

OOI 2nd Cycle

Distributed Vending Machine

TEAM1

202111290 박범식

202211382 조휘원

202211373 정지원

Table of Contents

- 0. 동작 화면
- 1. Specification Review
 - 1-1. Stage 2050
- 2. Category Partition Test
- 3. Brute Force Test
- 4. 피드백 반영 후 Unit Test 결과

0 동작화면

RUC 01
초기 메뉴 선택

안녕하세요 , Team1 DVM 입니다 .

희망하는 옵션을 선택해주세요 .

1. 음료 구매하기

2. 선결제 한 음료 받아가기

Enter menu :

0 동작화면

RUC 02

음료 종류 및 수량 선택

```
Enter menu : 1

구매 하고 싶은 메뉴와 수량을 입력해주세요 .
ex) 02 10

-----

콜라 (01)
사이다 (02)
녹차 (03)
홍차 (04)
밀크티 (05)
탄산수 (06)
보리차 (07)
캔커피 (08)
물 (09)
에너지드링크 (10)
유자차 (11)
식혜 (12)
아이스티 (13)
딸기주스 (14)
오렌지주스 (15)
포도주스 (16)
이온음료 (17)
아메리카노 (18)
핫초코 (19)
카페라떼 (20)

-----

Enter menu : 01 1
```

0 동작화면

RUC 03

카드 정보 입력과 확인,
결제 에러 처리

RUC 04

음료 판매 처리
(현재 자판기)

```
Enter menu : 01 1
음료 가격 총 1500원 (콜라 (01) 1개 1500원 * 1)
```

```
Enter menu : 01 1
음료 가격 총 1500원 (콜라 (01) 1개 1500원 * 1)

카드 정보를 입력해주세요.

Enter your card : 3422-abcd

결제가 불가능한 카드 정보입니다. 다시 입력해주세요.
```

```
카드 정보를 입력해주세요.

Enter your card : 3422-abcd

결제가 완료되었습니다.

메인 화면으로 돌아갑니다.
계속하려면 Enter를 누르세요...
```

0 동작화면

RUC 05

다른 자판기 조회
및 위치 안내

RUC 06

선결제 구입
(다른 자판기)

```
Enter menu : 03 1

현재 해당 자판기에서 구매가 불가능합니다 .
(3, 4) 위치의 자판기에서 구매가 가능합니다 .

카드 정보를 입력해주세요 .

Enter your card : 4322-abcd

결제가 완료되었습니다 .
=====
자판기 위치 : (3, 4)
인증코드 : 73GJr
=====

메인 화면으로 돌아갑니다 .
계속하려면 Enter를 누르세요 ...
```


0 동작화면

RUC 07

선결제 음료 수령

안녕하세요 , Team1 DVM 입니다 .

희망하는 옵션을 선택해주세요 .

- 1. 음료 구매하기
- 2. 선결제 한 음료 받아 가기

Enter menu : 2

선결제 인증코드를 입력해주세요 .

Enter your prepayment code : 73GJr
선결제 한 음료가 제공되었습니다 .

메인 화면으로 돌아갑니다 .
계속하려면 Enter를 누르세요 ...

1 Specification Review

1-1. Stage 2050

01
DVM

REVIEW

“Purpose : 다른 자판기”와 “overview : 타 자판기" 용어를 통일시켜야 함
Overview : 내부적으로 Private을 말하는건지 그렇다면 어떻게든 공개를 안한다고 하기에는 좌표, 판매목록, 판매 가능한 품목은 공개를 해야 해야함
= 명시 제대로 해야함

Overview

자판기 ID, 좌표, 재고, 판매 목록, 판매 가능한 품목 목록, 타 자판기 리스트를 내부적으로 보유한다.



Overview

자판기 ID, 좌표, 재고, 판매 목록, 판매 가능한 품목 목록, **다른 자판기 리스트 정보를 보유한다.**

1 Specification Review

1-1. Stage 2050

02
DVM::queryItems

REVIEW

전체 자판기의 의미 (주변 자판기 포함인지 전국에 있는 모든 DVM인지) Purpose와 Overview 부분 하고자 하는 방향이 다름
Overview에서 말하고자 하는 " 판매 가능 하다는건" -> 재고가 0이면 조회 안 한 다는 것인지 명확히 나와있지않음

Purpose	전체 자판기에서 취급 중인 음료 목록을 조회한다.
Overview	자판기에서 판매 가능한 전체 음료 아이템 목록을 조회한다.



Purpose	음료 목록을 조회한다.
Overview	자판기에서 실제 취급하고 있는 음료가 아닌, 판매 가능한 20가지의 전체 음료 아이템 목록을 조회한다.

1 Specification Review

1-1. Stage 2050

03
DVM::queryStocks

REVIEW

다른 DVM에서 재고 확인 실패 시 메시지 부분 보안 필요
Abstract Operation 부분 3번이 실행하기전 신호를 받은 DVM에 재고 확인
전달된 인증코드가 선결제 정보와 일치하는지 부분에서 선결제 정보를 어떤형식으로 가져올지 정확히 구분 필요

Input	itemCode: string, count: int
Abstract Operation	1. DVM 재고 확인 2. (1) 없을 시, Other DVM에게 질의 3. 가장 가까운 위치 계산 4. 결과 메시지 생성
Exceptional Courses of Events	itemCode 존재하지 않을 시, findItem() 내부에서 예외 발생



Input	itemCode: string (인증코드 형식: 5자리 영문+숫자 ex) ab23A), count: int
Abstract Operation	1. DVM 재고 확인 2. (1) 없을 시, Other DVM에게 순차적으로 질의 3. 응답 중 재고 있는 DVM 중 가장 가까운 위치 계산 4. 결과 메시지 생성
Exceptional Courses of Events	타 DVM 재고 없음 or 연결 실패 시 → flag:not_available 반환 itemCode 미존재 시 → findItem() 내부에서 예외 발생 가능

1 Specification Review

1-1. Stage 2050
05
DVM::requestOrder

REVIEW

선 결제 실패시 (인증코드 실패, 다른 DVM과의 통신 문제, 소켓 오류)등에 대한 오류 구분이 필요
결과 반환 부분 포맷 추가 필요
반환이라고 하면 기존에 가지고 있던 인증코드를 준다는 의미 X => 생성 및 반환 이라 명시해야함

Abstract Operation	1. 대상 DVM 탐색 및 선결제 가능 확인 2. Sale::createSaleForDvm() : 선결제 Sale 생성 3. askForPrepayment() 4. 결과 반환
Exceptional Courses of Events	결제 가능 DVM 존재하지 않거나 (DVMNotFoundException), 선결제 실패 시 예외 발생



Abstract Operation	1. 대상 DVM 탐색 및 선결제 가능 확인 2. Sale::createSaleForDvm() : 선결제 Sale 생성 3. askForPrepayment() 4. 선결제 결과 생성 및 반환
Exceptional Courses of Events	결제 가능 DVM 존재하지 않거나 (DVMNotFoundException), 선결제 실패(인증코드 생성 실패, 다른 DVM과의 소켓통신 네트워크 문제) 시 예외 발생

1 Specification Review

1-1. Stage 2050
10

OtherDVM::
askForPrepayment

REVIEW

다른 자판기 or 외부 자판기 명칭 통일 필요
응답 DVM에게 전송 후 전송 DVM의 상태 설정 필요

Purpose	외부 자판기에 선결제 가능 여부를 확인한다.
---------	--------------------------



Purpose	다른 자판기에 선결제 가능 여부를 확인한다.
---------	--------------------------

+ 물리적으로 음료를 제공하는 DVM은 전송 DVM이 아닌 응답 DVM이라고 판단하여,
전송 DVM의 상태 변경은 별도로 수행하지 않으며, 기존 로직을 유지하여 응답 DVM에서만 재고 차감 및 상태 변경

1 Specification Review

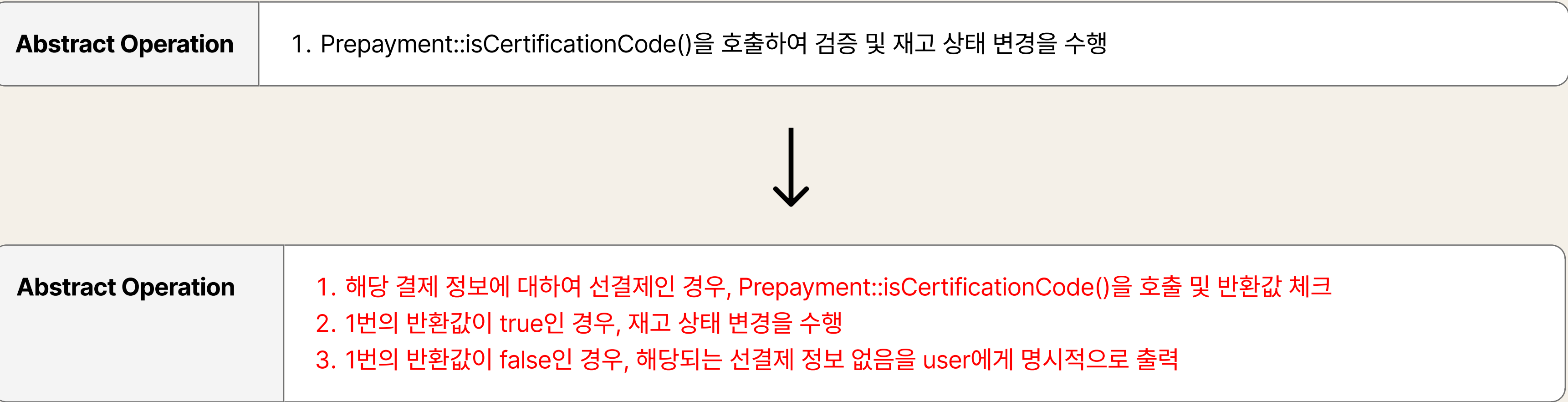
1-1. Stage 2050

17

Sale::
receivePrepaidItem

REVIEW

선 결제 실패시 (인증코드 실패, 다른 DVM과의 통신 문제, 소켓 오류)등에 대한 오류 구분이 필요해보임 => 해당 되지 않는 에러 처리
결과 반환 부분 포맷 추가 필요
선 결제 정보를 가져오는 부분이 음료의 카테고리 형태로 나뉘서 가져오는지 전체 선결제 정보를 탐색하는지 구체적인 부분이 필요 함



2 Category Partition Test

Category Partition Test



No	Group	Category	Value	Constraint	P/F	
1	Stock Status	no stock	Handles lack of inventory	error	Pass	
2	Target DVM	other DVM	Request to other vending machine		Fail	타 자판기와 연동 불가
3	Certification Code	used code	Reused code error	error	Fail	인증 중복확인 불가
4	Certification Code	invalid format	Wrong format handling	error	Pass	
5	Item Code	invalid item code	Invalid item code handling	error	Pass	
6	Quantity Request	over stock	Ordering more than available	error	Pass	
7	Card Format	invalid format	Card format error	error	Pass	
8	Socket Response	delayed	Slow response case		Fail	
9	Message Type	prepayment request	Selects prepayment path		Pass	
10	Certification Code	already used code	Reuse error	error	Fail	중복 확인 부분 불가
11	Certification Code	invalid code format	Invalid regex format	error	Pass	
12	Input Code Format	too short	Less than required length	error	Pass	
13	Input Code Format	symbols present	Contains special characters	error	Pass	
14	Card Format	invalid format	Card input error	error	Fail	카드 형식에 대한 설명이 없음 및 카드 형식에 맞춰도 오류
15	Purchase Flow	direct sale + all valid	Standard direct purchase flow		Fail	카드 결제부분에서 막힘
16	Purchase Flow	prepaid sale + all valid	Standard prepaid flow		Fail	카드 결제부분에서 막힘
17	Certification Code	valid unused code	Code is valid and unused		Pass	
18	Input Code Format	5-char alphanumeric	Properly formatted code		Pass	

Pass Rate = 11 /18 = 61%

2 Category Partition Test

CT02

RUC06 선결제 구입: 특정 DVM 대상의 선결제를 처리하고 인증 코드를 반환한다.

Test Report

타자판기와 연동 불가

DVM 간 연결을 위한 명령행 인수 전달 방식이 미흡하여 발생한 테스트 환경 구성상의 문제로 판단 (수정완료)

1. 명령행 인수 기반의 연결 방식 README 에 추가
2. main.cpp 포트 및 IP 유효성 검증 기능 추가

```
명령행 인수로 다른 DVM 설정 지정

# 기본 설정으로 실행 (다른 DVM 없이)
./build/app_program

# 다른 DVM 하나와 연결
./build/app_program 172.20.10.2:9001

# 여러 DVM과 연결
./build/app_program 172.20.10.2:9001 192.168.1.100:9002

# 사용법: ./app_program [다른_DVM_IP:PORT] [다른_DVM_IP:PORT] ...
# 주의: 자신의 포트는 9000으로 고정되어 있습니다.
```

```
pair<string, int> parseIpPort(const string& ipPort) {
    size_t colonPos = ipPort.find(':');
    if (colonPos == string::npos) {
        throw invalid_argument("IP:PORT 형식이 아닙니다: " + ipPort);
    }

    string ip = ipPort.substr(0, colonPos);
    string portStr = ipPort.substr(colonPos + 1);

    if (ip.empty() || portStr.empty()) {
        throw invalid_argument("IP 또는 PORT가 비어있습니다: " + ipPort);
    }

    int port = stoi(portStr);
    if (port <= 0 || port > 65535) {
        throw invalid_argument("포트 번호가 잘못되었습니다: " + portStr);
    }

    return make_pair(ip, port);
}
```

2 Category Partition Test

CT03, CT10

RUC07 선결제 음료 수령: 인증코드를 입력받아 선결제된 음료를 수령 처리

Test Report

인증코드 중복 확인 불가

해당 이슈는 실제 문제라기보다는 메시지 표현 방식으로 인한 이해 오해로 판단 (유지사항)

```
bool result = dvm->processPrepaidItem(certCode);
if (!result)
{
    cout << "이 자판기에서 선결제한 음료가 없습니다.\n" << endl;
}
```

인증코드가 존재하지 않거나, 이미 사용된 코드인 경우 모두
“이 자판기에서 선결제한 음료가 없습니다.” 가 출력된다.

이미 사용된 인증코드는 내부적으로 무효 처리되어
유효성 검사를 통과하지 못하는 것이 정상동작이라고 판단되어 유지한다.

2 Category Partition Test

CT08

RUC05 다른 자판기 조회 및 위치 안내: 특정 DVM 대상의 선결제를 처리하고 인증 코드를 반환한다.

RUC06 선결제 구입: 다른 DVM으로부터 수신한 선결제 판매를 저장한다.

Test Report

응답이 느린 케이스에 대한 대응이 필요

OOPT1000에 기술했던 대로 3초 내 응답 관련 timeout 처리를 함. (수정완료)

Controller::runServer → 수신 측 서버 소켓 타임 아웃 설정 (서버에서 응답을 받을 때

OtherDVM::findAvailableStocks → 응답 수신 제한 시간 설정 (다른 DVM의 응답을 기다릴 때)

OtherDVM::estabilishConnection → 연결 시도 시간 제한 설정 (다른 DVM과 연결을 시도할 때)

```
struct timeval timeout;
timeout.tv_sec = 3;
timeout.tv_usec = 0;
setsockopt(sock, SOL_SOCKET, SO_RCVTIMEO, &timeout, sizeof(timeout));
```

2 Category Partition Test

CT14, 15, 16

RUC03 카드 정보 입력과 확인, 결제 에러 처리: 카드 정보를 입력하고 유효성 검증 및 결제를 시도한다.

Test Report

카드 형식에 대한 오류
카드 결제 막힘

1. 카드 형식 명시 (유지사항)
숫자4자리-영문4자리 **ex) 1583-aDGz**
PFR 상 '카드는 휘발성 정보' 라는 기준에 따라 실제 결제 시스템 없이 유효성 검사만 수행

2. 카드 결제 막힘 (유지사항)
다른 자판기와의 통신을 재시도 한 후, 카드 결제 재시도 필요

카드 정보를 입력해 주세요 .
Enter your card : 1234-abcd
결제가 완료되었습니다 .

3 Brute Force Test

TestNum	Use Case	Test Description	Expected Output
BF-01	RUC 01 초기메뉴 선택	콘솔에 선택지를 보여주고, 사용자가 유효한 숫자(1 또는 2)를 입력할 때까지 반복 입력을 받는다.	1이나 2가 아니면 잘못 입력했다는 메시지와 함께 다시 입력하도록 해야함
BF-02	RUC 01 초기메뉴 선택	각 번호에 맞게 입력했을 때 올바르게 출력값이 나오는지 확인	1번 입력했을 경우 음료 구매 화면으로, 2번 입력 했을 경우 선결제 화면으로 이동
BF-03	RUC 02 음료 종류 및 수량 선택	유효하지 않은 번호 입력 시 결과 확인	음료 이외 번호 입력 시 화면 변화없이 재입력 요청
BF-04	RUC 02 음료 종류 및 수량 선택	입력 한 후 초과시간 지났을 때 반응 확인	초기화면으로 돌아가야 함
BF-05	RUC 02 음료 종류 및 수량 선택	숫자가 아닌 문자열을 넣었을 때 반응 확인	문자열이 들어왔으니 화면 변화없이 재입력 요청 및 안정적 화면 출력
BF-06	RUC 02 음료 종류 및 수량 선택	수량을 0 및 음수로 입력했을 때 반응 확인	0과 음수 수량은 유효하지 않으므로 재입력 요청
BF-07	RUC 02 음료 종류 및 수량 선택	구매하고 싶은 수량은 자라수 제한이 있는지	02 256 과 같이 수량이 너무 큰 값이 입력될 경우 재고가 부족하니 최대한 이하로 입력해달라고 메시지 출력, 최대값이 뺏으로 되어 있는가
BF-08	RUC 02 음료 종류 및 수량 선택	자판기에서 판매 가능한 전체 음료 아이템 목록을 출력한다.	1부터 20까지의 메뉴 display
BF-09	RUC 03 카드 정보 입력과 확인	음료 가격이 올바르게 계산되는지 확인	음료 가격 * 수량으로 문제 없음
BF-10	RUC 03 카드 정보 입력과 확인	유효한 카드정보와 유효하지 않은 카드정보 입력 시 결과 확인	유효한 카드 정보 입력시 결제가 완료되며, 유효하지 않은 카드 정보는 다시 입력 요청
BF-11	RUC 03 카드 정보 입력과 확인	카드번호에 공백, 특수문자가 포함 된 경우	잘못된 형식으로 간주되어 재입력 요청
BF-13	RUC 04 음료 판매 처리(현재 자판기)	결제 완료 후 Enter 누를 경우와 다른 key를 눌렀을 때 확인	Enter를 입력하면 메인 화면으로 돌아감
BF-14	RUC 04 음료 판매 처리(현재 자판기)	음료 결제 후 자판기에서 해당 수량만큼 차감되는지 확인	해당 수량만큼 차감 됨
BF-15	RUC 05 다른 자판기 조회 및 위치 안내	현재 자판기 및 다른 자판기의 재고 상태를 확인하고 본인의 재고가 충분할 경우 판매한다.	재고가 충분하지 않다면 가장 가까운 다른 판매기의 위치를 출력 및 선결제 진행
BF-16	RUC 05 다른 자판기 조회 및 위치 안내	재고가 충분하지 않을 경우 다른 자판기를 알려주거나 재고가 없을 경우 not available 출력	다른 자판
BF-17	RUC 05 다른 자판기 조회 및 위치 안내	카드 정보 잘못 입력 시, 초기화면으로 돌아가건 아닌지 확인	카드
BF-18	RUC 05 다른 자판기 조회 및 위치 안내	다른 자판기에도 재고가 없으면 어떤 메시지가 나오는지	재고가 1
BF-19	RUC 05 다른 자판기 조회 및 위치 안내	재고가 있는 다른 자판기 위치가 제대로 출력되는지	(3, 4) 위치의 지
BF-20	RUC 05 다른 자판기 조회 및 위치 안내	인증코드가 정상적으로 S자리로 생성되는지 확인	
BF-21	RUC 05 다른 자판기 조회 및 위치 안내	구매 가능한 자판기 위치와 인증 코드 출력되는지	위치 (3, 4)
BF-22	RUC 06 선결제 구입	특정 OVM 대상의 선결제를 처리하고 인증 코드를 반환한다.	선결제 금액을 Sale 객체에 판매 아이
BF-23	RUC 07 선결제 음료 수령	유효한 인증코드와 유효하지 않은 코드 입력 시 결과 확인	유효한
BF-24	RUC 07 선결제 음료 수령	음료 수령 시, 어떤 자판기에서 수량이 빠져나가는지 확인	선결제
BF-25	RUC 07 선결제 음료 수령	음료 제공 후 메인 화면으로 돌아갈 때 Enter가 아닌 다른 키가 입력했을 경우 확인	Er
BF-26	RUC 07 선결제 음료 수령	인증코드를 입력받아 선결제된 음료를 수령 처리	sales 내에 3
BF-27	RUC 07 선결제 음료 수령	인증 코드를 한 번만 사용할 수 있도록 사용 여부 관리를 한다.	사용 여부
BF-28	RUC 07 선결제 음료 수령	입력된 인증코드가 일치하고, 아직 사용되지 않았다면 사용 처리를 한다.	사용

TestNum	pass/fail	비고
BF-01	Pass	
BF-02	Pass	홈으로 되돌아가기 버튼이 있었으면 좋을 것 같음 / 잔여수량도 가격도 안 뜨는 상황
BF-03	Fail	오류값 입력 시, 재입력 요청은 있지만 초기화면으로 돌아감, 또한 음료번호는 01,02가 아닌 1,2 으로 할 시 무한 입력 요청 루프 반복
BF-04	Fail	SRS에는 안 들어가있지만 입력시간 초과 인 경우 대응 필요
BF-05	Fail	재입력 요청은 있지만 마찬가지로 홈버튼으로 돌아가고, 음료 + 문자열로 수량 입력 시 출력값이 이상하게 됨
BF-06	Pass	재입력 요청은 있지만 마찬가지로 홈버튼으로 돌아가고, 수량을 -1로 했을 경우에는 홈이 아니라 다시 음료 선택하는 화면 출력(?)
BF-07	Fail	높은 값으로 수량 입력 시, 무한 입력 요청 루프 반복 현상
BF-08	Fail	수량, 가격 미출력
BF-09	Fail	음료 03번(녹차)부터는 결과를 출력 없이 무한 입력 요청 루프 반복 현상
BF-10	Pass	
BF-11	Fail	유효한 카드번호 3422-abcd인데 대문자로 3422-abcD로 했을 때도 결제가 되는 issue 발생
BF-13	Pass	임의의 값 넣었을 때는 변화없지만 그 후 enter 누르면 초기화면으로 돌아가긴 하지만, 출력에서 이상한 문자 출력
BF-14	Pass	잔여 수량이화면에 뜨진 않지만 임의로 값을 넣어서 확인 완료
BF-15	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-16	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-17	Pass	
BF-18	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-19	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-20	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-21	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-22	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-23	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-24	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-25	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-26	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-27	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가
BF-28	Fail	다른 자판기와 통신이 connection failed 라서 확인이 불가

Pass Rate = 7/28=25%

3 Brute Force Test

BF02

RUC01 초기메뉴 선택:각 번호에 맞게 입력했을 때 올바르게 출력값이 나오는지 확인

Test Report

홈으로 되돌아가기 버튼이 있었으면 좋을 것 같음
잔여수량/가격 나타나지 않는 상황

Controller.cpp

코드 반영 완료
→ 모든 입력에서 00 및 0을 입력하면 초기화면으로 돌아가도록 수정

잔여 수량 및 가격은 PFR 및 SRS에 기재되어 있지 않은 사항

```
114 int Controller::displayMenu()  
115 {  
116     cout << "\n안녕하세요, Team1 DVM 입니다.\n"  
117         << endl;  
118     cout << "희망하는 옵션을 선택해주세요.\n"  
119         << endl;  
120     cout << "1. 음료 구매하기" << endl;  
121     cout << "2. 선결제 한 음료 받아 가기" << endl;  
122     cout << "0. dvm 종료\n"  
123         << endl;  
124  
125     int choice;  
126     while (true)  
127     {  
128         cout << "Enter menu : ";  
129         if (cin >> choice)  
130         {  
131             if (choice == 0)  
132             {  
133                 exit(0);  
134             }
```

```
208  
209  
210     if (menu == "00")  
211     {  
        exit(0);  
    }
```

```
325  
326  
327     cout << "Enter your prepayment code : ";  
328     string certCode;  
329     cin >> certCode;  
330     if (certCode == "00")  
331     {  
        exit(0);  
    }
```

3 Brute Force Test

BF03

RUC02 음료 종류 및 수량 선택 : 유효하지 않은 번호 입력 시 결과 확인

Test Report

오류값 입력 시, 재입력 요청은 있지만 초기화면으로 돌아감
또한 음료 번호는 01,02가 아닌 1,2로 입력 시 무한 입력 요청 루프 반복

Controller.cpp

코드 반영 완료

- 기존 return 문에서 continue문으로 변경하여
- 초기 화면이 아닌 음료 구매 프로세스로 돌아가도록 수정

```
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
try  
{  
    int menuInt = stoi(menu);  
    if (menu.length() != 2 || menuInt < 1 || menuInt > 20)  
    {  
        cout << "\n음료를 잘못 입력하셨습니다. 다시 입력해주세요. \n";  
        cout << endl;  
        continue;  
    }  
    if (count < 1)  
    {  
        cout << "\n수량을 잘못 입력하셨습니다. 다시 입력해주세요. \n";  
        cout << endl;  
        continue;  
    }  
}
```

3 Brute Force Test

BF04

RUC02 음료 종류 및 수량 선택 : 입력 안 한 후 초과시간 지났을 때 반응 확인

Test Report

SRS에는 안 들어가 있지만 입력시간 초과인 경우 대응 필요

Controller.cpp

코드 반영 완료

→ 입력 무응답으로 20초의 시간이 경과하면 초기화면으로 돌아감

```
184 struct pollfd pfd = {STDIN_FILENO, POLLIN, 0};
185 int ret = poll(&pfd, 1, 20000);
186
187 if (ret == 0)
188 {
189     cout << "\n입력 시간이 초과되어 초기화면으로 돌아갑니다.\n";
190     break;
191 }
192 else if (ret < 0)
193 {
194     perror("poll 실패");
195     break;
196 }
197
```


3 Brute Force Test

BF05

RUC02 음료 종류 및 수량 선택 : 숫자가 아닌 문자열을 넣었을 때 반응 확인

Test Report

재입력 요청은 있지만 마찬가지로 홈버튼으로 돌아가고, 음료 + 문자열로 수량 입력 시 출력 이상하게 됨

Controller.cpp

코드 반영 완료

→ 수량 입력 시 문자열로 입력하게 되면 에러 처리 정상적으로 하도록 수정

```
200         if (!(cin >> menu >> count))
201         {
202             cin.clear();
203             cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
204             cout << "\n잘못된 입력입니다. 다시 입력해주세요.\n";
205             << endl;
206             continue;
207         }
```

3 Brute Force Test

BF07

RUC02 음료 종류 및 수량 선택 : 구매하고 싶은 수량은 자리수 제한이 있는지

Test Report

높은 값으로 수량 입력 시, 무한 입력 요청 루프 반복 현상

Controller.cpp

코드 반영 완료

→BF03, BF05에서 함께 해결

3 Brute Force Test

BF08

RUC02 음료 종류 및 수량 선택 : 자판기에서 판매 가능한 전체 음료 아이템 목록을 출력한다.

Test Report

수량 가격 미출력

SRS, PFR에 기재되지 않은 사항이기에 적용 불필요

3 Brute Force Test

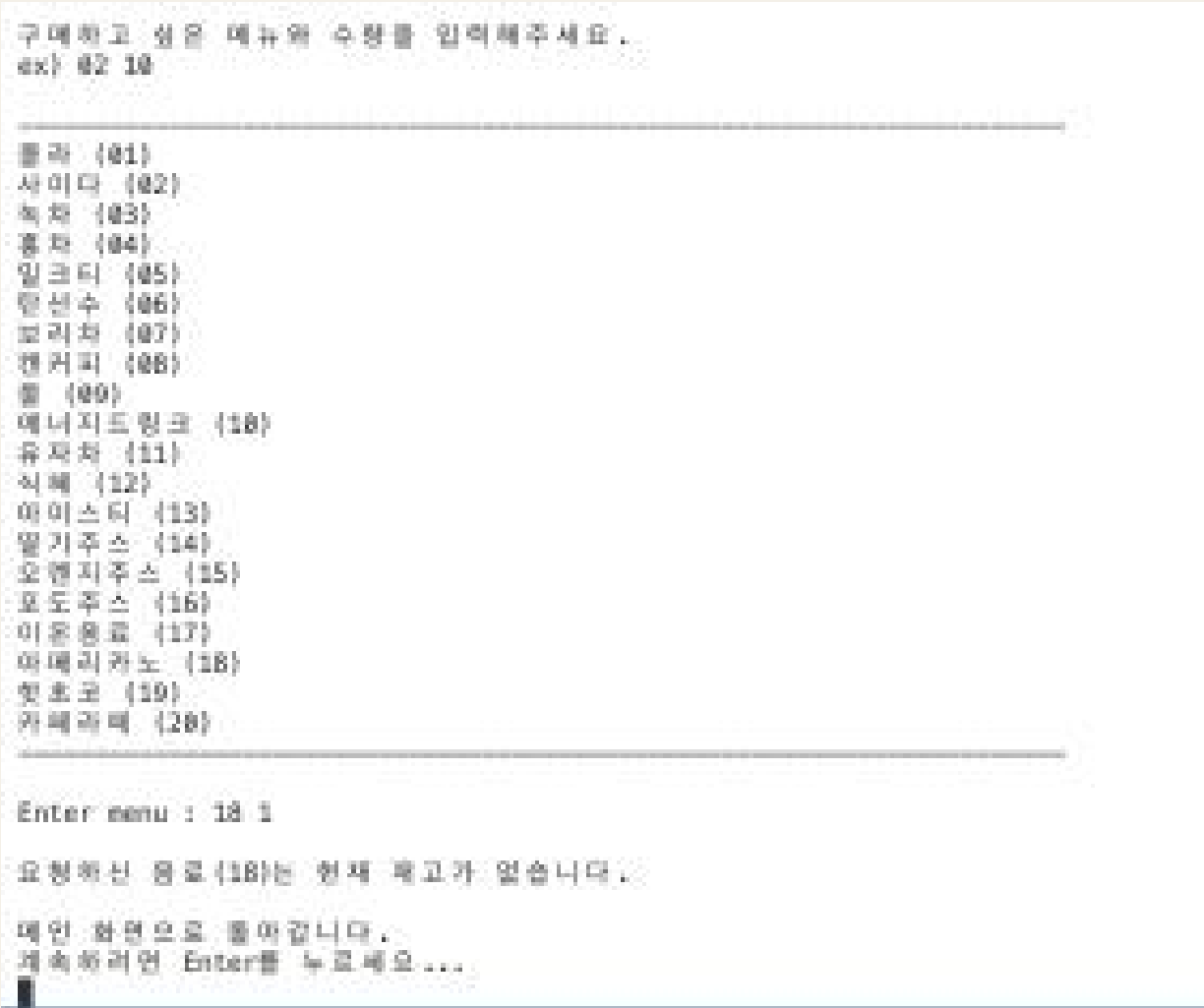
BF09

RUC03 카드 정보 입력과 확인 : 음료 가격이 알맞게 계산되는지 확인

Test Report

음료 03번(녹차)부터는 결과물 출력 없이 무한 입력 요청 루프 반복 현상

확인되지 않은 오류 사항



3 Brute Force Test

BF11

RUC03 카드 정보 입력과 확인 : 카드 번호에 공백, 특수문자가 포함된 경우

Test Report

유효한 카드번호 3422-abcd인데 대문자로 3422-abcD로 했을 때도 결제가 되는 issue

외부 인터페이스인 Card 클래스에서 취급하고 있는 카드 정보 형식은
(숫자4자리)-(영문 대/소문자 4자리)이기에 해당 test case는 정상 상태로 간주

3 Brute Force Test

BF15-BF28
RUC05 ~ RUC07

Test Report

다른 자판기와 통신이 connection failed라서 확인이 불가

dvm 프로세스 인스턴스를 여러 개 실행하였을 때에 대한
port 및 ip 설정에 대한 문서가 누락되어 검증팀과의 소통 실패
해당 부분 전달 완료 및 테스트 방법 수정 필요

빌드 및 소켓 통신 실행 방법

One-command Build & Test

To build the project and run all tests at once, use the following commands:

```
mkdir -p build && cd build
cmake ..
make
./dvm_test && ./sale_test && ./domain_test && ./otherdvm_test && ./controller_test && ./integration_test
```

Individual Components

1. Build only:

```
mkdir -p build && cd build
cmake ..
```


4 피드백 반영 후 Unit Testing 결과

```
● peter_mac ~/Desktop/advancedsw/build ↗ develop ./dvm_test -tail 5 | tail -n 3
[-----] Global test environment tear-down
[=====] 57 tests from 1 test suite ran. (3 ms total)
[ PASSED ] 57 tests.

● peter_mac ~/Desktop/advancedsw/build ↗ develop ./sale_test -tail 5 | tail -n 3
[-----] Global test environment tear-down
[=====] 40 tests from 1 test suite ran. (1 ms total)
[ PASSED ] 40 tests.

● peter_mac ~/Desktop/advancedsw/build ↗ develop ./domain_test -tail 5 | tail -n 3
[-----] Global test environment tear-down
[=====] 78 tests from 5 test suites ran. (1 ms total)
[ PASSED ] 78 tests.

● peter_mac ~/Desktop/advancedsw/build ↗ develop ./otherdvm_test -tail 5 | tail -n 3
[-----] Global test environment tear-down
[=====] 33 tests from 1 test suite ran. (0 ms total)
[ PASSED ] 33 tests.

● peter_mac ~/Desktop/advancedsw/build ↗ develop ./controller_test -tail 5 | tail -n 3
[-----] Global test environment tear-down
[=====] 36 tests from 1 test suite ran. (3 ms total)
[ PASSED ] 36 tests.
```

총 5개의 모듈별 단위 테스트
총 224개의 테스트 케이스 통과

4 dvm test (57개/All Passed)

1. UI

- 전체 음료 목록 출력 및 특정 음료 포함 여부 확인
- 재고 수량, 가격, 이름 포함 여부 확인
- 재고 초과 요청 시 처리 확인

2. 판매

- 재고 내 요청 시 정상 처리 (requestOrder)
- 재고 초과 시 runtime_error 예외 발생 검증
- 0개/음수/미존재 아이템 요청 처리 확인

3. 선결제

- 선결제 저장 후 인증 코드 사용 처리 (processPrepaidItem)
- 유효한 코드 → true / 잘못된 코드 → false / 중복 사용 → false 반환 확인
- 특수 문자 포함, 숫자 인증 코드 등 다양한 입력 테스트

4. 타 DVM 판매 저장 (saveSaleFromOther)

- 재고 감소 여부 확인
- 유효하지 않은 아이템 또는 재고 초과 시 예외 발생 검증
- 인증코드 중복 처리 가능 여부 확인

5. 타 DVM 통신 예외 처리

- 존재하지 않는 DVM ID 요청 시 DVMNotFoundException 발생 확인

6. 결제 흐름 시나리오

- 다양한 가격/수량 조합에 대한 결제 시도 (간접 테스트 포함)

7. 모의 시나리오 테스트

- 연속 구매 및 재고 소진 후 예외 처리
- 선결제와 직접 구매를 혼합하여 사용하는 복합 시나리오
- 여러 인증 코드의 등록 및 사용, 중복 사용 처리

4 sale test (40개/All Passed)

1. 판매 객체 생성

- 다양한 수량 (1개, 10개, 0개, 음수 등)에 대해 정상 생성 여부 확인
- 고가, 저가, 무료 아이템도 테스트

2. 선결제 생성

- 대상 DVM ID에 따라 고유한 인증코드 생성 여부 검증
- 다양한 DVM ID (0, 음수, 대형 숫자) 사용 가능 여부 확인
- 생성된 인증코드가 5자리 영숫자 포맷을 만족하는지 확인

3. 인증코드 기반 생성

- 다양한 포맷의 인증코드 (영문, 숫자, 특수문자, 공백 포함 등)에 대해 처리 가능 여부 확인
- 잘못된 인증코드로는 인증 실패, 올바른 경우에만 정상 수령 처리

4. 인증 처리

- 올바른 인증코드로만 1회 인증 가능
- 재사용, 오타자, 대소문자 차이, 공백 포함 등의 변형 → 모두 인증 실패
- 일반 판매 (Standalone)에서는 인증 불가 확인

5. 시나리오 테스트

- 복수 판매 생성 후 인증 → 각각 대응되는 코드로만 인증 가능
- 인증 실패 → 최종에 올바른 코드로 인증 성공 → 이후 재사용 실패
- 일반 판매 vs 선결제 판매의 인증 가능 여부 비교

6. 소멸자 예외 처리

- 선결제 포함/미포함 판매 객체의 소멸자(~Sale()) 실행 시 예외 발생 없음 확인

4 domain test (78개/All Passed)

1. card_test

- 카드 포맷 유효성 검증 (Card::isValid) 로직의 정확성 테스트
- 유효한 카드 형식: 1234-abcd, 대소문자 혼합 등 → true 반환
- 잘못된 형식: 자리 수 부족/초과, 숫자-문자 규칙 위반, 특수문자 포함, 공백 포함 등 → false 반환

2. certification_code_test

- 인증코드 생성 및 검증 기능 (CertificationCode)의 정상 작동 여부 확인
- 랜덤 코드 생성: 길이 5자 보장, 충돌 없음 확인
- 코드 검증: 올바른 코드만 인증 성공, 중복 사용 차단
- 대소문자 구분, 공백 포함, 특수문자 포함 등 예외 케이스 테스트

3. item_test

- 아이템 정보 저장, 가격 계산, 출력 등 기본 기능 검증
- 다양한 이름/코드/가격 조합으로 객체 생성
- 가격 계산 (수량 곱셈), 출력 문자열 확인
- 연산자 비교 (<)의 ID 기준 동작 확인

4. location_test

- 맨해튼 거리 계산 및 좌표값 처리 검증
- 기본 거리 계산, 양방향 동등성 확인
- 음수 좌표, 큰 좌표, INT_MAX/INT_MIN 처리 등
- 같은 좌표 여부 및 좌표 게터 확인

5. prepayment_test

- Prepayment 객체를 통한 인증코드 관리 로직 검증
- 고정/랜덤 코드 생성 시도 및 검증 흐름 테스트
- 잘못된 코드 다중 시도 후 올바른 코드 인증 확인
- 다양한 DVM ID 대응력 (음수, 0, 매우 큰 값)

4 other dvm test (33개/All Passed)

1. 위치

- 다양한 좌표(양수, 음수, 최대/최소값 포함)에 대한 Location 객체 반환 정확성 검증
- getLocation()의 동작이 설정된 위치와 일치하는지 확인

2. 거리 계산

- 자판기 간 맨해튼 거리 계산 정확성 테스트
- 동일 위치, 축 일치, 음수 좌표 등 다양한 케이스 적용

3. DVM 식별자 검증

- getDvmId()로 반환되는 DVM ID의 정확성 확인
- 음수, 0, INT_MAX/MIN 등 극단값도 포함하여 경계값 테스트 수행

4. 네트워크 응답 메시지 직렬화 테스트

- SocketMessage 포맷을 기반으로 한 문자열 메시지 생성 함수 검증
- 메시지 내 필드 포함 여부 및 형식 검증

5. Mock 기반 원격 자판기 응답 테스트

- findAvailableStocks() 및 askForPrepayment() 함수에 대한 가짜 응답 반환 동작 확인
- 요청 아이템 코드, 수량, 인증코드 등에 따라 분기되는 응답값 검증

6. 선결제 인증 코드 유효성

- 인증코드의 길이와 형식에 따라 성공/실패 여부가 올바르게 처리되는지 확인

7. 거리 동일 시 낮은 DVM ID 우선 로직

- 동일 거리일 때 ID가 더 낮은 자판기가 우선 선택되는 정책 간접 검증

8. 응답 시간 (성능)

- 네트워크 통신 응답이 1초 이내에 반환되는지 확인 (간접 테스트)

9. 주변 자판기 안내 기능

- 여러 자판기 중 거리 계산을 통해 가장 가까운 자판기를 올바르게 탐색하는지 확인

4 controller test (36개/All Passed)

1. 응답 파싱 기능

- flag, item_code, item_name, count, total_price 등의 key-value 형태 파싱 검증
- 잘못된 형식, 빈 입력, 중복 키, 공백 및 특수문자 포함 등의 엣지 케이스 포함

2. 재고 요청 처리

- 정상 요청 처리 시 msg_type:resp_stock, item_code, item_num, src_id, coor_x/y 등이 포함되는지 확인
- 재고 부족 (item_code:003) 상황에 대한 처리
- 요청 포맷이 잘못된 경우와 누락된 필드에 대한 예외 대응

3. 선결제 요청 처리

- 정상 응답 시 msg_type:resp_prepay, item_code, item_num, availability:T/F 포함 확인
- 재고 없음 처리 및 잘못된 형식/필드 누락 상황 처리

4. 경계값 테스트

- 매우 긴 문자열 (1000자), 음수/0/큰 수(999999)의 수량, INT_MAX(2147483647) 등 경계 조건 입력에 대한 응답 안정성 검증

5. 한국어 및 특수문자 대응

- 응답 파싱 시 한글 및 특수문자(!@#\$%^&*) 포함 항목 처리 여부 확인

Thank you!