OOPT 2050

객체지향개발방법론 T8
202212354 박수진
202212372 임세희
202011285 나형진
201914182 이영규

목차

개발 환경(CI/CD, UT)

CI/CD

- 서버: AWS EC2
- CI/CD 도구: Jenkins
- 빌드 도구: Gradle

UT

- Junit
- 코드 커밋 시 자동으로 unit test
- 빌드 과정에서 테스트 통과 여부 확인

개발 환경(CI/CD, UT)

Workflow

- 로컬 환경에서 기능 구현 및 테스트
- 개별 브랜치에서 기능 구현 완료 후 main 브랜치에 코드 push
- Github Webhook이 Jenkins 빌드 트리거
- Jenkins가 Gradle을 이용해 자동 빌드 + JUnit 테스트
- 테스트 통과 시 EC2 서버에 자동 배포

UT 시나리오 (DisplayManager)

Test No.	Test Name	method	Purpose	Unit Test Scenario	Pass
1	Test PrintMainScene method	printMainScene	mainScene에 해당하는 JPanel이 실행되는지 test	JPanel의 component가 일치하는 지 check	P
2	Test clickInputCard method	clickInputCard	mainScene에 해당하는 JPanel이 실행되는지 test	JPanel의 component가 일치하는 지 check	P
3	Test clickInputCode method	clickInputCode	mainScene에 해당하는 JPanel이 실행되는지 test	JPanel의 component가 일치하는 지 check	P
4	Test clickInputAdmin method	clickInputAdmin	mainScene에 해당하는 JPanel이 실행되는지 test	JPanel의 component가 일치하는 지 check	P
5	Test printMsgAndMainScene method	printMsgAndMainScene	mainScene에 해당하는 JPanel이 실행되는지 test	JPanel의 component가 일치하는 지 check	P
6	Test showErrorMessage method	showErrorMessage	label의 setVisible 기능 test	error 상황에서 해당 label이 visible이 되는지 확인	P
7	Test showItems method	showItems	mainScene에 해당하는 JPanel이 실행되는지 test	JPanel의 component가 일치하는 지 check	P
8	test testFailPayment method	testFailPayment	Fail Label이 올바르게 뜨는지 test	해당 기능중 restoreStock이 작동하는지 panel이 null이 아닌지, 해당 label의 글씨가 payment fail인지 check	

UT 시나리오 (StockManager)

Test No.	Test Name	method	Purpose	Unit Test Scenario	Pass
9	Test prePaymentUI_noDVM method	prePaymentUI	noDVM 상황에서 prepaymentUI의 올바른 출력 test	nearest DVM이 없는 상황에서 display되는지, panel이 존재하는지, 해당 label에 no DVM이라고 텍스트가 적히는지 check	P
10	Test prePaymentUI_yesDVM method	prePaymentUI	yesDVM 상황에서 prepaymentUI의 올바른 출력 test	DVM위치가 리턴된 상황에서 display되는지, panel이 존재하는지, 해당 label에 Would you like to make a payment for the DVM located at a distance of (x : 1, y: 1)이 출력되는지 check	P
11	Test init method	init	init함수를 통해 변수가 초기화되었는지 test	인자로 들어오는 stockManager가 null인지 check	P
12	Test checkStock method	checkStock	itemcode 를 넣어 올바른 리턴인지 test	itmecode를 넣었을때 0인지 check	P
13	Test restoreStock method	restoreStock	임의의 인자를 넣었을때 올바른 itemNum이 나오는지 test	code 를 1, Num 를 5로 넣고 restoreStock를 확인해서 예상한 num이 나오는지 check	P
14	Test rechargeStock method	rechargeStock	재고가 올바르게 다 충전되는지 test	재고를 충전한 후 정해진 부분의 itemNum이 99개인지 check	P
15	Test selectStock method	selectStock	재고를 요청한 만큼 빼두는지 test	재고를 충전한 후 정한 만큼 가지고 가는지 갯수 check	P
16	test printStockList method	printStockList	print 예외가 발생하지 않는지 test	print 콘솔에 예외가 발생하지 않으면 정상 출력으로 판단하여 그러한지 check	

UT 시나리오 (PrepaymentManager)

Test No.	Test Name	method	Purpose	Unit Test Scenario	Pass
17	Test askStockRequest method	askStockRequest	재고 요청 메시지가 전송되는지 test	item code와 item num를 입력했을 때 1초 안에 메시지가 보내지는지 check	P
18	Test putMsg method	putMsg	재고 요청에 대한 응답 메시지를 받았을 때, DVM의 생성 여부 test	protocol에 맞고, item num개수가 요청한 item num개수 이상인 message를 인자값으로 넘겼을 때, dvmList size가 1값과 같은지, id가 msg에서 보낸대로 제대로 들어갔는지 check	P
19	Test sendAskPrepaymentMsg method	senAskPrepaymentMsg	가까운 DVM에게 선결제 요청 메시지가 전송되는지 test	좌표 1, 1인 dvm을 dvmList에 넣어놓고 선결제 요청이 갔는지 check	P
20	Test respPrepayment method	respPrepayment	선결제가 가능하다고 돌아왔을 때, 결제가 잘 실행되는지 test	availability가 T인 메시지를 받았을 때, PaymentManager 클래스의 startPayment가 호출되고, 그 값이 true일 때 printMsgAndMainScene 함수가 1초안에 호출되는지 check	P
21	Test otherVMPrepaymentRequest method	otehrVMPrepaymentRequest	재고 요청이 들어왔을 때 재고를 잘 검사하는지 test	StockManager의 checkStock이 10을 return할 때, otherVMPrepaymentRequest의 return값도 10과 같은지 check	P
22	Test otherVMPrepayemntResponse method	otherVMPrepaymentResponse	선결제 요청이 들어왔을 때 재고를 차감하고 인증코드를 잘 저장하는지 test	selectStock이 true를 return했을 때 otherVMPrepaymentResponse이 호출됐을 때 result값이 true인지, addCode가 제대로 불렸는지 check	P
23	Test nearestDVMCoordinate method	nearestDVMcoordinate	dvmList 정렬해서 가짱 가까운 DVM의 좌표를 반환하는지 test	DVMList에 좌표값이 1, 1과 2, 2인 dvm 두개를 넣어놓고 2, 2를 return하는지 check	P

UT 시나리오 (Network)

Test No.	Test Name	method	Purpose	Unit Test Scenario	Pass
24	Test run method	run	서버가 제대로 열리는지 test	Network의 start()를 호출하고 clientSocket을 생성해서 연결되는지 check	P
25	Test clientStart method with clientID 0	clinetStart	broadcast로 msg가 보내지는지 test	clientStart에 인자를 0으로 호출해서 메시지가 전송되는지 test	P
26	Test clientStart method with non-zero clientID	clentStart	특정 dvm에게만 msg가 보내지는지 test	clientStart에 인자를 5로 넘겨서 dvmList[4]에게 전송되는지 test	P
27	Test clientHandler run method with req_stock message	run	다른 dvm에서 req_stock message가 왔을때 수신하는지 test	protocol에 맞는 message를 생성해서 clientHandler에게 넘겼을 때, message가 잘 파싱되면 불리는 otherPrepaymentRequest가 불리는지 test	P
28	Test clientHandler run method with req_prepay message	run	다른 dvm에서 req_prepay message가 왔을때 수신하는지 test	protocol에 맞는 message를 생성해서 clientHandler에게 넘겼을 때, message가 잘 파싱되면 불리는 otherPrepaymentResponse가 불리는지 test	P
29	Test ServerResponseThread run method with resp_stock	run	다른 dvm에게 req_stock message을 보낸 후 응답을 받는지 test	protocol에 맞는 message를 생성해서 server에게 넘겼을 때, message가 잘 파싱되면 불리는 putMsg가 불리는지 test	P
30	Test ServerResponseThread run method with resp_prepay	run	다른 dvm에게 req_prepay message을 보낸 후 응답을 받는지 test	protocol에 맞는 message를 생성해서 server에게 넘겼을 때, message가 잘 파싱되면 불리는 respPrepayment가 불리는지 test	P

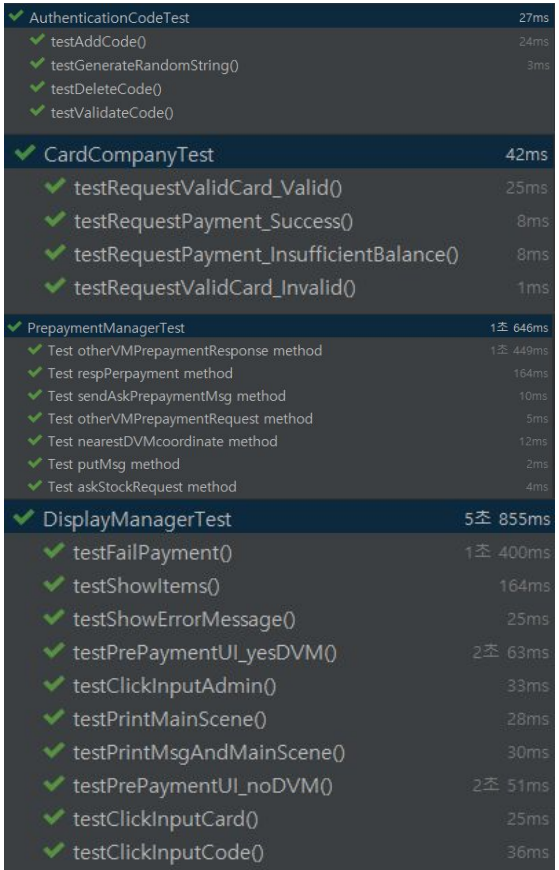
UT 시나리오 (AdminManager & AuthenticationCode & CardCompany)

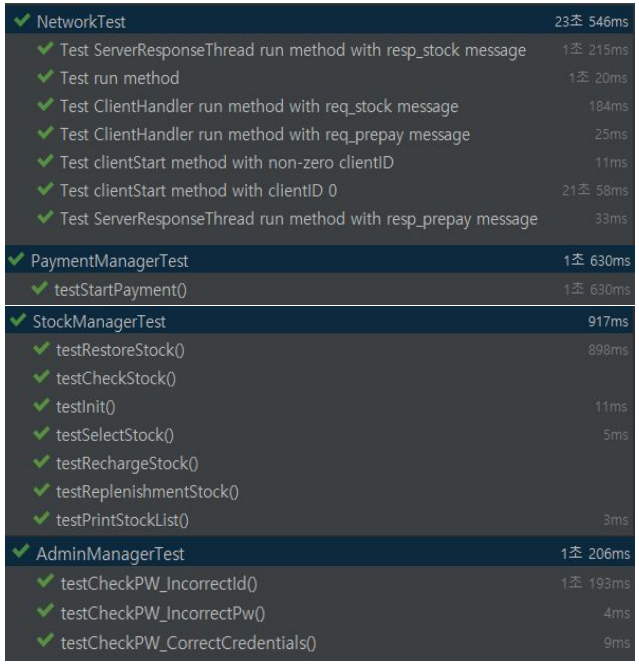
Test No.	Test Name	method	Purpose	Unit Test Scenario	Pass
31	Test testValidateCodemethod	validateCode	validate가 정상판별을 하는지 test	임의 코드를 add 하고 validate한지 check	P
32	Test checkPW_incorrectPW method	addCode	code가 add 하고 존재하는지 test	add code를 하고 validate인지 check	P
33	Test checkID_incorrectID method	DeleteCode	code가 delete되고 존재하는지 test	code를 delete 시키고 validate인지 check	P
34	Test generageRandomString method	generageRandomString	String을 정해진 형식으로 만들어내는지 test	정해진 양식에 맞쳐서 만들어져 양식과 형태가 맞는지 check	P
35	Test requestValidCard method	requestValidCard	valid한 카드가 들어왔을때 정상이라 판단하는지 test	올바른 카드를 받아서 1을 반환하는 지 check	P
36	Test requestValidCard method	requestValidCard	invalid한 카드가 들어왔을 때 비정상이라 판단하는지 test	올바르지 않은 카드를 받아서 -1을 반환하는지 check	P
37	Test requestPayment_sucess_method	requestPayment	올바른 카드를 받아 payment를 진행하는 지 test	올바른 카드번호와 price를 입력받았을때 payment가 정상 작동하는 지 check	P
38	Test requestPayment_insufficientBalance_method	requestPayment	올바른 카드를 받아 payment를 진행하는 지 test	올바른 카드번호와 비싼 price를 입력받았을때 payment의 부족한것이 정상작동하는 지 check	P

UT 시나리오 (paymentManager)

Test No.	Test Name	method	Purpose	Unit Test Scenario	Pass
39	Test startPayment method	startPayment	payment가 정상진행되는지 test	적당한 itemcode와 itemNum을 입력받아서 totalPrice을 계산하여 구매를 한다. 이때 카드번호도 올바르게 입력받아서 구매를 진행하며 item의 price와 결제가 정상인지 check	P

UT 결과


AuthenticationCodeTest 27ms
testAddCode() 24ms
testGenerateRandomString() 3ms
testDeleteCode()
testValidateCode()
CardCompanyTest 42ms
testRequestValidCard_Valid() 25ms
testRequestPayment_Success() 8ms
testRequestPayment_InsufficientBalance() 8ms
testRequestValidCard_Invalid() 1ms
PrepaymentManagerTest 1초 646ms
Test otherVMPrepaymentResponse method 1초 449ms
Test respPerpayment method 164ms
Test sendAskPrepaymentMsg method 10ms
Test otherVMPrepaymentRequest method 5ms
Test nearestDVMcoordinate method 12ms
Test putMsg method 2ms
Test askStockRequest method 4ms
DisplayManagerTest 5초 855ms
testFailPayment() 1초 400ms
testShowItems() 164ms
testShowErrorMessage() 25ms
testPrePaymentUI_yesDVM() 2초 63ms
testClickInputAdmin() 33ms
testPrintMainScene() 28ms
testPrintMsgAndMainScene() 30ms
testPrePaymentUI_noDVM() 2초 51ms
testClickInputCard() 25ms
testClickInputCode() 36ms


NetworkTest 23초 546ms
Test ServerResponseThread run method with resp_stock message 1초 215ms
Test run method 1초 20ms
Test ClientHandler run method with req_stock message 184ms
Test ClientHandler run method with req_prepay message 25ms
Test clientStart method with non-zero clientID 11ms
Test clientStart method with clientID 0 21초 58ms
Test ServerResponseThread run method with resp_prepay message 33ms
PaymentManagerTest 1초 630ms
testStartPayment() 1초 630ms
StockManagerTest 917ms
testRestoreStock() 898ms
testCheckStock()
testInit() 11ms
testSelectStock() 5ms
testRechargeStock()
testReplenishmentStock()
testPrintStockList() 3ms
AdminManagerTest 1초 206ms
testCheckPW_IncorrectId() 1초 193ms
testCheckPW_IncorrectPw() 4ms
testCheckPW_CorrectCredentials() 9ms

System Test 시나리오 및 결과

Test No.	System Function	Use Case	Test 항목	description	Pass
1	2.1	인증코드로 음료 수령	초기화면 출력 시험	초기화면 동작 test	P
2	2.1	인증코드로 음료 수령	인증코드 입력 버튼 시험	인증코드를 입력 후 버튼 동작 test	P
3	2.1	인증코드로 음료 수령	인증코드 유효성 결과값 검사	다양한 코드값을 입력하여 원하는 값 출력여부 Test	P
4	3.5	인증코드로 음료 수령	상품 제공 시험	구매한 음료가 출력되는지 test	P
5	1.1	카드 투입	초기화면 출력 시험	초기화면 동작 test	P
6	1.1	카드 투입	카드 번호 입력 시험	카드 번호 입력 후 버튼 동작 test	P
7	1.2	카드 유효성 확인	카드 유효성 검사	카드 정보를 입력하여 카드 유효성 검사 test	P

System Test 시나리오 및 결과

Test No.	System Function	Use Case	Test 항목	description	Pass
8	5.1	카드 결제로 음료 수령	음료 선택정보 검사	음료 선택정보가 정해진 범위내로 들어오고 정확하게 저장되는 지 Test	P
9	3.5	카드 결제로 음료 수령	음료 제공 검사	음료가 사용자가 선택했던 대로 제대로 제공되는지 Test	P
10	3.1	현 자판기 재고 확인	요청된 item의 정확한 수량 검사	정확하게 요청된 item의 수량을 확인하는 지 test	P
11	3.1	현 자판기 재고 확인	요청된 만큼 수량 확보 검사	요청된 수량만큼 재고에서 빼두는지 Test	P
12	1.3	카드 결제	잔액 확인 및 카드사 결제 검사	카드사에서 카드 잔액을 확인하여 정확히 차감되는지 Test	P
13	1.3	카드 결제	재고 복구 검사	카드잔액이 없을 경우 재고가 정확히 복구되는지 Test	P

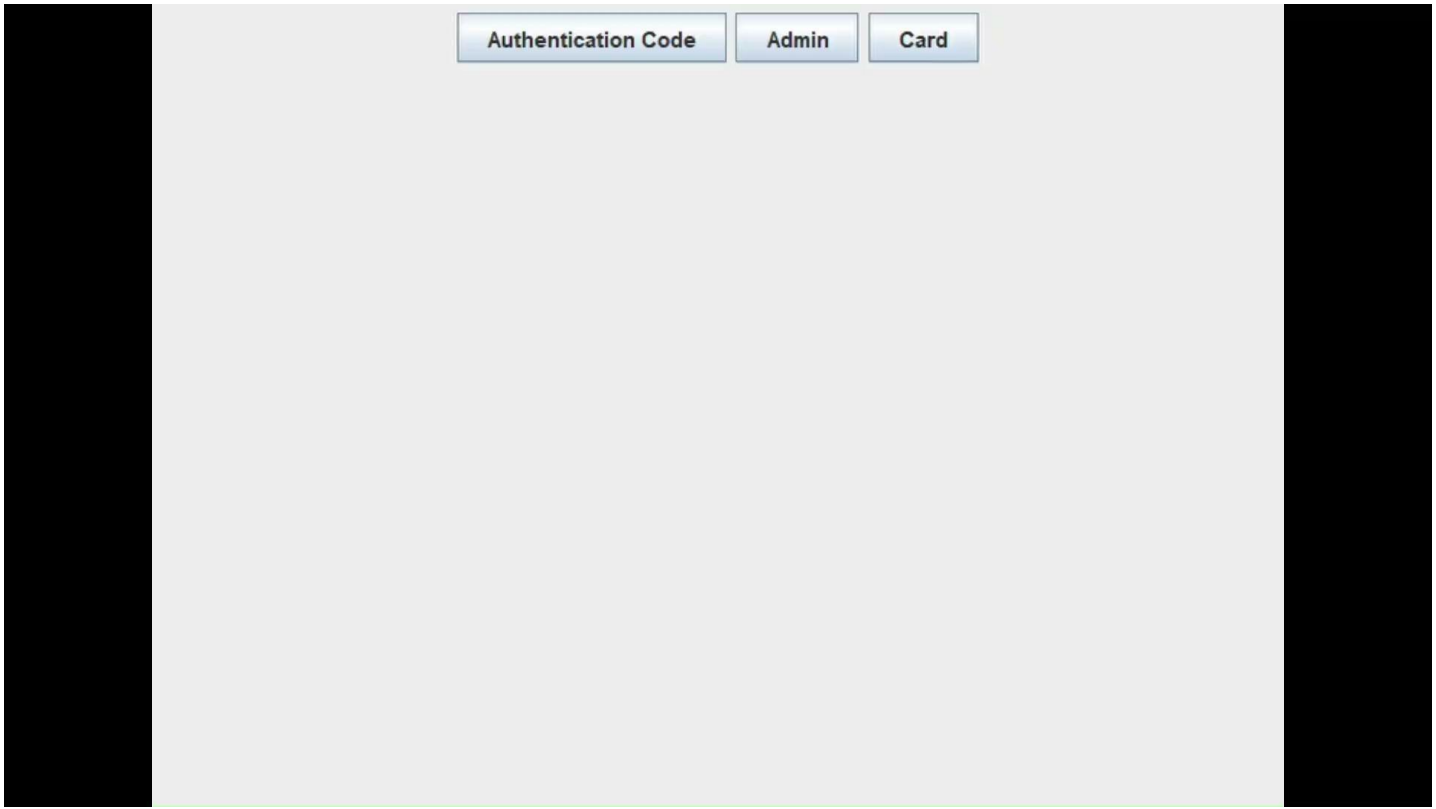
System Test 시나리오 및 결과

Test No.	System Function	Use Case	Test 항목	description	Pass
14	3.2	타 자판기 재고 확인	재고 확인 요청 발신 시험	다른 자판기들에게 우리가 보낸 재고 확인 요청 메세지 발신 여부 test	P
15	3.2	타 자판기 재고 확인	재고 확인 응답 수신 시험	다른 자판기에서 보낸 재고 확인 응답 메시지 수신 여부 test	P
16	6.1	타 자판기 재고 확인	선결제 여부 질의, 위치 출력 시험	사용자에게 선결제 여부를 질의하고 가장 자까운 자판기의 위치를 출력하는지 test	P
17	4.1	선결제	선결제 요청 메시지 발신 시험	다른 자판기들에게 우리가 보낸 선결제 요청 메시지 발신 여부 test	P
18	4.1	선결제	선결제 가능 여부 응답 메시지 수신 시험	다른 자판기들에서 보낸 선결제 가능 여부 응답 메시지 수신 여부 test	P
19	4.1	선결제	인증코드 발급 시험	주어진 형식(알파벳 대소문자, 숫자로 구성된 10자리)의 인증코드가 정상적으로 생성되는지 test	P
20	4.1	선결제	가장 가까운 자판기 계산 시험	가장 가까운 자판기 확인 test	P
21	6.1	선결제	거리 계산 시험	자판기의 좌표를 통한 거리 계산 test	P
22	2.2	선결제	입력코드 발급 시험	해당 자판기에 입력할 인증코드가 올바르게 발급됐는지 test	P

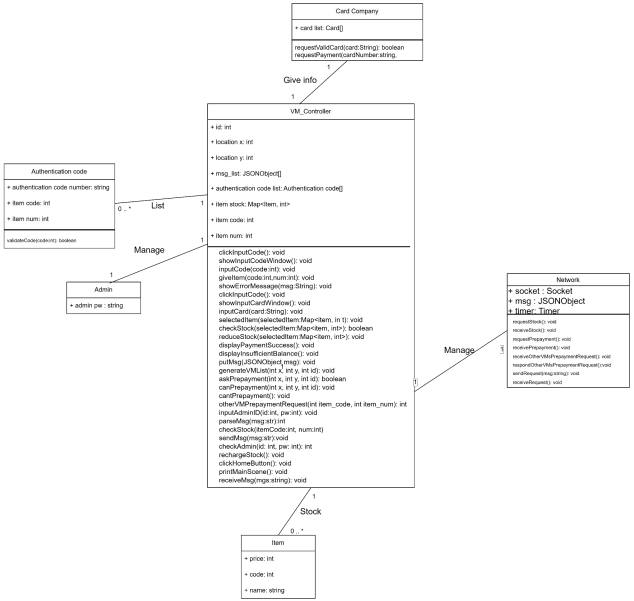
System Test 시나리오 및 결과

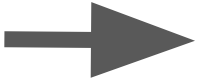
Test No.	System Function	Use Case	Test 항목	description	Pass
23	4.2	선결제 요청	선결제 요청 수신	다른 VM에서 보낸 선결제 요청 메시지 수신 여부 test	P
24	4.2	선결제 요청	선결제 가능 여부 응답 발신	다른 VM에게 우리가 보낸 선결제 가능 여부 응답 메시지 발신 여부 test	P
25	4.2	선결제 요청	상품 확보 시험	입력 수량을 재고에서 차감 test	P
26	3.3	재고 정보 요청	재고 확인 요청 수신	다른 VM에서 보낸 재고 확인 요청 메시지 수신 여부 test	P
27	3.3	재고 정보 요청	재고 확인 응답 발신	다른 VM에게 우리가 보낸 재고 확인 응답 메시지 발신 여부 test	P
28	3.4	재고 보충	관리자 아이디 시험	관리자 아이디를 입력하여 아이디가 유효한지 검사 test	P
29	3.4	재고 보충	상품 재고 보충 시험	초기 화면에서 관리자 아이디를 입력할 시, 현 자판기에서 관리하는 음료 품목들 모두 99개 충전 test	P
30	5.2	초기 화면으로 돌아가기	초기화면 출력 시험	초기화면 동작 test	P

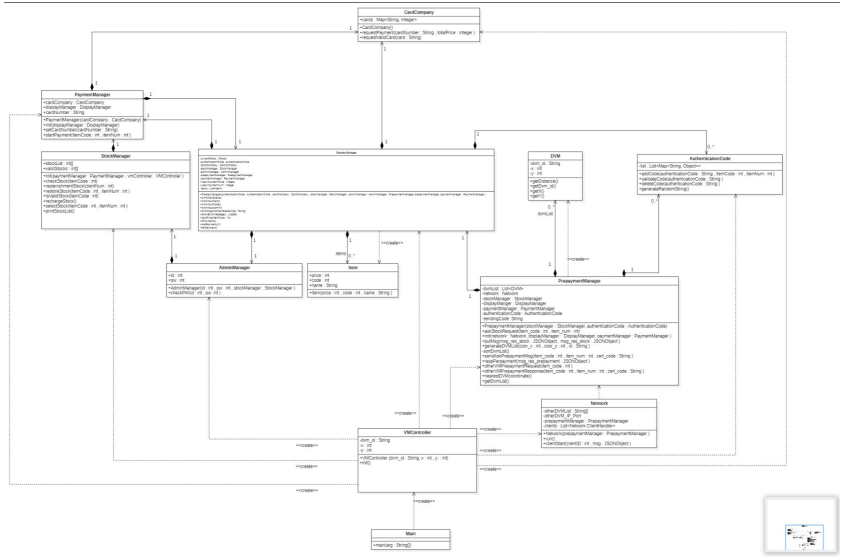
시스템동작 Demo 화면 또는 영상



OOD 2040에서 변경/수정된 부분







구현시 예상보다 어려웠던 점

1. get, set 함수를 아예 안 쓰고 구현은 하지 못해서 아쉽다.
2. 통신이 날라올때의 그 찰나의 타이밍 차이로 코드의 실행이 달라져 시간을 좀 잡아 먹었다. thread에 타이머를 걸어서 해결을 하였다.
3. swing 코드를 처음 다뤄보아 UI 구현에 시간이 쓰였다.
4. class diagram 으로 짠 내용만으로는 모든 구현이 되지않아 함수가 추가되었고 object 간의 상호작용을 조금 더 신경써서 구현을 해야 했다.
5. ec2 가 너무 느리고 결론적으로 통신이 되지 않았다. json parser
6. 인코딩 문제

구현시 예상보다 쉬웠던 점

1. 서버 클라이언트 통신 자체가 어렵진 않았다.
2. 클래스와 그 클래스 내의 메소드들의 내용을 미리 정해두고 구현을 하니까, 분업이 잘 됐다.
3. 객체마다 역할이 분할되어 정의되어 있어 메소드 구현이 상대적으로 쉬웠다.

객체지향방법론의 장단점

장점 :

- 전체 프로그램의 방향성 유지에 용이하고 상속, 다형성과 같은 특성을 이용한 코드의 재사용성이 높다.
- 유지보수가 쉽고, 복잡한 코드를 효과적으로 관리할 수 있다.

단점 :

- 객체에 대한 이해가 필요하여 설계에 많은 시간이 소요된다.
- 절차형 프로그래밍에 비해 코드의 길이가 길어져 프로그램 실행 속도가 느려진다.
- 변경점이 있을 경우 이전에 작성한 모든 문서를 수정해야해서 힘들다
- 초기에 충분한 논의를 하지 않으면 갈수록 수정량이 배로 증가한다.

소감

- 개발 프로세스와 CI/CD 환경을 배울수있어서 좋은 경험이었습니다.
- 초기 설계 단계의 중요성을 깨달았고, 충분한 분석 후 객체간의 관계와 역할을 정의해야 했음을 깨달았습니다.

감사합니다.