

DVM 시스템

Architecture Description

-SEI Style-

소프트웨어공학특강

-팀 1-

윤상혁, 최슬아, 루카이

목차

| | |
|--|----|
| Stakeholder/View Table..... | 3 |
| Part 1. DVM System Overview through Module View | 4 |
| Part 2. DVM Management System Overview through Module View | 10 |
| Part 3. DVM System Overview through Component and Connector View | 13 |
| Part 4. DVM Management System Overview through Component and Connector View | 21 |
| Part 5. Interaction Diagram of DVM and DVM Management System through Component & Connector View | 23 |
| Part 6. Allocation View | 29 |
| Part 7. Documentation Beyond Views | 32 |

Stakeholder/View Table

Stakeholder/View Table

| | Module Views | | | C&C Views | | | | Allocation Views |
|----------|---------------|------|---------|-------------------|---------------|------------------|-------------|------------------|
| | Decomposition | Uses | Layered | Publish-Subscribe | Client-Server | Service-oriented | Shared-Data | Deployment |
| 판매 담당자 | o | o | o | o | s | o | s | s |
| 사용자 | | | | | | | | |
| 재고관리 담당자 | | | | | o | | | o |
| 개발자 | d | d | D | d | d | d | d | d |
| 네트워크 관리자 | s | s | D | s | d | d | d | d |
| 테스터 | s | s | s | s | s | s | s | s |

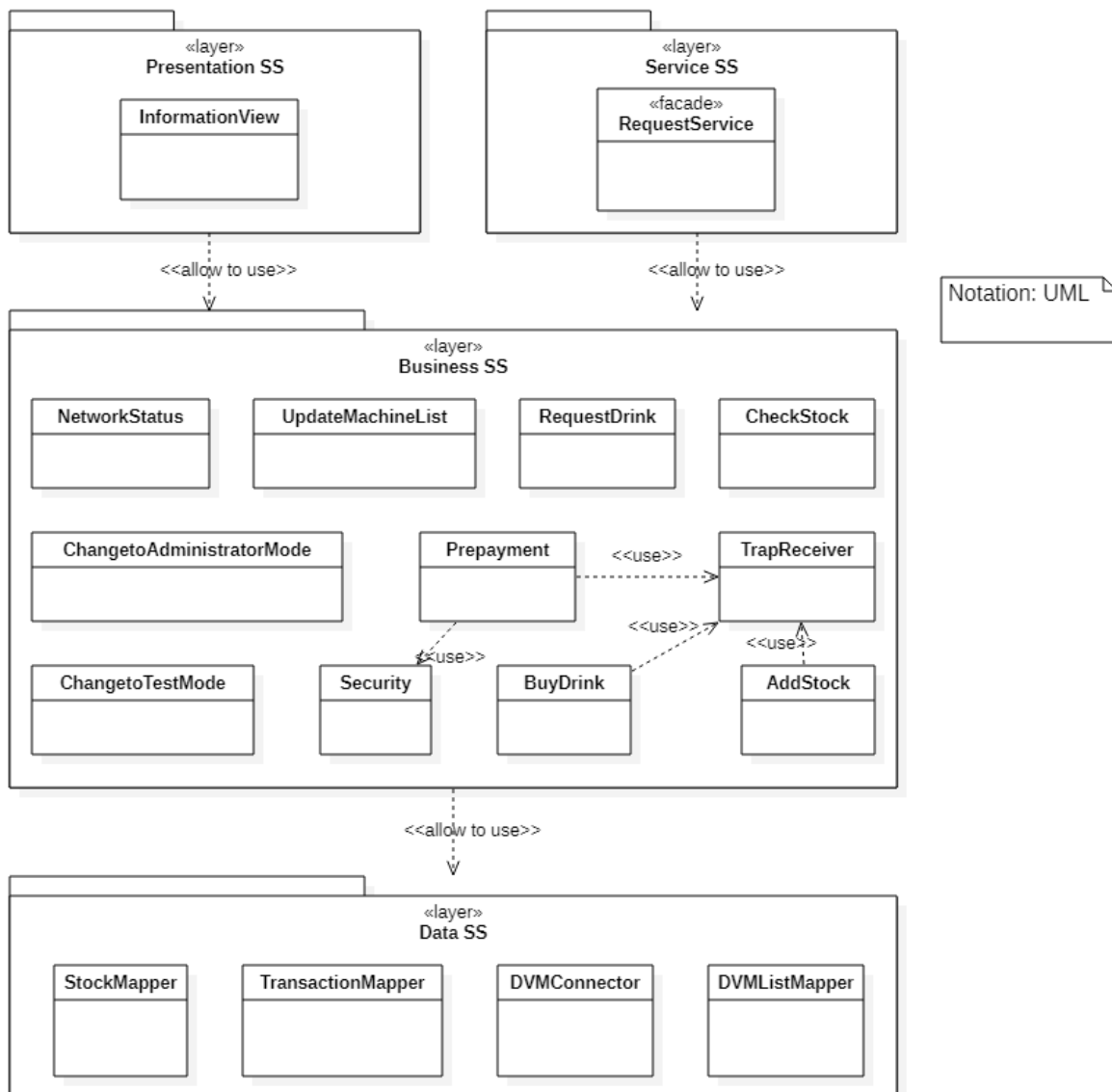
Key: d = detailed information, s = some details, o = overview Information, x = anything

Part 1.

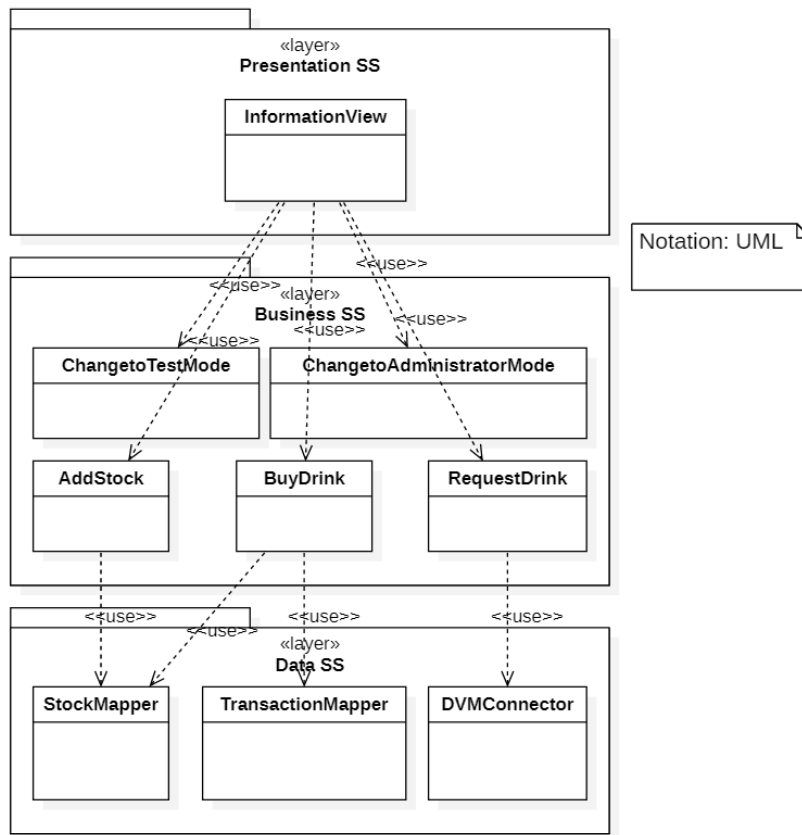
DVM System Overview through Module View

Section 1. Primary Presentation

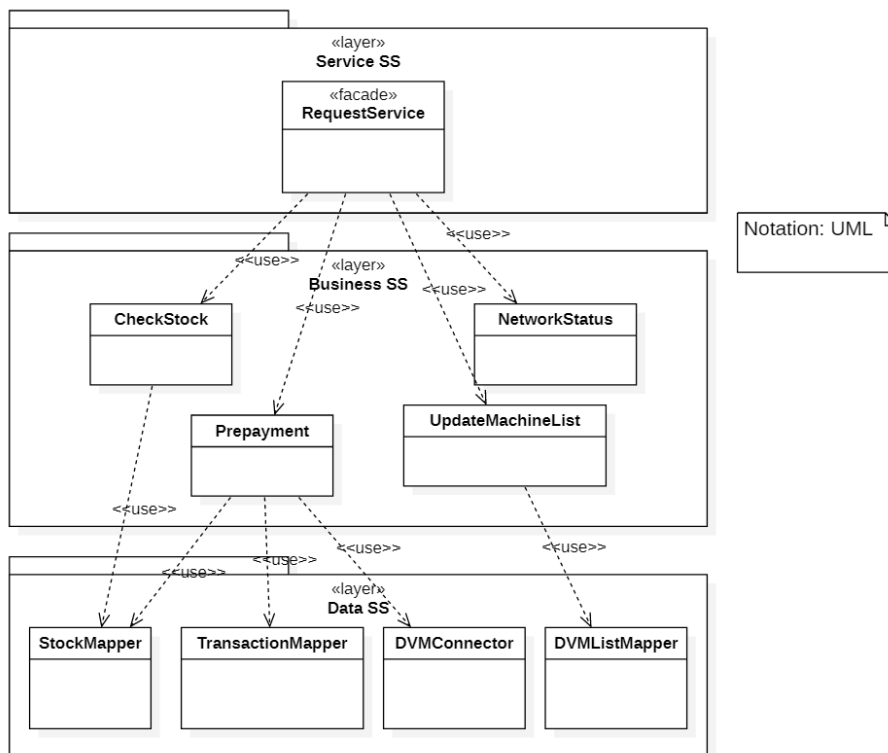
1.1.1 Primary presentation of DVM System



1.1.2 Use style view in terms of InformationView module



1.1.3 Use style view in terms of RequestService module



Section 2. Element Catalog

1.2.1 Element and their properties

| Layer | Element | Properties |
|--------------|---------------------------|-----------------------------------|
| Presentation | InformationView | 자판기를 사용하는 사용자에게 화면을 표시한다. |
| Service | RequestService | 다른 자판기 혹은 관리 시스템으로부터 받은 요청을 처리한다. |
| Business | AddStock | 재고를 추가한다. |
| | BuyDrink | 음료수를 구매 서비스를 제공한다. |
| | RequestDrink | 다른 자판기에 음료 사항 확인을 요청한다. |
| | ChangetoAdministratorMode | 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. | |
| Data | StockMapper | 재고 사항을 관리한다. |
| | TransactionMapper | 판매 사항을 업데이트한다. |
| | DVMConnector | 다른 자판기에 서비스를 요청한다. |
| | DVMListMapper | 자판기 리스트를 관리한다. |

1.2.2 Relations and their properties

| Element (caller) | Element (callee) | Relation and Property |
|------------------------------------|---|--|
| Prepayment | Security | 다른 자판기에 결제 정보를 송수신 하기 위해 결제정보를 암호화 및 복호화 한다. |
| Prepayment BuyDrink AddStock | TrapReceiver | 요청받은 이벤트들을 메시지 큐로부터 읽어 오기 위해 TrapReceiver를 호출한다. |
| InformationView | ChangetoAdministratorMode ChangetoTestMode AddStock BuyDrink RequestDrink | 자판기 사용자로부터 입력 받은 이벤트를 수행하기 위해 알맞은 모듈들을 호출한다. |

| | | |
|--|--|---|
| RequestService | CheckStock NetworkStatus Prepayment UpdateMachineList | 다른 자판기 혹은 매니지먼트 시스템으로부터 받은 요청사항을 처리하기 위해 알맞은 모듈들을 호출한다. |
| AddStock BuyDrink CheckStock Prepayment | StockMapper | 재고 사항 확인 혹은 음료 구매, 추가로 인한 재고 사항 변경이 생겼을 때 재고 정보를 업데이트 하기 위해 호출한다. |
| BuyDrink Prepayment | TransactionMapper | 음료를 판매하여 발생한 거래 내역을 업데이트 한다. |
| RequestDrink Prepayment | DVMConnector | 다른 자판기에 서비스 요청을 하기 위해 호출한다. |
| UpdateMachineList | DVMListMapper | 자판기 목록을 업데이트하기 위해 호출한다. |

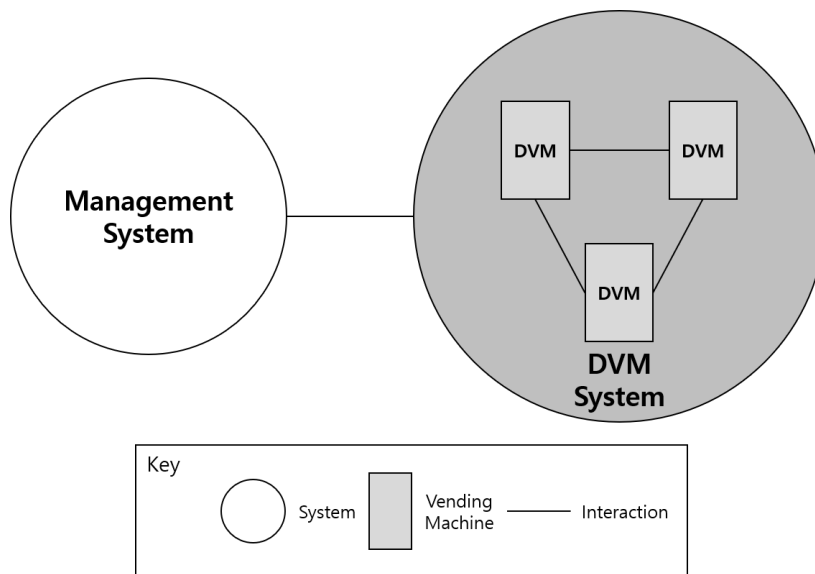
1.2.3 Element interfaces

| Element | Interface | Description |
|--------------------------|---------------------------|----------------------------|
| InformationView | requireAccessToken | 관리자 확인을 위해 접근 토큰 입력을 요구한다. |
| | diplayInfo | 업데이트된 화면 정보를 출력한다. |
| | request | 사용자가 요청한 작업을 처리한다. |
| TrapReceiver | Trap | 새로운 이벤트를 입력 받는다. |
| | transformAndEnqueue | 입력 받은 이벤트를 올바른 큐에 넣는다. |
| | consume | 메시지 큐의 이벤트를 반환한다. |
| StockMapper | updateStock | 재고 사항을 업데이트 한다. |
| TransactionMapper | registerTransaction | 새로운 판매 기록을 등록한다. |
| RequestMapper | sendRequest | 외부 요청 사항을 처리한다. |
| CheckStock | checkStock | 재고 사항을 확인한다. |
| ChangetoAddmistratorMode | changetoAdministratorMode | 관리자 모드로 변환한다. |
| ChangetoTestMode | changetoTestMode | 테스트 모드로 변환한다. |
| RequestDrink | requestDrink | 다른 자판기에 음료 재고 사항 확인을 요청한다. |
| Prepayment | requestPrepayment | 다른 자판기에 선결제를 요청한다. |

| | | |
|-------------------|-------------------|----------------------------|
| | prepayment | 다른 자판기로부터 요청받은 선 결제를 처리한다. |
| Security | encrypt | 결제 정보를 암호화한다. |
| | decrypt | 결제 정보를 복호화한다. |
| UpdateMachineList | updateMachineList | 등록된 자판기 리스트를 업데이트 한다. |
| DVMListMapper | updateMachineList | 등록된 자판기 리스트를 업데이트 한다. |
| NetworkStatus | getNetworkStatus | 현재 자판기의 네트워크 상태를 반환한다. |

Section 3. Context Diagram

1.3.1 Context Diagram



Section 4. Variability Guide

Section 5. Rationale

1.5.1 Design decision for DVM System and Rationale

| Design Decisions and Location | Rationale |
|---|---|
| DVM을 위한 논리적 구조로 Service Application Reference | UI 지원은 없지만 클라이언트와 서버가 loosely coupled 되어 있어 분산 시스템에서 사용이 적합하다. Message Broadcasting |

| | |
|-----------------------------|---|
| Architecture 를 사용한다. | 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. |
|-----------------------------|---|

1.5.2 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 UI보단 자판기간의 연결성이 더 중요하므로 선택하지 않았다. |

1.5.3 Selected Style and Rationale

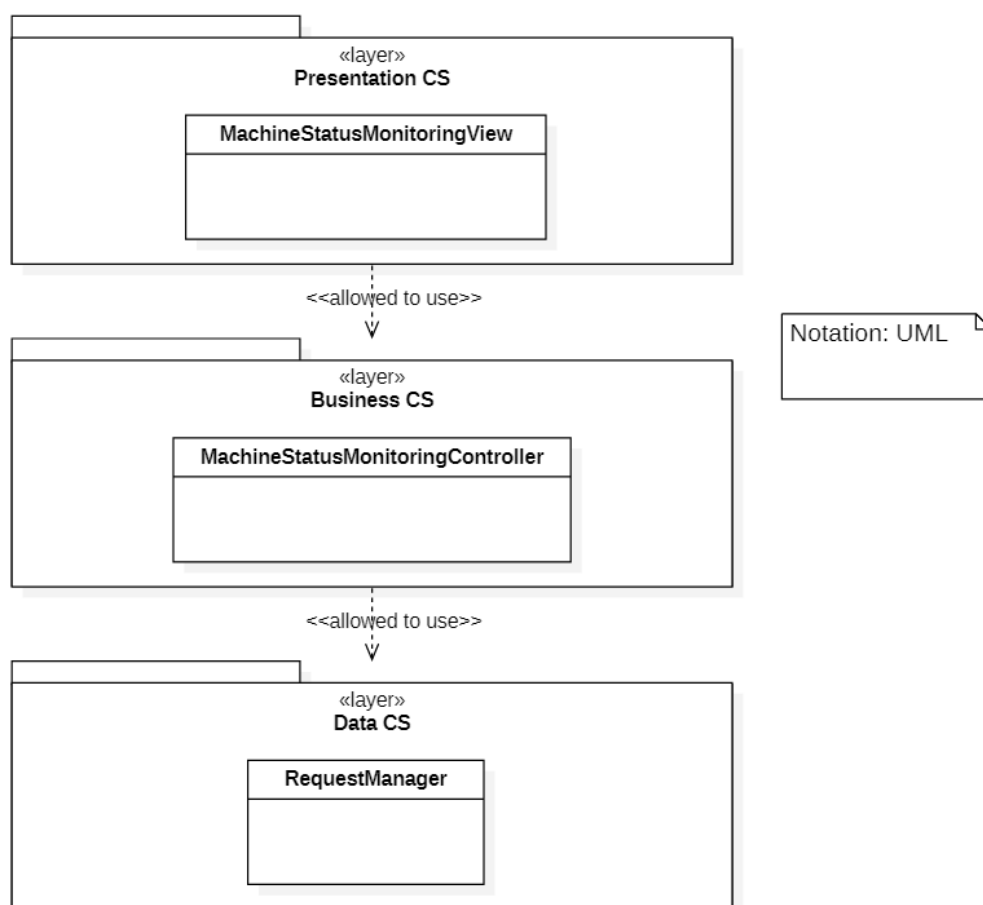
| Selected Style | Rationale |
|---|---|
| Module View with Decomposition, Uses, Layered Style | DVM 시스템의 전체적인 구성을 나타내기 위해 Decomposition Style을 사용하였다. DVM 시스템은 Presentation, Service, Business, Data 레이어로 구성되어 있다. 이를 표현하기 위해 Layered Style을 사용하였다. DVM 시스템의 모듈들은 서로를 사용한다. 이를 나타내기 위해 Uses Style을 사용하였다. |

Part 2.

DVM Management System Overview through Module View

Section 1. Primary Presentation

2.1.1 Primary presentation of DVM Management System



Section 2. Element Catalog

2.2.1 Element and their properties

| Layer | Element | Property |
|--------------|-----------------------------|-------------------|
| Presentation | MachineStatusMonitoringView | DVM 관리자와 상호작용 한다. |

| | | |
|----------|-----------------------------------|------------------------|
| Business | MachineStatusMonitoringController | 관리자의 요청 사항을 처리한다. |
| Data | RequestManager | 요청 사항을 DVM 시스템으로 전달한다. |

2.2.2 Relations and their properties

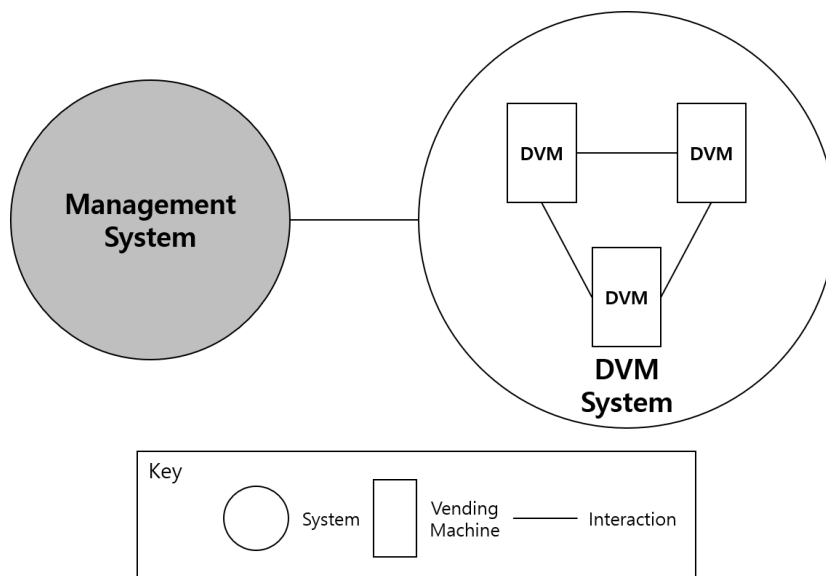
| Element (caller) | Element (callee) | Relation and Property |
|-----------------------------------|-----------------------------------|---------------------------|
| MachineStatusMonitoringView | MachineStatusMonitoringController | 관리자의 요청 사항을 처리하기 위해 호출한다. |
| MachineStatusMonitoringController | RequestManager | 요청 사항을 자판기로 전달하기 위해 호출한다. |

2.2.3 Element interfaces

| Element | Interface | Description |
|-----------------------------------|-------------------|-----------------------------------|
| MachineStatusMonitoringController | checkStock | 자판기의 재고 사항을 확인한다. |
| | updateMachineList | 모든 자판기들에 새로운 자판기의 등록 혹은 삭제를 처리한다. |
| | getNetworkStatus | 자판기의 네트워크 상황을 확인한다. |
| RequestManager | requestStock | 자판기의 재고사항 확인을 요청한다. |
| | requestDVMList | 자판기 리스트 업데이트를 요청한다. |
| | requestStatus | 자판기 네트워크 상태 확인을 요청한다. |

Section 3. Context Diagram

2.3.1 Context Diagram



Section 4. Variability Guide

Section 5. Rationale

2.5.1 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. |

2.5.2 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. |

2.5.3 Selected Style and Rationale

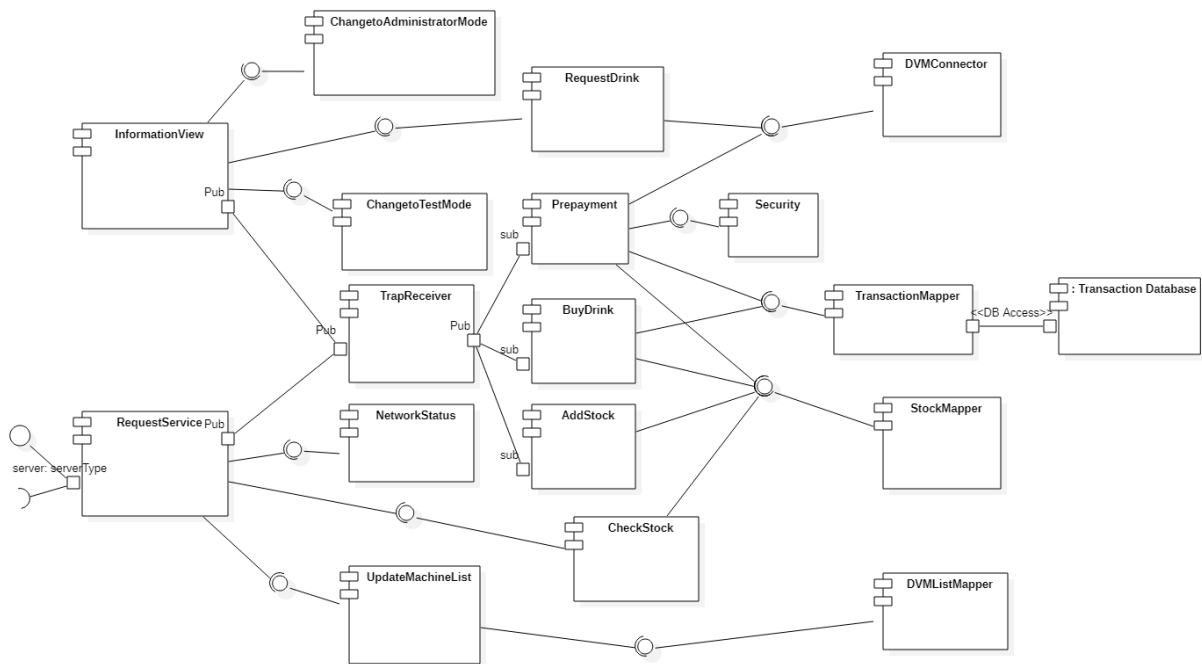
| Selected Style | Rationale |
|---|--|
| Module View with Decomposition, Uses, Layered Style | DVM 매니지먼트 시스템의 전체적인 구성을 나타내기 위해 Decomposition Style을 사용하였다. DVM 매니지먼트 시스템은 Presentation, Business, Data 레이어로 구성되어 있다. 이를 표현하기 위해 Layered Style을 사용하였다. DVM 매니지먼트 시스템의 모듈들은 서로를 사용한다. 이를 나타내기 위해 Uses Style을 사용하였다.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

Part 3.

DVM System Overview through Component and Connector View

Section 1. Primary Presentation

3.1.1 Component Diagram of DVM System



Section 2. Element Catalog

1.2.1 Element and their properties

| 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. |
| ChangoAddAdministratorMode | 관리자 모드로 전환한다. |
| ChangoTestMode | 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. |

1.2.2 Relations and their properties

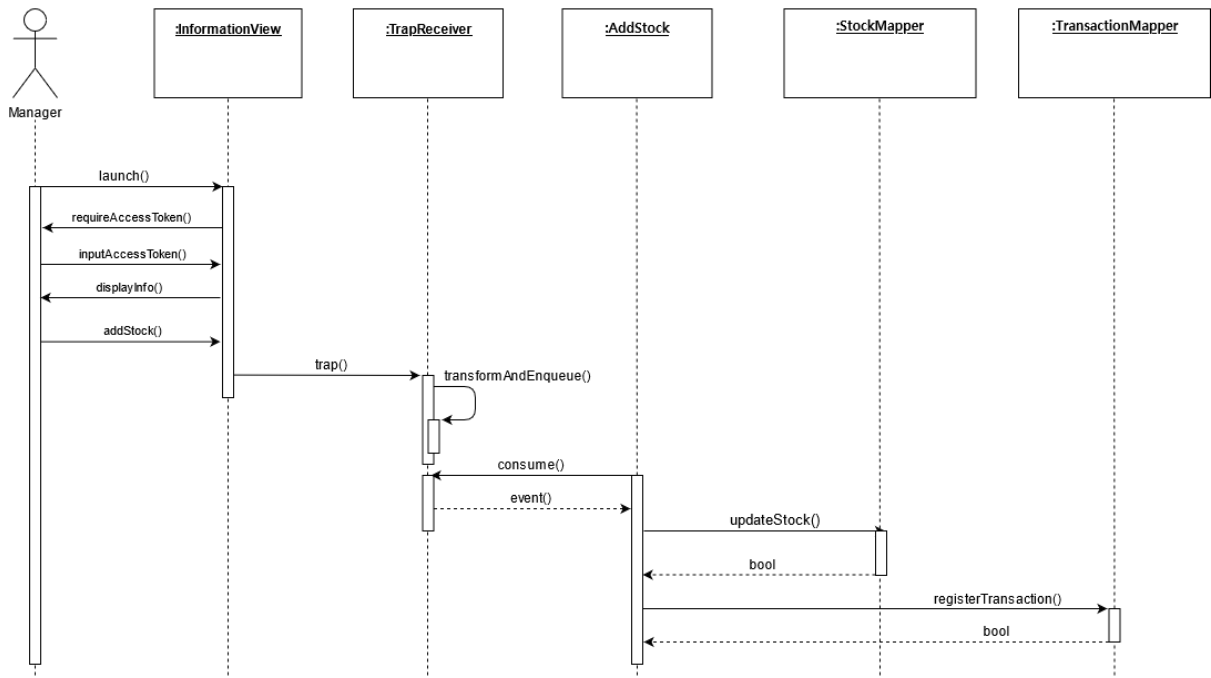
| Element (caller) | Element (callee) | Relation and Property |
|--|---|--|
| InformationView | ChangetoAdministratorMode ChangetoTestMode RequestDrink | 자판기 화면에서 입력 받은 사용자의 요청 사항을 처리한다. |
| RequestService | NetworkStatus CheckStock UpdateMachineList | 다른 자판기 혹은 매니지먼트 시스템으로부터 받은 요청사항을 처리한다. |
| InformationView RequestService | TrapReceiver | 재고 변경과 관련된 이벤트를 메시지 큐에 추가한다. |
| Prepayment BuyDrink AddStock | TrapReceiver | 자신이 처리해야 할 이벤트를 메시지 큐로부터 읽어온다. |
| Prepayment | Security | 결제정보를 암호화 및 복호화한다. |
| RequestDrink Prepayment | DVMConnector | 다른 자판기에 서비스를 요청한다. |
| Prepayment BuyDrink | TransactionMapper | 거래가 발생하면 거래 내역을 저장한다. |
| Prepayment BuyDrink AddStock CheckStock | StockMapper | 재고상황을 확인 혹은 업데이트 한다. |
| UpdateMachineList | DVMListMapper | 자판기 리스트를 업데이트 한다. |

1.2.3 Element interfaces

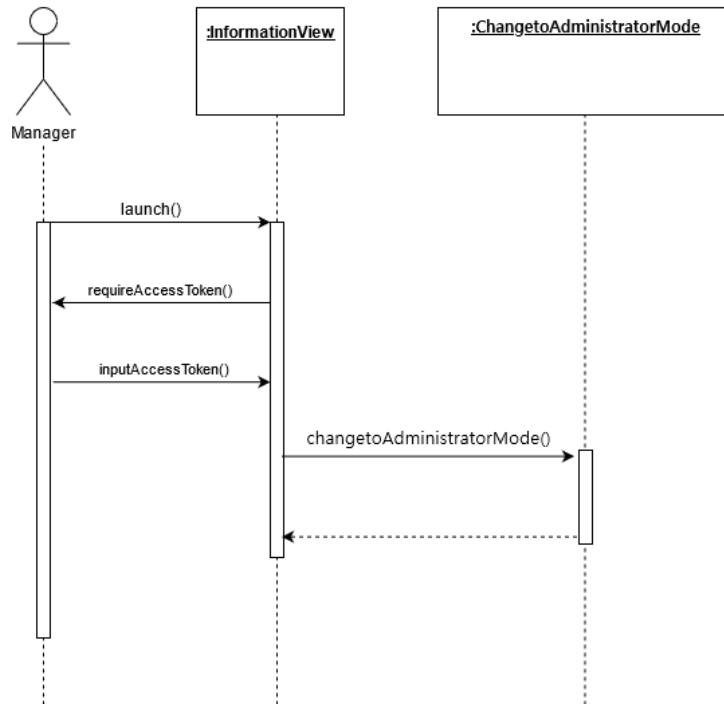
| Element | Interface | Description |
|---------------------------|---------------------------|----------------------------|
| InformationView | requireAccessToken | 관리자 확인을 위해 접근 토큰 입력을 요구한다. |
| | diplayInfo | 업데이트된 화면 정보를 출력한다. |
| | request | 사용자가 요청한 작업을 처리한다. |
| TrapReceiver | Trap | 새로운 이벤트를 입력 받는다. |
| | transformAndEnqueue | 입력 받은 이벤트를 올바른 큐에 넣는다. |
| | consume | 메시지 큐의 이벤트를 반환한다. |
| StockMapper | updateStock | 재고 사항을 업데이트 한다. |
| TransactionMapper | registerTransaction | 새로운 판매 기록을 등록한다. |
| RequestMapper | sendRequest | 외부 요청 사항을 처리한다. |
| CheckStock | checkStock | 재고 사항을 확인한다. |
| ChangetoAdministratorMode | changetoAdministratorMode | 관리자 모드로 변환한다. |
| ChangetoTestMode | changetoTestMode | 테스트 모드로 변환한다. |
| RequestDrink | requestDrink | 다른 자판기에 음료 재고 사항 확인을 요청한다. |
| Prepayment | requestPrepayment | 다른 자판기에 선결제를 요청한다. |
| | prepayment | 다른 자판기로부터 요청받은 선결제를 처리한다. |
| Security | encrypt | 결제 정보를 암호화한다. |
| | decrypt | 결제 정보를 복호화한다. |
| UpdateMachineList | updateMachineList | 등록된 자판기 리스트를 업데이트 한다. |
| DVMListMapper | updateMachineList | 등록된 자판기 리스트를 업데이트 한다. |
| NetworkStatus | getNetworkStatus | 현재 자판기의 네트워크 상태를 반환한다. |

3.2.4 Element behavior

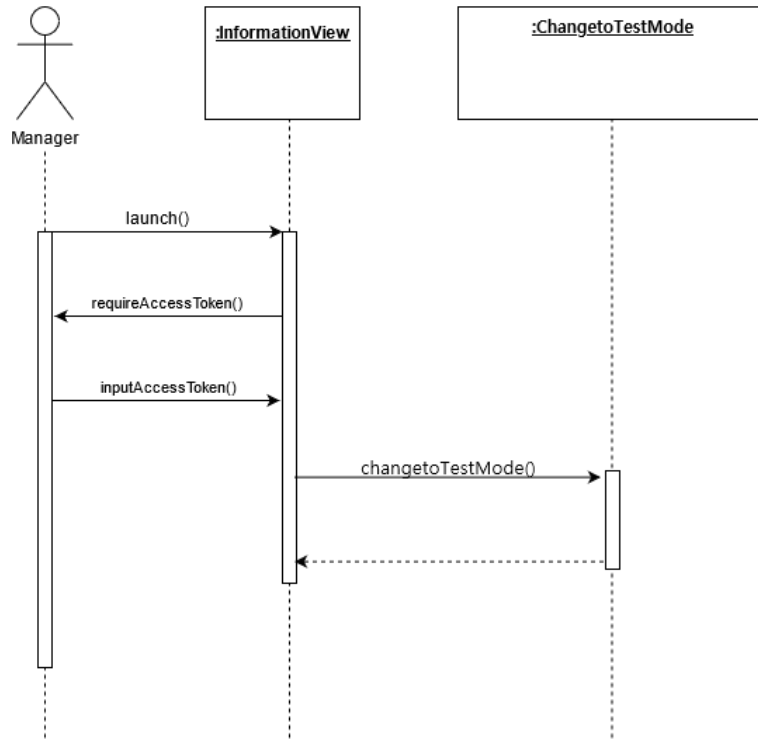
Sequence Diagram 1. (Use Case: Add Stock)



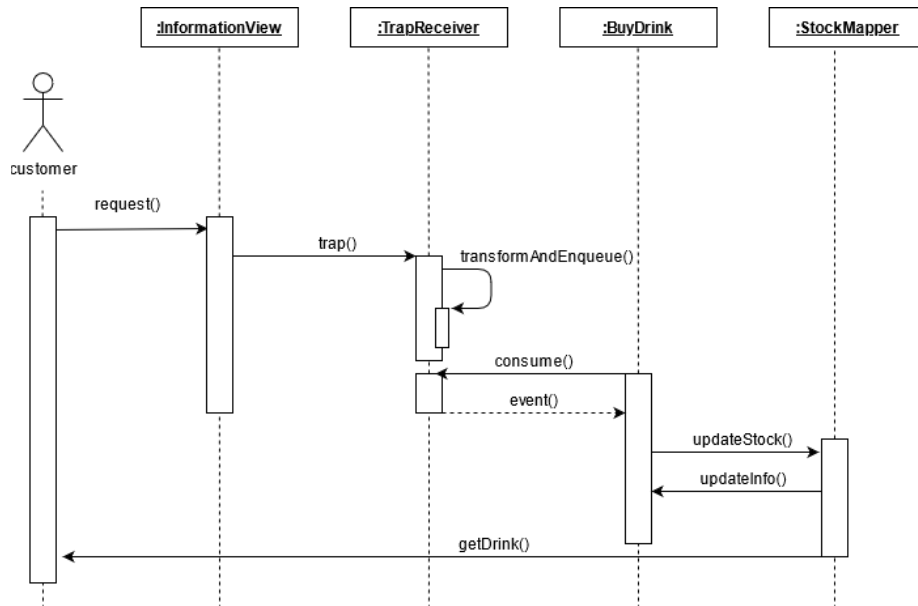
Sequence Diagram 2. (Use Case: Change to Administrator Mode)



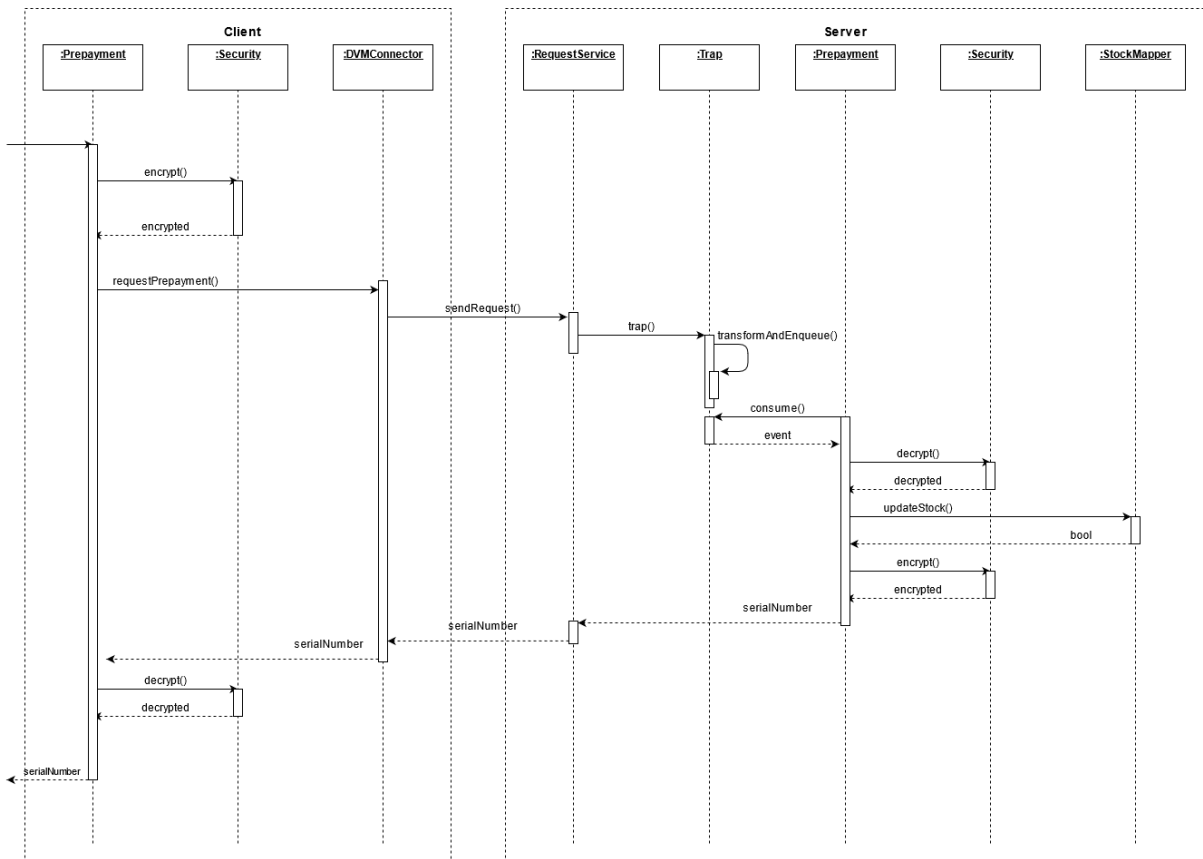
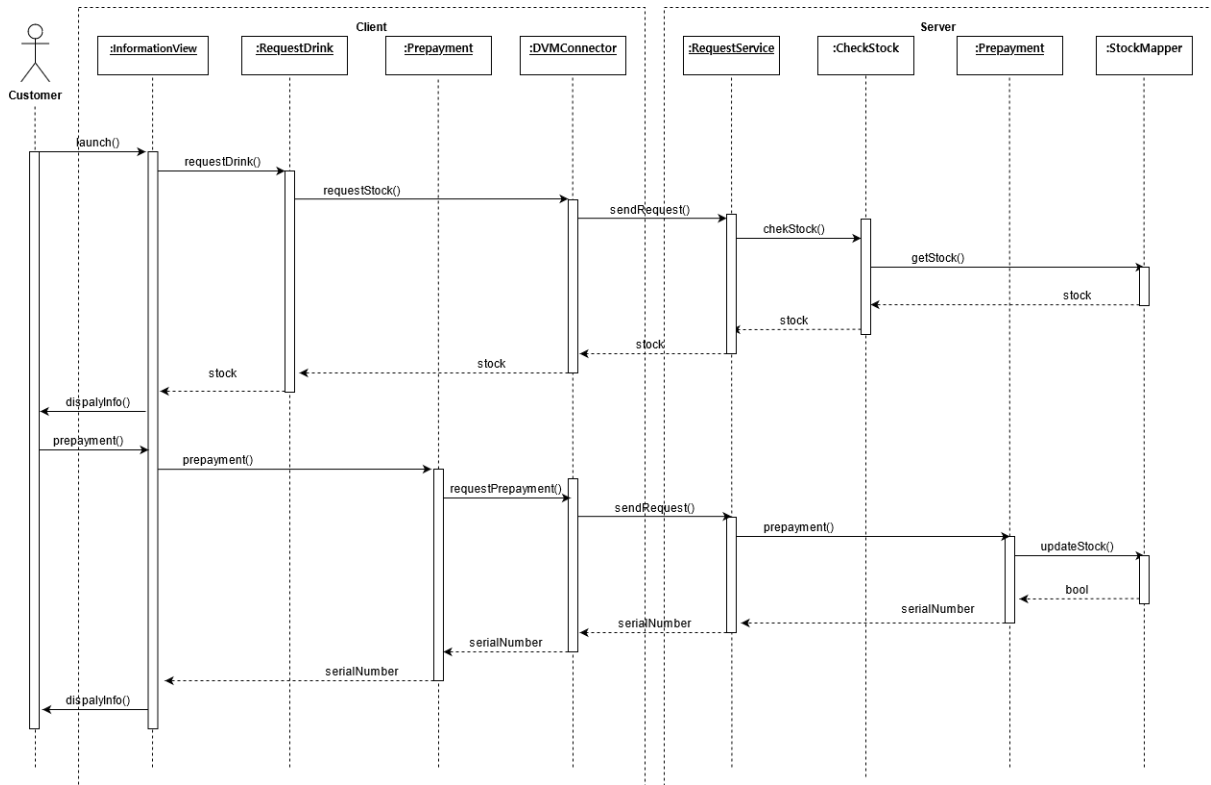
Sequence Diagram 3. (Use Case: Change to Test Mode)

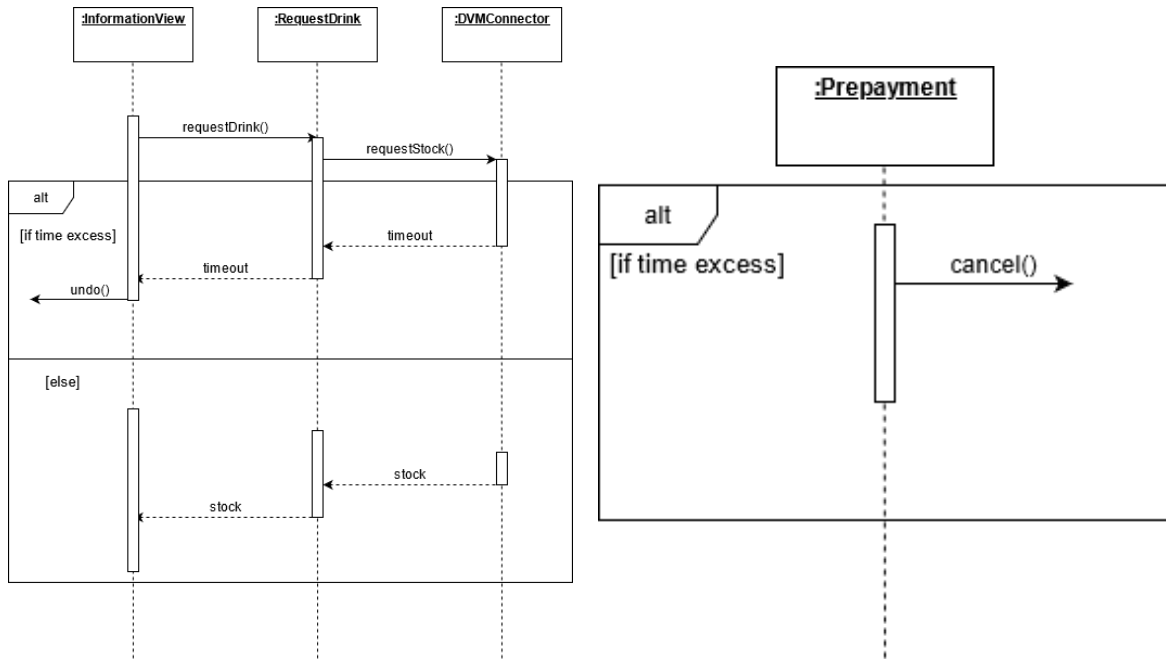


Sequence Diagram 4. (Use Case: Buy Drink)



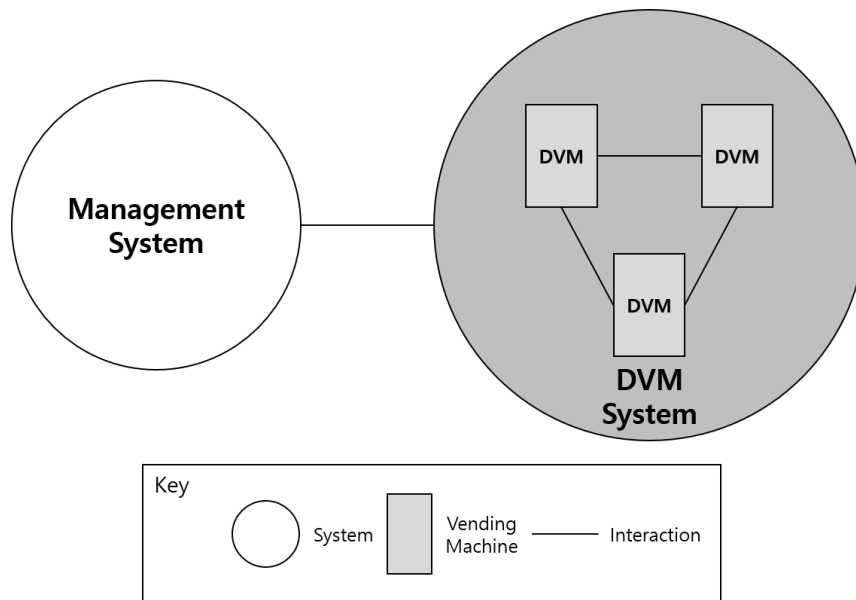
Sequence Diagram 5. (Use Case: Prepayment)





Section 3. Context Diagram

2.3.1 Context Diagram



Section 4. Variability Guide

Section 5. Rationale

3.5.1 Selected Style and Rationale

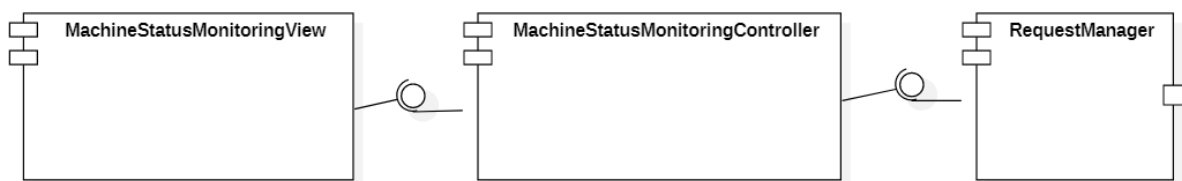
| Selected Style | Rationale |
|---|--|
| C&C View with Publisher-Subscribe Style | DVM 시스템의 컴포넌트간 호출 관계를 표현하기 위해 C&C View를 선택했다. 재고 사항에 변경이 생길 경우 정확한 재고 사항을 나타내기 위해 메시지 큐를 사용한다. 이를 C&C View에서 표현하기 위해 Publisher-Subscribe Style을 사용하였다. |

Part 4.

DVM Management System Overview through Component and Connector View

Section 1. Primary Presentation

4.1.1 Primary presentation of DVM Management System



Section 2. Element Catalog

4.2.1 Element and their properties

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

4.2.2 Relations and their properties

| Element (caller) | Element (callee) | Relation and Property |
|-----------------------------------|-----------------------------------|---------------------------|
| MachineStatusMonitoringView | MachineStatusMonitoringController | 관리자의 요청 사항을 처리하기 위해 호출한다. |
| MachineStatusMonitoringController | RequestManager | 요청 사항을 자판기로 전달하기 위해 호출한다. |

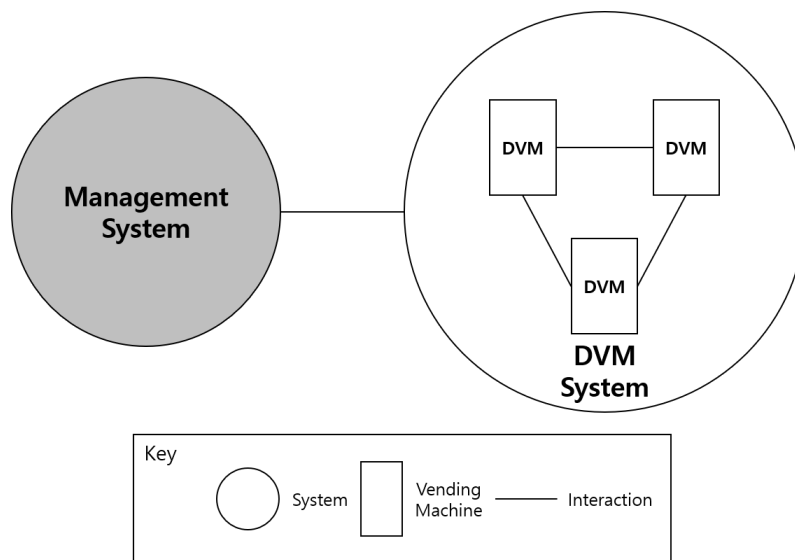
4.2.3 Element interfaces

| Element | Interface | Description |
|-----------------------------------|------------|-------------------|
| MachineStatusMonitoringController | checkStock | 자판기의 재고 사항을 확인한다. |

| | | |
|----------------|-------------------|-----------------------------------|
| | updateMachineList | 모든 자판기들에 새로운 자판기의 등록 혹은 삭제를 처리한다. |
| | getNetworkStatus | 자판기의 네트워크 상황을 확인한다. |
| RequestManager | requestStock | 자판기의 재고사항 확인을 요청한다. |
| | requestDVMList | 자판기 리스트 업데이트를 요청한다. |
| | requestStatus | 자판기 네트워크 상태 확인을 요청한다. |

Section 3. Context Diagram

4.3.1 Context Diagram



Section 4. Variability Guide

Section 5. Rationale

4.5.1 Selected Style and Rationale

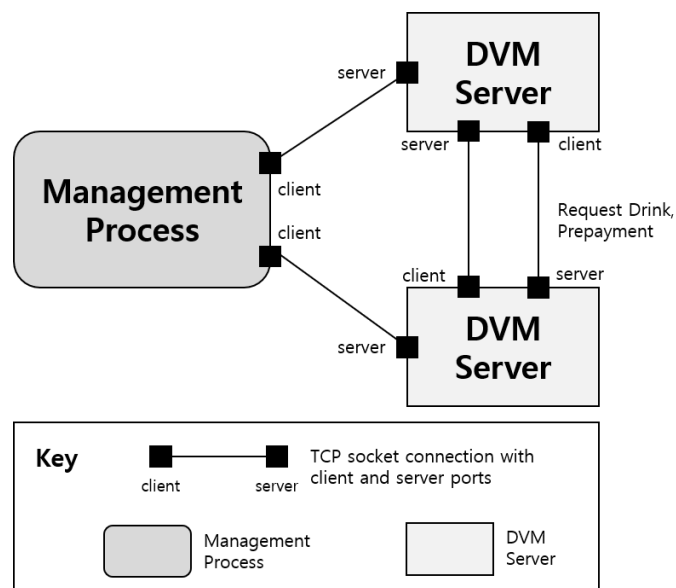
| Selected Style | Rationale |
|----------------|---|
| C&C View | DVM 매니지먼트 시스템의 컴포넌트간 호출 관계를 표현하기 위해 C&C View를 선택했다. |

Part 5

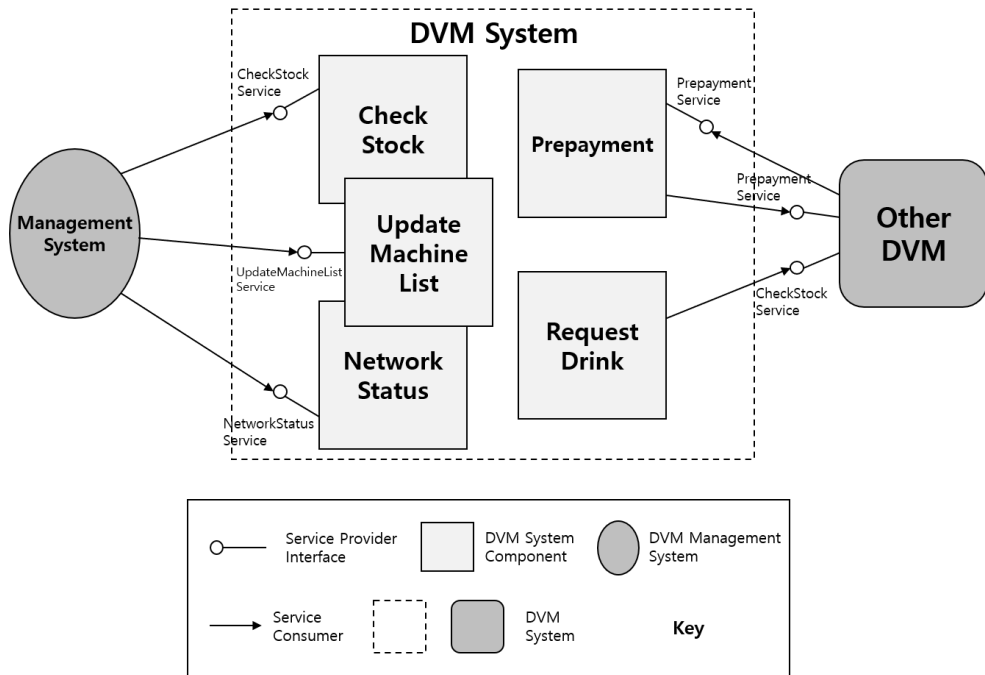
Interaction Diagram of DVM and DVM Management System through Component & Connector View

Section 1. Primary Presentation

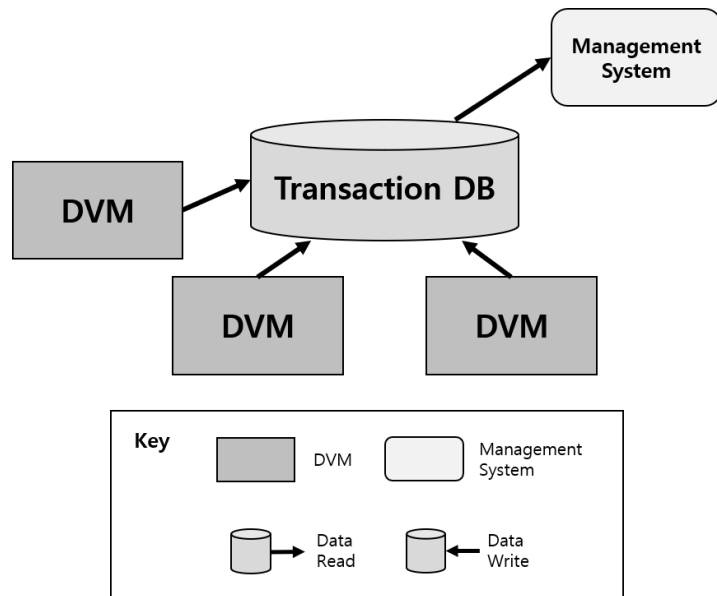
5.1.1 DVM and DVM Management System C&C View with Client-Server Style



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



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



Section 2. Element Catalog

5.2.1 Element and their properties

| Element | Property |
|--------------------|---------------------------------------|
| Management Process | DVM 시스템을 관리하기 위한 매니지먼트 프로세스이다. DVM과 통 |

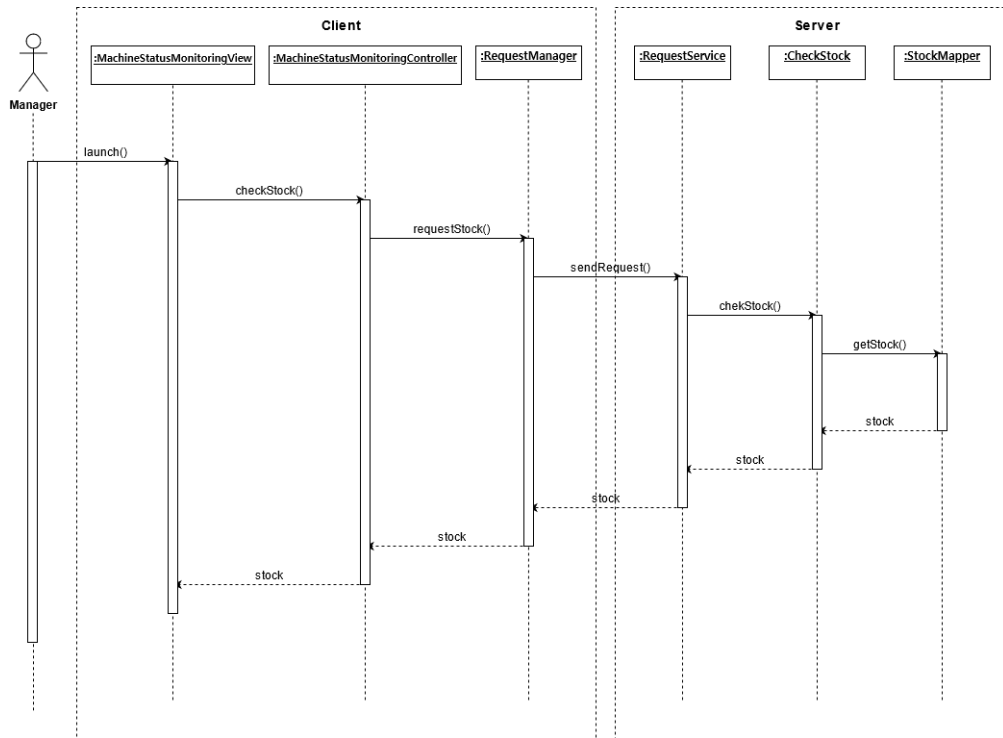
| | |
|-------------------|--|
| | 신하며 이때 매니지먼트 시스템은 클라이언트이다. |
| DVM Server | DVM에서 작동하며 DVM 서비스를 제공하는 서버이다. 매니지먼트 시스템과 다른 DVM에 서비스를 제공할 때는 서버 역할이며 다른 DVM에 서비스를 요청하면 클라이언트로 동작한다. |
| Check Stock | 매니지먼트 시스템에 재고 확인을 위한 서비스를 제공한다. |
| UpdateMachineList | 매니지먼트 시스템에 자판기 리스트 업데이트를 위한 서비스를 제공한다. |
| Network Status | 매니지먼트 시스템에 네트워크 상황 확인을 위한 서비스를 제공한다. |
| Prepayment | 다른 자판기에 선결제 서비스를 제공하거나 이용한다. |
| RequestDrink | 다른 자판기에 재고 확인을 위한 서비스를 요청한다. |
| Transaction DB | 모든 거래 내역을 저장하는 데이터베이스이다. |

5.2.2 Relations and their properties

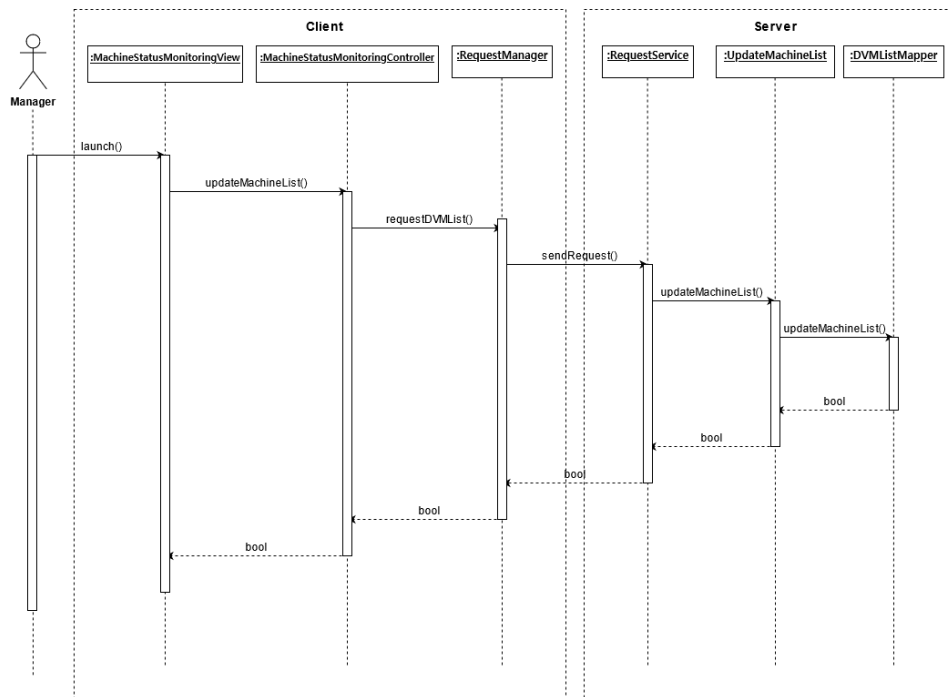
| Element | Element | Relation and Property |
|-------------------|----------------|---|
| Management System | DVM System | 매니지먼트 시스템은 자판기의 재고 사항을 확인, 자판기 리스트 업데이트, 네트워크 상황 확인을 위해 자판기에 서비스를 요청할 수 있다. |
| DVM System | DVM System | 자판기는 다른 자판기에 재고 사항 확인을 요청하고 선결제를 진행하기 위해 서비스를 요청할 수 있다. |
| Management System | Transaction DB | 매니지먼트 시스템은 거래 내역을 확인하기 위해 데이터베이스를 읽을 수 있다. |
| DVM System | Transaction DB | 자판기는 거래가 발생하면 거래 내역을 저장하기 위해 데이터베이스에 거래 내역을 쓸 수 있다. |

5.2.3 Element behavior

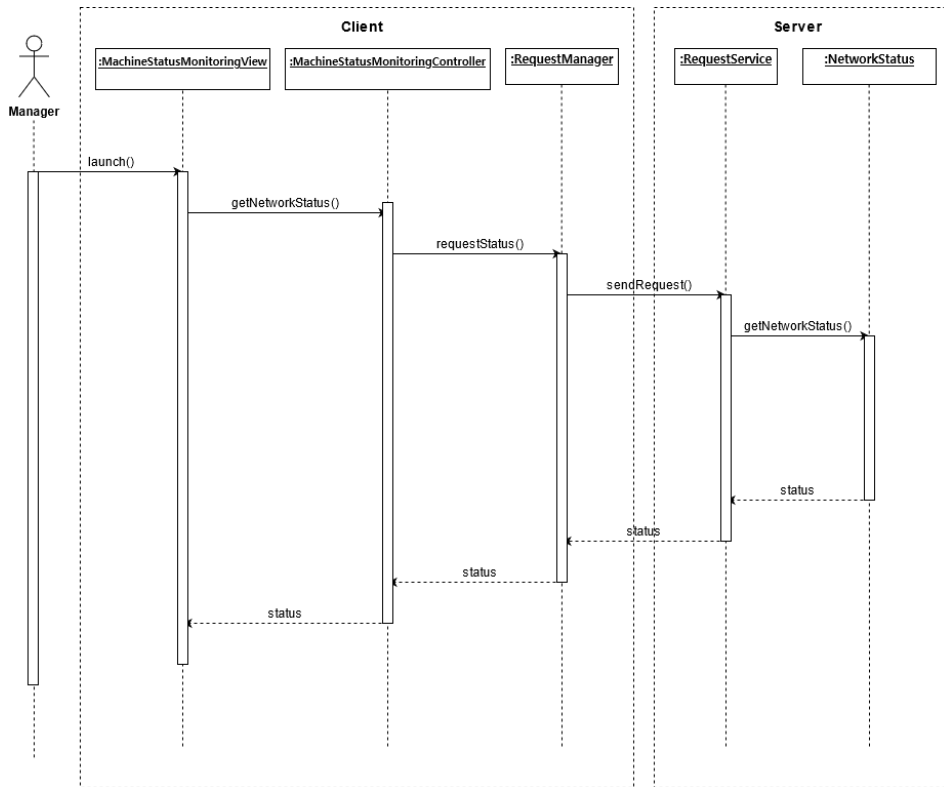
Sequence Diagram 1. (Use Case: Check Stock)



Sequence Diagram 2. (Use Case: Update Machine List)

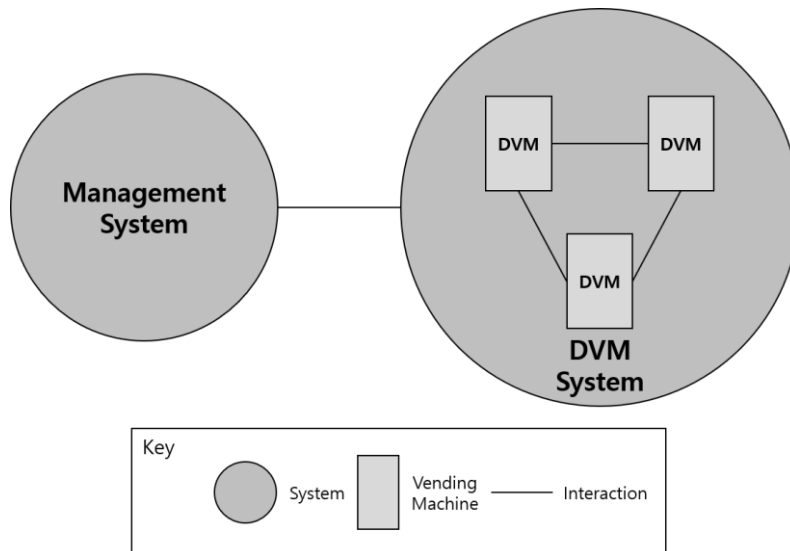


Sequence Diagram 3. (Use Case: Check Network Status)



Section 3. Context Diagram

5.3.1 Context Diagram



Section 4. Variability Guide

Section 5. Rationale

5.5.1 Selected Style and Rationale

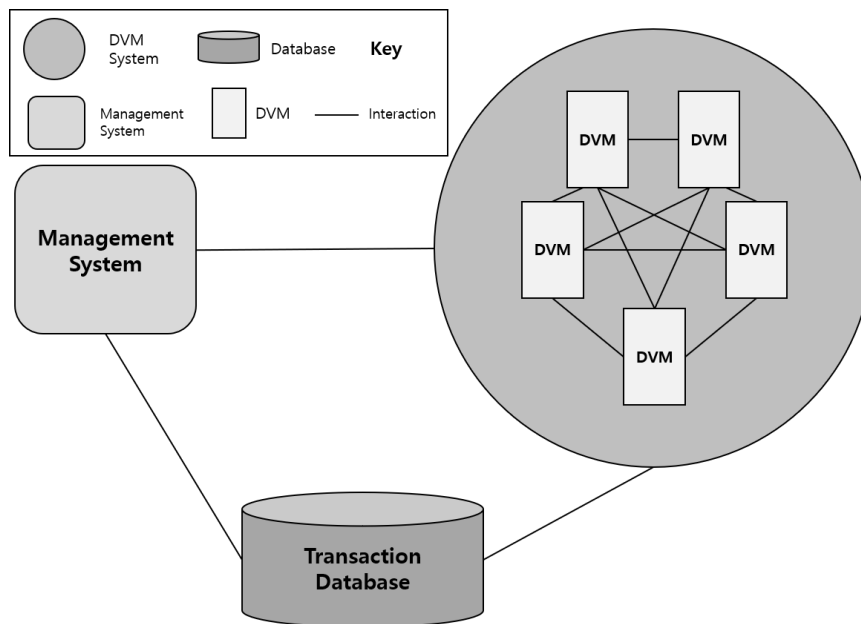
| Selected Style | Rationale |
|---|---|
| C&C View with Client-Server Style | DVM 매니지먼트 시스템은 DVM 시스템에 여러가지 서비스를 요청한다. 이 때 매니지먼트 시스템은 클라이언트 DVM 시스템은 서버 역할을 하게 된다. 또한 DVM 시스템은 서비스 요청과 처리 사항에 따라 클라이언트 혹은 서버가 된다. 이를 표현하기 위해 Client-Server Style을 사용하였다. |
| C&C View with Service-Oriented Architecture Style | 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. |
| 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. |

part 6.

Allocation View

Section 1. Primary Presentation

6.1.1 Deployment Diagram



Section 2. Element Catalog

6.2.1 Element and their properties

| 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. |

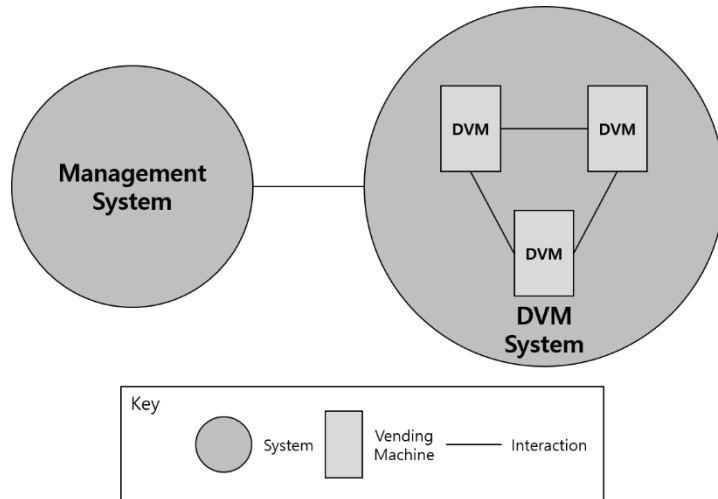
6.2.2 Relations and their properties

| Element (caller) | Element (callee) | Relation and Property |
|-------------------|------------------|-----------------------------|
| Management System | DVM System | 각 자판기들과 연결하여 DVM System을 관리 |

| | | |
|-------------------|----------------------|-----------------------|
| | | 한다. |
| Management System | Transaction Database | 모든 거래 내역들을 확인한다. |
| DVM System | Transaction Database | 거래가 발생하면 거래 내역을 저장한다. |

Section 3. Context Diagram

6.3.1 Context Diagram



Section 4. Variability Guide

Section 5. Rationale

6.5.1 Deployment Pattern for DVM System

| 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. |

6.5.2 Selected Style and Rationale

| Selected Style | Rationale |
|----------------------|--|
| Allocation View with | Deployment Pattern으로 Three-tier deployment pattern을 사용하였다. |

| | |
|------------------|-------------------------------------|
| Deployment Style | 이를 나타내기 위해 Deployment Style을 사용하였다. |
|------------------|-------------------------------------|

part 7

Documentation Beyond Views

Section 1. Documentation roadmap

7.1.1 Purpose of Documentation

DVM(Distributed Vending Machine) 시스템과 DVM 시스템 관리를 위한 DVM 매니지먼트 시스템의 아키텍처를 설명한다.

7.1.2 Documentation Organization

본 문서는 DVM을 위한 DVM 시스템과 DVM 매니지먼트 시스템을 정의한다. 파트 1과 3에서 DVM 시스템의 구성을 설명한다. 파트 2와 4에서는 DVM 매니지먼트 시스템의 구성을 설명한다. 파트 5에서 DVM 시스템, DVM 매니지먼트 시스템의 상호 작동 방식에 대해 설명한다. 파트 6에서 전체 시스템의 배포 방법에 대해 정의한다.

7.1.3 View Overview

| Part | View | Description | |
|------|---|-------------|---|
| 1 | DVM System Overview through Module View | View style | Decomposition, Use, Layered |
| | | Graphics | UML |
| 2 | DVM Management System Overview through Module View | View style | Decomposition, Use, Layered |
| | | Graphics | UML |
| 3 | DVM System Overview through Component and Connector View | View style | Client – Server, Publish – Subscribe |
| | | Graphics | UML |
| 4 | DVM Management System Overview through Component and Connector View | View style | C&C View |
| | | Graphics | UML |
| 5 | Interaction Diagram of DVM and DVM Management System through Component and Connector View | View style | Client-Server, Service Oriented Architecture, Shared Data |
| | | Graphics | Informal notation |
| 6 | Allocation View | View style | Deployment |
| | | Graphics | Informal notation |

Section 2. How a view is documented

각 파트 별 구성은 다음과 같다. 파트 1, 2는 DVM 시스템과 DVM 관리 시스템의 전체적인 구성을 Module Decomposition View로 나타낸다. 파트 3, 4는 파트 1, 2에서 정의한 시스템 모듈들의 실제 구성과 인터페이스를 Component & Connector View로 설명한다. 파트 5는 DVM 시스템과 DVM 관리 시스템 및 데이터베이스의 상호작용과 동작 방식을 설명한다. 파트 6에서는 정의한 시스템의 배포 방법을 Allocation View로 나타낸다.

Section 3. System overview

7.3.1 System Overview

DVM(Distributed Vending Machine)은 여러 자판기가 네트워크상에서 연결되어 운영되는 자판기 시스템이다. 사용자는 자판기에서 음료를 구매할 수 있으며 다른 자판기의 음료 구매 또한 가능하다. DVM 시스템에는 관리자가 존재하며 모든 자판기의 재고 사항과 거래 내역을 확인할 수 있다.

7.3.2 Primary Functionality by Use Case

| Use Case | Description |
|-----------------|---|
| UC-1: 재고 추가 | 재고를 추가하면 재고 상태를 업데이트 한다. 만약 품질된 음료의 재고를 추가하면 음료를 판매 가능 상태로 변경한다. |
| UC-2: 재고 상태 확인 | 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. |
| UC-3: 관리자 모드 접근 | 자판기의 제품 판매 상태를 활성화 혹은 비활성화를 할 수 있다. 제품 판매 기록을 확인할 수 있다. |
| UC-4: 테스트 모드 접근 | 테스트를 위한 데모 시스템이 존재해야 한다. 테스트 모드에서는 제품 판매 상태 변경이 가능하며 결제를 자유롭게 수행할 수 있다. |
| UC-5: 음료 구매 | 사용자는 자판기의 화면을 보고 음료를 카드로 결제한다. 잔액이 부족한 경우 결제되지 않는다. |
| UC-6: 선결제 구매 | 만약 다른 자판기에 있는 음료를 구매할 경우 해당 자판기에서 결제 후 인증 코드를 발급한다. 발급받은 인증 코드를 |

| | |
|----------------------|---|
| | 입력하면 음료를 받을 수 있다. |
| UC-7: 새로운 자판기 추가, 삭제 | 자판기를 추가 혹은 삭제를 하면 주변 자판기에 상태를 업데이트 한다. |
| UC-8: 네트워크 점검 | 네트워크 관리자가 특정 자판기에 상태 확인 요청을 할 경우 자판기는 상태 메시지를 전송한다. |

Section 4. Mapping Between Views

7.4.1 Mapping Between Part 1 View and Part 3 View

| Element in C&C View 3 | Element in Module View 1 |
|---------------------------|---------------------------|
| InformationView | InformationView |
| RequestService | RequestService |
| AddStock | AddStock |
| BuyDrink | BuyDrink |
| RequestDrink | RequestDrink |
| ChangetoAdministratorMode | ChangetoAdministratorMode |
| ChangetoTestMode | ChangetoTestMode |
| CheckStock | CheckStock |
| Prepayment | Prepayment |
| NetworkStatus | NetworkStatus |
| UpdateMachineList | UpdateMachineList |
| TrapReceiver | TrapReceiver |
| Security | Security |
| StockMapper | StockMapper |
| TransactionMapper | TransactionMapper |
| DVMConnector | DVMConnector |
| DVMListMapper | DVMListMapper |

7.4.2 Mapping Between Part 2 View and Part 4 View

| Element in C&C View 4 | Element in Module View 2 |
|-----------------------------------|-----------------------------------|
| MachineStatusMonitoringView | MachineStatusMonitoringView |
| MachineStatusMonitoringController | MachineStatusMonitoringController |
| RequestManager | RequestManager |

7.4.3 Mapping Between Part 3, 4 View and Part 5 View

| | |
|---|-----------------------|
| Element in C&C View 3, 4 | Element in C&C View 5 |
| MachineStatusMonitoringView MachineStatusMonitoringController RequestManager | Management Process |
| RequestService RequestDrink CheckStock Prepayment NetworkStatus UpdateMachineList StockMapper TransactionMapper DVMConnector DVMListMapper | DVM Server |
| CheckStock | CheckStock |
| Prepayment | Prepayment |
| NetworkStatus | NetworkStatus |
| UpdateMachineList | UpdateMachineList |
| RequestDrink | RequestDrink |

Section 5. Rationale

7.5.1 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 판 | 전용 클라이언트가 필요하며 모바일 환경에서 사용이 불가능하다. |

| | | |
|--|----------------------|--|
| | 매 기록 및 분석을 확인할 수 있다. | |
|--|----------------------|--|

7.5.2 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. |

Section 6. Directory