의 판매 재고에 따라 원하는 제품을 구매할 수 없는 경우를 막고자 한다.

현재 이용하고 있는 자판기에서 판매하지 않는 제품도 구매할 수 있게 해 편의성을 높이고자 한다.

분산 시스템을 이용해 자판기 간의 효율적인 통신과 재고 관리를 가능하게 한다.

Project Objectives

기존 자판기의 상품 구매 및 반환 기능을 기본으로 갖추되 타 자판기의 재고까지 고려한 자판기 분산 시스템을 제작한다.

사용자가 눈으로 쉽게 이해할 수 있는 UI를 제공하고, 타 자판기의 제품과 선결제 시스템을 연계해 사용자 편의성을 더욱 높인다.

타 자판기 제품을 선결제하면 해당 자판기 위치와 인증번호를 제공한다.

Activity 1001. Define Draft Plan

Functional Requirements

판매 가능한 제품을 모두 보여준다.

사용자에게 주문을 받는다.

한번에 구매 가능한 최대 음료 개수와 재고 현황은 자판기마다 다르다.

분산 시스템을 이용해 제품 판매 가능 여부를 확인한다.

타 자판기에 있는 제품 구매를 원한다면 해당 자판기의 위치를 제공하고 사용자에게 선결제 여부를 물어본다.

자판기 선정은 사용자가 위치한 곳의 자판기에서 가장 가깝고 음료의 재고가 있는 자판기로 한다.

결제는 카드만 가능하다.

사용자에게 결제를 요구한다.

카드사로부터 정보를 받아 결제를 승인 받는다.

결제는 카드 잔액이나 한도에 따라 실패할 수 있으며, 결제가 승인되면 제품이나 인증코드를 제공한다.

제품 재고 현황을 업데이트 한다.

타 자판기의 재고 확인 요청에 응한다.

Activity 1001. Define Draft Plan

Non-Functional Requirements

제품의 가격은 적당해야 한다.

통신시간이 너무 지연되지 않게 한다.

사용자가 통신 시스템 자판기 사용에 거부감을 갖지 않도록 사용자 친화적인 UI를 제공한다.

제품 종류와 재고에 대한 변경 가능성을 염두하며 설계한다.

Resource

인적 자원: 4명

작업 기간 : 2개월

투자 금액 : 인당 400만원, 총 1600만원

Activity 1002. Create Preliminary Investigation Report

Alternative Solutions

일반적인 재판기 판매 프로세스 코드를 구입한 후 기능을 추가한다.

분산 시스템을 이용한 통신 코드를 아웃소싱한다.

한번에 구매할 수 있는 최대 음료 개수로 인한 문제가 발생한다면 구매 가능 수량을 변경한다.

통신 딜레이 등으로 인한 재고 파악에 문제가 발생한다면 멀티 프로세싱을 최소화한다.

Activity 1002. Create Preliminary Investigation Report

Project Justification (Business Demands)

- Duration

제작기간 : 24.05 말까지 기본적인 소프트웨어 완성

24.09 말까지 자판기 시범 제작 및 구현

24.12 말까지 최종 시험 및 상용 준비 완료

- Risk

사용자가 원하는 음료수를 얻기 위해 다른 자판기까지 가야 하는 번거로움

자판기까지 향하는 경로 내의 모든 음료 소매업체가 경쟁사

자판기 간 통신 지연이 사용자의 불편을 유발할 수 있음

자판기마다 제공할 수 있는 서비스 조건이 모두 다름

분산 시스템 및 통신을 다뤄본 팀원이 없음

자판기마다 운용하는 코드가 다름

- Cost

자판기 한 대 평균 판매가격 : 500~650만원

자판기 내 디스플레이 설치 및 운송, 초기 투자비용 - 상당함

설치 완료시 재고 보충 및 유지관리 - 상대적 저렴

초기투자비용이 크지만 유지관리 비용은 비교적 저렴한 편

- Effect

분산 시스템이라는 미래지향적 인프라를 제공하며 자판기의 새로운 방향을 제시한다.

특정 상품의 판매 위치를 쉽게 제공할 수 있으므로 보다 사용자 친화적이다.

타 자판기에서의 제품 수령을 유도함으로써 판매 가능 제품 범위를 늘리고 제품 부족으로 인한 고객 손실을 막을 수 있다

Activity 1002. Create Preliminary Investigation Report

Risk Reduction Plan

다른 자판기까지의 이동 기피 및 포기를 막기 위해 선결제시 제품 가격을 인하하여 고객 손실을 막는다.

제품 수령 가능 기한을 넉넉하게 설정해 경쟁사로의 불가피한 고객 유출을 감내하되 매출 손실을 최소화한다.

자체제작과 아웃소싱을 동시에 진행해 신속한 분산 시스템 통신을 보장하되 오류를 최소화한다.

최대한 통일된 자판기 구현 코드를 위해 매주 지속적인 모니터링을 실시하고 팀 회의를 진행한다.

통신을 위한 팀 스터디를 진행한다.

Market Analysis

자판기 간 통신을 통해 음료 판매가 이뤄지고 있는 경우는 아직 없다.

편의점 증가로 인한 자판기 소비가 줄어들었다.

일반 매장에 비해 자리 차지가 적어 공간을 효율적으로 사용할 수 있다.

Other Managerial Issues

OOAD를 통해 개발해야 한다.

소프트웨어 제작 마감 기한이 하드하다.

Activity 1003. Define Requirements

개요: 분산형 자판기 시스템을 위한 controller SW를 개발한다.

고객: 자판기 음료를 구매하고자 한다. 이 때 사용하고 있는 자판기에 재고가 있는지는 고려하지 않는다.

Activity 1003. Define Requirements

Functional Requirements

- 사용자에게 정보를 제공한다.
 - 총 20종류의 음료수 리스트(이름, 가격)를 화면에 제공한다.
 - 음료수 종류 및 개수에 대한 선택지를 제공한다.
 - 사용자가 선택한 개수만큼 구매 가능한지에 대한 여부를 제공한다.
 - 이용 가능한 가장 가까운 타 자판기의 선결제 가능 여부를 고지한다.
 - 이용 가능한 가장 가까운 타 자판기 위치를 제공한다.
- 사용자가 음료수를 구매할 수 있도록 결제 시스템을 제공한다.
 - 카드 결제 시스템을 제공한다.
 - 선 결제 시스템을 제공한다.

Activity 1003. Define Requirements

Functional Requirements

- 사용자가 음료수를 수령할 수 있는 시스템을 제공한다.
 - 해당 자판기에서 음료수를 제공한다.
 - 다른 자판기에서 음료수를 수령할 수 있도록 인증 코드를 제공한다.
 - 타 자판기에 인증 코드를 제시할 시 음료수를 제공한다.
- 다른 자판기와 통신한다.
 - 다른 자판기의 재고 상태를 물어보는 송신 기능을 제공한다.
 - 본 자판기의 재고 상태를 응답하는 송신 기능을 제공한다.
 - 다른 자판기의 메시지를 수신하는 기능을 제공한다.
 - 다른 자판기에게 선결제를 요청하는 송신 기능을 제공한다.
 - 본 자판기의 선결제 가능 여부를 송신하는 기능을 제공한다.

Activity 1003. Define Requirements

Non-Functional Requirements

- 자판기에서 제공하는 GUI는 처음 사용하는 사용자도 직관적으로 사용법을 알 수 있어야 한다.
- 사용자가 필요한 정보들을 전부 제공한다.
- 타 자판기와의 통신에 따른 지연은 5초 미만이어야 한다.
- 사용자가 자판기를 사용함에 있어서 불편함을 느끼지 않도록 GUI가 매끄러워야 한다.

Activity 1004. Record Terms in Glossary

Term	Description
client	사용자
drink	음료수
credit card	카드
advance payment	선결제
card limit	카드 한도
get drink	음료수 수령
authentication code	인증코드
menu	음료수 정보
vending machine	자판기
location	자판기 위치

Activity 1005. Implement Prototype

User-Interface 기준

Authority

Exit

Choose

제품과 수량 선택 및 취소

제품 구매 가능 여부 확인

Pay

결제 금액 확인

제품 결제

결제 결과 확인

필요에 따른 선결제 여부 선택 & 인증코드 확인

Maintenance

재고 추가

재고 업데이트

재고 삭제

자판기 간 통신

자판기 간 통신 수정 및 업데이트

Receipt

사용자가 구매한 모든 제품과 수량

사용자가 지불한 금액

Activity 1006. Define Business Use Case

- Define System Boundary



- Identify the actors
 - Client : 음료수를 구매하기 위해 자판기와 상호작용하는 객체
 - Another Vending Machine Controller : 재고가 없는 음료수를 판매하기 위해 상호작용하는 다른 자판기 시스템

Activity 1006. Define Business Use Case

- Identify goals by client
 - 음료수의 종류 및 가격 확인
 - 음료수 구매
 - 음료수 선택
 - 카드 결제
 - 다른 자판기 위치 확인
 - 다른 자판기 음료수 선결제
 - 인증 코드 확인
 - 인증 코드 수령
 - 음료수 수령

Activity 1006. Define Business Use Case

- Identify goals by another vending machine
 - 자판기 위치 정보 확인
 - 음료수 재고 상황 정보 확인
 - 음료수 선결제 요청

Activity 1006. Define Business Use Case

- Identify use cases by actor-based
 - Client
 - Check menu information
 - Buy drink
 - Select drink
 - Pay by credit card
 - Check another vending machine location
 - Pay in advance
 - Confirm authentication code
 - Another vending machine
 - Get location
 - Get drinks quantity status
 - Request for advance payment

Activity 1006. Define Business Use Case

- Identify use cases by event-based
 - Issue authentication code
 - Get drink
 - Exceed card limits

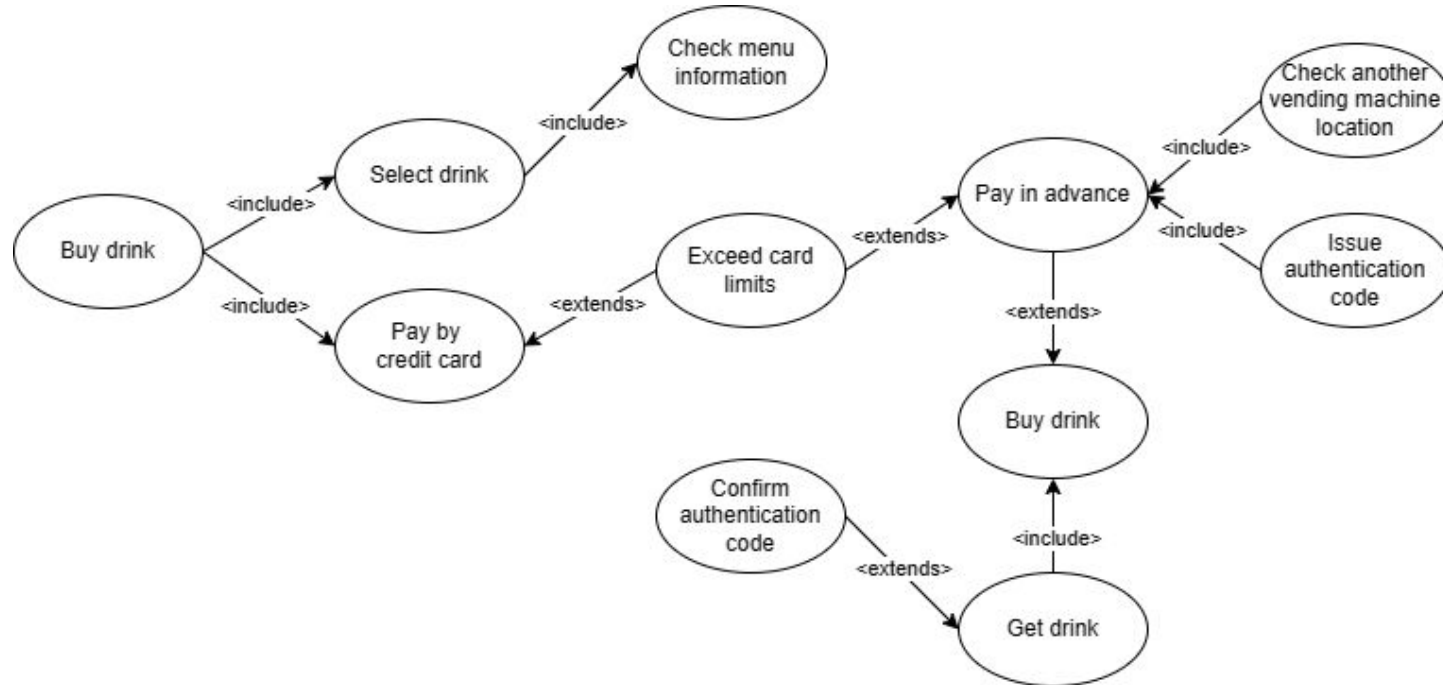
Activity 1006. Define Business Use Case

- Allocate system functions into related use cases & Categorize & Rank

Ref. #	Function	Use Case Number & Name	Category	Rank
R 1.1	Buy drink	1. Buy drink	Primary	High
R 1.2	Select drink	2. Select drink	Primary	High
R 1.3	Pay by credit card	3. Pay by credit card	Primary	High
R 1.4	Pay in advance	4. Pay in advance	Primary	High
R 1.5	Exceed card limits	5. Exceed card limits	Primary	High
R 2.1	Get drink	6. Get drink	Primary	High
R 2.2	Confirm authentication code	7. Confirm authentication code	Primary	High
R 2.3	Issue authentication code	8. Issue authentication code	Primary	High
R 3.1	Check menu information	9. Check menu information	Primary	High
R 3.2	Check another vending machine location	10. Check another vending machine location	Primary	High
R 4.1	Get location	11. Get location	Primary	High
R 4.2	Get drinks quantity status	12. Get drinks quantity status	Primary	High
R 4.3	Request for advance payment	13. Request for advance payment	Primary	High

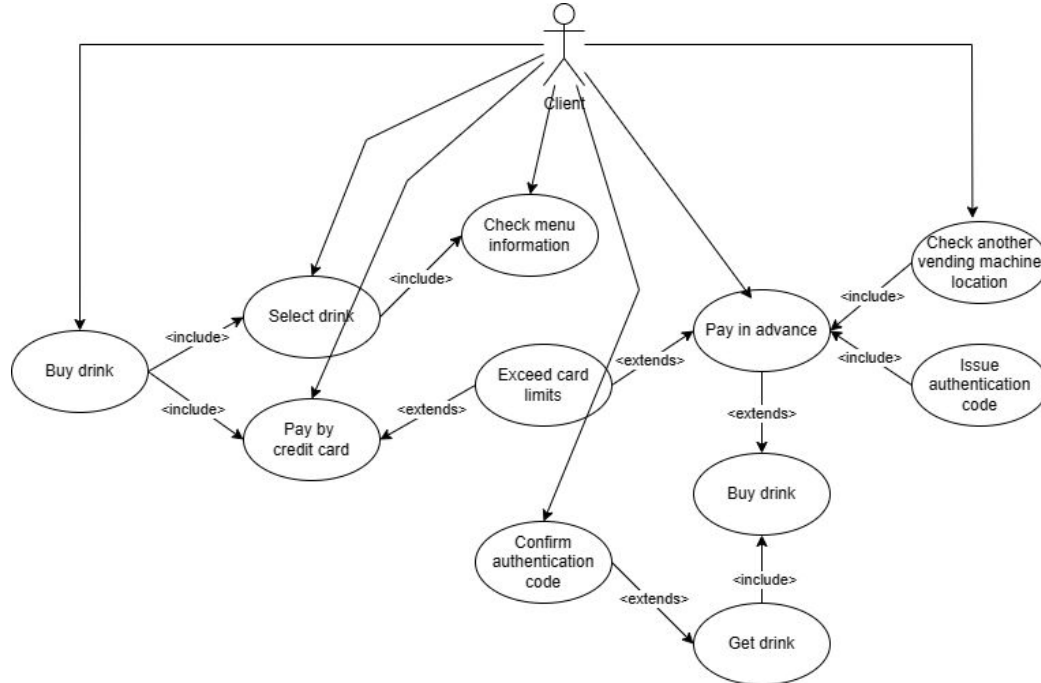
Activity 1006. Define Business Use Case

- Identify relationships between use cases



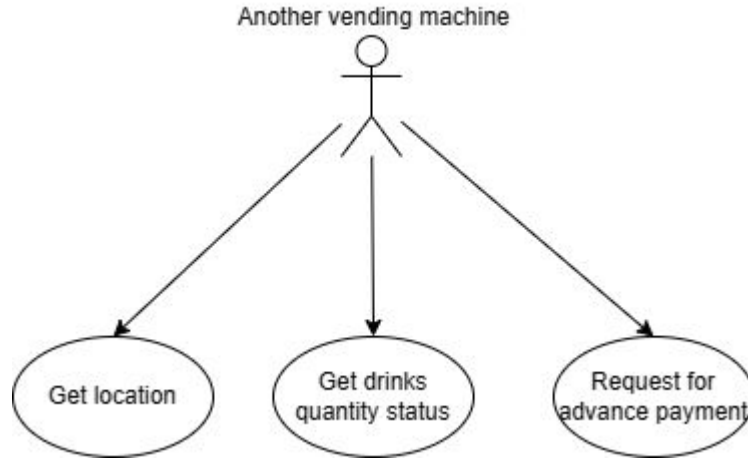
Activity 1006. Define Business Use Case

- Draw a use case diagram



Activity 1006. Define Business Use Case

- Draw a use case diagram



Activity 1006. Define Business Use Case

- Describe use cases

Use Case	1. Buy drink
Actors	Client
Description	음료수를 구매한다.

Use Case	2. Select drink
Actors	Client
Description	구매하고자 하는 음료수 종류를 UI에서 선택한다.

Activity 1006. Define Business Use Case

- Describe use cases

Use Case	3. Pay by credit card
Actors	Client
Description	구매할 가격 만큼 사용자의 카드에서 결제한다.

Use Case	4. Pay in advance
Actors	Client
Description	다른 자판기의 음료수를 사용자의 카드로 선결제한다.

Activity 1006. Define Business Use Case

- Describe use cases

Use Case	5. Exceed card limits
Actors	Event
Description	만약 구매하려는 금액이 사용자 카드의 잔액보다 크거나, 카드의 한도를 초과할 경우 결제하지 못한다.

Use Case	6. Get drink
Actors	Event
Description	음료수를 수령한다.

Activity 1006. Define Business Use Case

- Describe use cases

Use Case	7. Confirm authentication code
Actors	Client
Description	사용자의 인증 코드를 입력받고 확인한다.

Use Case	8. Issue authentication code
Actors	Event
Description	인증 코드를 사용자에게 발급한다.

Activity 1006. Define Business Use Case

- Describe use cases

Use Case	9. Check menu information
Actors	Client
Description	음료수 종류와 각 가격의 리스트를 확인한다.

Use Case	10. Check another vending machine location
Actors	Client
Description	선결제한 다른 자판기의 위치 정보를 확인한다.

Activity 1006. Define Business Use Case

- Describe use cases

Use Case	11. Get location
Actors	Another vending machine
Description	타 자판기에서 우리 자판기의 위치 정보를 받는다. 타 자판기의 아이디와 우리 자판기의 아이디를 함께 전달한다.

Use Case	12. Get drinks quantity status
Actors	Another vending machine
Description	타 자판기에서 우리 자판기의 음료수의 재고 현황을 받는다. 타 자판기의 아이디와 우리 자판기의 아이디를 함께 전달한다.

Activity 1006. Define Business Use Case

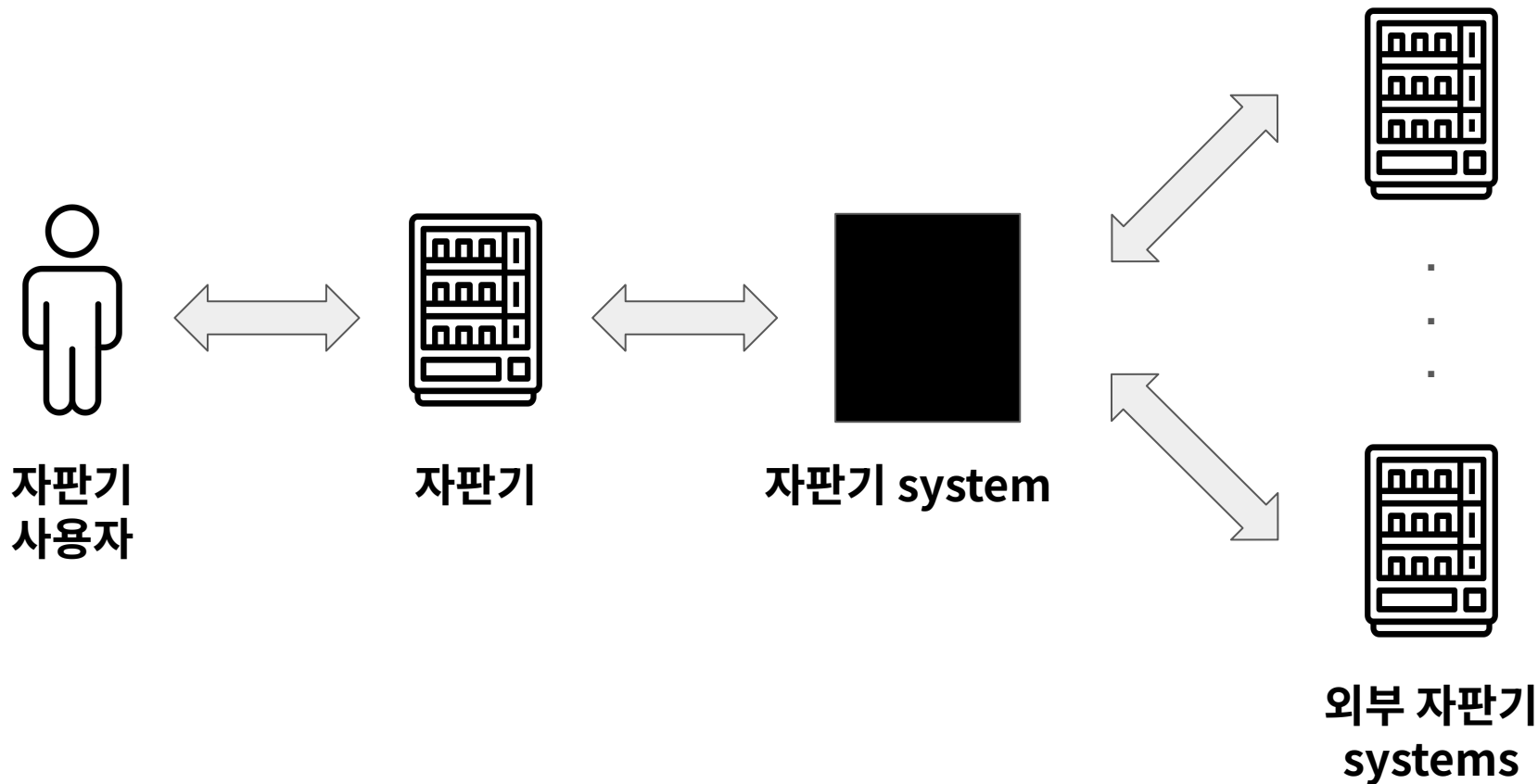
- Describe use cases

Use Case	13. Request for advance payment
Actors	Another vending machine
Description	타 자판기는 우리 자판기 시스템에 현재 선결제가 가능한 지 묻는 요청을 보낸다. 타 자판기의 아이디와 우리 자판기의 아이디를 함께 전달한다.

Activity 1007. Define Business Concept Model

- Identify 'Concepts' in the target domain.
 - Drink
 - Pay
 - Location
 - Authentication-Code

Activity 1008. Define Draft System Architecture



Activity 1009. Define System Test Case

NUM	User Case	Test Case	Description
1	Select drink	음료 개수 선택	사용자가 유효한 개수의 음료를 선택하였는지 확인
2		음료 종류 선택	사용자가 판매중인 음료를 선택하였는지 확인
3	Pay by credit card	결제 금액 차감	카드의 잔고에서 결제한 금액만큼 차감됐는지 확인
4		결제	카드 혹은 잔고에 문제가 없을 때 결제가 정상적으로 이루어지는지 확인
5		결제 가격	실제 총 구입금액과 결제된 가격이 일치하는지 확인
6	Get drink	음료 반출	결제한 음료가 수량과 종류를 맞춰 반출하는지 확인
7		음료 제공 자판기	음료가 사용자가 선택한 자판기에서 반출되는지 확인
8	Pay in advance	선결제 메시지 발신	선 결제 메시지가 문제없이 발신되었는지 확인
9		선결제	타 자판기에서 판매하는 음료를 현재 자판기에서 선결제 할 수 있는지 확인

Activity 1009. Define System Test Case

NUM	User Case	Test Case	Description
10	Exceed card limits	결제 오류 제어	카드 혹은 잔고에 문제가 생겼을 때 오류를 제어할 수 있는지 확인
11	Confirm authentication code	인증코드 입력	알맞은 인증코드 입력 시 음료가 문제없이 반출되는 지 확인
12	Issue authentication code	인증코드 생성	선결제 시 인증코드가 규칙대로 생성되는지 확인
13		인증코드 중복	해당 인증코드가 미사용된 인증코드와 중복되는지 확인.
14	Check menu information	음료 디스플레이	음료 정보가 실제 자판기에 저장되어 있는 것과 일치하는지 확인
15	Check another vending machine location	음료 반출	결제한 음료가 수량과 종류를 맞춰 반출하는지 확인
16	Get location	음료 제공 자판기	음료가 사용자가 선택한 자판기에서 반출되는지 확인
17	Get drinks quantity status	선결제 메시지 발신	선 결제 메세지가 문제없이 발신되었는지 확인
18	Request for advance payment	선결제 메시지 수신	선 결제 메세지가 문제없이 수신되었는지 확인

감사합니다