

DVM 시스템

Architecture Description

-HP Style-

소프트웨어공학특강

-팀 1-

윤상혁, 최슬아, 루카이

목차

1. Introduction	3
2. System Purpose.....	5
3. Structure	7
4. Dynamic Behavior	24
5. Other Views.....	34
6. Conceptual Framework.....	39
7. Conclusion	41

1. Introduction

1.1 Purpose

분산 자판기 시스템을 위한 DVM 시스템과 이를 관리하기 위한 DVM 관리 시스템의 아키텍처를 정의한다. 이 문서는 아래 Stakeholder 들을 위해 작성되었다.

1.2 Stakeholders

- 판매 담당자
- 사용자
- 재고관리 담당자
- 개발자
- 네트워크 관리자
- 테스터

1.3 Creating and Modification History

버전	작성자	내용	일자
0.1	윤상혁	제정	21.06.07
0.2	루카이	Section 2 작성	21.06.08
0.3	최슬아	Section 4 작성	21.06.10
1.0	윤상혁	완료	21.06.13

1.4 Selected Viewpoints

View Type	View Style	Section
Module View Type	Decomposition Style	3.1.1
	Uses Style	3.1.2
	Layered Style	
C&C View Type	Publisher-Subscribe Style	5.1.1

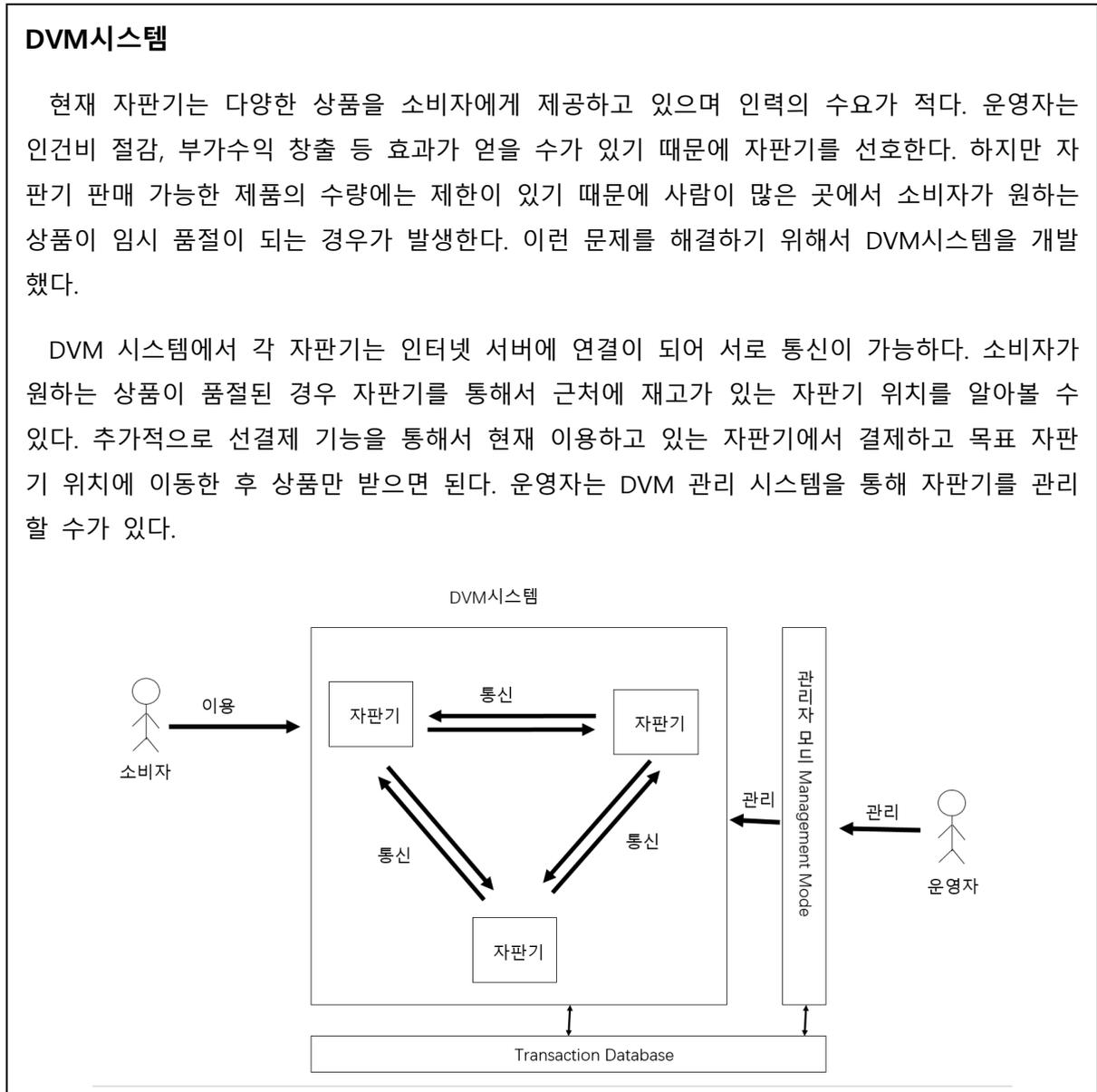
	Client-Server Style	5.1.3
	Service-Oriented Architectural Style	5.1.4
	Shared Data Style	5.1.5
Allocation View Type	Deployment Style	5.2.1

1.5 Name of Architecture

- DVM System Architecture (Service Application Reference Architecture)
- DVM Management System Architecture (Rich Client Application Reference Architecture)

2. System Purpose

2.1 Context



2.2 Interface

Interface	Description
Check Stock	매니지먼트 시스템에 재고 확인을 위한 서비스를 제공한다.
UpdateMachineList	매니지먼트 시스템에 자판기 리스트 업데이트를 위한 서비스를 제공한다.

Network Status	매니지먼트 시스템에 네트워크 상황 확인을 위한 서비스를 제공한다.
Prepayment	다른 자판기에 선결제 서비스를 제공하거나 이용한다.
RequestDrink	다른 자판기에 재고 확인을 위한 서비스를 요청한다.

2.3 Non-functional Requirements

DVM시스템: Qualities

- 인증 코드 발급 시 해당 자판기에 코드 발급을 확인한 후 사용자에게 코드를 발급한다.
- 새로운 자판기를 추가하면 모든 자판기에 1분 안에 자신의 정보를 업데이트 한다.
- 현재 자판기에 재고가 없을 경우 항상 구매 가능한 자판기를 찾아 출력해야 한다. 만약 해당 자판기가 멀리 떨어져 있더라도 결과는 10초 안에 출력 되어야 한다.
- 자판기의 네트워크 연결 상태를 실시간으로 확인한다.
- 재고의 상태는 실시간으로 확인 가능해야 하며 재고가 부족한 경우 1분 안에 알림이 와야 한다.
- 모든 결제 정보는 암호화가 되어 전송 및 저장이 되어야 한다.
- 테스트 모드에서 자판기의 상태를 변경하였을 때 즉시 반영되어야 하며 다른 자판기에서도 변경된 상태가 보여야 한다.
- 새로운 사용자가 자판기 사용 시 5분 안에 구매가 가능해야 한다.
- 동시에 여러 사용자가 한 자판기의 음료 구매를 원할 경우에도 항상 정확한 판매 수량을 보여주어야 한다.

DVM시스템: Constraints

- 모든 음료를 제공하기 위해선 최소 3개의 자판기가 존재해야 한다.
- 사용자의 구매 내역은 최소 45일동안 저장되어야 한다.

DVM시스템: Principles

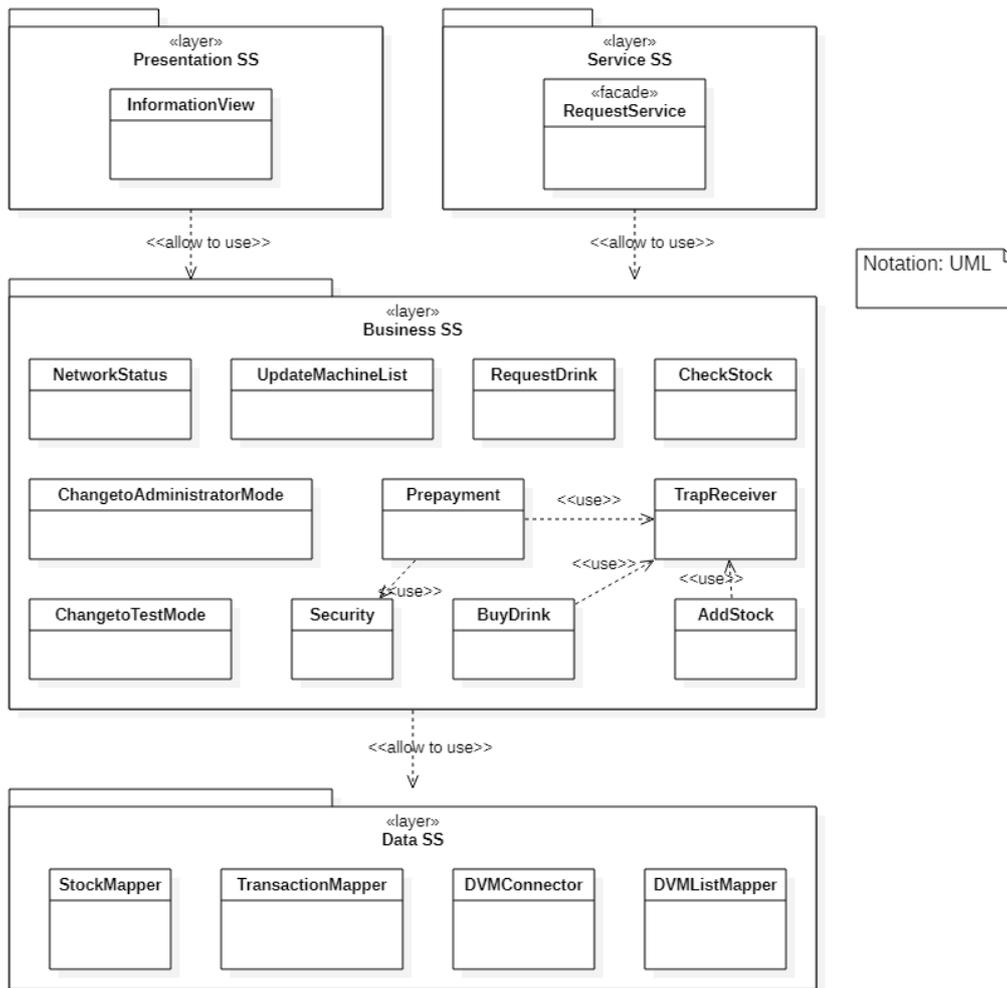
- 일정 시간 이상 반응이 없을 경우 원래 상태로 회복하기 위해 컴포넌트들에 Timestamp와 Rollback 기법을 사용한다.
- 결제 컴포넌트들에 Encrypt Data 기법을 적용한다.
- 재고 확인 및 추가 컴포넌트에 Message queue, Prioritize events 기법을 적용한다.

3. Structure Section

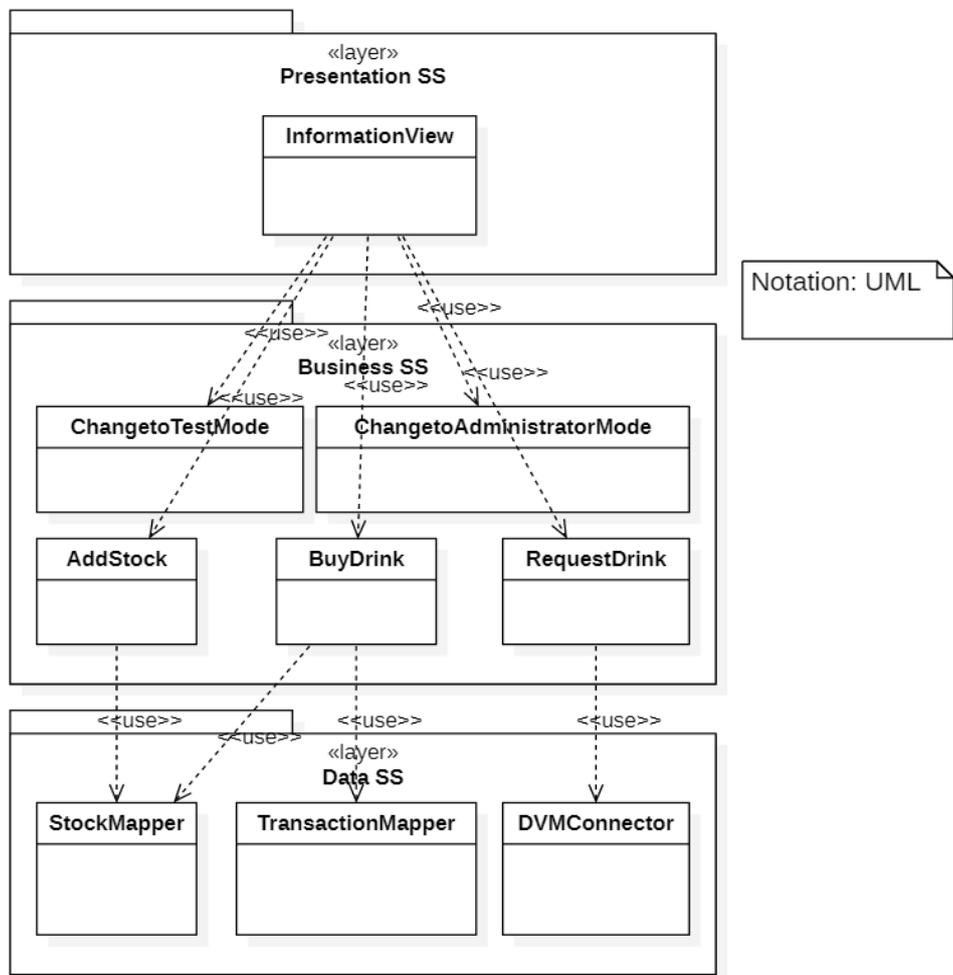
3.1 Overview

3.1.1 Overview Architecture of DVM System

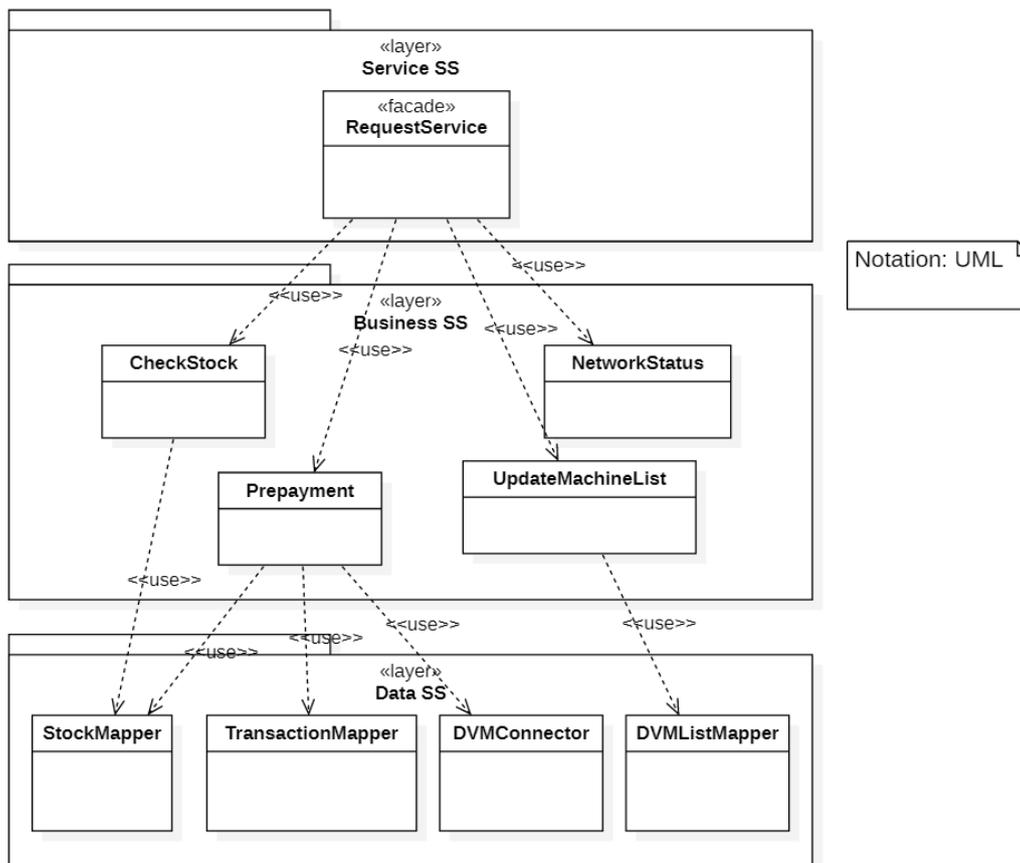
3.1.1.a Primary presentation of DVM System



3.1.1.b Use style view in terms of InformationView Module



3.1.1.c Use style view in terms of RequestService Module



3.1.1.d Design decision for DVM System and Rationale

Design Decisions and Location	Rationale
DVM을 위한 논리적 구조로 Service Application Reference Architecture 를 사용한다.	UI 지원은 없지만 클라이언트와 서버가 loosely coupled 되어 있어 분산 시스템에서 사용이 적합하다. Message Broadcasting 으로 재고 확인을 하는 DVM과 맞게 메시지 기반으로 통신이 가능하며 관리 시스템과 같은 다른 시스템과도 동작 가능하다 (Interoperability). 자판기를 위한 UI는 외부 시스템 연결로 구현이 필요하다.

3.1.1.e Alternative design for DVM System and reason for discarding

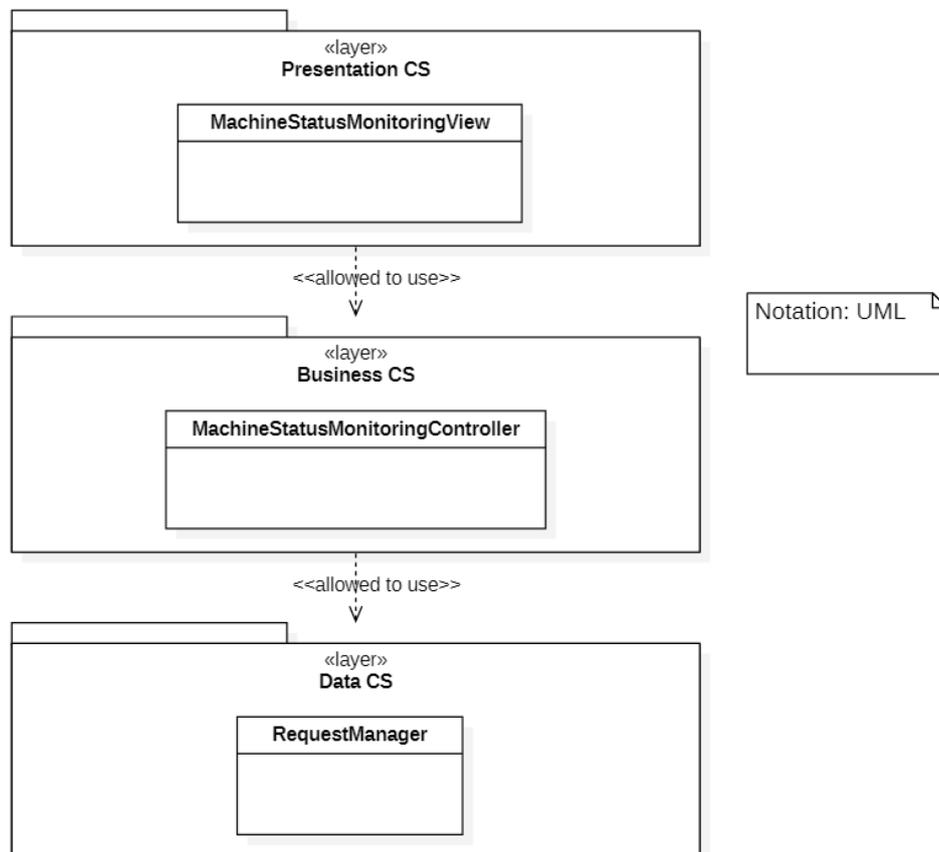
Alternative	Reason for Discarding
Rich Client Application Reference Architecture	사용자와 풍부한 상호작용과 빠른 반응성을 보여주지만 DVM에서 중요한 것은 UI보다 자판기간의 연결성이 더 중요하므로 선택하지 않았다.

3.1.1.f Selected Style and Rationale

Selected Style	Rationale
Module View with Decomposition, Uses, Layered Style	<p>DVM 시스템의 전체적인 구성을 나타내기 위해 Decomposition Style을 사용하였다.</p> <p>DVM 시스템은 Presentation, Service, Business, Data 레이어로 구성되어 있다. 이를 표현하기 위해 Layered Style을 사용하였다.</p> <p>DVM 시스템의 모듈들은 서로를 사용한다. 이를 나타내기 위해 Uses Style을 사용하였다.</p>

3.1.2 Overview Architecture of DVM Management System

3.1.2.a Primary Presentation of DVM Management System



3.1.2.b Design decision for DVM Management System and rationale

Design Decisions and Location	Rationale
-------------------------------	-----------

DVM 관리 시스템을 위한 논리적 구조로 Rich Client Application Reference Architecture 를 사용한다.	관리자가 자판기의 재고 수량을 확인하기 쉽게 풍부한 UI 환경을 제공한다. 또한 자판기와 네트워크 통신을 통해 재고 사항을 확인할 수 있다. 데이터베이스에 저장되어 있는 데이터를 읽어와 판매 기록 및 분석을 확인할 수 있다.
--	---

3.1.2.c Alternative designs for DVM Management System and reason for discarding

Alternative	Reason for Discarding
Rich Internet Application Reference Architecture	웹 브라우저로 접근이 가능하여 접근성은 높지만 여러 자판기 관리, 재고 관리, 재고 부족 시 알림 등 사용자와 풍부한 상호작용에 제약사항이 존재하여 선택하지 않았다.
Web Application Reference Architecture	웹 브라우저로 접근이 가능하여 접근성은 높지만 여러 자판기 관리, 재고 관리, 재고 부족 시 알림 등 사용자와 풍부한 상호작용에 제약사항이 존재하여 고르지 않았다.
Mobile Application Reference Architecture	여러 자판기와 통신하여 재고 사항을 읽어와야 하기 때문에 성능과 네트워크 연결 사항에 제약이 생겨 선택하지 않았다.

3.1.2.d Selected Style and Rationale

Selected Style	Rationale
Module View with Decomposition, Uses, Layered Style	DVM 매니지먼트 시스템의 전체적인 구성을 나타내기 위해 Decomposition Style을 사용하였다. DVM 매니지먼트 시스템은 Presentation, Business, Data 레이어로 구성되어 있다. 이를 표현하기 위해 Layered Style을 사용하였다. DVM 매니지먼트 시스템의 모듈들은 서로를 사용한다. 이를 나타내기 위해 Uses Style을 사용하였다.

3.2 Components Section

3.2.1 Components of DVM System

Component	InformationView
Responsibilities	자판기를 사용하는 사용자에게 화면을 표시한다. 제공하는 인터페이스: (1) requireAccessToken 관리자 모드 혹은 테스트 모드에 접근을 원하는 사용자에게 접근 토큰을 요구한다. (2) displayInfo

	관리자 모드 혹은 테스트 모드로 전환하면 새로 업데이트된 화면을 표시한다. (3) request 사용자가 요청한 이벤트를 알맞은 컴포넌트에 요청한다.
Collaborators	Component: ChangetoAdministratorMode, ChangetoTestMode, AddSock, BuyDrink, RequestDrink
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	RequestService
Responsibilities	다른 자판기 혹은 관리 시스템으로부터 받은 요청을 처리한다. 제공하는 인터페이스: (1) SendRequest DVM 관리 시스템 혹은 다른 자판기에서 요청한 서비스를 처리한다.
Collaborators	Component: TrapReceiver, NetworkStatus, UpdateMachineList
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	AddStock
Responsibilities	재고를 추가한다.
Collaborators	Component: TrapReceiver, StockMapper
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	BuyDrink
Responsibilities	음료수를 구매 서비스를 제공한다.
Collaborators	Component: TrapReceiver, StockMapper, TransactionMapper
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	RequestDrink
Responsibilities	다른 자판기에 음료 사항 확인을 요청한다. 제공하는 인터페이스: (1) requestDrink

	현재 자판기에 없는 음료를 주변 자판기에 재고 확인을 요청한다.
Collaborators	Component: DVMConnector
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	ChangetoAdministratorMode
Responsibilities	관리자 모드로 전환한다. 제공하는 인터페이스: (1) changetoAdministratorMode 관리자 모드로 전환한다.
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	ChangetoTestMode
Responsibilities	테스트 모드로 전환한다. 제공하는 인터페이스: (1) changetoTestMode 테스트 모드로 전환한다.
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	CheckStock
Responsibilities	재고를 확인한다. 제공하는 인터페이스: (1) checkStock 현재 자판기의 재고를 확인한다.
Collaborators	Component: StockMapper
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	Prepayment
-----------	------------

Responsibilities	<p>선결제를 지원한다.</p> <p>제공하는 인터페이스:</p> <p>(1) requestPrepayment 다른 자판기에 선결제를 요청한다. 사용자의 결제정보를 다른 자판기에 전송한다.</p> <p>(2) prepayment 다른 자판기로부터 요청받은 선결제를 진행한다. 현재 자판기에서만 사용 가능한 시리얼 넘버를 생성하여 요청한 자판기로 전송한다.</p>
Collaborators	Component: TrapReceiver, Security, DVMConnector, StockMapper, TransactionMapper
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	NetworkStatus
Responsibilities	<p>현재 자판기의 네트워크 상황을 확인한다.</p> <p>제공하는 인터페이스:</p> <p>(1) getNetworkStatus 현재 자판기의 네트워크 상태를 반환한다.</p>
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	UpdateMachineList
Responsibilities	<p>자판기 리스트를 업데이트 한다.</p> <p>제공하는 인터페이스:</p> <p>(1) updateMachineList 현재 자판기에 등록 되어있는 주변 자판기들의 리스트를 업데이트한다.</p>
Collaborators	Component: DVMListMapper
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	TrapReceiver
Responsibilities	요청 사항을 메시지 큐에 추가한다.

	<p>제공하는 인터페이스:</p> <p>(1) Trap 큐에 삽입하고 싶은 메시지를 입력 받는다.</p> <p>(2) transformAndEnqueue 입력 받은 이벤트를 적절한 큐에 삽입한다.</p> <p>(3) consume 각 컴포넌트에서 원하는 이벤트를 큐에서 반환한다.</p>
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	Security
Responsibilities	<p>결제 정보 암호화를 위한 기능을 제공한다.</p> <p>제공하는 인터페이스:</p> <p>(1) encrypt 결제 정보를 암호화한다.</p> <p>(2) decrypt 결제 정보를 복호화한다.</p>
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	StockMapper
Responsibilities	<p>재고 사항을 관리한다.</p> <p>제공하는 인터페이스:</p> <p>(1) updateStock 현재 자판기의 재고사항을 업데이트한다.</p>
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	TransactionMapper
Responsibilities	<p>판매 사항을 업데이트한다.</p> <p>제공하는 인터페이스:</p>

	(1) registerTransaction 새로 발생한 거래 내역을 데이터베이스에 저장한다.
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	DVMConnector
Responsibilities	다른 자판기에 서비스를 요청한다. 제공하는 인터페이스: (1) requestStock 다른 자판기에 재고사항 확인을 요청한다.
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	DVMListMapper
Responsibilities	자판기 리스트를 관리한다. 제공하는 인터페이스: (1) updateMachineList 현재 자판기에 저장되어 있는 자판기 리스트를 업데이트한다.
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

3.2.2 Components of DMV Management System

Component	MachineStatusMonitoringView
Responsibilities	DVM 관리자와 상호작용 한다.
Collaborators	Component: MachineStatusMonitoringController
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	MachineStatusMonitoringController
Responsibilities	관리자의 요청 사항을 처리한다.

	<p>제공하는 인터페이스:</p> <p>(1) checkStock 자판기의 재고 사항을 확인을 요청한다.</p> <p>(2) updateMachineList 모든 자판기들에 새로운 자판기의 등록 혹은 삭제를 요청한다.</p> <p>(3) getNetworkStatus 자판기의 네트워크 상태를 표시를 요청한다.</p>
Collaborators	Component: RequestManager
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

Component	RequestManager
Responsibilities	<p>요청 사항을 DVM 시스템으로 전달한다.</p> <p>제공하는 인터페이스:</p> <p>(1) requestStock 자판기의 재고 사항을 확인을 요청한다.</p> <p>(2) requestDVMList 모든 자판기들에 새로운 자판기의 등록 혹은 삭제를 요청한다.</p> <p>(3) requestStatus 자판기의 네트워크 상태를 표시를 요청한다.</p>
Collaborators	
Notes	해당 컴포넌트는 시스템이 시작될 때 생성되어 종료될 때까지 상주한다.
Issues	

3.3 Interface Section

3.3.1 Interface of DVM System

Interface	requireAccessToken
Description	관리자 모드 혹은 테스트 모드로 접근을 원하는 관리자에게 접근 토큰을 입력 받는다.
Services	<p>Operation: requireAccessToken()</p> <p>Description: 접근 토큰을 입력 받아 유효성 검사를 진행한다.</p>
Protocol	제약 사항 없음
Notes	Supported by InformationView
Issues	

Interface	diplayInfo
Description	업데이트된 정보를 출력한다.
Services	Operation: diplayInfo() Description: 업데이트된 정보를 출력한다.
Protocol	제약 사항 없음
Notes	Supported by InformationView
Issues	

Interface	request
Description	사용자가 요청한 이벤트를 처리하기 위해 적절한 컴포넌트를 호출한다.
Services	Operation: request() Description: 사용자가 요청한 이벤트를 처리하기 위해 적절한 컴포넌트를 호출한다.
Protocol	제약 사항 없음
Notes	Supported by InformationView
Issues	

Interface	Trap
Description	새로운 이벤트를 입력 받는다.
Services	Operation: Trap() Description: 새로운 이벤트를 입력 받는다.
Protocol	제약 사항 없음
Notes	Supported by TrapReceiver
Issues	

Interface	transformAndEnqueue
Description	입력 받은 이벤트를 적절한 큐에 삽입한다.
Services	Operation: transformAndEnqueue() Description: 입력 받은 이벤트를 적절한 큐에 삽입한다.
Protocol	제약 사항 없음
Notes	Supported by TrapReceiver
Issues	

Interface	consume
Description	메시지 큐의 이벤트를 반환한다.
Services	Operation: consume()

	Description: 메시지 큐의 이벤트를 반환한다.
Protocol	제약 사항 없음
Notes	Supported by TrapReceiver
Issues	

Interface	updateStock
Description	자판기의 재고 사항을 업데이트 한다.
Services	Operation: updateStock() Description: 자판기의 재고 사항을 업데이트 한다.
Protocol	제약 사항 없음
Notes	Supported by StockMapper
Issues	

Interface	registerTransaction
Description	새로 발생한 거래 내역을 등록한다.
Services	Operation: registerTransaction() Description: 새로 발생한 거래 내역을 등록한다.
Protocol	제약 사항 없음
Notes	Supported by TransactionMapper
Issues	

Interface	sendRequest
Description	매니지먼트 시스템 혹은 다른 자판기로부터 요청 받은 서비스를 처리한다.
Services	Operation: sendRequest() Description: 매니지먼트 시스템 혹은 다른 자판기로부터 요청 받은 서비스를 처리한다.
Protocol	제약 사항 없음
Notes	Supported by RequestMapper
Issues	

Interface	checkStock
Description	자판기의 재고 사항을 확인한다.
Services	Operation: checkStock() Description: 자판기의 재고 사항을 반환한다.
Protocol	제약 사항 없음
Notes	Supported by CheckStock

Issues	
--------	--

Interface	changetoAdministratorMode
Description	관리자 모드로 변환한다.
Services	Operation: changetoAdministratorMode() Description: 관리자 모드로 변환한다.
Protocol	제약 사항 없음
Notes	Supported by ChangetoAdministratorMode
Issues	

Interface	changetoTestMode
Description	테스트 모드로 변환한다.
Services	Operation: changetoTestMode() Description: 테스트 모드로 변환한다.
Protocol	제약 사항 없음
Notes	Supported by ChangetoTestMode
Issues	

Interface	requestDrink
Description	다른 자판기에 음료 재고 사항 확인을 요청한다.
Services	Operation: requestDrink() Description: 다른 자판기에 음료 재고 사항 확인을 요청한다.
Protocol	제약 사항 없음
Notes	Supported by RequestDrink
Issues	

Interface	requestPrepayment
Description	다른 자판기에 선결제를 요청한다.
Services	Operation: requestPrepayment() Description: 다른 자판기에 선결제를 요청한다.
Protocol	제약 사항 없음
Notes	Supported by Prepayment
Issues	

Interface	prepayment
-----------	------------

Description	다른 자판기로부터 요청 받은 선결제를 처리한다.
Services	Operation: prepayment() Description: 다른 자판기로부터 요청 받은 선결제를 처리한다.
Protocol	제약 사항 없음
Notes	Supported by Prepayment
Issues	

Interface	encrypt
Description	결제 정보를 암호화한다.
Services	Operation: encrypt() Description: 결제 정보를 암호화한다.
Protocol	제약 사항 없음
Notes	Supported by Security
Issues	

Interface	decrypt
Description	결제 정보를 복호화한다.
Services	Operation: decrypt() Description: 결제 정보를 복호화한다.
Protocol	제약 사항 없음
Notes	Supported by Security
Issues	

Interface	updateMachineList(UpdateMachineList)
Description	자판기에 등록 되어있는 자판기 리스트를 업데이트한다.
Services	Operation: updateMachineList() Description: 자판기에 등록 되어있는 자판기 리스트를 업데이트한다.
Protocol	제약 사항 없음
Notes	Supported by UpdateMachineList
Issues	

Interface	updateMachineList(DVMMListMapper)
Description	자판기에 등록 되어있는 자판기 리스트를 업데이트한다.
Services	Operation: updateMachineList() Description: 자판기에 등록 되어있는 자판기 리스트를 업데이트한다.
Protocol	제약 사항 없음

Notes	Supported by DVMListMapper
Issues	

Interface	getNetworkStatus
Description	현재 자판기의 네트워크 상태를 반환한다.
Services	Operation: getNetworkStatus() Description: 현재 자판기의 네트워크 상태를 반환한다.
Protocol	제약 사항 없음
Notes	Supported by NetworkStatus
Issues	

3.3.2 Interface of DVM Management System

Interface	checkStock
Description	자판기의 재고 상황을 확인한다.
Services	Operation: checkStock() Description: 자판기의 재고 상황을 확인한다.
Protocol	제약 사항 없음
Notes	Supported by MachineStatusMonitoringController
Issues	

Interface	updateMachineList
Description	모든 자판기에 등록되어 있는 자판기 리스트를 업데이트한다.
Services	Operation: updateMachineList() Description: 모든 자판기에 등록되어 있는 자판기 리스트를 업데이트한다.
Protocol	제약 사항 없음
Notes	Supported by MachineStatusMonitoringController
Issues	

Interface	getNetworkStatus
Description	자판기의 네트워크 상태를 확인한다.
Services	Operation: getNetworkStatus() Description: 자판기의 네트워크 상태를 확인한다.
Protocol	제약 사항 없음
Notes	Supported by MachineStatusMonitoringController
Issues	

Interface	requestStock
Description	자판기의 재고 상태를 확인을 요청한다.
Services	Operation: requestStock() Description: 자판기의 재고 상태를 확인을 요청한다.
Protocol	TCP/IP
Notes	Supported by RequestManager
Issues	

Interface	requestDVMList
Description	모든 자판기에 등록되어 있는 자판기 리스트 업데이트를 요청한다.
Services	Operation: requestDVMList() Description: 모든 자판기에 등록되어 있는 자판기 리스트 업데이트를 요청한다.
Protocol	TCP/IP
Notes	Supported by RequestManager
Issues	

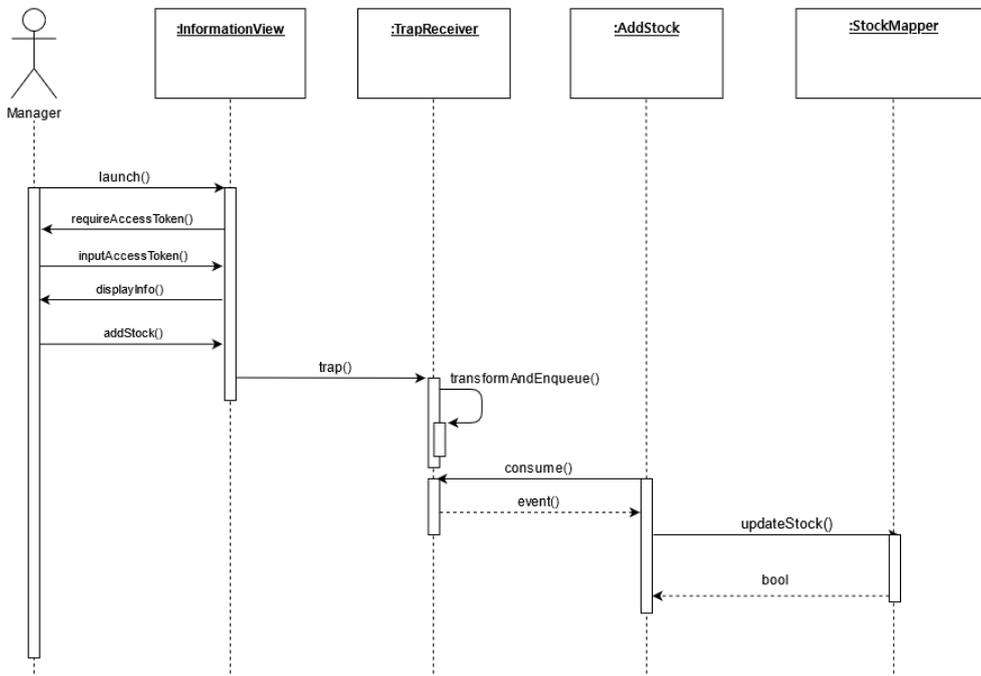
Interface	requestStatus
Description	자판기의 네트워크 상태를 확인을 요청한다.
Services	Operation: requestStatus() Description: 자판기의 네트워크 상태를 확인을 요청한다.
Protocol	TCP/IP
Notes	Supported by RequestManager
Issues	

4. Dynamic Behavior Section

4.1 Scenario Section

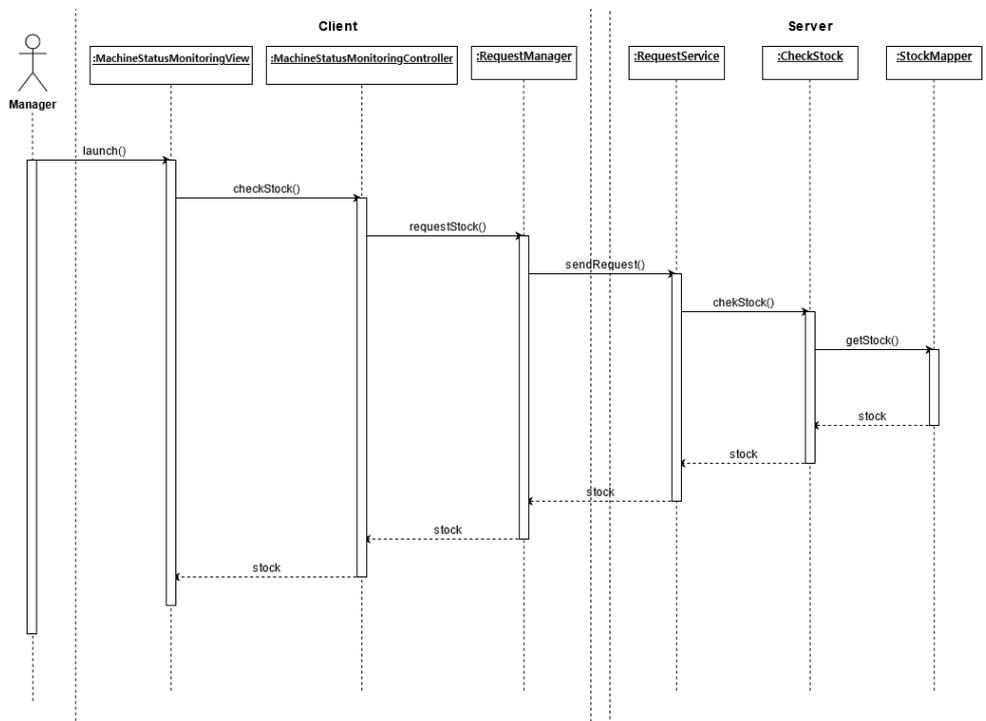
Component Interaction Model for Add Stock

Use Case	UC-1: 재고 추가
Description	재고를 추가하면 재고 상태를 업데이트 한다. 만약 품절된 음료의 재고를 추가하면 음료를 판매 가능 상태로 변경한다.
Actors	Manager, DVM
Components	:InformationView, :TrapReceiver, :AddStock, :StockMapper
Assumptions	
Steps	<ol style="list-style-type: none"> 1. 관리자가 재고 추가를 요청하면 :InformationView는 접근 토큰 입력을 요구한다. 관리자가 접근 토큰을 입력하면 재고 추가 화면으로 전환된다. 2. 관리자가 재고 추가를 요청하면 :TrapReceiver에 의해 이벤트가 큐에 입력된다. 3. :AddStock은 큐에서 이벤트를 꺼내 :StockMapper를 통해 재고 사항을 업데이트 한다.
Variations	



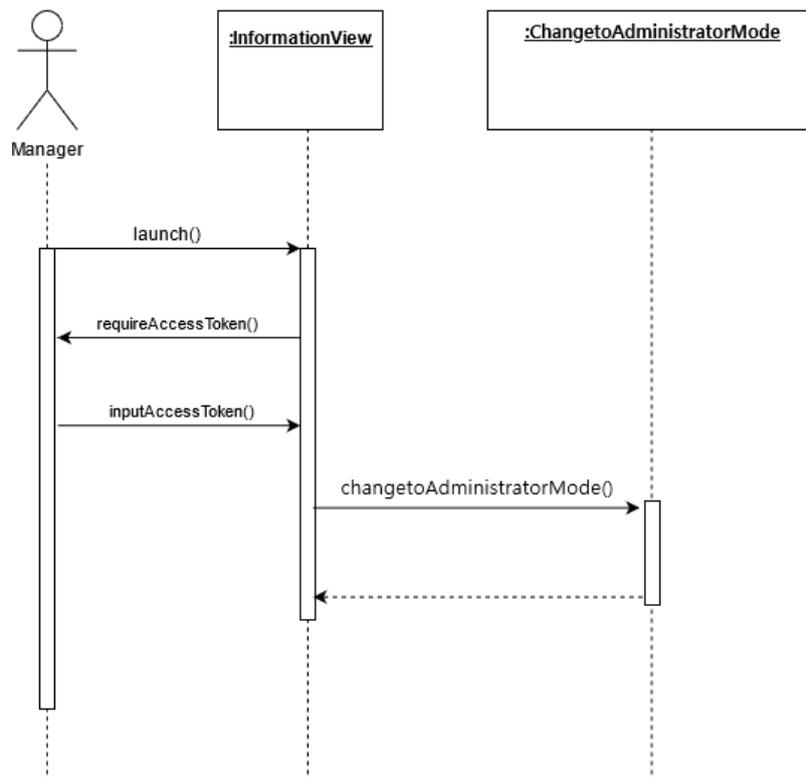
Component Interaction Model for Check Stock

Use Case	UC-2: 재고 상태 확인
Description	실제 업체에서 보충 가능한 재고의 양에 따라 일시 품질/품질의 구분이 가능해야 한다. 모든 자판기의 재고 확인이 가능해야 한다. 재고가 부족할 경우 알림이 와야 한다.
Actors	Manager, DVM Management System, DVM
Components	:MachineStatusMonitoringView, :MachineStatusMonitoringController, :RequestManager, :RequestService, :CheckStock, :StockMapper
Assumptions	
Steps	<ol style="list-style-type: none"> 1. 관리자는 :MachineStatusMonitoringView를 통해 특정 자판기의 재고 확인을 요청한다. 2. :MachineStatusMonitoringController와 :RequestManager를 통해 자판기에 서비스 요청을 보낸다. 3. :RequestManager는 요청받은 서비스를 처리하기 위해 :CheckStock, :StockMapper를 통해 현재 자판기의 재고 사항을 관리자 시스템에 반환한다.
Variations	



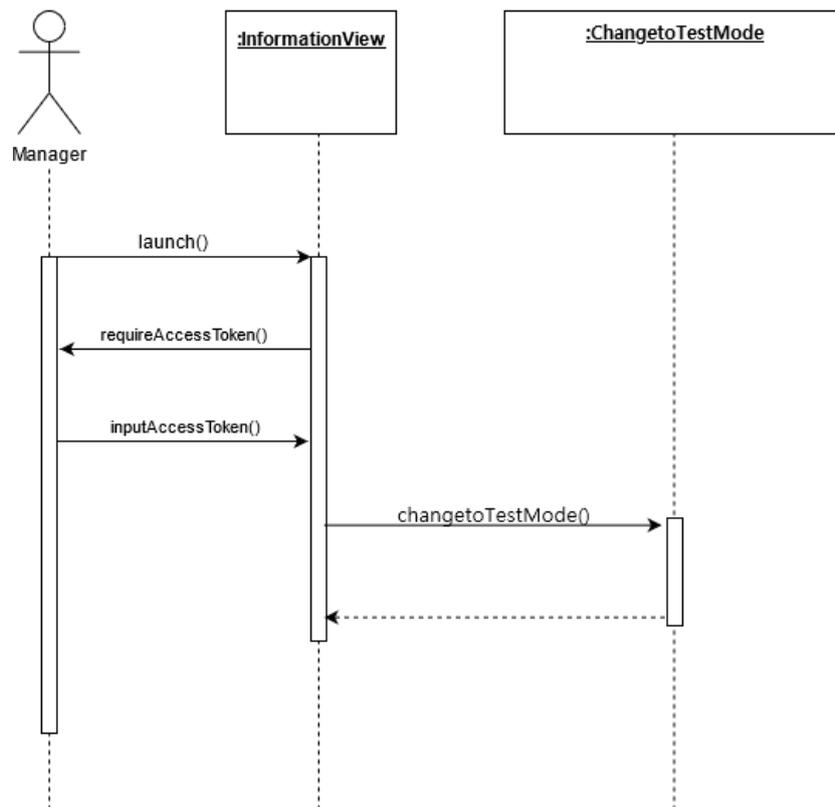
Component Interaction Model for Change to Administrator Mode

Use Case	UC-3: 관리자 모드 접근
Description	자판기의 제품 판매 상태를 활성화 혹은 비활성화를 할 수 있다. 제품 판매 기록을 확인할 수 있다.
Actors	Manager, DVM
Components	ChangetoAdministratorMode, launch, requireAccessToken, inputAccessToken
Assumptions	
Steps	<ol style="list-style-type: none"> 1. 관리자가 관리자 모드로 변환을 요청하면 :InformationView는 접근 토큰 입력을 요구한다. 2. 입력된 토큰을 확인하면 :ChangetoAdministratorMode는 관리자 모드로 변환한다.
Variations	



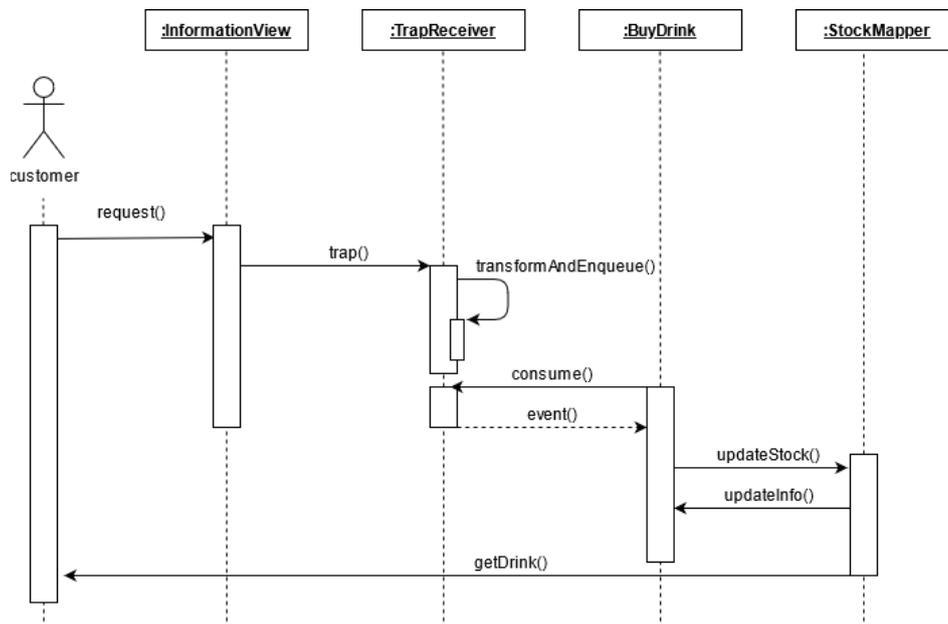
Component Interaction Model for Change to Test Mode

Use Case	UC-4: 테스트 모드 접근
Description	테스트를 위한 데모 시스템이 존재해야 한다. 테스트 모드에서는 제품 판매 상태 변경이 가능하며 결제를 자유롭게 수행할 수 있다.
Actors	Manager, DVM
Components	Lanuch, requireAccessToken, inputAccessToken, changetoTestMode
Assumptions	
Steps	<ol style="list-style-type: none"> 관리자가 테스트 모드로 변환을 요청하면 :InformationView는 접근 토큰 입력을 요구한다. 입력된 토큰을 확인하면 :ChangetoTestMode는 관리자 모드로 변환한다.
Variations	



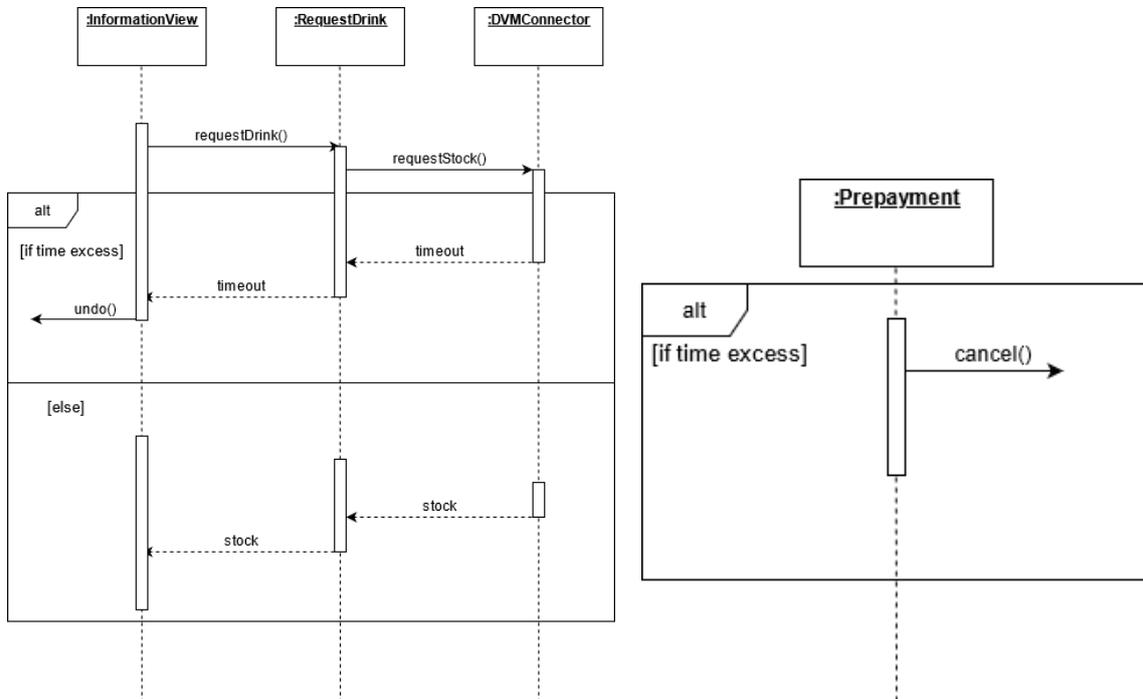
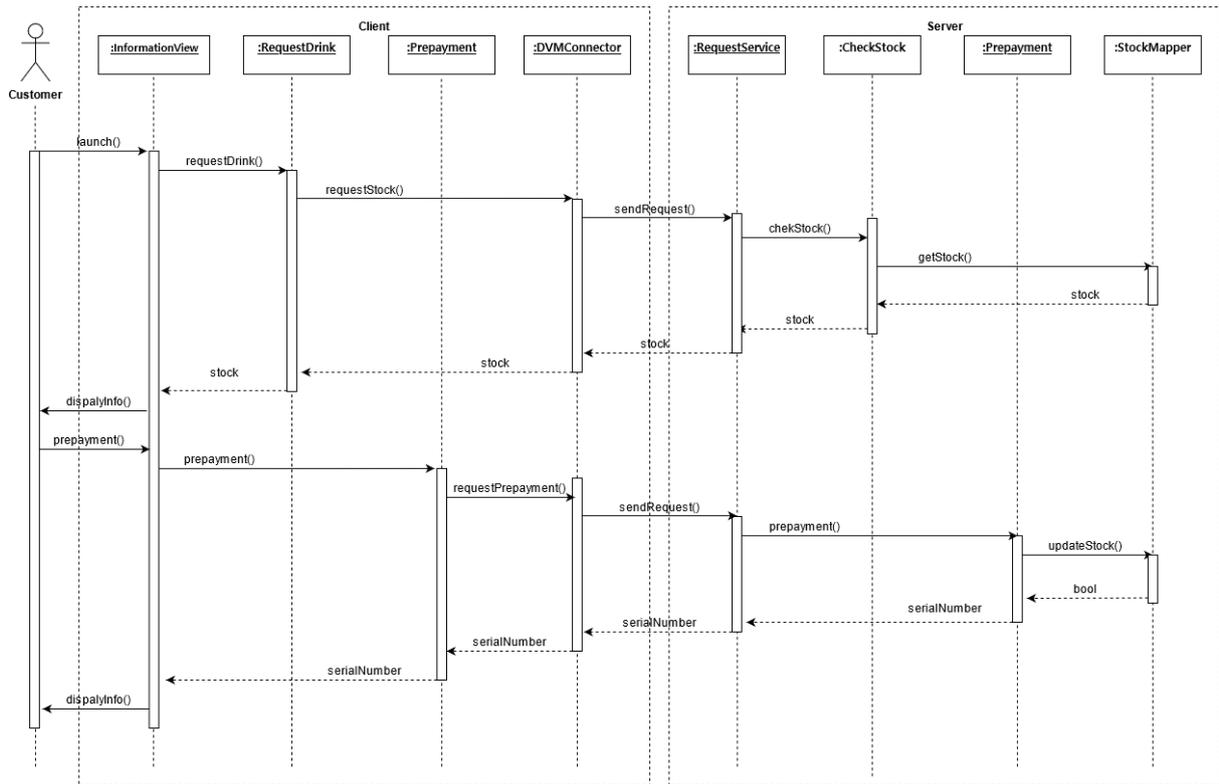
Component Interaction Model for Buy Drink

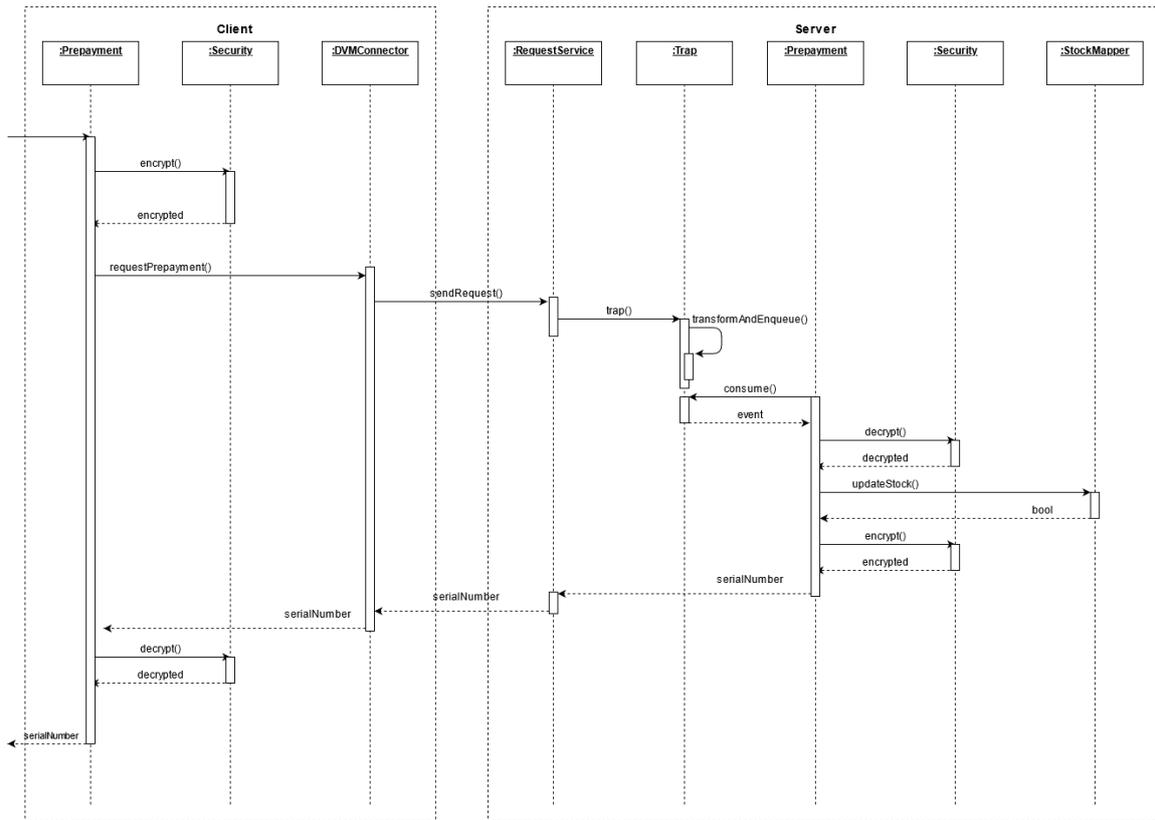
Use Case	UC-5: 음료 구매
Description	사용자는 자판기의 화면을 보고 음료를 카드로 결제한다. 잔액이 부족한 경우 결제되지 않는다.
Actors	Customer, DVM
Components	:InformationView, :TrapReceiver, :BuyDrink, :StockMapper
Assumptions	
Steps	<ol style="list-style-type: none"> 1. 사용자가 화면에서 원하는 음료와 수량을 선택하여 음료 구매를 요청한다. 2. :InformationView는 사용자가 발생시킨 이벤트를 :TrapReceiver를 통해 메시지 큐에 추가한다. 3. :TrapReceiver는 발생한 이벤트를 음료 구매에 해당하는 큐에 추가한다. 4. :BuyDrink는 새로 발생한 이벤트를 큐에서 확인하고 :StockMapper에 음료 재고 업데이트를 요청한다. 5. :StockMapper는 사용자가 구매한 개수만큼 재고를 감소시키고 사용자에게 음료를 반환한다.
Variations	



Component Interaction Model for Prepayment

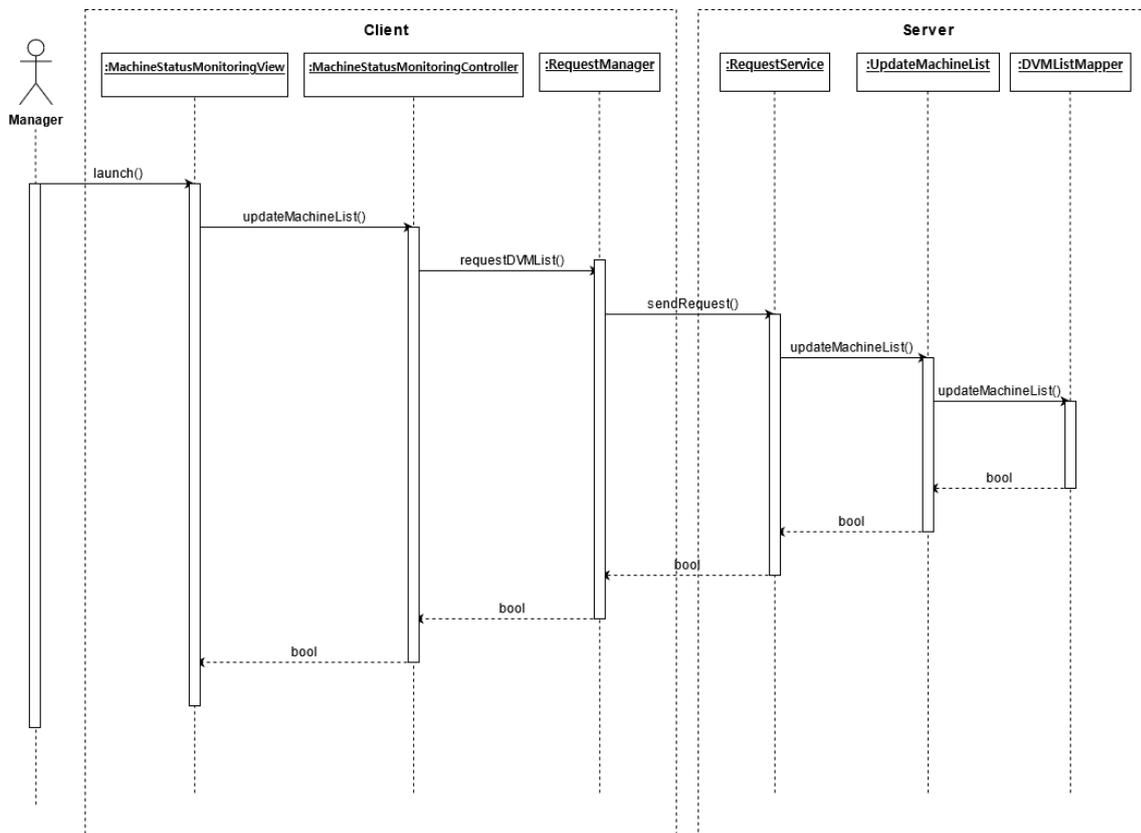
Use Case	UC-6: 선결제 구매
Description	만약 다른 자판기에 있는 음료를 구매할 경우 해당 자판기에서 결제 후 인증 코드를 발급한다. 발급받은 인증 코드를 입력하면 음료를 받을 수 있다.
Actors	Customer, DVM
Components	:InformationView, :RequestDrink, :Prepayment, :DVMConnector, :RequestService, :CheckStock, :TrapReceiver, :Security
Assumptions	
Steps	<ol style="list-style-type: none"> 1. 사용자가 자판기에 없는 음료 구매를 요청하면, :InformationView는 주변 자판기의 재고 사항을 알아내기 위해 :ReqeustDrink를 호출한다. :ReqeustDrink는 :DVMConnector를 통해 다른 자판기에 서비스 요청을 보낸다. 2. 서비스 요청을 받은 자판기는 :RequestService를 통해 서비스를 해석하고 :CheckStock을 호출하여 재고 사항 확인을 요청한다. :CheckStock은 :StockMapper를 호출하여 재고사항을 반환한다. 재고 사항을 반환 받은 :InformationView는 사용자에게 자판기의 위치와 재고 사항을 보여주고 선결제를 진행할 것인지 물어본다. 3. 사용자가 선결제를 요청하면 :Prepayment는 :Security를 통해 결제정보를 암호화 하여 :DVMConnector를 통해 해당 자판기에 선결제를 요청한다. 4. 서비스 요청을 받은 자판기는 :RequestService를 통해 서비스를 해석하고 :TrapReceiver를 통해 큐에 이벤트를 삽입한다. :Prepayment는 큐에 있는 이벤트를 읽어 와 :Security를 통해 결제 정보를 복호화 한다. 결제 정보를 확인하면 :StockMapper를 호출하여 재고 사항을 업데이트 하고 serialNumber를 생성한다. 생성한 SerialNumber를 암호화 하여 선결제를 요청한 자판기에 반환한다. 5. serialNumber를 받은 자판기는 :Security를 통해 복호화 하여 사용자에게 출력한다.
Variations	<p>#2: 일정 시간까지 요청한 답이 오지 않으면 자판기는 처음 화면으로 되돌아간다.</p> <p>#5. 일정 시간까지 사용자가 발급받은 시리얼번호를 사용하지 않으면 결제를 취소한다.</p>





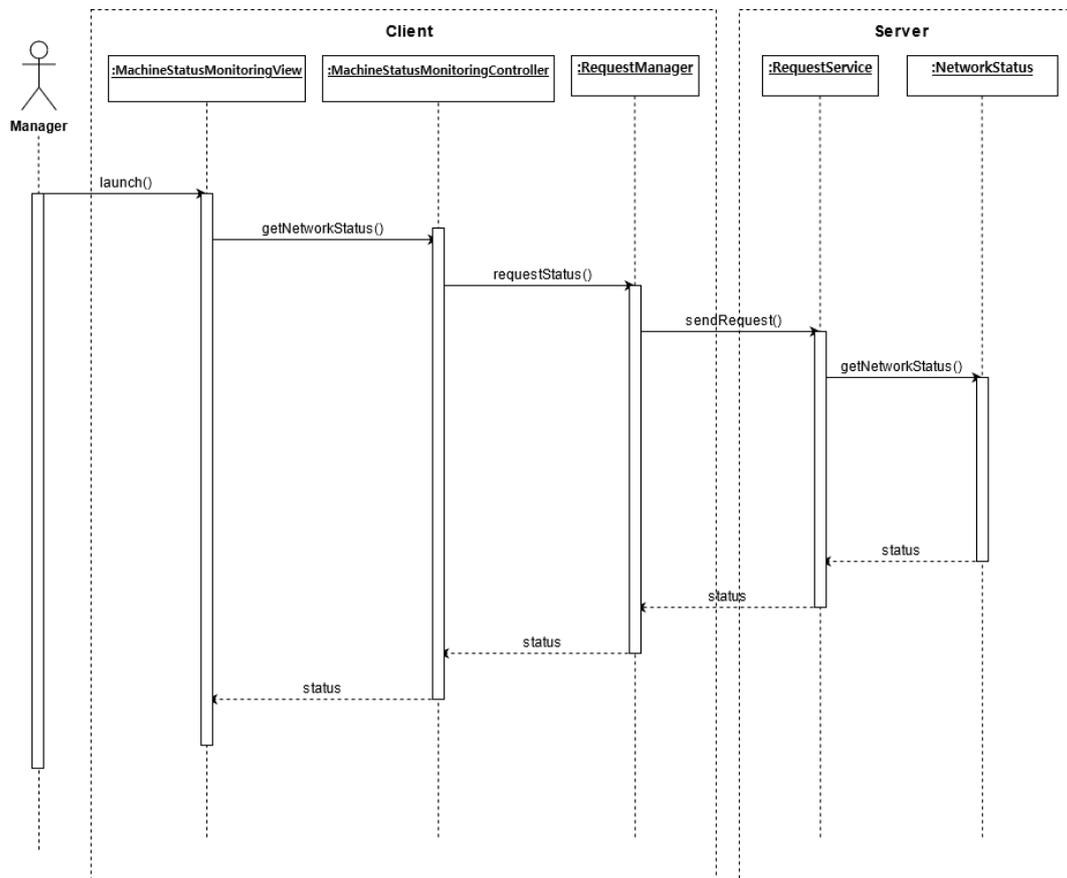
Component Interaction Model for Update Machine List

Use Case	UC-7: 새로운 자판기 추가, 삭제
Description	자판기를 추가 혹은 삭제를 하면 주변 자판기에 상태를 업데이트 한다.
Actors	Manager, Management System, DVM
Components	:MachineStatusMonitoringView, :MachineStatusMonitoringController, :RequestManager, :RequestService, :UpdateMachineList, :DVMListMapper
Assumptions	
Steps	<ol style="list-style-type: none"> 1. 매니저가 매니지먼트 시스템을 통해 자판기 추가 혹은 삭제를 요청하면 :MachineStatusMonitoringView는 MachineStatusMonitoringController, :RequestManager를 통해 모든 자판기들에 자판기 리스트 업데이트 요청을 보낸다. 2. 서비스 요청을 받은 자판기는 :RequestService, :UpdateMachineList, :DVMListMapper를 거쳐 자판기 리스트를 업데이트 한다.
Variations	



Component Interaction Model for Check Network Status

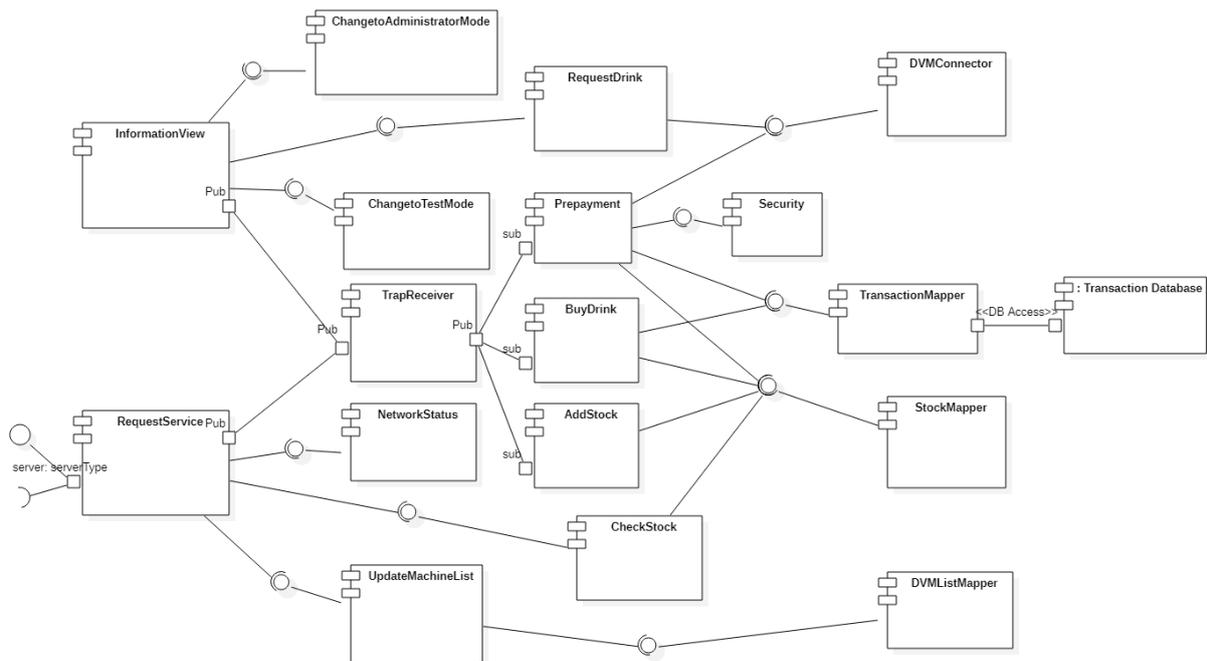
Use Case	UC-8: 네트워크 점검
Description	네트워크 관리자가 특정 자판기에 상태 확인 요청을 할 경우 자판기는 상태 메시지를 전송한다.
Actors	Manager, Management System, DVM
Components	:MachineStatusMonitoringView, :MachineStatusMonitoringController, :RequestManager, :RequestService, :NetworkStatus
Assumptions	
Steps	<ol style="list-style-type: none"> 매니저가 자판기의 네트워크 상태 확인을 요청하면 :MachineStatusMonitoringView는 MachineStatusMonitoringController, :RequestManager를 통해 자판기에 네트워크 상태 확인 서비스를 요청한다. 서비스 요청을 받은 자판기는 :RequestService, :NetworkStatus를 통해 자판기의 네트워크 상태를 반환한다.
Variations	



5. Other View Section

5.1 Component View

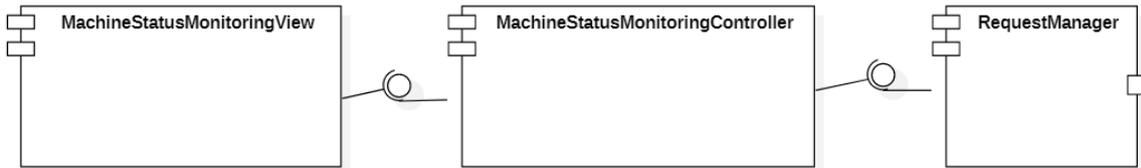
5.1.1 Component Diagram of DVM System



Element	Properties
InformationView	자판기를 사용하는 사용자에게 화면을 표시한다.
RequestService	다른 자판기 혹은 관리 시스템으로부터 받은 요청을 처리한다.
AddStock	재고를 추가한다.
BuyDrink	음료수를 구매 서비스를 제공한다.
RequestDrink	다른 자판기에 음료 사항 확인을 요청한다.
ChangetoAddmistratorMode	관리자 모드로 전환한다.
ChangetoTestMode	테스트 모드로 전환한다.
CheckStock	재고를 확인한다.
Prepayment	선결제를 지원한다.
NetworkStatus	현재 자판기의 네트워크 상황을 확인한다.
UpdateMachineList	자판기 리스트를 업데이트 한다.
TrapReceiver	요청 사항을 메시지 큐에 추가한다.
Security	결제 정보 암호화를 위한 기능을 제공한다.
StockMapper	재고 사항을 관리한다.

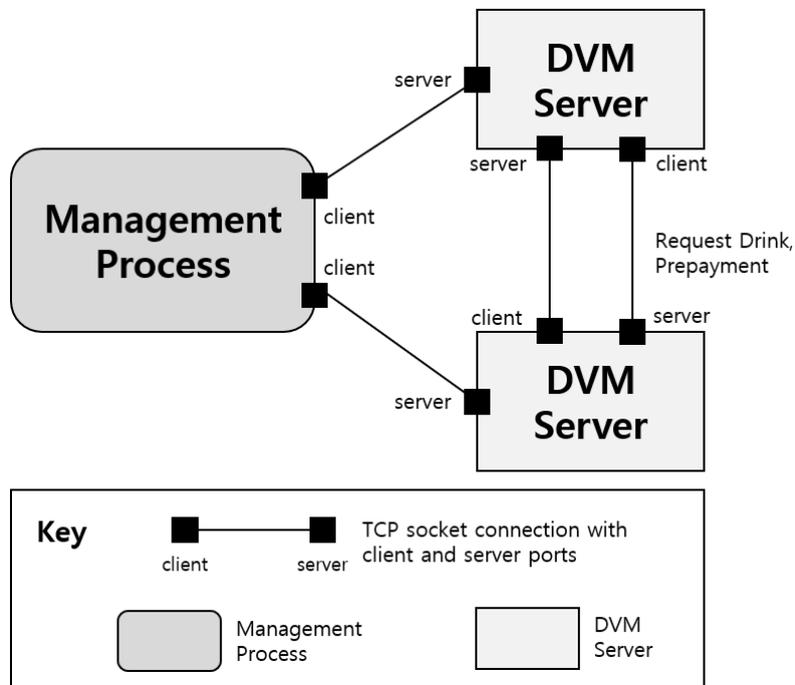
TransactionMapper	판매 사항을 업데이트한다.
DVMConnector	다른 자판기에 서비스를 요청한다.
DVMListMapper	자판기 리스트를 관리한다.

5.1.2 Component Diagram of DVM Management System



Element	Property
MachineStatusMonitoringView	DVM 관리자와 상호작용 한다.
MachineStatusMonitoringController	관리자의 요청 사항을 처리한다.
RequestManager	요청 사항을 DVM 시스템으로 전달한다.

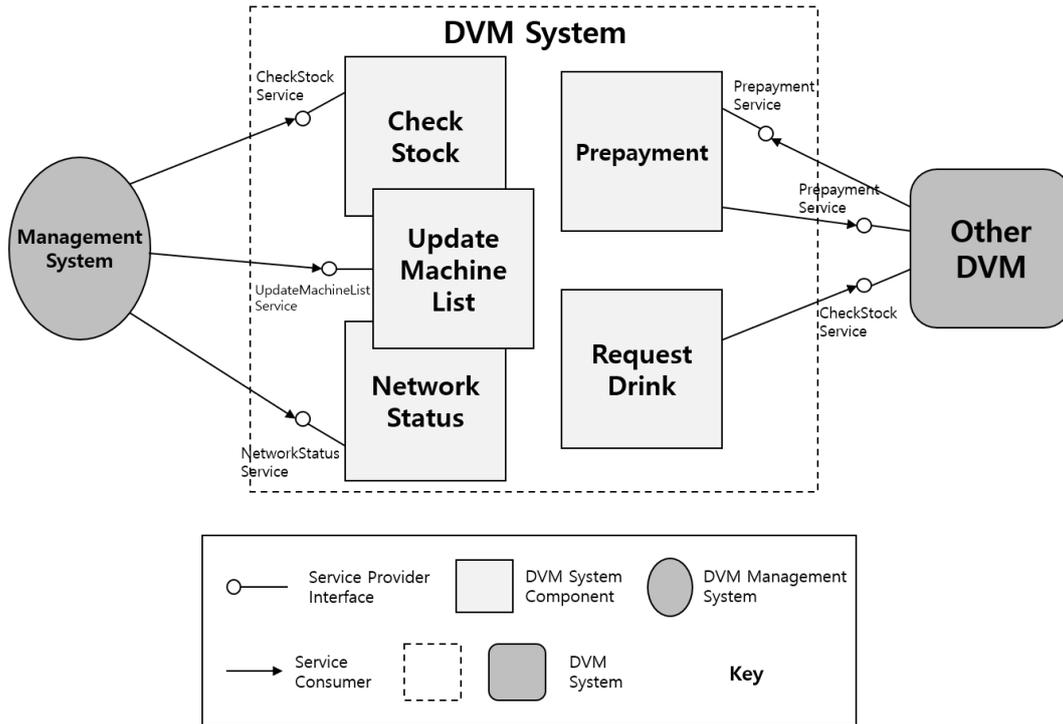
5.1.3 DVM and DVM Management System C&C View with Client-Server Style



Element	Property
---------	----------

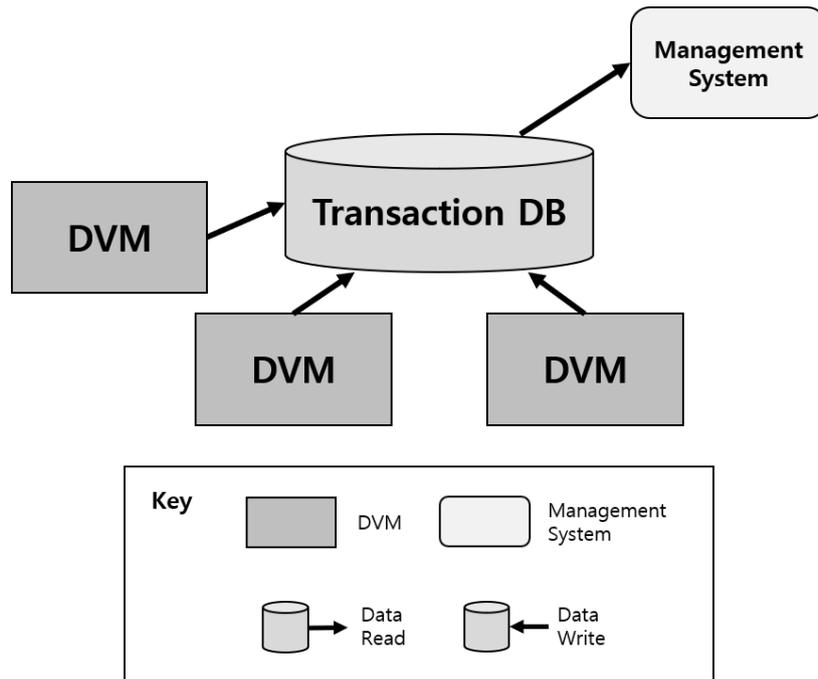
Management Process	DVM 시스템을 관리하기 위한 매니지먼트 프로세스이다. DVM과 통신하며 이때 매니지먼트 시스템은 클라이언트이다.
DVM Server	DVM에서 작동하며 DVM 서비스를 제공하는 서버이다. 매니지먼트 시스템과 다른 DVM에 서비스를 제공할 때는 서버 역할이며 다른 DVM에 서비스를 요청하면 클라이언트로 동작한다.

5.1.4 DVM System C&C View with Service-Oriented Architecture Style



Element	Property
Check Stock	매니지먼트 시스템에 재고 확인을 위한 서비스를 제공한다.
UpdateMachineList	매니지먼트 시스템에 자판기 리스트 업데이트를 위한 서비스를 제공한다.
Network Status	매니지먼트 시스템에 네트워크 상황 확인을 위한 서비스를 제공한다.
Prepayment	다른 자판기에 선결제 서비스를 제공하거나 이용한다.
RequestDrink	다른 자판기에 재고 확인을 위한 서비스를 요청한다.

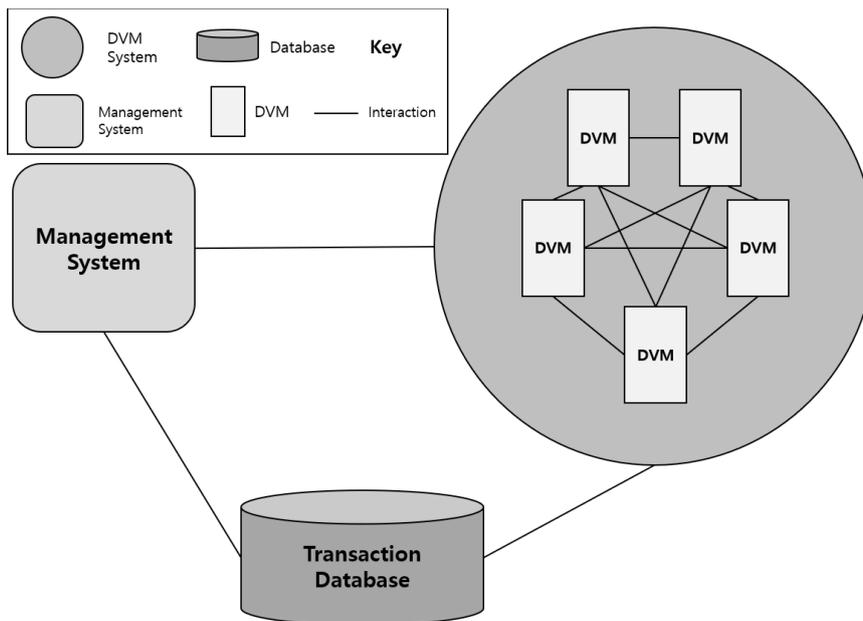
5.1.5 DVM, DVM Management System and Database C&C View with Shared-Data Style



Element	Property
Transaction DB	모든 거래 내역을 저장하는 데이터베이스이다.

5.2 Physical View

5.2.1 Deployment Diagram



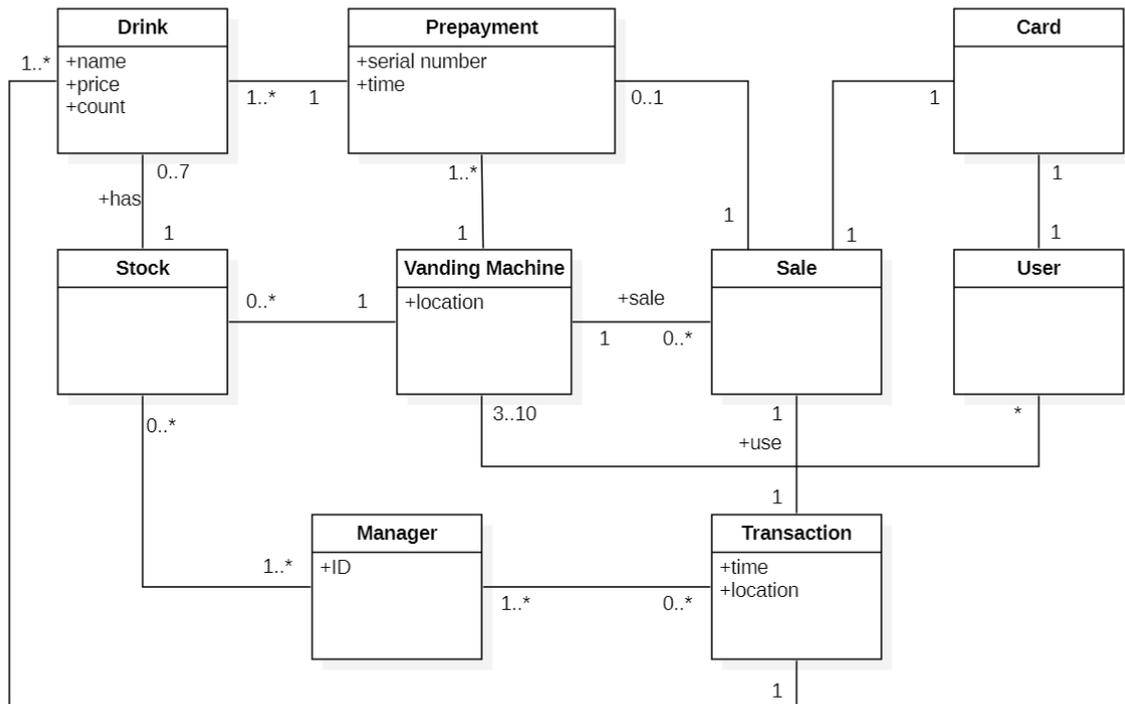
Element	Property
Management System	DVM System을 관리한다.
DVM System	자판기들이 연결되어 DVM System을 구축한다. 연결된 자판기들은 선결제 구매와 같은 기능을 제공한다.
Transaction Database	모든 자판기에서 발생한 거래 내역을 저장한다.

Design Decisions and Location	Rationale
물리적 구조로 Three-tier deployment pattern을 사용한다.	Con-2 를 만족시키기 위해 외부 데이터베이스가 필요하며 UC-2 를 위해 외부에서 접속 가능한 클라이언트가 필요하다. DVM은 분산 시스템이기 때문에 여러 web/app tier가 존재해야 CRN-3, QA-6, uc-2를 만족시키기 위해 별도의 클라이언트가 필요하다.

6. Conceptual Framework Section

6.1 Domain Model

6.1.1 Domain Model of DVM System



6.1.2 Domain Lexicon

Lexicon	Description
Vending Machine	음료를 판매하는 자판기이다. 자판기는 사용자에게 직접 음료를 판매하기도 하고 다른 자판기의 음료를 미리 결제하는 선결제 시스템도 제공한다. 모든 자판기는 네트워크로 연결되어 있다.
User	자판기를 이용하는 사용자이다. 사용자는 자판기의 음료를 구매할 수 있으며 다른 자판기의 음료를 선결제 가능하다.
Manager	자판기를 관리하는 관리자이다. 관리자는 모든 자판기들에 네트워크로 접근하여 자판기의 재고 사항과 네트워크 상태를 확인 가능하며 모든 거래 내역을 볼 수 있다.
Card	사용자가 음료를 구매하는 수단인 카드이다. 자판기의 선결제 시스템은 일정 시간 이상 사용자가 음료를 교환하지 않을 경우 환불이 진행되므로 음료 구매는 카드 사용을 우선으로 한다.

Sale	음료의 구매를 나타낸다. 사용자는 카드를 이용하여 음료를 구매하고 모든 음료 구매는 즉시 거래 내역으로 저장된다.
Transaction	음료 구매 내역이다. 사용자가 자판기에서 직접 음료를 구매하거나 선결제를 진행하면 거래 내역이 발생하여 저장된다.
Drink	자판기에서 판매하는 음료이다. 음료의 종류는 총 20가지가 있으며 한 자판기당 보유할 수 있는 음료의 최대 종류는 7가지이다.
Stock	자판기의 재고 상황을 나타낸다. 모든 자판기는 각자의 재고 상황을 저장하며 서로 확인 가능하다.
Prepayment	사용자가 구매를 원하는 음료가 현재 자판기에 없을 경우 자판기는 다른 자판기의 음료를 구매할 수 있는 선결제 기능을 지원한다. 사용자가 다른 자판기의 음료를 구매할 경우 해당 자판기에서 사용 가능한 시리얼 번호를 받게 된다. 선결제 후 일정 시간동안 음료를 교환하지 않으면 구매가 취소된다.

7. Conclusion Section

7.1 Assessment

7.1.1 Chosen Architecture and Assessment

Chosen Architecture	Advantage	Limitation
Service Application Reference Architecture	클라이언트와 서버가 loosely coupled 되어 있어 분산 시스템에서 사용이 적합하다. Message Broadcasting으로 재고 확인을 하는 DVM과 맞게 메시지 기반으로 통신이 가능하며 관리 시스템과 같은 다른 시스템과도 동작 가능하다(Interoperability).	UI 지원이 부족하여 Presentation Layer를 추가하여 사용자와의 상호작용 방법을 추가해야 한다.
Rich Client Reference Architecture	관리자가 자판기의 재고 수량을 확인하기 쉽게 풍부한 UI 환경을 제공한다. 또한 자판기와 네트워크 통신을 통해 재고 사항을 확인할 수 있다. 데이터베이스에 저장되어 있는 데이터를 읽어와 판매 기록 및 분석을 확인할 수 있다.	전용 클라이언트가 필요하며 모바일 환경에서 사용이 불가능하다.

7.1.2 Non-functional Requirements and Assessment

ID	Quality Attribute	Scenario
QA-1	Availability	만약 주변 자판기의 재고 확인이 10초 이상 걸릴 경우 구매 목록 화면으로 돌아간다.
QA-2	Availability	인증 코드 발급 시 해당 자판기에 코드 발급을 확인한 후 사용자에게 코드를 발급한다. 인증 코드의 유효 시간은 10분이며 넘어갈 경우 결제 취소가 진행된다.
QA-4	Performance	현재 자판기에 재고가 없을 경우 항상 구매 가능한 자판기를 찾아 출력해야 한다. 만약 해당 자판기가 멀리 떨어져 있더라도 결과는 10초 안에 출력 되어야 한다.
QA-7	Security	모든 결제 정보는 암호화가 되어 전송 및 저장이 되어야 한다.
QA-10	Usability	동시에 여러 사용자가 한 자판기의 음료 구매를 원할 경우에도 항상 정확한 판매 수량을 보여주어야 한다.

Design Decision and Location	Rationale and Assumption
Request, Prepayment 컴포넌트에 Timestamp 기법을 적용한다. (QA-1, 2, 4)	Request, Prepayment 컴포넌트에 Timestamp 기법을 적용하여 정해진 시간보다 요청 응답이 길어질 경우 요청을 취소시키고 이전 상태로 되돌린다.
Security 컴포넌트를 추가하여 Encrypt Data 기법을 적용시킨다. (QA-7)	BuyDrink, Prepayment 컴포넌트에서 사용자가 결제 정보를 입력하였을 때 Security 컴포넌트를 통하여 결제 정보를 암호화 및 복호화 한다.
StockMapper 컴포넌트를 호출하기 전 항상 모든 컴포넌트들은 TrapReceiver를 호출하여 Message queue, Prioritize events 기법이 적용되게 한다. (QA-10)	음료 구매 및 추가 상황에서 항상 정확한 재고 사항을 보여주기 위해 Message queue와 Prioritize events 기법을 적용한다. 선결제 요청이 올 경우 결제 요청을 받은 순서대로 처리하며 현장에서 구매 및 재고 추가는 항상 우선적으로 처리한다. 단순 재고 확인에는 적용하지 않는다.

7.1.3 Viewpoints

Selected style	Rationale	Section
Module View with Decomposition, Uses, Layered Style	DVM 시스템의 전체적인 구성을 나타내기 위해 Decomposition Style을 사용하였다. DVM 시스템은 Presentation, Service, Business, Data 레이어로 구성되어 있다. 이를 표현하기 위해 Layered Style을 사용하였다. DVM 시스템의 모듈들은 서로를 사용한다. 이를 나타내기 위해 Uses Style을 사용하였다.	3.1.1
Module View with Decomposition, Uses, Layered Style	DVM 매니지먼트 시스템의 전체적인 구성을 나타내기 위해 Decomposition Style을 사용하였다. DVM 매니지먼트 시스템은 Presentation, Business, Data 레이어로 구성되어 있다. 이를 표현하기 위해 Layered Style을 사용하였다. DVM 매니지먼트 시스템의 모듈들은 서로를 사용한다. 이를 나타내기 위해 Uses Style을 사용하였다.	3.1.2
C&C View with Publisher-Subscribe Style	DVM 시스템의 컴포넌트간 호출 관계를 표현하기 위해 C&C View를 선택했다. 재고 사항에 변경이 생길 경우 정확한 재고 사항을 나타내기 위해 메시지 큐를 사용한다. 이를 C&C View에서 표현하기 위해 Publisher-Subscribe Style을 사용하였다.	5.1.1
C&C View	DVM 매니지먼트 시스템의 컴포넌트간 호출 관계를 표현하기 위해 C&C View를 선택했다.	5.1.2

C&C View with Client-Server Style	DVM 매니지먼트 시스템은 DVM 시스템에 여러가지 서비스를 요청한다. 이 때 매니지먼트 시스템은 클라이언트 DVM 시스템은 서버역할을 하게 된다. 또한 DVM 시스템은 서비스 요청과 처리 사항에 따라 클라이언트 혹은 서버가 된다. 이를 표현하기 위해 Client-Server Style을 사용하였다.	5.1.3
C&C View with Service-Oriented Architecture Style	DVM 시스템은 매니지먼트 시스템과 다른 DVM 시스템에 여러가지 서비스를 제공한다. 또한 다른 DVM 시스템에 서비스를 요청하기도 한다. 이를 표현하기 위해 Service-Oriented Architecture Style을 사용하였다.	5.1.4
C&C View with Shared Data Style	DVM 시스템은 거래가 발생하면 거래 내역을 Transaction 데이터베이스에 저장하고 매니지먼트 시스템은 모든 자판기에서 발생한 거래 내역을 읽을 수 있다. 이를 표현하기 위해 Shared Data Style을 사용하였다.	5.1.5
Allocation View with Deployment Style	Deployment Pattern으로 Three-tier deployment pattern을 사용하였다. 이를 나타내기 위해 Deployment Style을 사용하였다.	5.2.1