

ADD

Team3 : 유혜리 이동훈 송승현

1st Iteration

1. 입력물 검토.....	3p
2. 요인을 선택하여 Iteration 목표 수립.....	5p
3. 정제할 시스템 요소 선택.....	7p
4. 선택된 요인을 충족시키는 설계 개념 선택.....	9p
5. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의.....	11p
6. 뷰 스케치 및 설계 결정 기록.....	13p
7. 현재 설계 분석 수행 및 Iteration 목표와 설계 목표 달성 검토.....	16p
2 nd Iteration.....	18p

1. 입력물 검토

1. 입력물 검토

카테고리	세부 사항																		
설계 목적	기존의 자판기 보다 효율적인 분산형 자판기 관리 시스템이다. 목적은 분산형 자판기 시스템의 구축을 지원하는 충분히 상세한 설계를 산출해내는 것이다.																		
최우선 기능 요구	UC-1: 상품선택 버튼을 누른다. UC-2: 코드입력 버튼을 누른다. UC-3: 코드를 환불한다. UC-4: 관리자 버튼을 누른다. UC-5: 상품을 선택한다. UC-6: 자판기에 재고가 있거나, 다른 재고가 있는 자판기 조회를 마친 경우 QR결제를 요청한다. UC-7: 재고가 없을 경우, 재고가 있는 다른 자판기를 조회한다. UC-8: 자판기의 재고정보를 네트워크 상에 동기화한다. UC-9: 자판기의 네트워크 상태를 점검한다. UC-10: 결제가 이루어진 자판기에서 선결제한 상품을 제공할 자판기로 코드를 전송한다. UC-11: 관리자 상태에서 자판기의 재고를 변경한다.																		
품질 속성 시나리오	<table border="1" data-bbox="682 796 1421 999"> <thead> <tr> <th>시나리오 ID</th> <th>고객 중요성</th> <th>아키텍트 구현 난이도</th> </tr> </thead> <tbody> <tr> <td>QA-1</td> <td>높음</td> <td>높음</td> </tr> <tr> <td>QA-2</td> <td>높음</td> <td>높음</td> </tr> <tr> <td>QA-3</td> <td>중간</td> <td>낮음</td> </tr> <tr> <td>QA-4</td> <td>높음</td> <td>중간</td> </tr> <tr> <td>QA-5</td> <td>높음</td> <td>높음</td> </tr> </tbody> </table>	시나리오 ID	고객 중요성	아키텍트 구현 난이도	QA-1	높음	높음	QA-2	높음	높음	QA-3	중간	낮음	QA-4	높음	중간	QA-5	높음	높음
시나리오 ID	고객 중요성	아키텍트 구현 난이도																	
QA-1	높음	높음																	
QA-2	높음	높음																	
QA-3	중간	낮음																	
QA-4	높음	중간																	
QA-5	높음	높음																	
제약 사항	CON-1: 최소 1000명의 동시 사용자를 지원해야 한다. CON-2: 보안 관련한 문제가 발생할 경우 이와 관련한 정보가 관리자에게 빠르게 전달되어야 한다. CON-3: 네트워크 성능은 5분 간격으로 점검할 수 있어야 한다. CON-4: 총 개발인원이 3명을 넘지 않아야 한다.																		
아키텍처 관심사	CRN-1: 최소 사양의 하드웨어에서 작동해야 한다. CRN-2: 자판기의 네트워크 연결을 관리하는 모듈이 필요하다 CRN-3: 결제 완료 시, 실시간으로 재고상태가 동기화가 되어야 한다.																		

2.요인을 선택하여 Iteration 목표 수립

2.요인을 선택하여 Iteration 목표 수립

시스템 구조에 영향을 미칠 수 있는 요인들

CRN-1 :

최소 사양의 하드웨어에서도 원활히 작동해야 한다.

이터레이션 1의 목표:

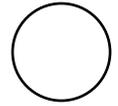
전체 시스템 구조 수립

ID	Description
CON-1	최소 1000명의 동시 사용자를 지원해야 한다.
CON-2	보안 관련한 문제가 발생할 경우 이와 관련한 정보가 관리자에게 빠르게 전달되어야 한다.
CON-3	네트워크 성능은 5분 간격으로 점검할 수 있어야 한다.
CON-4	총 개발인원이 3명을 넘지 않아야 한다.
CRN-2	자판기의 네트워크 연결을 관리하는 모듈이 필요하다
CRN-3	결제 완료 시, 실시간으로 재고상태가 동기화가 되어야 한다.

3. 정제할 시스템 요소 선택

3. 정제할 시스템 요소 선택

범례



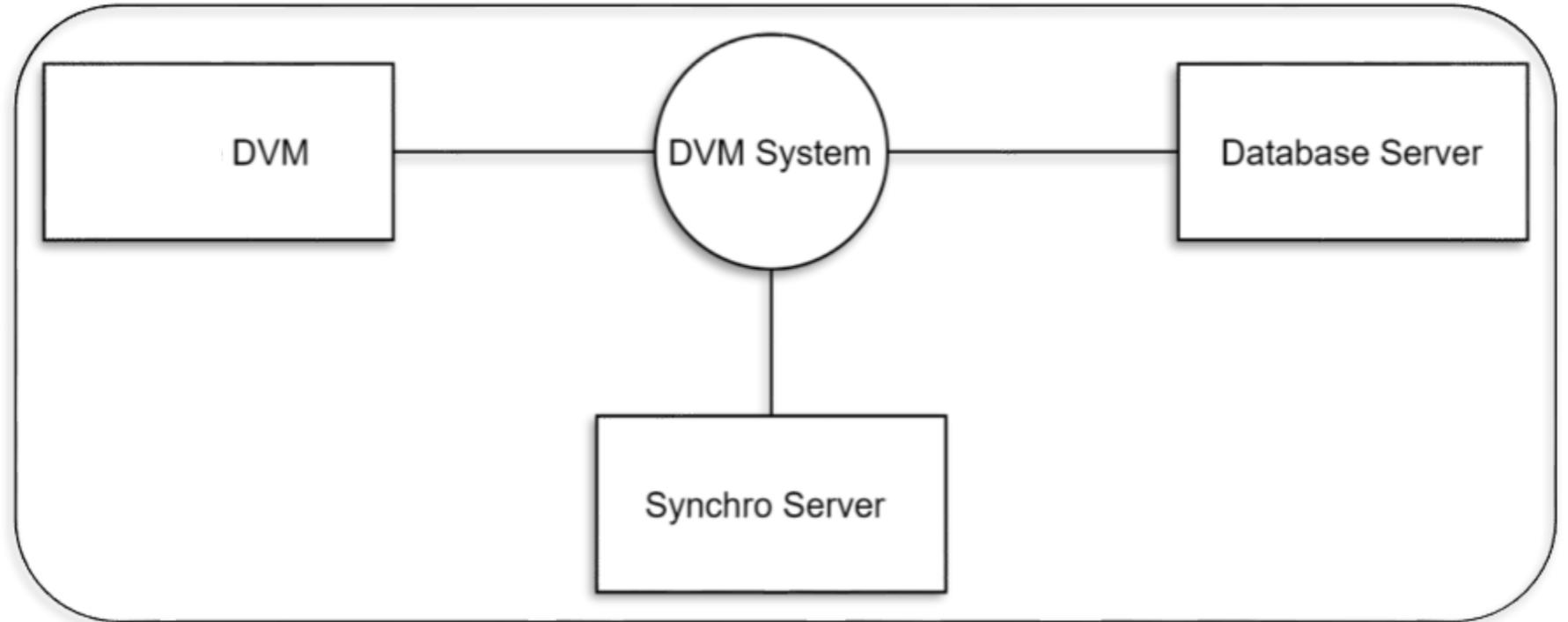
개발중인 시스템



외부 시스템



데이터 흐름



4. 선택된 요인을 충족시키는 설계 개념 선택

4. 선택된 요인을 충족시키는 설계 개념 선택

설계 결정과 위치	근거
<p>논리적구조 클라이언트 부분 : 리치 클라이언트 애플리케이션을 선택한다.</p>	<p>리치 클라이언트 애플리케이션 참조 아키텍처는 사용자 머신에 설치하는 애플리케이션으로, 이 시스템에서 필요한 사용자 인터페이스는 모두 제공할 수 있다. 가벼운 클라이언트 애플리케이션을 사용(CRN-1)하므로 주로 프레젠테이션 로직으로 구성되고, 사용자 데이터를 획득하여 서버가 처리할 수 있도록 넘기는 형태로 작동한다. 애플리케이션은 간헐적으로 네트워크 연결성을 지원한다.</p> <p>폐기된 대안</p> <ol style="list-style-type: none"> 1. 리치 인터넷 애플리케이션 : 이 참조 아키텍처는 웹 브라우저에서 실행되는 풍부한 사용자 인터페이스를 갖는 애플리케이션을 개발하는 것을 지향한다. 풍부한 사용자 인터페이스가 필요하지 않고 사용자가 배치된 하드웨어에 접근하는 방식을 가진 이 시스템에 특성 상 사용자 경험을 제공하기에 적절치 않아 폐기되었다. 2. 모바일 애플리케이션 : 이 참조 아키텍처는 핸드헬드 디바이스에 배포되는 애플리케이션을 개발하는 것을 지향한다. 핸드 헬드 디바이스의 사용이 사전에 고려되지 않았으므로 폐기하였다. 3. 웹 애플리케이션 : 이 참조 아키텍처는 웹 브라우저에서 접근할 수 있는 애플리케이션을 개발하는 것을 지향한다. 배포와 업데이트가 쉬워지만 현재 시스템의 구조상, 사용자가 배치된 하드웨어에 접근하는 방식이기 때문에 사용자 경험을 제공하기에 알맞지 않아 폐기되었다.
<p>논리적인구조 서버 부분 : 서비스 애플리케이션 참조 아키텍처를 사용한다.</p>	<p>서비스 애플리케이션은 자판기에서 유저가 호출하여 동작하여 자판기 내부에 서버가 불필요하므로 외부에 서버를 별도로 구성해야한다. (QA-4)</p>
<p>물리적 구조 3티어 배포 패턴을 사용한다.</p>	<p>데이터베이스를 호스팅하는 티어와 분리된 티어에 배포되므로 3티어 배포가 적당하다. 폐기된 대안으로, 2티어 대안은 싱크로서버가 동기화를 지원해야 하므로 직접적으로 데이터베이스를 호스팅하는 티어와 분리되어야 하여 폐기되었다. 3티어 이상의 경우 다른 서버가 이 시스템에 추가적으로 필요하지 않기 때문에 폐기되었다.</p>
<p>swing 자바 프레임워크와 다른 자바 기술을 사용하여 클라이언트 애플리케이션의 사용자 인터페이스를 구축한다.</p>	<p>자바 리치 클라이언트 구축을 위한 표준 프레임워크는 이식성을 보장하며, 개발자들이 익숙하여 인원이 많이 필요하지 않는(CON-4)이다. 또한 Java는 Type Safe한 언어이므로 QA-1에서 비정상적 입력을 해결 할 수 있다. 또한 기존의 개발자들이 익숙한 Java Framework 이므로 기존 개발인원으로 개발을 진행 할 수 있다.</p>
<p>QR코드 결제 모듈을 사용하여 결제 모듈 부분을 해결한다.</p>	<p>이미 개발된 결제 모듈을 사용하여 시스템 적으로도 안정적이고, 추가적인 개발비용 및 인력 낭비를 막을 수 있으므로 QR코드 결제 모듈을 DVM에서 사용한다.</p>

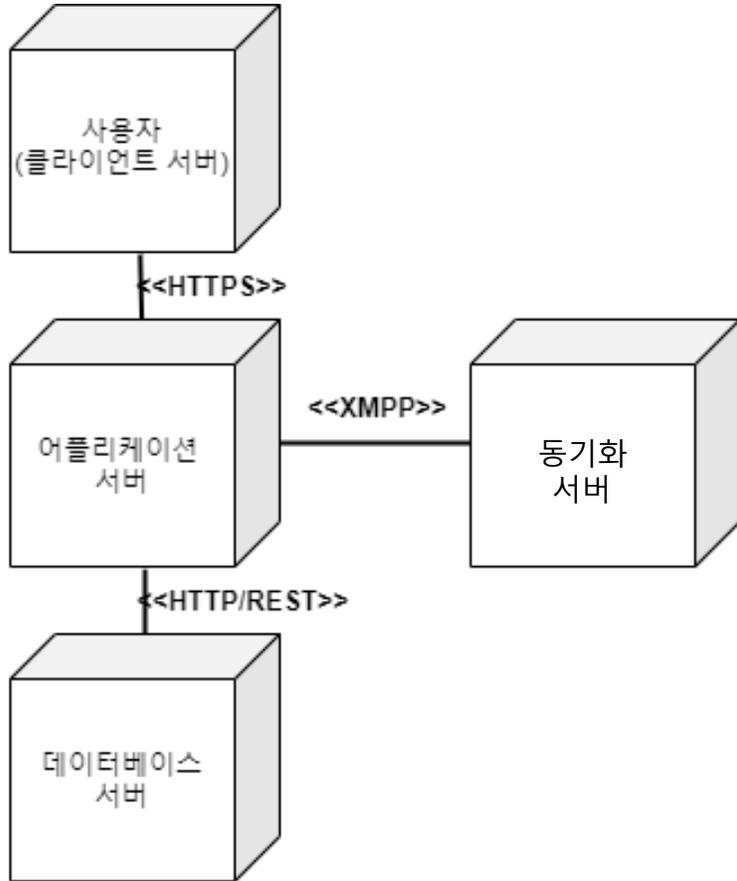
5. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의

5. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의

설계 결정과 위치	근거
리치 클라이언트 애플리케이션에서 로컬 데이터 소스를 제거함	일반적으로 네트워크 연결이 신뢰할 수 있으므로, 데이터를 로컬에서 저장할 필요가 없다고 판단된다.
서비스 애플리케이션 참조 아키텍처의 데이터 레이어에 싱크로 서버가 접근하는 것을 전담하는 모듈을 생성함	참조 아키텍처의 서비스 에이전트 컴포넌트가 싱크로 서버의 접근을 추상화하기 위해 채택되었다.(CON-3)

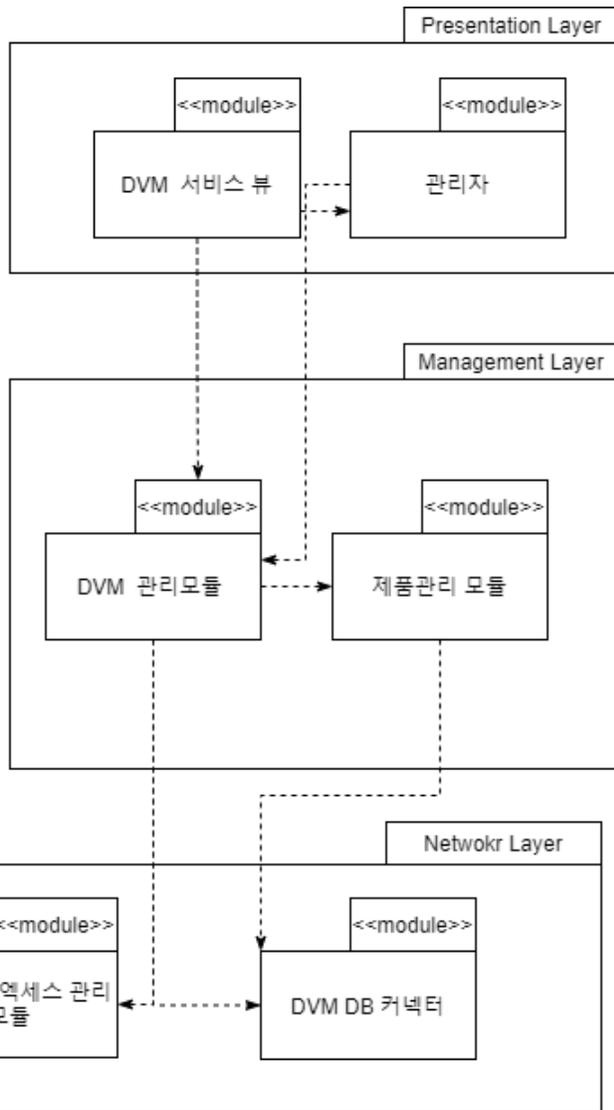
6. 뷰 스케치 및 설계 결정 기록

6. 뷰 스케치 및 설계 결정 기록



요소	책임
어플리케이션	사용자 상호작용 가능한 UI를 표현
데이터베이스 서버	데이터베이스를 호스팅하고, DVM 기기의 네트워크를 관리하는 서버.
동기화 서버	클라이언트 어플리케이션과 네트워크 데이터 베이스 서버 사이의 동기화를 관리하는 서버(QA-4, CRN-2)

6. 뷰 스케치 및 설계 결정 기록



요소	책임
DVM 서비스 뷰	이 모듈은 Spring을 사용하여 구현되며, 사용자가 DVM에서 판매되는 제품을 구매하기 위해 DVM 하드웨어, 제품 종류를 선택할 수 있도록 하는 뷰를 표시한다.
DVM 관리 모듈	이 모듈은 Spring를 사용하여 구현되며, DVM의 동작 상태를 관리한다. 또한 재고와 종류를 관리하며, 제품 추가 및 재고 부족 시 알람 및 추가 관리를 할 수 있다.
네트워크 관리 액세스 모듈	이 모듈은 DVM의 네트워크 신호를 관리하며, 네트워크 간의 지연이 발생할 경우 지연 시간동안 거래 처리를 중단시킨다. 네트워크 신호 탈취, 위변조로부터 네트워크를 보호한다. 외부 서버와 통신을 통해 결제 프로세스를 관리한다.
DVM DB	DVM의 판매 제품 종류, 제품의 재고량, 작동 상태, DVM위치를 저장하는 데이터베이스이다. 다른 DVM이 판매 제품 종류, 제품의 재고량, 작동 상태, DVM위치에 접근하는 것이 가능하다.
관리자	이 모듈은 Spring을 사용하여 구현되며, DVM, DB, 네트워크를 관리할 수 있는 관리자 권한을 처리한다. 또한, 기존 사용자 서버와 통합된다.

7. 현재 설계 분석 수행 및 Iteration 목표와 설계 목표 달성 검토

7. 현재 설계 분석 수행 및 Iteration 목표와 설계 목표 달성 검토

해결되지 않음	부분적으로 해결	해결됨	이테레이션 동안 이루어진 설계 설정
	QA-1		Java 기반의 프로그램 설계로 비정상적인 타입의 입력을 제한함
QA-2, QA-3			아직 결정되지 않음.
	QA-4		서비스 애플리케이션 참조 아키텍처를 사용하여 동기화 서버를 사용하여, 동기화를 지원할 모듈이 인식되었다.
QA-5			아직 결정되지 않음.
CON-1, CON-2			아직 결정되지 않음.
	CON-3		모듈이 식별됨
		CON-4	Java 개발자에게 익숙한 프레임워크를 사용하여 최소인원으로 개발이 가능하게 됨
		CRN-1	하드웨어에 설치하는 리치 클라이언트 애플리케이션 참조 아키텍처를 선택하여 이 시스템의 논리적 구조로서 인식되었다.
CRN-2			아직 결정되지 않음.
	CRN-3		서비스 애플리케이션 참조 아키텍처를 사용하여 동기화 서버를 사용하여, 동기화를 지원할 모듈이 인식되었다.
	UC-1, UC-2, UC-3, UC-4, UC-5, UC-6, UC-7, UC-8, UC-9, UC-10, UC-11		이 유스케이스들을 지원할 모듈이 인식됨

2nd Iteration

1. (1,2단계 통합)요인을 선택하여 이터레이션 목표 수립.....19p
2. 정제할 시스템 요소 선택.....21p
3. 선택된 요인을 충족시키는 설계 개념 선택.....23p
4. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의.....25p
5. 뷰 스케치 및 설계 결정 기록.....27p
6. 현재 설계 분석 수행 및 Iteration 목표와 설계 목표 달성 검토.....41p

1. 요인을 선택하여 이터레이션 목표 수립

1. 요인을 선택하여 이터레이션 목표 수립

CRN-2, 3 :

자판기의 네트워크 연결을 관리하는 모듈 필요
결제 완료 시, 실시간으로 재고상태가 동기화가 되어야 한다.

이터레이션 2의 목표:

구현단위에 관하여 추론

최우선 UC :

UC-1, UC-2, UC-3, UC-4, UC-5, UC-6, , UC-7, UC-8, UC-9, UC-10, UC-11

2. 정제할 시스템 요소(모듈)

2. 정제할 시스템 요소(모듈)

“지난 이터레이션1의 참조 아키텍처에 의해 정의된 다른 레이어 안에 위치한 모듈을 정제한다.”

3. 선택된 요인을 충족시키는 설계 개념 선택

3. 선택된 요인을 충족시키는 설계 개념 선택

설계 결정 및 위치	근거와 가정
애플리케이션 도메인 모델 생성	주요 엔티티와 엔티티간 관계를 식별하여 초기 도메인 모델을 생성한다. 도메인 모델을 생성하지 않으면 유지보수가 어려워 진다.
기능 요구에 매핑할 도메인 객체 식별	애플리케이션 요소와, 각 기능 요소를 도메인 객체로서 식별한다. 도메인 객체를 고려하지 않고 레이어를 직접 모듈로 분할할 경우, 기능 요구를 고려하지 않아 위험이 증가한다.
도메인 객체를 컴포넌트로 분할	각 기능을 지원하는 모듈로 분할하기 위해 레이어 내에 컴포넌트 모듈로 더욱 자세히 분할한다.
Java rich client 사용	Java rich client는 클라이언트 어플리케이션의 개발을 지원하는 플랫폼이다. 완전한 어플리케이션을 처음 부터 작성하는 대신 플랫폼에서 제공하는 프레임 워크의 검증되고 테스트 된 기능을 활용할 수 있다.
Spring 프레임워크 사용	Java 기반의 Spring 프레임워크는 type safe하며 Null safety를 지원한다. 이를 통해 QA-1을 해결가능하다.
SMTP 프로토콜 사용	문제가 발생한 경우 관리자에게 이메일로 알림을 전송한다. (CON-2, QA-2)

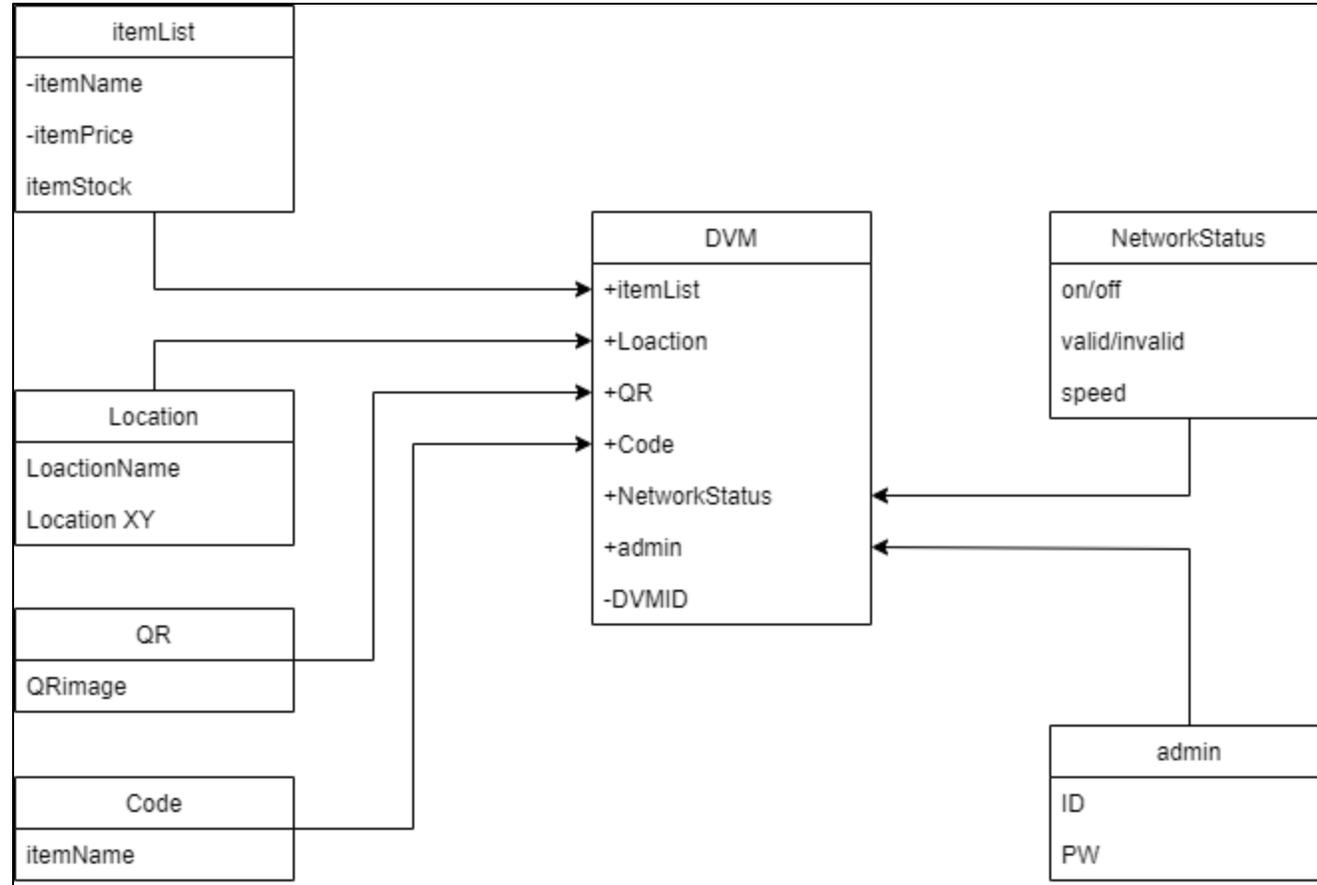


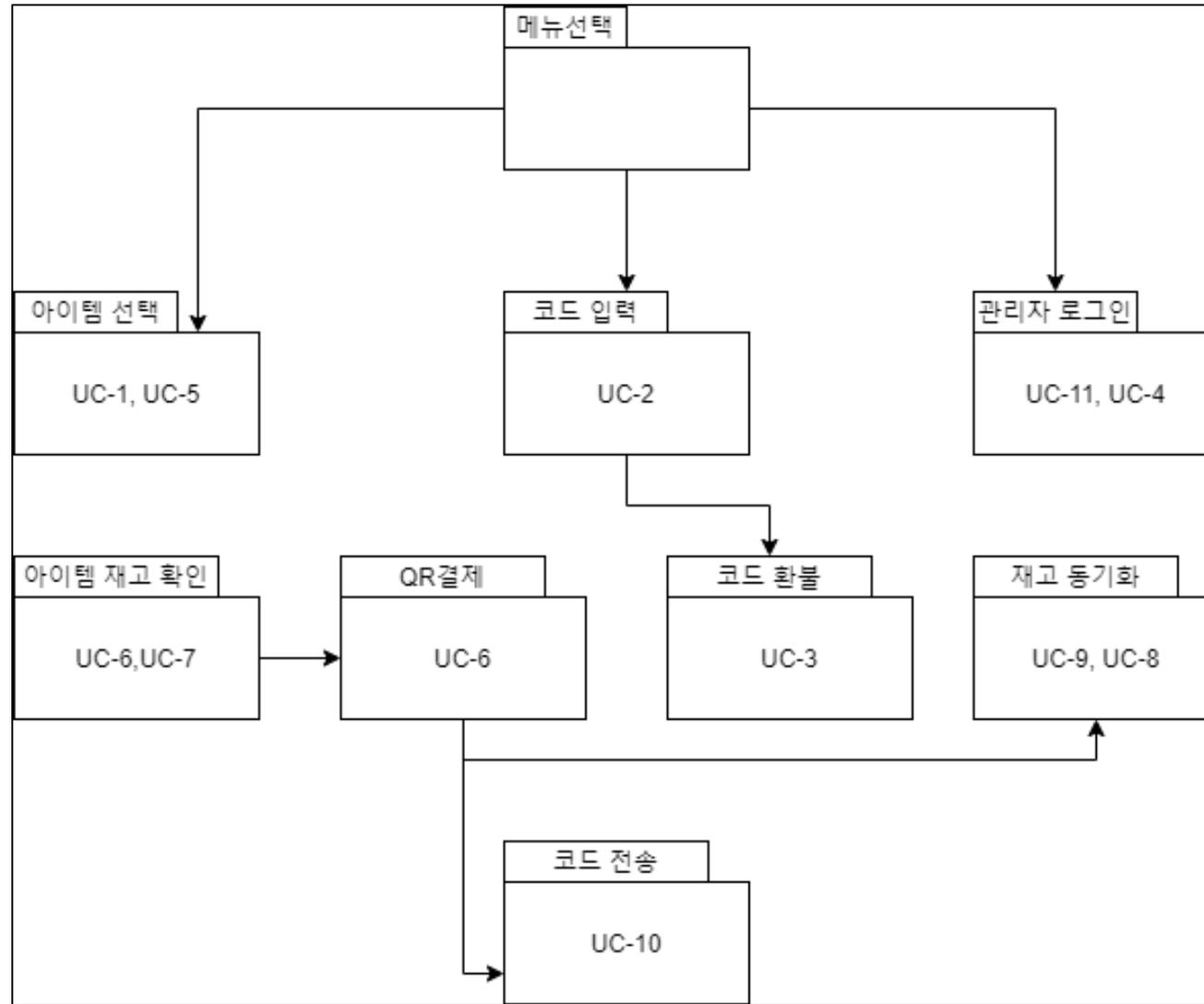
4. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의

4. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의

설계 결정 및 위치	근거
초기 도메인 모델만 생성	최우선 유스케이스에 entity는 식별되고, 모델링될 필요가 있지만, 이 설계 단계를 가속화하기 위해 초기 모델만 생성된다.
시스템 유스케이스를 도메인 객체에 맵핑	시스템의 유스케이스를 분석함으로써 초기에 도메인 객체를 식별 할 수 있다. CRN-2, 3을 해결하기 위해 모든 유스케이스에 대한 도메인 객체가 식별된다.
Spring을 데이터 레이어에 있는 apache web server와 연결시킴	Apache webserver를 사용하여 HTTP 통신을 manage 한다. 또한 관리자에게 오류 보고를 하기 위해 SMTP도 지원한다.

5. 뷰 스케치 및 설계 결정 기록





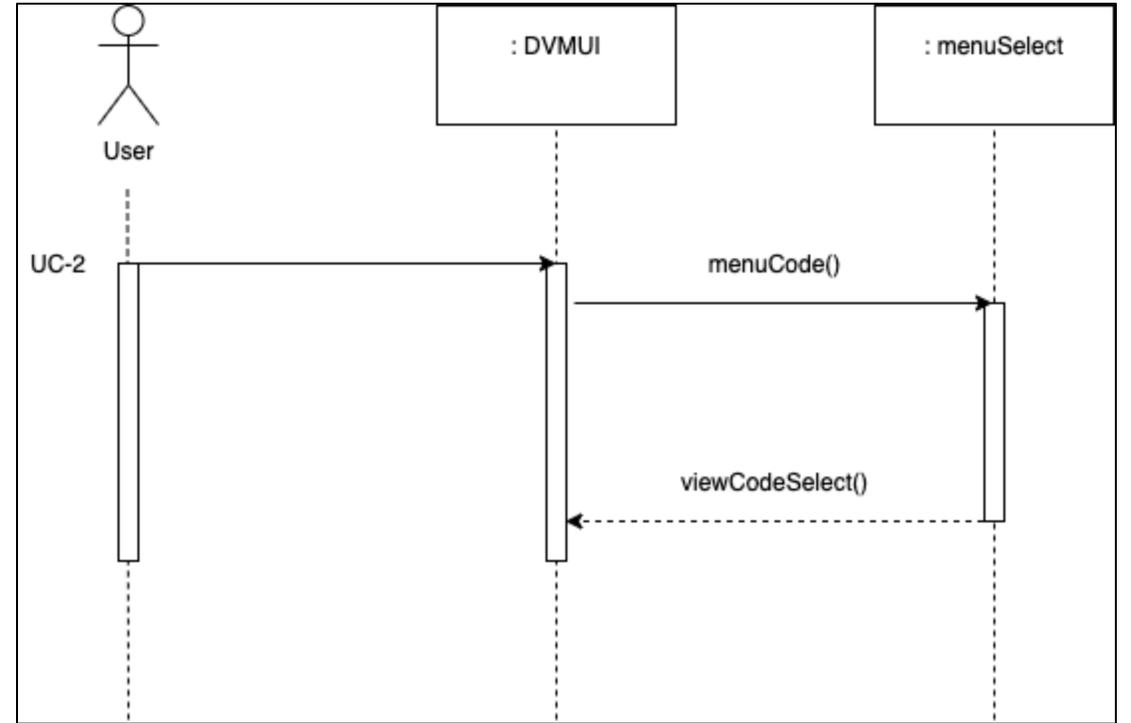
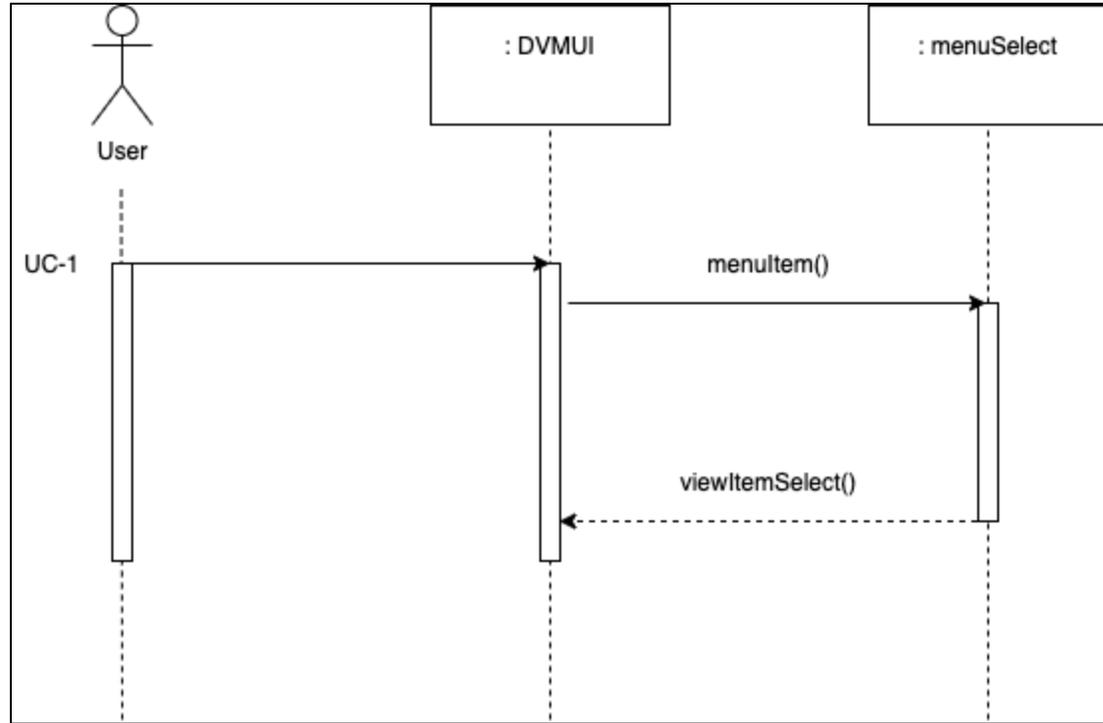
메서드명	설명
요소 : menuSelect	
menuItem()	DVMUI 로부터 '상품선택' 메뉴 선택 정보를 받는다.
viewItemSelect()	DVMUI 에 '상품 선택 화면 데이터'를 반환한다.
menuCode()	DVMUI 로부터 '코드입력' 메뉴 선택 정보를 받는다.
viewCodeSelect()	DVMUI 에 '코드 입력 화면 데이터'를 반환한다.
menuAdmin()	DVMUI 로부터 '관리자 모드' 선택 정보를 받는다.
viewAdminSelect()	DVMUI 에 '관리자 모드 화면 데이터'를 반환한다.

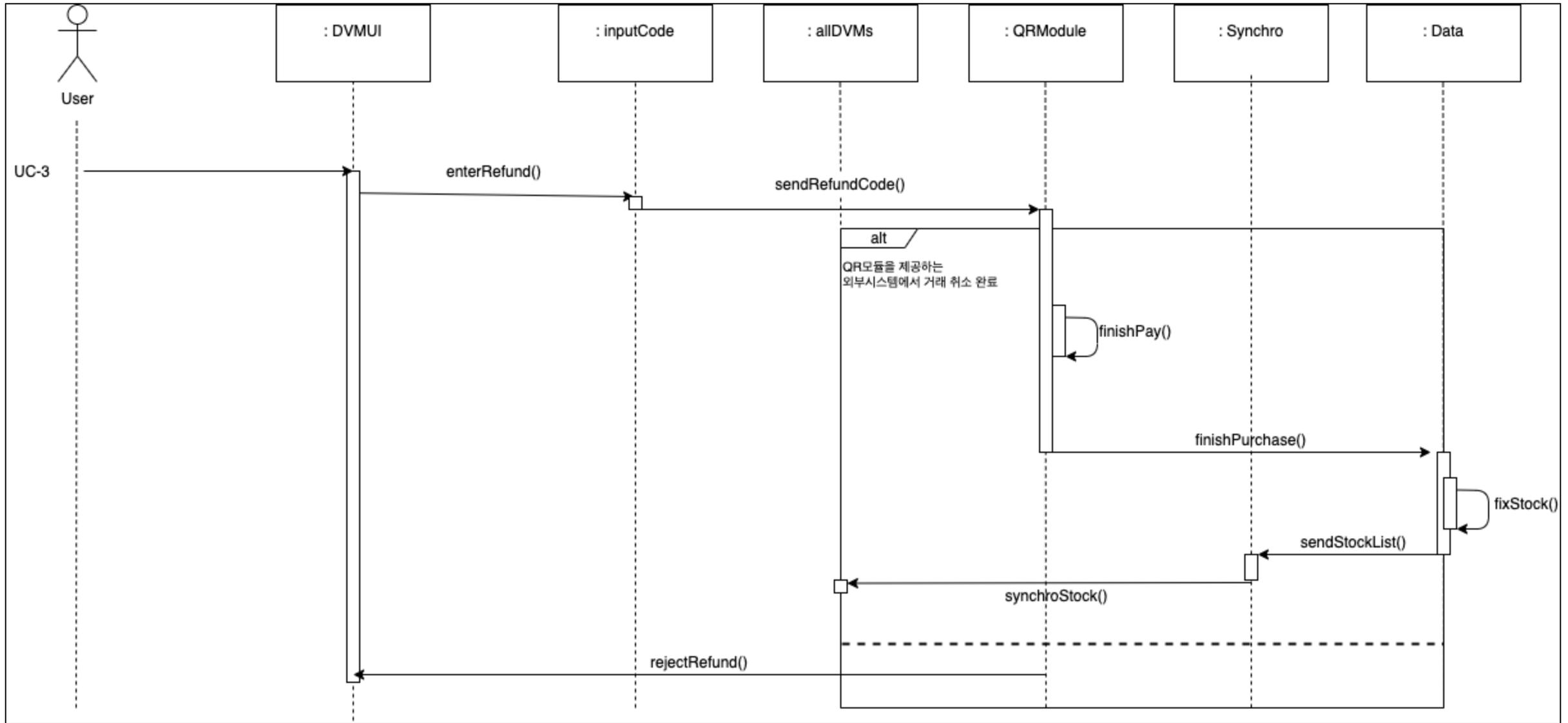
메서드명	설명
요소 : admin	
ChangeStock()	재고 변경 정보를 요청한다.

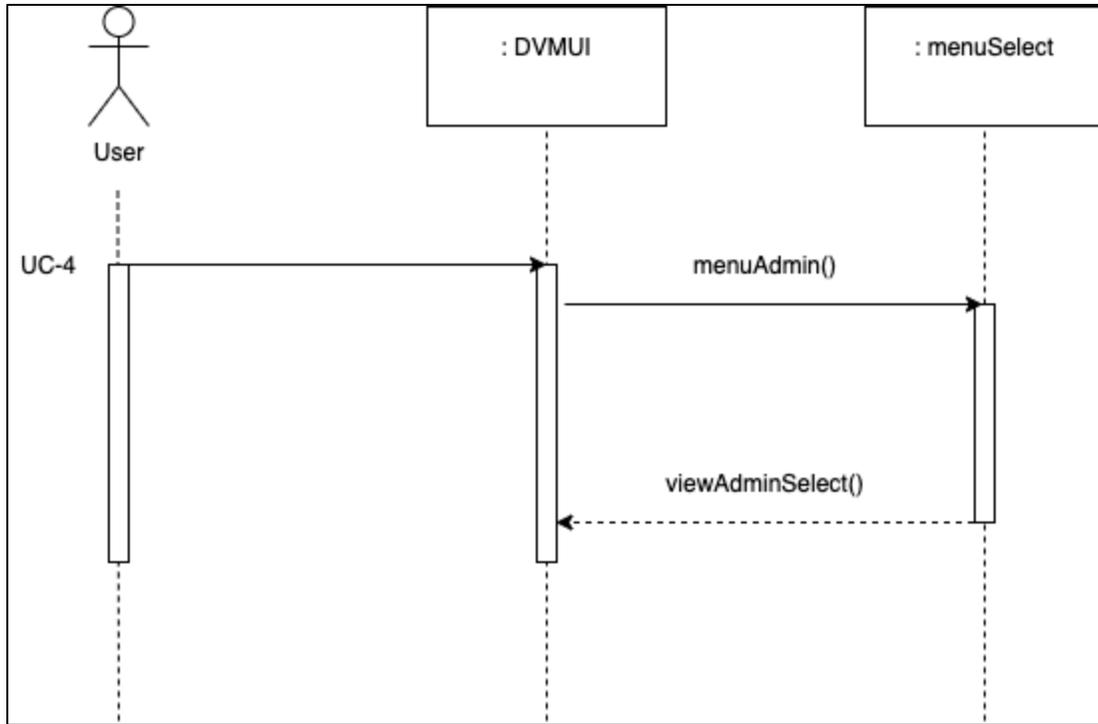
메서드명	설명
요소 : inputCode	
enterCode()	DVMUI 로부터 결제된 코드를 받는다
enterRefund()	DVMUI 로부터 환불 여부 정보를 요청한다.
selectItem()	ItemSelect 로부터 '선택된 제품' 정보를 요청한다.

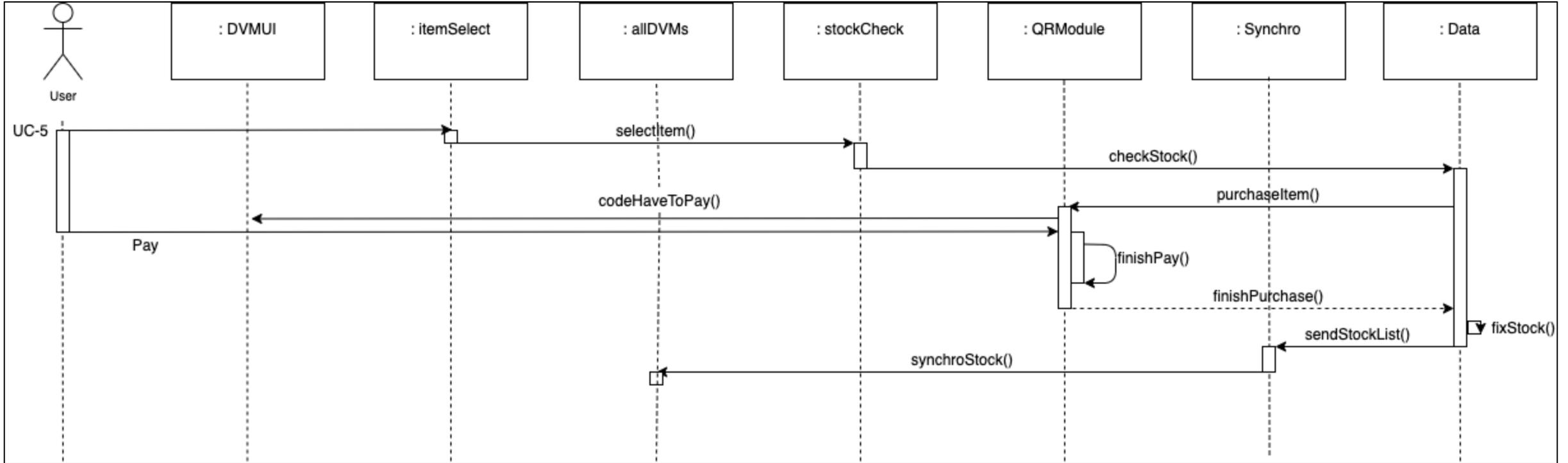
메서드명	설명
요소 : Syncro	
synchroStock()	sendStockList로부터 받은 재고 정보를 allDVMs 로 보낸다.
IsStockDVM()	allDVMs에 IsStockDVMConfirm으로 부터 받은 정보에 해당하는 재고 정보를 요청한다.
IsStockInfo()	allDVMs 중 재고가 있는 자판기로부터 해당 자판기 정보를 받는다.
networkStatus()	네트워크 상태를 요청한다.
warningNetReport()	불안정한 네트워크 상태 정보를 관리자에게 반환한다.
sendCodeInfo()	SendCodeAndSelectedDVMInfo 로부터 받은 코드와 DVM정보를 selectedDVM으로 보낸다.

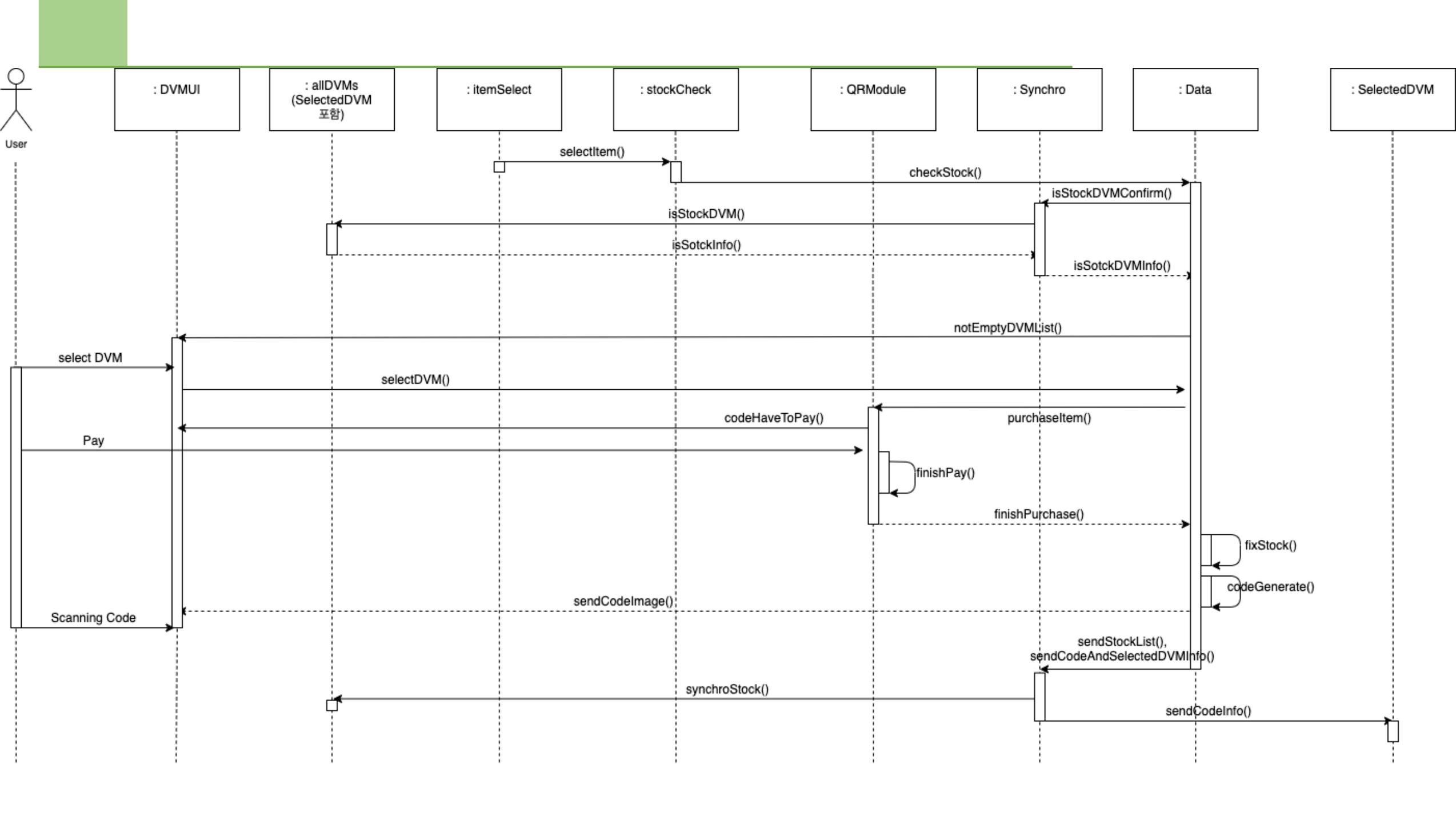
메서드명	설명
요소 : Data	
finishPurchase()	QRModule로부터 결제 완료 정보를 받는다.
fixStock()	재고를 변경한다.
sendStockList()	Synchro에 변경된 재고 정보를 보낸다.
checkStock()	StockCheck 로부터 자판기의 제품 재고 정보를 받는다.
adminChangeStock()	DVMMManagementUI 로부터 관리자 권한 재고 변경 정보를 받는다.
adminChangeStockFinishInfo()	DVMMManagementUI 에 관리자 권한 재고 변경 완료 정보를 보낸다.
purchaseItem()	QRModule 에 결제해야하는 정보를 보낸다.
isStockDVMConfirm()	CheckStock 으로부터 받은 제품 재고 정보를 바탕으로 Synchro 에 재고가 있는 자판기 정보를 요청한다.
isStockDVMInfo()	Synchro 로부터 재고가 있는 자판기 정보를 받는다.
notEmptyDVMIList()	DVMUI 에 재고가 있는 자판기 정보를 보낸다.
selectDVM()	DVMUI 로부터 사용자가 선택한 '재고가 있는 다른 자판기'를 받는다.
codeGenerate()	SelectDVM 이후 대기하다가 QRModule로부터 fixStock 요청이 오면, 재고가 있는 자판기로 보낼 코드를 생성한다.
sendCodeImage()	DVMUI에 생성된 코드 정보를 보낸다.
sendCodeAndSelectedDVMInfo()	Synchro 에 코드를 보내야 할 자판기정보와 코드를 보낸다.
manageStock()	관리자가 변경한 재고 정보를 요청한다.

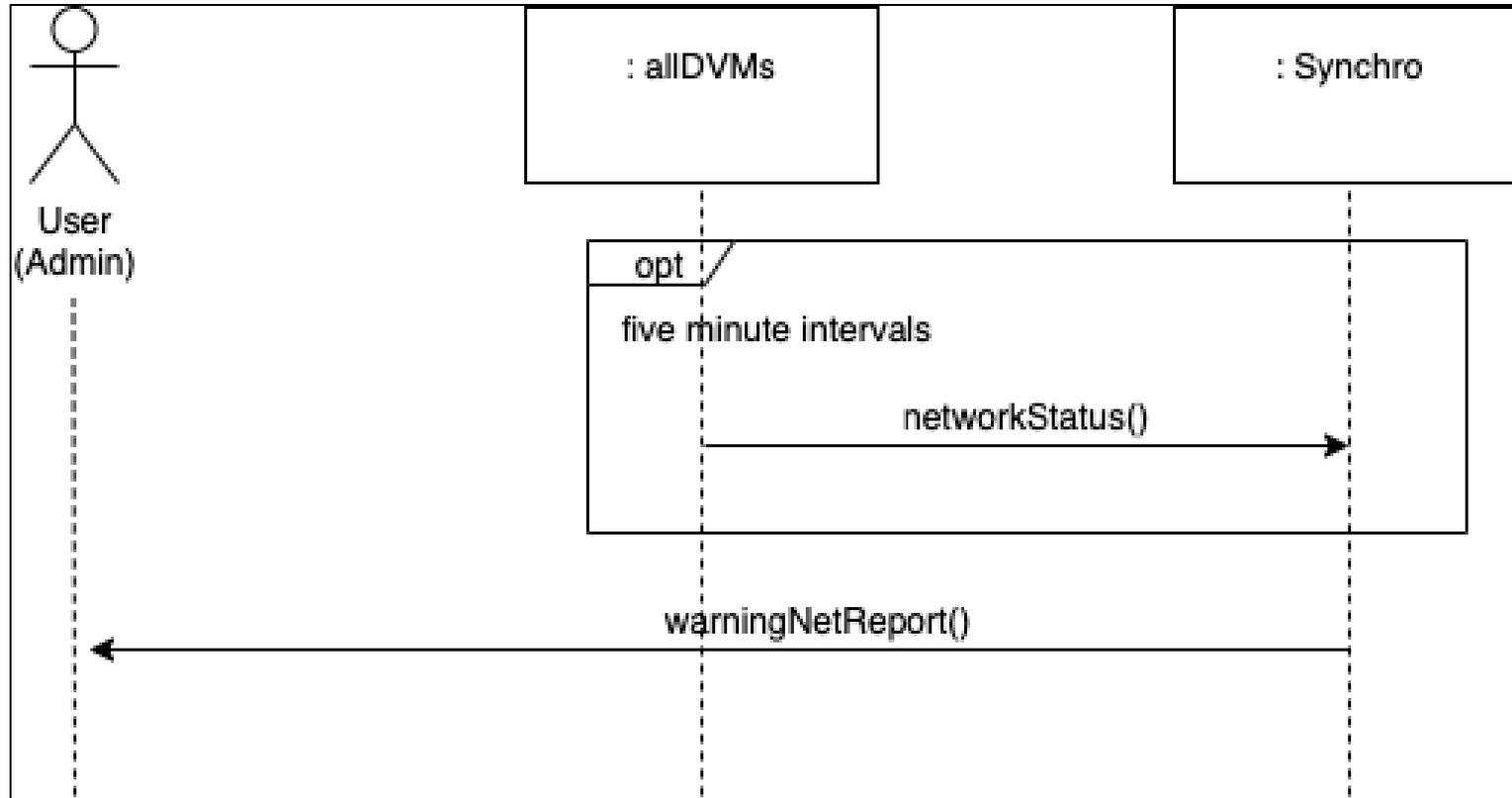


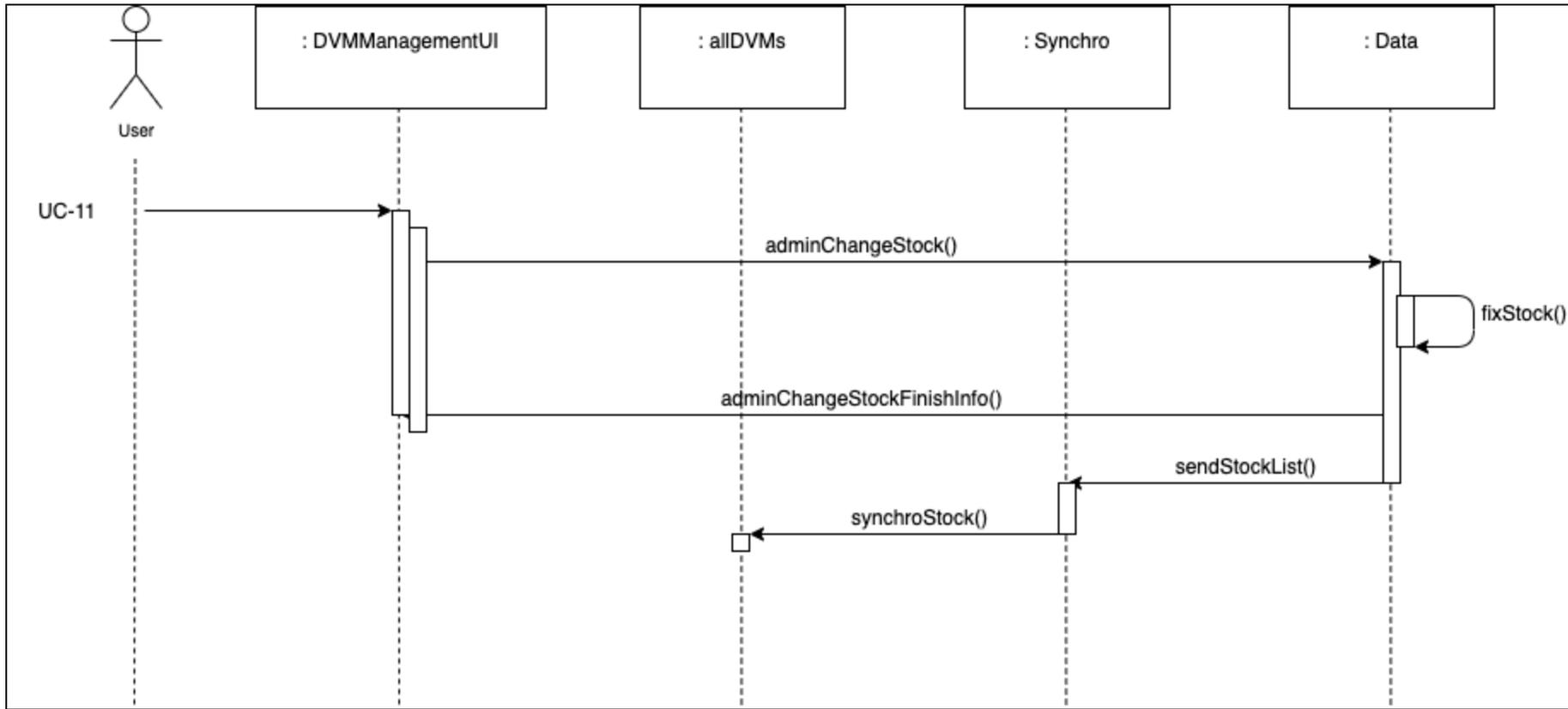












6. 현재 설계 분석 수행 및 이터레이션 목표와 설계 목표 달성 검토

해결되지 않음	부분적으로 해결	해결됨	이테레이션 동안 이루어진 설계 결정
		UC-1, UC-2, UC-3, UC-4, UC-5, UC-6, UC-7, UC-8, UC-9, UC-10, UC-11	이 유스케이스들을 지원하는 레이어에 걸친 모듈과 예비 인터페이스가 식별되었음.
		QA-1, QA-2, QA-4	해당 QA 관련된 모듈이 식별되었고, 관련 유스케이스에 해당되는 메서드를 포함한 시퀀스 다이어그램이 생성됨.
	QA-3		해당 QA 관련된 모듈이 식별되었고, 관련 유스케이스 해당되는 메서드를 포함한 시퀀스 다이어그램이 생성됨. 유효 기간에 대한 정의가 아직 없다.
QA-5			결정되지 않음.
CON-1			결정되지 않음.
		CON-2	SMTP 프로토콜을 사용하여 해결함.
	CON-3		해당 관련 모듈이 식별되었고 메서드를 포함한 시퀀스 다이어그램도 형성되었다. 점검 시간 단위가 정해지지 않았다.
			결정되지 않음.
		CRN-2, CRN-3	해당 고려사항과 관련된 모듈이 식별되었고 관련 유스케이스에 해당되는 메서드를 포함한 시퀀스 다이어그램이 생성됨.

3rd Iteration

1. (1,2단계 통합)요인을 선택하여 이터레이션 목표 수립.....44p
2. 정제할 시스템 요소 선택.....46p
3. 선택된 요인을 충족시키는 설계 개념 선택.....48p
4. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의.....50p
5. 뷰 스케치 및 설계 결정 기록.....52p
6. 현재 설계 분석 수행 및 Iteration 목표와 설계 목표 달성 검토.....54p



1. 요인을 선택하여 이터레이션 목표 수립

1. 요인을 선택하여 이터레이션 목표 수립

QA-3,5 :

코드의 유효기간은 일주일이며, 유효기간 내에 코드를 환불할 수 있다.
자판기 수가 늘어나도 전체 시스템 성능에 영향을 주지 않아야 한다.

이터레이션 3의 목표:

품질 속성의 달성에 집중한다.

2. 정제할 시스템 요소(모듈)

2. 정제할 시스템 요소(모듈)

이전 이터레이션에서 식별된 물리적 노드

동기화 서버,
데이터베이스 서버

3. 선택된 요인을 충족시키는 설계 개념 선택

3. 선택된 요인을 충족시키는 설계 개념 선택

이번 이터레이션의 설계 활동은 요소와 관련된 기술의 설정을 포함한다.
따라서 어떤 새로운 설계 개념이 선택되지 않았으며,
모든 결정은 인스턴스화 카테고리에 속한다.



4. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의

4. 아키텍처 요소 인스턴스화 및 책임 할당과 인터페이스 정의

설계 결정 및 위치	근거와 가정
동기화서버에 스케일링과 로드밸런싱 사용	스케일링과 로드밸런싱을 사용하여 시스템 성능 상 동기화에 문제가 없도록 지원하며, 최소 1000명 이상의 트래픽 상황에서도 성능에 문제가 없도록 한다.(QA-5, CON-1)
인증유효기간을 도입하여 재고 무한 정지 상황을 방지	코드를 사용하지 않아 DVM에서의 재고 무한 홀딩 상황이 발생 할 수 있어 생성된 코드의 유효기간을 부여하여 해당 상황을 방지하도록 한다. (QA-3)
DVM 하드웨어와 동기화 서버에 타이머 적용	네트워크의 모니터링이 상시 이루어지도록 신호전송에 주기를 부여한다. (CON-3)

5. 뷰 스케치 및 설계 결정 기록

메서드명	설명
요소 : Synchro	
NetworkStatus()	네트워크 성능 데이터를 요청한다.
Timer()	5분마다 네트워크 성능 데이터 요청을 반환한다.

6. 현재 설계 분석 수행 및 이터레이션 목표와 설계 목표 달성 검토

해결되지 않음	부분적으로 해결	해결됨	이테레이션 동안 이루어진 설계 결정
		QA-5, CON-1	동기화서버에 스케일링과 로드밸런싱 사용하여, 트래픽 상황에서도 성능에 문제가 없도록 하였다.
		QA-3	해당모듈이 식별되었고, 기존의 시퀀스 다이어그램에 메서드가 추가되었다.
		CON-3	해당 관련 모듈이 식별되었고 메서드를 포함한 시퀀스 다이어그램도 형성되었다. 메서드 작동에 주기를 도입하였다.