

HP Software Architecture Document

유혜리 이동훈 송승현

Division of Computer Science & Engineering

gpfl111@konkuk.ac.kr dlehdgns56@konkuk.ac.kr uion93@konkuk.ac.kr

Index

1. Introduction Section
2. System Purpose
 - 2-1. Context
 - 2-2. System Interface
 - 2-3. Non-functional requirement
3. Structure
 - 3-1. Overview
 - 3-2. Components
 - 3-3. Interface
4. Dynamic Behavior
 - 4-1. Scenarios
 - 4-1-1. Scenario Specification
 - 4-1-2. Component Interaction Model
 - 4-2. Mechanisms
5. Other view
 - 5-1. Process View
 - 5-2. Development View
 - 5-3. Physical View
6. Conceptual Framework
 - 6-1. Domain Lexicon
 - 6-2. Lexicon Diagram
7. Conclusion

1. Introduction Section

Introduction Section에서는 Architecture와 Stakeholders를 식별하고, 문서의 생성과 subsequent modification를 기록하였다.

Name of the architecture	분산형 자판기 아키텍처(Distributed Vending Machine Architecture)																
Team information	유혜리(건국대학교 일반대학원 컴퓨터공학부 HCI Lab) 송승현(건국대학교 일반대학원 컴퓨터 공학부 DS Lab) 이동훈(건국대학교 일반대학원 컴퓨터공학부 INNS Lab)																
Creation and modification	None																
Audience and purpose	DVM 제조회사, DVM System Architecture description																
Selected viewpoints	<table border="1"> <thead> <tr> <th>Name of viewpoint</th> <th>Models</th> </tr> </thead> <tbody> <tr> <td>Context and System Interface Viewpoint</td> <td>-Context Diagram(UML 유스케이스 다이어그램) -System interface description</td> </tr> <tr> <td>Concepts Viewpoint</td> <td>-Domain lexicon -Lexicon diagram</td> </tr> <tr> <td>Logical Structure Viewpoint</td> <td>-architecture diagram -Commentary -Component specification -Interface specification</td> </tr> <tr> <td>Logical Behavior Viewpoint</td> <td>-Scenario specification -Component interaction model</td> </tr> <tr> <td>Process Viewpoint</td> <td>-UML object model with active objects</td> </tr> <tr> <td>Physical Viewpoint</td> <td>-UML deployment diagram</td> </tr> <tr> <td>Development Viewpoint</td> <td>-UML component diagram</td> </tr> </tbody> </table>	Name of viewpoint	Models	Context and System Interface Viewpoint	-Context Diagram (UML 유스케이스 다이어그램) -System interface description	Concepts Viewpoint	-Domain lexicon -Lexicon diagram	Logical Structure Viewpoint	-architecture diagram -Commentary -Component specification -Interface specification	Logical Behavior Viewpoint	-Scenario specification -Component interaction model	Process Viewpoint	-UML object model with active objects	Physical Viewpoint	-UML deployment diagram	Development Viewpoint	-UML component diagram
Name of viewpoint	Models																
Context and System Interface Viewpoint	-Context Diagram (UML 유스케이스 다이어그램) -System interface description																
Concepts Viewpoint	-Domain lexicon -Lexicon diagram																
Logical Structure Viewpoint	-architecture diagram -Commentary -Component specification -Interface specification																
Logical Behavior Viewpoint	-Scenario specification -Component interaction model																
Process Viewpoint	-UML object model with active objects																
Physical Viewpoint	-UML deployment diagram																
Development Viewpoint	-UML component diagram																

2. System Purpose

이 섹션에서는 시스템의 목적과 시스템이 해결하고자 하는 문제를 문서화하였다. 아래와 같은 사항들이 포함된다.

- 시스템에 사용될 Context와 해결될 문제
- 시스템이 인터페이스에서 제공하는 서비스
- 서비스의 퀄리티

2-1. Context

이 섹션은 시스템의 컨텍스트와 해결되는 문제를 briefly and informally 하게 소개하였다.

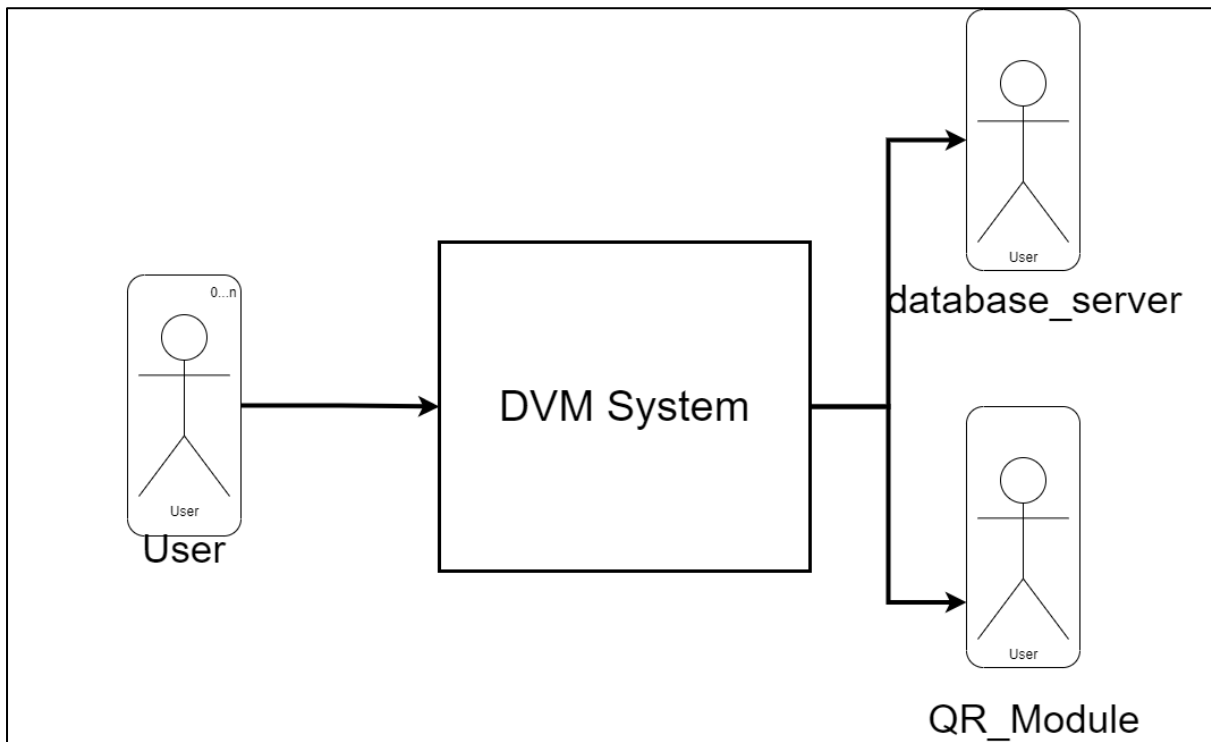


Figure 1. Workflow Diagram of DVM

DVM System은 사용자와 DVM들, DVM의 정보를 가지고 있는 Database Server와 결제를 담당하는 QR 모듈로 이루어져 있다. 앞으로의 문서는 위의 그림1의 구조를 좀더 세분화하여 구체적으로 설명하게 될 것이다.

2-2. System Interface

이번 섹션에서는 DVM의 Interface에 대하여 기술하였다. Use Case에 따라서 제공되는 Interface를 작성하였다.

Interface	DVMDisplayManager
Use Cases	<p>UC-1: 상품선택 버튼을 누른다: 사용자가 상품을 구매하기 위해 DVM에 접근하여 초기 화면의 '상품선택', '코드입력', '관리자' 세가지 옵션 중 '상품선택' 버튼을 누른 경우</p> <p>System operations of this use case: <i>menuItem</i>(상품선택화면 보기 요청을 받는다), <i>viewItem</i>(상품 선택 화면을 반환한다)</p>
	<p>UC-2: 코드입력 버튼을 누른다: 사용자가 다른 DVM에서 재고가 없어 상품을 수령 받지 못하고 코드를 받아, 현재 DVM에 접근하여 초기 화면의 '상품선택', '코드입력', '관리자' 세가지 옵션 중 '코드입력' 버튼을 누른 경우</p>

	<p>System operations of this use case: <i>menuCode</i>(코드결제 화면 보기 요청을 받는다), <i>viewCode</i>(코드 결제 화면을 반환한다)</p>
	<p>UC-3: 코드를 환불한다: 사용자가 '코드입력' 버튼을 눌러 코드입력 화면으로 넘어간 경우, 코드를 입력하지 않고 코드 환불을 위해 '코드 환불' 버튼을 누른 경우</p> <p>System operations of this use case: <i>menuCode</i>(코드결제 화면 보기 요청을 받는다), <i>viewCode</i>(코드 결제 화면을 반환한다), <i>menuRefund</i>(코드환불 요청 정보를 보낸다), <i>viewRefundFinish</i>(코드 환불 완료 정보를 화면으로 반환한다)</p>
	<p>UC-4: 관리자 버튼을 누른다: 관리자가 DVM 관리를 위해 DVM에 접근하여 초기 화면의 '상품선택', '코드입력', '관리자' 세가지 옵션 중 '관리자' 버튼을 누른 경우</p> <p>System operations of this use case: <i>menuAdmin</i>(관리자 화면 보기 요청을 받는다), <i>viewAdmin</i>(관리자 화면을 반환한다.)</p>
	<p>UC-5: 상품을 선택한다: 사용자가 상품을 구매하기 위해 DVM에 접근하여 '상품선택' 버튼을 눌러 상품 목록을 확인 후 구매를 원하는 상품을 선택한 경우</p> <p>System operations of this use case: <i>menuItem</i>(상품선택 화면 보기 요청을 받는다), <i>viewItem</i>(상품선택 화면을 반환한다.), <i>selectItem</i>(사용자가 선택한 상품 정보를 보낸다)</p>
	<p>UC-6: DVM에 재고가 있거나, 다른 재고가 있는 DVM 조회를 마친 경우 QR결제를 요청한다: 사용자가 구매하려는 상품을 선택하여 결제를 요청하는 과정</p> <p>System operations of this use case: <i>viewQR</i>(사용자가 선택한 상품의 QR결제 코드를 화면으로 반환한다), <i>finishRequest</i>(결제완료 정보를 보낸다), <i>viewFinish</i>(상품구매 완료 화면을 반환한다),</p>
	<p>UC-7: 재고가 없을 경우, 재고가 있는 다른 DVM을 조회한다: 사용자가 구매하려는 제품의 재고가 없는 경우에 해당 제품의 재고가 있는 다른 DVM을 조회하는 과정</p> <p>System operations of this use case: <i>menuItem</i>(상품선택 화면 보기 요청을 받는다), <i>viewItem</i>(상품선택 화면을 반환한다), <i>selectItem</i>(사용자가 선택한 상품 정보를 보낸다), <i>viewDVMList</i>(재고가 있는-상품수령이 가능한 DVM리스트를 화면에 반환한다), <i>selectDVM</i>(사용자가 선택한 DVM정보를 넘긴다),</p>

	<p><i>viewSendCode</i>(상품수령 코드를 화면으로 반환한다)</p> <p>UC-11: 관리자 상태에서 DVM의 재고를 변경한다: 관리자가 DVM에 재고를 재정보기한 후 이 정보를 시스템에 입력하기 위해 DVM에 접근하여 초기 화면에서 '관리자' 버튼을 선택해 재고를 변경하는 경우</p> <p>System operations of this use case: <i>MenuAdmin</i>(관리자 화면 보기 요청을 받는다), <i>viewAdmin</i>(관리자 화면을 반환한다), <i>stockInfoRequest</i>(재고 정보를 요청한다), <i>viewStock</i>(재고정보를 화면으로 반환한다), <i>stockChangeRequest</i>(재고 변경 정보를 보낸다), <i>viewChangeStockFinish</i>(재고 갱신 완료 정보를 반환한다)</p>
--	--

Interface	NetworkManager-Client
Use Cases	<p>UC-1: 상품선택 버튼을 누른다. : 사용자가 상품을 구매하기 위해 DVM에 접근하여 초기 화면의 '상품선택', '코드입력', '관리자' 세가지 옵션 중 '상품선택' 버튼을 누른 경우</p> <p>System operations of this use case: <i>itemRequest</i>(상품정보를 요청한다), <i>itemResponse</i>(상품정보를 DB에서 받아 반환한다)</p>
	<p>UC-3: 코드를 환불한다: 사용자가 '코드입력' 버튼을 눌러 코드입력 화면으로 넘어간 경우, 코드를 입력하지 않고 코드 환불을 위해 '코드 환불' 버튼을 누른 경우</p> <p>System operations of this use case: <i>refundRequest</i>(코드환불요청 정보를 보낸다), <i>refundResponse</i>(코드환불 완료정보를 보낸다)</p>
	<p>UC-5: 상품을 선택한다: 사용자가 상품을 구매하기 위해 DVM에 접근하여 '상품선택' 버튼을 눌러 상품 목록을 확인 후 구매를 원하는 상품을 선택한 경우</p>

System operations of this use case:

itemRequest(상품정보를 요청한다),

itemResponse(상품정보를 DB에서 받아 반환한다),

selectItem(사용자가 선택한 상품정보를 받는다)

UC-6: DVM에 재고가 있거나, 다른 재고가 있는 DVM 조회를 마친 경우 QR 결제를 요청한다. 사용자가 구매하려는 상품을 선택하여 결제를 요청하는 과정

System operations of this use case:

selectItem(사용자가 선택한 상품정보를 받는다),

selectItemQR(사용자가 선택한 상품에 대한 결제정보를 통해 QR모듈로부터 받은 QR코드를 반환한다),

finishRequest(결제 완료 정보를 보낸다),

finishResponse(결제 완료정보를 반환한다)

UC-7: 재고가 없을 경우, 재고가 있는 다른 DVM을 조회한다. 사용자가 구매하려는 제품의 재고가 없는 경우에 해당 제품의 재고가 있는 다른 DVM을 조회하는 과정

System operations of this use case:

sendDVMList(재고가 있는 DVM 정보들을 반환한다),

selectDVM(사용자가 선택한 DVM 정보를 넘긴다)

UC-8: DVM의 재고정보를 네트워크 상에 동기화한다. DVM에서 상품 구매 완료 등으로 인해 재고가 변경될 경우 이 정보가 네트워크 상에서 동기화되는 경우

System operations of this use case:

finishResponse(결제완료정보를 DB에 넘겨 재고정보를 갱신한다),

sendCode(사용자가 선택한 DVM 과 상품에 대한 결제완료정보를 DB에 넘기고 상품수령 코드를 반환한다)

	<p>UC-9: DVM의 네트워크 상태를 점검한다: 5분간격으로 네트워크 상태를 점검하고, 문제 발생 시 관리자에게 메시징한다.</p> <p>System operations of this use case:</p> <p>networkStatus(네트워크 상태를 넘긴다),</p> <p>networkStatusResponse(네트워크 문제가 발생한 경우, DB에 전송하고 관리자에게 메시징한다. 문제가 발생하지 않은 경우 확인 정보를 반환한다)</p> <p>UC-11: 관리자 상태에서 DVM의 재고를 변경한다. : 관리자가 DVM에 재고를 재정비한 후 이 정보를 시스템에 입력하기 위해 DVM에 접근하여 초기 화면에서 '관리자' 버튼을 선택해 재고를 변경하는 경우</p> <p>System operations of this use case:</p> <p>stockInfoRequest(재고정보를 요청한다),</p> <p>stockInfoResponse(재고정보를 반환한다),</p> <p>stockChangeRequest(재고 변경 정보를 보낸다),</p> <p>changeStockResponse(DB의 재고정보 갱신 후 받은 재고 갱신 완료정보를 반환한다)</p>
--	--

Interface	RequestService
Use Cases	<p>UC-1: 상품선택 버튼을 누른다. 사용자가 상품을 구매하기 위해 DVM에 접근하여 초기 화면의 '상품선택', '코드입력', '관리자' 세가지 옵션 중 '상품선택' 버튼을 누른 경우</p> <p>System operations of this use case:</p> <p>itemRequest(상품정보를 요청한다),</p> <p>itemResponse(상품정보를 DB에서 받아 반환한다)</p>

UC-3: 코드를 환불한다. 사용자가 '코드입력' 버튼을 눌러 코드입력 화면으로 넘어간 경우, 코드를 입력하지 않고 코드 환불을 위해 '코드 환불' 버튼을 누른 경우

System operations of this use case:

refundRequest(코드환불요청 정보를 보낸다),

refundResponse(코드환불 완료정보를 보낸다)

UC-5: 상품을 선택한다. 사용자가 상품을 구매하기 위해 DVM에 접근하여 '상품선택' 버튼을 눌러 상품 목록을 확인 후 구매를 원하는 상품을 선택한 경우

System operations of this use case:

itemRequest(상품정보를 요청한다),

itemResponse(상품정보를 DB에서 받아 반환한다),

selectItem(사용자가 선택한 상품정보를 받는다)

UC-6: DVM에 재고가 있거나, 다른 재고가 있는 DVM 조회를 마친 경우 QR 결제를 요청한다. 사용자가 구매하려는 상품을 선택하여 결제를 요청하는 과정

System operations of this use case:

selectItem(사용자가 선택한 상품정보를 받는다),

selectItemQR(사용자가 선택한 상품에 대한 결제정보를 통해 QR모듈로부터 받은 QR코드를 반환한다),

finishRequest(결제 완료 정보를 보낸다),

finishResponse(결제 완료정보를 반환한다)

UC-7: 재고가 없을 경우, 재고가 있는 다른 DVM을 조회한다. 사용자가 구매하려는 제품의 재고가 없는 경우에 해당 제품의 재고가 있는 다른 DVM을 조회하는 과정

System operations of this use case:

sendDVMList(재고가 있는 DVM 정보들을 반환한다),

selectDVM(사용자가 선택한 DVM 정보를 넘긴다)

UC-8: DVM의 재고정보를 네트워크 상에 동기화한다.DVM에서 상품 구매 완료 등으로 인해 재고가 변경될 경우 이 정보가 네트워크 상에서 동기화되는 경우

System operations of this use case:

finishResponse(결제완료정보를 DB에 넘겨 재고정보를 갱신한다),

sendCode(사용자가 선택한 DVM 과 상품에 대한 결제완료정보를 DB에 넘기고 상품수령 코드를 반환한다)

UC-9: DVM의 네트워크 상태를 점검한다: 5분간격으로 네트워크 상태를 점검하고, 문제 발생 시 관리자에게 메시징 한다.

System operations of this use case:

networkStatus(네트워크 상태를 넘긴다),

networkStatusResponse(네트워크 문제가 발생한 경우, DB에 전송하고 관리자에게 메시징한다. 문제가 발생하지 않은 경우 확인 정보를 반환한다)

UC-11: 관리자 상태에서 DVM의 재고를 변경한다. 관리자가 DVM에 재고를 재정비한 후 이 정보를 시스템에 입력하기 위해 DVM에 접근하여 초기 화면에서 '관리자' 버튼을 선택해 재고를 변경하는 경우

System operations of this use case:

stockInfoRequest(재고정보를 요청한다),

stockInfoResponse(재고정보를 반환한다),

stockChangeRequest(재고 변경 정보를 보낸다),

changeStockResponse(DB의 재고정보 갱신후 받은 재고 갱신 완료정보를 반환한다)

Interface	DataManager
Use Cases	<p>UC-1: 상품선택 버튼을 누른다. 사용자가 상품을 구매하기 위해 DVM에 접근하여 초기 화면의 '상품선택', '코드입력', '관리자' 세가지 옵션 중 '상품선택' 버튼을 누른 경우</p> <p>System operations of this use case:</p> <p>itemRequest(상품정보를 요청한다),</p> <p>itemResponse(상품정보를 DB에서 받아 반환한다)</p>
	<p>UC-3: 코드를 환불한다. 사용자가 '코드입력' 버튼을 눌러 코드입력 화면으로 넘어간 경우, 코드를 입력하지 않고 코드 환불을 위해 '코드 환불' 버튼을 누른 경우</p> <p>System operations of this use case:</p> <p>refundRequest(코드환불요청 정보를 보낸다),</p> <p>refundResponse(코드환불 완료정보를 보낸다)</p>
	<p>UC-5: 상품을 선택한다. 사용자가 상품을 구매하기 위해 DVM에 접근하여 '상품선택' 버튼을 눌러 상품 목록을 확인 후 구매를 원하는 상품을 선택한 경우</p> <p>System operations of this use case:</p> <p>itemRequest(상품정보를 요청한다),</p> <p>itemResponse(상품정보를 DB에서 받아 반환한다),</p> <p>selectItem(사용자가 선택한 상품정보를 받는다)</p>
	<p>UC-6: DVM에 재고가 있거나, 다른 재고가 있는 DVM 조회를 마친 경우 QR 결제를 요청한다. 사용자가 구매하려는 상품을 선택하여 결제를 요청하는 과정</p> <p>System operations of this use case:</p>

selectItem(사용자가 선택한 상품정보를 받는다),
selectItemQR(사용자가 선택한 상품에 대한 결제정보를 통해 QR모듈로부터 받은 QR코드를 반환한다),
finishRequest(결제 완료 정보를 보낸다),
finishResponse(결제 완료정보를 반환한다)

UC-7: 재고가 없을 경우, 재고가 있는 다른 DVM을 조회한다. 사용자가 구매하려는 제품의 재고가 없는 경우에 해당 제품의 재고가 있는 다른 DVM을 조회하는 과정

System operations of this use case:

sendDVMList(재고가 있는 DVM 정보들을 반환한다),
selectDVM(사용자가 선택한 DVM 정보를 넘긴다)

UC-8: DVM의 재고정보를 네트워크 상에 동기화한다. DVM에서 상품 구매 완료 등으로 인해 재고가 변경될 경우 이 정보가 네트워크 상에서 동기화되는 경우

System operations of this use case:

finishResponse(결제완료정보를 DB에 넘겨 재고정보를 갱신한다),
sendCode(사용자가 선택한 DVM 과 상품에 대한 결제완료정보를 DB에 넘기고 상품수령 코드를 반환한다)

UC-9: DVM의 네트워크 상태를 점검한다: 5분간격으로 네트워크 상태를 점검하고, 문제 발생 시 관리자에게 메시징한다.

System operations of this use case:

networkStatus(네트워크 상태를 넘긴다),
networkStatusResponse(네트워크 문제가 발생한 경우, DB에 전송하고 관리자에게 메시징한다. 문제가 발생하지 않은 경우 확인 정보를 반환한다)

	<p>UC-11: 관리자 상태에서 DVM의 재고를 변경한다. 관리자가 DVM에 재고를 재정보기한 후 이 정보를 시스템에 입력하기 위해 DVM에 접근하여 초기 화면에서 '관리자' 버튼을 선택해 재고를 변경하는 경우</p> <p>System operations of this use case:</p> <p>stockInfoRequest(재고정보를 요청한다),</p> <p>stockInfoResponse(재고정보를 반환한다),</p> <p>stockChangeRequest(재고 변경 정보를 보낸다),</p> <p>changeStockResponse(DB의 재고정보 갱신 후 받은 재고 갱신 완료정보를 반환한다)</p>
--	--

Interface	NetworkManager(server)
Use Cases	<p>UC-1: 상품선택 버튼을 누른다. 사용자가 상품을 구매하기 위해 DVM에 접근하여 초기 화면의 '상품선택', '코드입력', '관리자' 세가지 옵션 중 '상품선택' 버튼을 누른 경우</p> <p>System operations of this use case:</p> <p>itemRequest(상품정보를 요청한다),</p> <p>itemResponse(상품정보를 DB에서 받아 반환한다)</p> <p>UC-3: 코드를 환불한다. 사용자가 '코드입력' 버튼을 눌러 코드입력 화면으로 넘어간 경우, 코드를 입력하지 않고 코드 환불을 위해 '코드 환불' 버튼을 누른 경우</p> <p>System operations of this use case:</p> <p>refundRequest(코드환불요청 정보를 보낸다),</p> <p>refundResponse(코드환불 완료정보를 보낸다)</p>

UC-5: 상품을 선택한다. 사용자가 상품을 구매하기 위해 DVM에 접근하여 '상품선택' 버튼을 눌러 상품 목록을 확인 후 구매를 원하는 상품을 선택한 경우

System operations of this use case:

itemRequest(상품정보를 요청한다),
itemResponse(상품정보를 DB에서 받아 반환한다),
selectItem(사용자가 선택한 상품정보를 받는다)

UC-6: DVM에 재고가 있거나, 다른 재고가 있는 DVM 조회를 마친 경우 QR 결제를 요청한다. 사용자가 구매하려는 상품을 선택하여 결제를 요청하는 과정

System operations of this use case:

selectItem(사용자가 선택한 상품정보를 받는다),
selectItemQR(사용자가 선택한 상품에 대한 결제정보를 통해 QR모듈로부터 받은 QR코드를 반환한다),
finishRequest(결제 완료 정보를 보낸다),
finishResponse(결제 완료정보를 반환한다)

UC-7: 재고가 없을 경우, 재고가 있는 다른 DVM를 조회한다. 사용자가 구매하려는 제품의 재고가 없는 경우에 해당 제품의 재고가 있는 다른 DVM를 조회하는 과정

System operations of this use case:

sendDVMList(재고가 있는 DVM 정보들을 반환한다),
selectDVM(사용자가 선택한 DVM 정보를 넘긴다)

UC-8: DVM의 재고정보를 네트워크 상에 동기화한다. DVM에서 상품 구매 완료 등으로 인해 재고가 변경될 경우 이 정보가 네트워크 상에서 동기화되는 경우

	<p>System operations of this use case:</p> <p>finishResponse(결제완료정보를 DB에 넘겨 재고정보를 갱신한다), sendCode(사용자가 선택한 DVM 과 상품에 대한 결제완료정보를 DB에 넘기고 상품수령 코드를 반환한다)</p> <hr/> <p>UC-9: DVM의 네트워크 상태를 점검한다: 5분간격으로 네트워크 상태를 점검하고, 문제 발생 시 관리자에게 메시징한다.</p> <p>System operations of this use case:</p> <p>networkStatus(네트워크 상태를 넘긴다), networkStatusResponse(네트워크 문제가 발생한 경우, DB에 전송하고 관리자에게 메시징한다. 문제가 발생하지 않은 경우 확인 정보를 반환한다)</p> <hr/> <p>UC-11: 관리자 상태에서 DVM의 재고를 변경한다. 관리자가 DVM에 재고를 재정비한 후 이 정보를 시스템에 입력하기 위해 DVM에 접근하여 초기 화면에서 '관리자' 버튼을 선택해 재고를 변경하는 경우</p> <p>System operations of this use case:</p> <p>stockInfoRequest(재고정보를 요청한다), stockInfoResponse(재고정보를 반환한다), stockChangeRequest(재고 변경 정보를 보낸다), changeStockResponse(DB의 재고정보 갱신후 받은 재고 갱신 완료정보를 반환한다)</p>
--	--

2-3. Non-functional requirement

DVM:Qualities

- Robustness: 사용자의 비정상적인 입력 값에 대하여 올바른 예외처리와 오류경고를 출력할 수 있어야한다.

- Security: 외부에서 네트워크에 접근하려고 할 때, 30초 내로 접근을 방지하고 관리자에게 관리자에게 외부 접근 알림을 전송한다.
- Availability: 인증코드의 유효기간은 일주일이며 유효기간내에는 환불이 가능하다
- Usability: DVM는 항상 동기화되어, 정확한 재고를 제공해야 한다.
- Scalability: DVM 수가 늘어날 경우 시스템 성능문제에 영향을 주지 않아야 한다.

DVM:Constrains

- 최소 1000 명의 동시 사용자를 지원해야 한다.
- 보안 관련한 문제가 발생할 경우 이와 관련한 정보가 관리자에게 빠르게 전달되어야 한다.
- 네트워크 성능은 5 분 간격으로 점검할 수 있어야 한다.
- 총 개발인원이 3 명을 넘지 않아야 한다.

DVM:Principle

- 사용자의 비정상적인 입력값은 올바른 형식으로 수정하고 재시도함으로써 결함을 잠재적으로 감추는 exception handling으로 처리한다. 이 원칙은 Robustness를 지원한다.
- 외부에서의 비정상적인 네트워크 접근은 보안 전술 중 detect intrusion으로 해결하며, 네트워크 트래픽 또는 시스템 안에서의 서비스 요청 패턴과 시그니처 집합, 또는 데이터베이스에 저장된 악의적 행위패턴과 비교하여 탐지한다. 또한 이를 통해 탐지한 공격에 대해 공격 대응 방법 중 inform actors, 즉 공격이 의심되거나 탐지될 때 운영자, 다른 직원, 현재 시스템에서는 관리자에게 알려준다. 이 두가지 원칙은 Security를 지원한다.
- 사용자가 cancel 명령을 내릴 때 시스템은 모든 사항에 대해 그 cancel명령을 듣고 종료되어야 하고, 어떤 리소스든 해제되어야 한다. 그리고 취소된 사항은 협력하는 컴포넌트에 고지되어야 한다. 이 시스템에서는 주문 취소에 대한 다른 DVM의 동기화를 의미하며, 이 원칙은 Availability를 지원한다.
- Maintain system model 원칙을 따라, DVM을 항상 동기화하여 정확한 재고를 정하여 명확한 모델을 유지한다. 이 원칙을 통해 Usability를 지원한다.
- 이 시스템은 increase resources를 항상 고려한다. DVM 수가 늘어날 경우가 항상 고려되어 추가 프로세서, 추가 메모리를 통해 시간을 감소시킬 수 있는 잠재력을 갖고있어 이 원칙을 통해 Scalability를 지원한다.

3. Structure

3-1. Overview

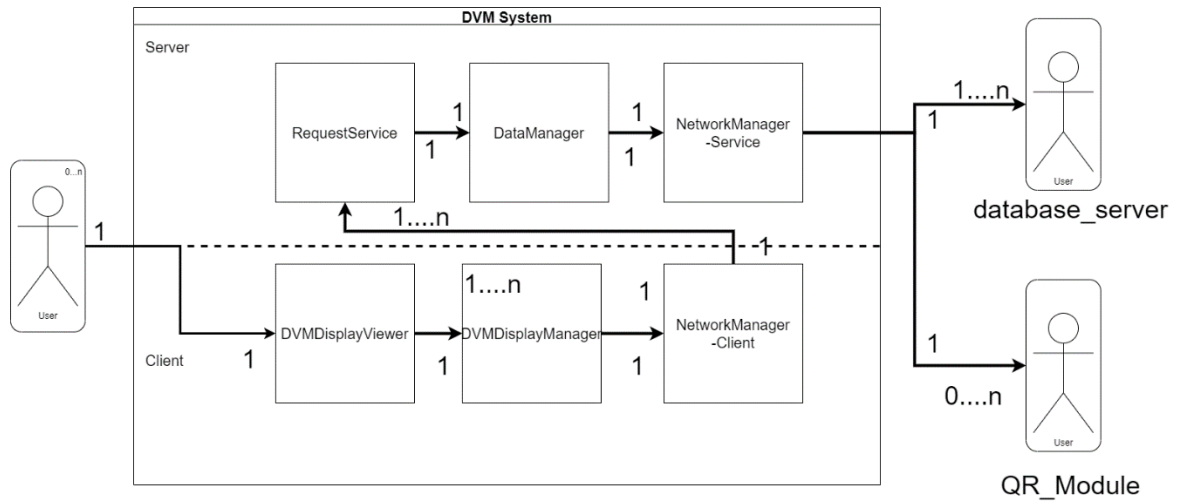


Figure 2. DVM overview architecture diagram

DVM system은 모든 DVM 기기와 연결되어 있는 단일 database server로 이루어져 있다. 각 DVM 마다 Client 구조를 포함하고 있으며 Client는 DVMDisplayViewer, DVMDisplayManager로 이루어져 있으며 NetworkManger-Client가 Server의 RequestService와 통신하여 DataManager와 NetworkManager-Server를 통해 DB와 외부모듈인 QR모듈에 Client의 request를 전달한다. QR모듈은 결제에 사용되는 모듈로 결제 성공-실패 여부를 전달해준다.

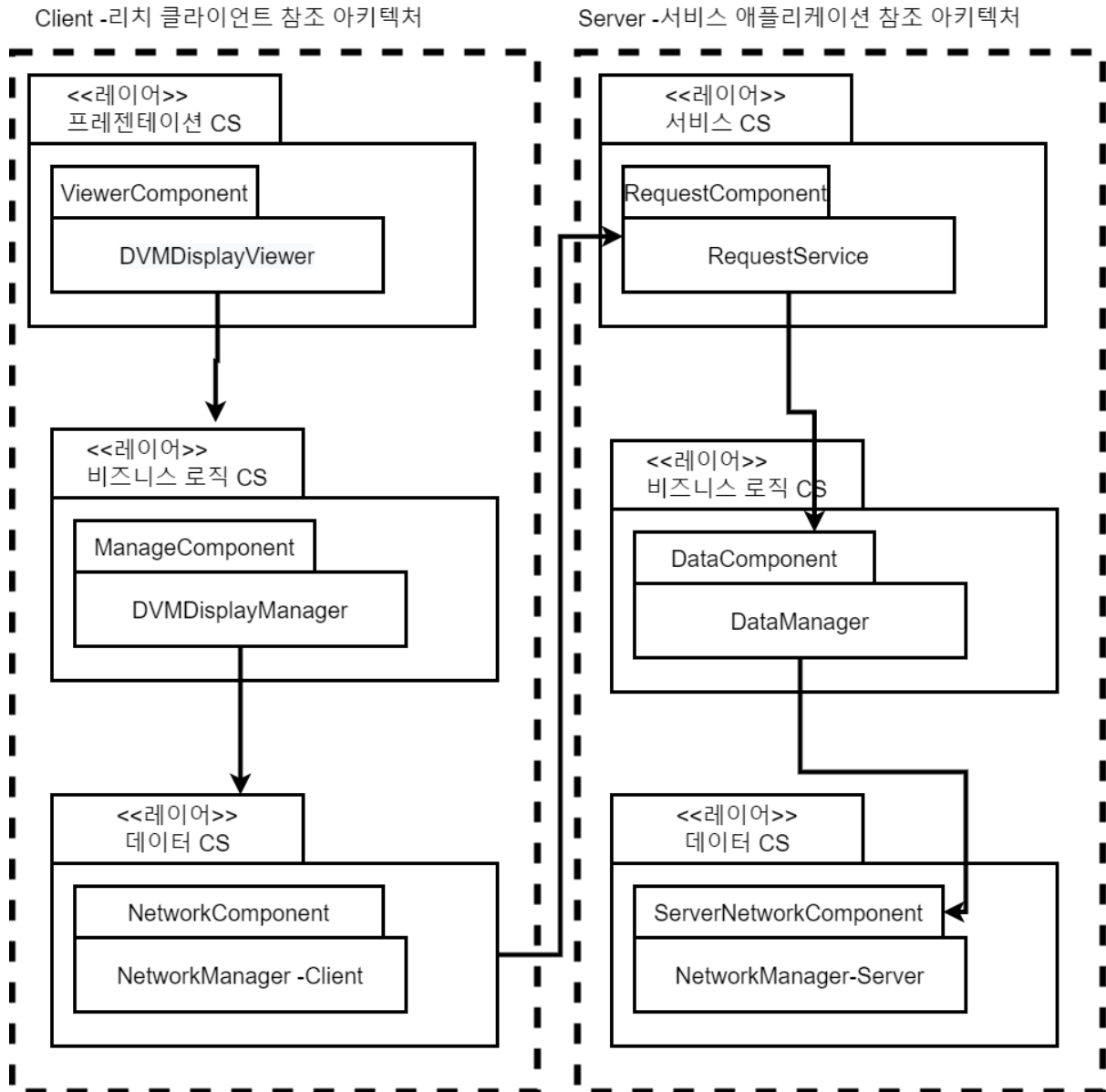


Figure 3 DVM Layered Diagram

Commentary
<ul style="list-style-type: none"> - Component : Components는 시스템 또는 응용프로그램 중 재활용가능한 일부분으로 소프트웨어의 복잡성을 관리 가능한 부분으로 나눌 수 있게 해준다. - Layering of Components : 상위 구조는 서버-클라이언트 구조이며, 클라이언트는 리치 클라이언트 참조 아키텍처를, 서버는 서비스 애플리케이션 참조 아키텍처를 사용한다. 각각의 아키텍처 내 구조는 layered Sytle 아키텍처가 선택되었다. : 프레젠테이션 layer의 컴포넌트는 비즈니스로직 layer의 컴포넌트에 저장된 정보에, 비즈니스로직 layer의 컴포넌트는 데이터 layer에 접근하고 정보를 업데이트한다.

3-2. Components

Component	ViewerComponent
Responsibilities	Viewer는 사용자의 입력에 따라 화면을 보여준다, 제공되는 interface: DVMDisplayViewer
Collaborators	Other components that the component interacts with.
Notes	Information about multiplicity, concurrency, persistency, parameterization, etc.
Issues	List of issues that remain to be resolved.

Component	ManageComponent
Responsibilities	ManageComponent는 사용자의 입력을 받은 Viewer에서 받은 값에 따라 화면의 상태를 전환한다. 제공되는 interface: NetworkManager-Client
Collaborators	Other components that the component interacts with.
Notes	Information about multiplicity, concurrency, persistency, parameterization, etc.
Issues	List of issues that remain to be resolved.

Component	ClientNetworkComponent
Responsibilities	ClientNetworkComponent는 RequestComponent에 신호를 보내는 역할을 한다. 제공되는 interface: NetworkManager-
Collaborators	Other components that the component interacts with.
Notes	Information about multiplicity, concurrency, persistency, parameterization, etc.
Issues	List of issues that remain to be resolved.

Component	RequestComponent
Responsibilities	RequestComponent는 ClientNetworkComponent에서 받은 신호를 DataComponent로 보내는 역할을 한다. 제공되는 interface: RequestService
Collaborators	Other components that the component interacts with.
Notes	Information about multiplicity, concurrency, persistency, parameterization, etc.
Issues	List of issues that remain to be resolved.

Component	DataComponent
Responsibilities	DataComponent는 RequestComponent에서 받은 신호를 ServerNetworkComponent로 보내는 역할을 한다. 제공되는 interface: Datamanager
Collaborators	Other components that the component interacts with.
Notes	Information about multiplicity, concurrency, persistency, parameterization, etc.
Issues	List of issues that remain to be resolved.

Component	ServerNetworkComponent
Responsibilities	ServerNetworkComponent는 Database를 업데이트하고, 변경 사항을 ManageComponent로 보내는 역할을 한다. 제공되는 interface: NetworkManager-Server
Collaborators	Other components that the component interacts with.
Notes	Information about multiplicity, concurrency, persistency, parameterization, etc.
Issues	List of issues that remain to be resolved.

3-3. Interfaces

Interface	DVMDisplayView
Description	DVM Display 화면 생성 인터페이스
Operations	x
Protocol	no restriction on order of operations
Notes	DVMDisplayManager 인터페이스로부터 화면 정보를 제공받는다.
Issues	x

Interface	DVMDisplayManager
Description	DVM의 Display 화면 정보 관리 인터페이스
Operations	MenuItem, viewItem, menuCode, viewCode, menuRefund, viewRefundFinish, menuAdmin, viewAdmin, selectItem, viewQR, finishRequest, viewFinish, viewDVMList, selectDVM, viewSendCode, stockInfoRequest, viewStock, stockChangeRequest, viewChangeStockFinish
Protocol	no restriction on order of operations
Notes	DVMDisplayView로부터 사용자 인터랙션을 받아 NetworkManager로 보내고, NetworkManager-Client로부터 받은 정보를 DVMDisplayView로 보낸다.
Issues	x

Interface	NetworkManager-Client
Description	서버와의 데이터 통신 인터페이스
Operations	ItemRequest, itemResponse, refundRequest, refundResponse, selectItem, selectItemQR, finishRequest, finishResponse, sendDVMList, selectDVM, sendCode, networkStatus, networkStatusResponse, stockInfoRequest, stockInfoResponse, stockChangeRequest, changeStockResponse
Protocol	no restriction on order of operations
Notes	DVMDisplayManager로부터 받은 정보를 RequestService를 통해 서버로 전달하고, RequestService를 통해 서버로부터 받은 정보를 DVMDisplayManager로 보낸다.
Issues	x

Interface	RequestService
Description	클라이언트와의 데이터 교류 인터페이스
Operations	ItemRequest, itemResponse, refundRequest, refundResponse, selectItem, selectItemQR, finishRequest, finishResponse, sendDVMList, selectDVM,

	finishResponse, sendCode, networkStatus, networkStatusResponse, stockInfoRequest, stockInfoResponse, stockChangeRequest, changeStockResponse
Protocol	no restriction on order of operations
Notes	클라이언트로부터 받은 정보를 DataManager로 전달하고, DataManager로부터 받은 정보를 클라이언트로 전송한다.
Issues	x

Interface	DataManager
Description	데이터베이스 서버와 교류되는 데이터 관리 인터페이스
Operations	ItemRequest, itemResponse, refundRequest, refundResponse, itemRequest, itemResponse, selectItem, selectItemQR, finishRequest, finishResponse, sendDVMList, selectDVM, sendCode, networkStatus, networkStatus, networkStautsResponse, stockInfoRequest, stockInfoResponse, stockChangeRequest, changeStockResponse
Protocol	no restriction on order of operations
Notes	RequestService로부터 받은 정보를 NetworkManager-Server를 통해 데이터베이스서버로 전송하고, NetworkManager-Server로부터 받은 정보를 RequestService로 보낸다.
Issues	x

Interface	Networkmanager-Server
Description	데이터베이스 서버와의 데이터 교류 인터페이스
Operations	ItemRequest, itemResponse, refundRequest, refundResponse, itemRequest, itemResponse, selectItem, selectItemQR, finishRequest, finishResponse, sendDVMList, selectDVM, sendCode, networkStatus, networkStatus, networkStautsResponse, stockInfoRequest, stockInfoResponse, stockChangeRequest, changeStockResponse
Protocol	no restriction on order of operations
Notes	데이터베이스서버로부터 받은 데이터를 DataManager로 보낸다.
Issues	x

4. Dynamic Behavior

4-1. Scenarios

4-1-1. Scenario Specification

- Scenarios

UC-1: 상품선택 버튼을 누른다. (all dvms)

UC-2: 코드입력 버튼을 누른다.(all dvms)

UC-3: 코드를 환불한다.(all ,data, syncro)

UC-4: 관리자 버튼을 누른다. (all)

UC-5: 상품을 선택한다. (all)

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.

Use case	상품선택 버튼을 누른다.
Description	사용자가 상품종류를 고를 수 있게 하는 오퍼레이션
Actors	유저
Steps	1. 상품을 고르기 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 상품선택 버튼 클릭 2. 화면에 상품종류 출력

Use case	코드입력 버튼을 누른다.
Description	사용자가 다른 DVM에서 받은 상품 수령 코드를 통해 상품을 수령받기 위한 오퍼레이션
Actors	유저
Steps	1. 다른 DVM에서 수령한 상품 수령코드를 통해 상품을 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 코드입력 버튼 클릭 2. '코드입력', '코드환불' 버튼 화면에 출력

Use case	코드를 환불한다.
Description	구매한 제품을 환불하기 위한 오퍼레이션
Actors	유저

Steps	<ol style="list-style-type: none"> 1. 다른 DVM에서 수령한 상품 수령코드를 통해 상품을 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 코드입력 버튼 클릭 2. '코드입력', '코드환불' 버튼 중 '코드환불'버튼 클릭 3. 환불할 상품 수령 코드 입력 4. QR모듈을 통한 환불
-------	---

Use case	관리자 버튼을 누른다.
Description	관리자가 DVM에 재고를 변경하기 위한 오퍼레이션
Actors	관리자
Steps	<ol style="list-style-type: none"> 1. 재고를 추가하기 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 관리자 버튼 클릭

Use case	상품을 선택한다.
Description	사용자가 상품을 구매하기 위해 상품을 선택하기 위한 오퍼레이션
Actors	유저
Steps	<ol style="list-style-type: none"> 1. 상품을 고르기 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 상품선택 버튼 클릭 2. 상품 목록에서 구매하려는 상품선택

Use case	자판기에 재고가 있거나, 다른 재고가 있는 자판기 조회를 마친 경우 QR결제를 요청한다.
Description	DVM의 재고 조회 및 결제를 위한 오퍼레이션
Actors	유저
Steps	<ol style="list-style-type: none"> 1. 상품을 고르기 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 상품선택 버튼 클릭 2. 상품목록 중 구매할 상품 선택 3. 현재 DVM의 해당 상품 재고조회 4. 재고가 있을 경우 QR결제 요청

Use case	재고가 없을 경우 재고가 있는 다른 자판기를 조회한다.
Description	다른 DVM의 재고 조회를 위한 오퍼레이션
Actors	유저
Steps	<ol style="list-style-type: none"> 1. 상품을 고르기 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 상품선택 버튼 클릭 2. 상품목록 중 구매할 상품선택 3. 현재 DVM의 해당 상품 재고 조회

	4. 재고가 없을 경우, 해당 상품의 재고가 있는 다른 DVM의 재고조회
--	--

Use case	자판기의 재고정보를 네트워크 상에 동기화한다.
Description	DVM의 재고정보가 변경된 경우 이를 네트워크 상에 동기화하는 오퍼레이션
Actors	유저
Steps	<ol style="list-style-type: none"> 1. 상품 구매 등으로 인한 재고정보의 변동 2. 네트워크를 통한 재고정보의 동기화

Use case	자판기의 네트워크 상태를 점검한다.
Description	각 DVM의 네트워크 상태를 점검하기 위한 오퍼레이션
Actors	관리자
Steps	<ol style="list-style-type: none"> 1. 5분 간격으로 각 DVM에서 네트워크로 현재 DVM의 네트워크 상태를 낸보낸다. 2. 이상이 있을 시에 관리자에게 이메일로 문제상황을 전달한다.

Use case	결제가 이루어진 자판기에서 선결제한 상품을 제공할 자판기로 코드를 전송한다.
Description	사용자가 구매하려는 상품의 재고가 없을 경우 다른 DVM에서 상품을 수령할 수 있도록 다른 DVM에 상품수령코드를 보내는 오퍼레이션
Actors	유저
Steps	<ol style="list-style-type: none"> 1. 상품을 고르기 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 상품선택 버튼 클릭 2. 상품목록 중 구매할 상품선택 3. 현재 DVM의 해당 상품 재고 조회 4. 재고가 없을 경우, 해당 상품의 재고가 있는 다른 DVM의 재고조회 5. 상품을 수령할 DVM 선택 6. 선택한 DVM으로 수령코드 전송

Use case	관리자 상태에서 자판기의 재고를 변경한다.
Description	DVM에 물류 정리 등으로 인한 외부적인 재고 변경 시 데이터베이스에 동기화하기 위한 오퍼레이션
Actors	관리자
Steps	<ol style="list-style-type: none"> 1. 재고를 추가하기 위해 초기 화면의 '상품선택', '코드입력', '관리자' 버튼 중 관리자 버튼 클릭 2. 변경할 재고 정보 입력 3. 동기화

4-1-2. Component Interaction Model

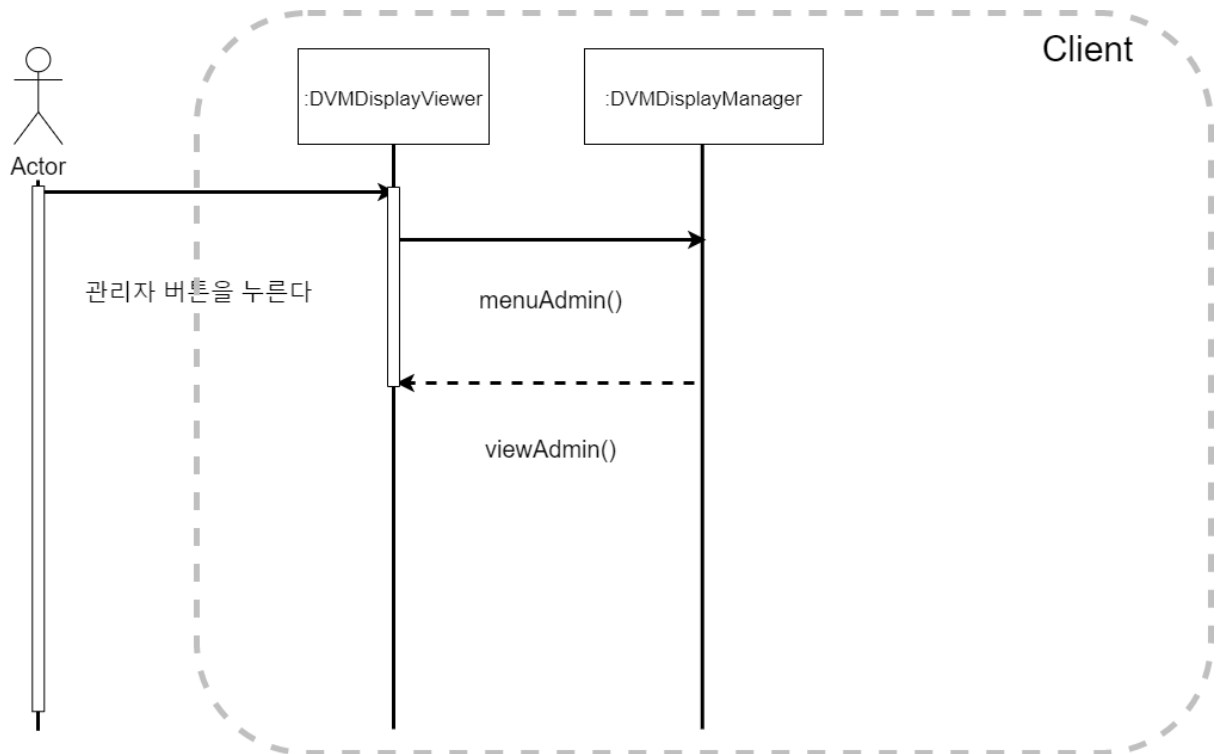


Figure 4. Use Case 2 Sequence Diagram

UC-2 사용자가 메뉴에서 관리자 로그인을 선택 화면에 관리자 로그인 화면 출력

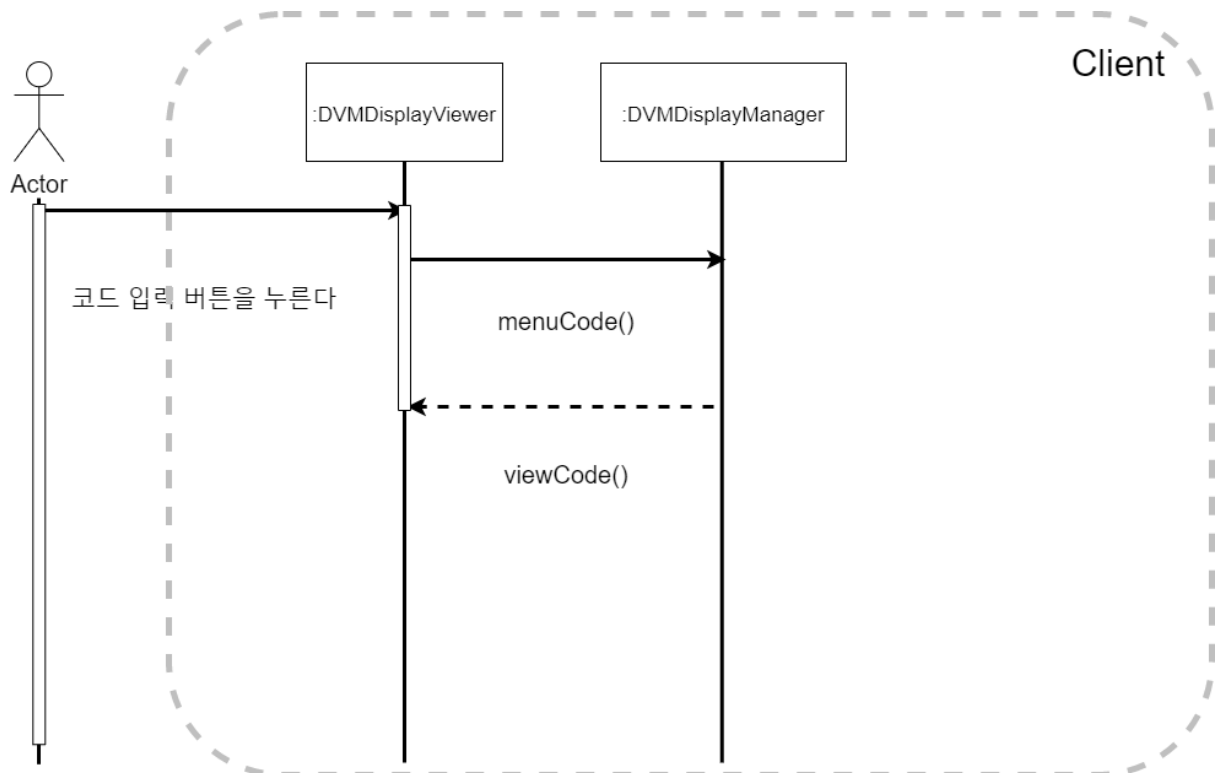


Figure 5. Use Case 4 Sequence Diagram

UC-4 사용자가 메뉴에서 코드 입력을 선택 화면에 코드 입력 화면 출력

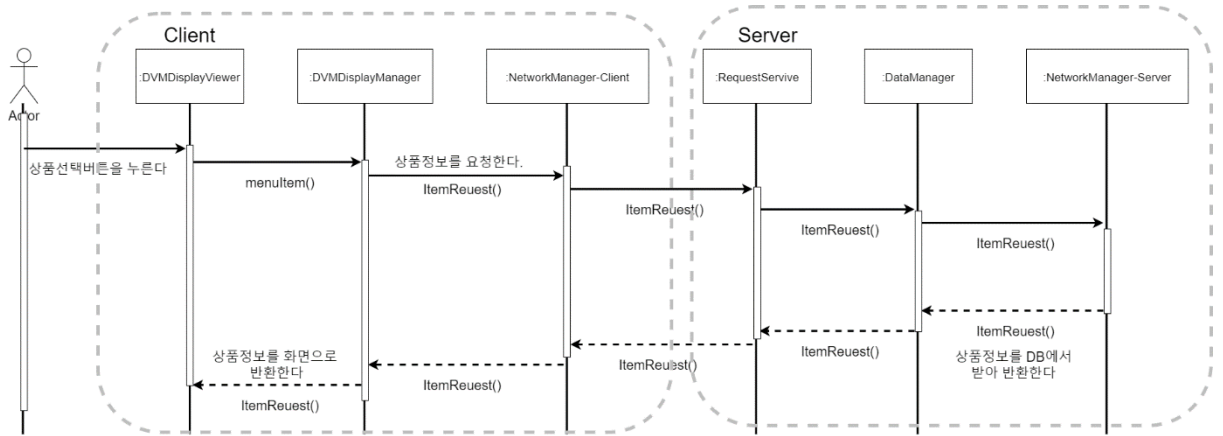


Figure 6. Use Case 1 Sequence Diagram

UC-1 사용자가 상품을 선택하고 해당 상품에 대한 정보를 Server가 DB에서 반환

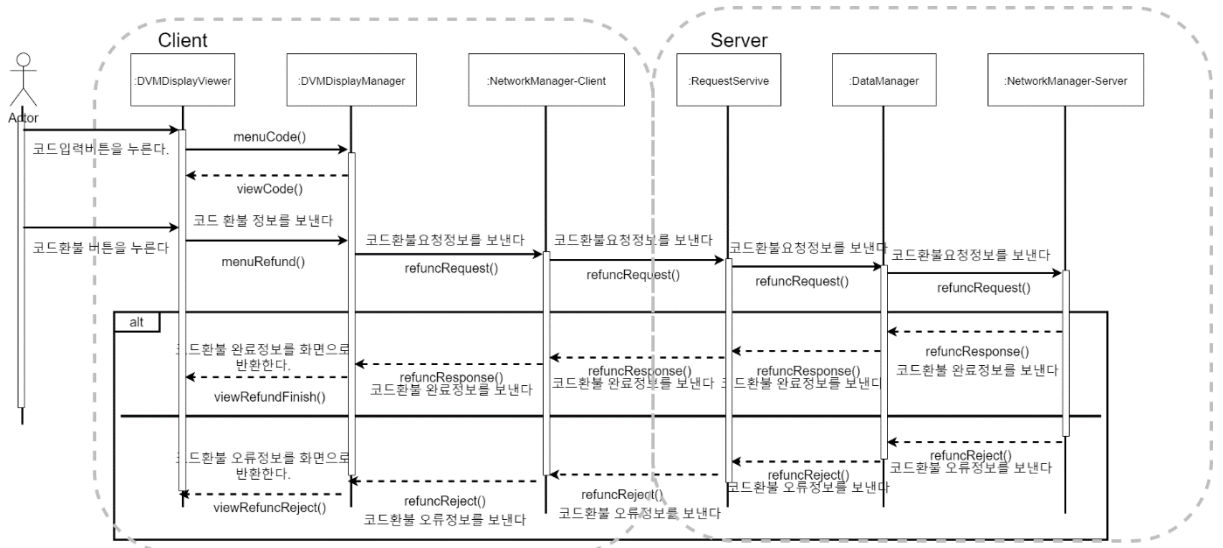
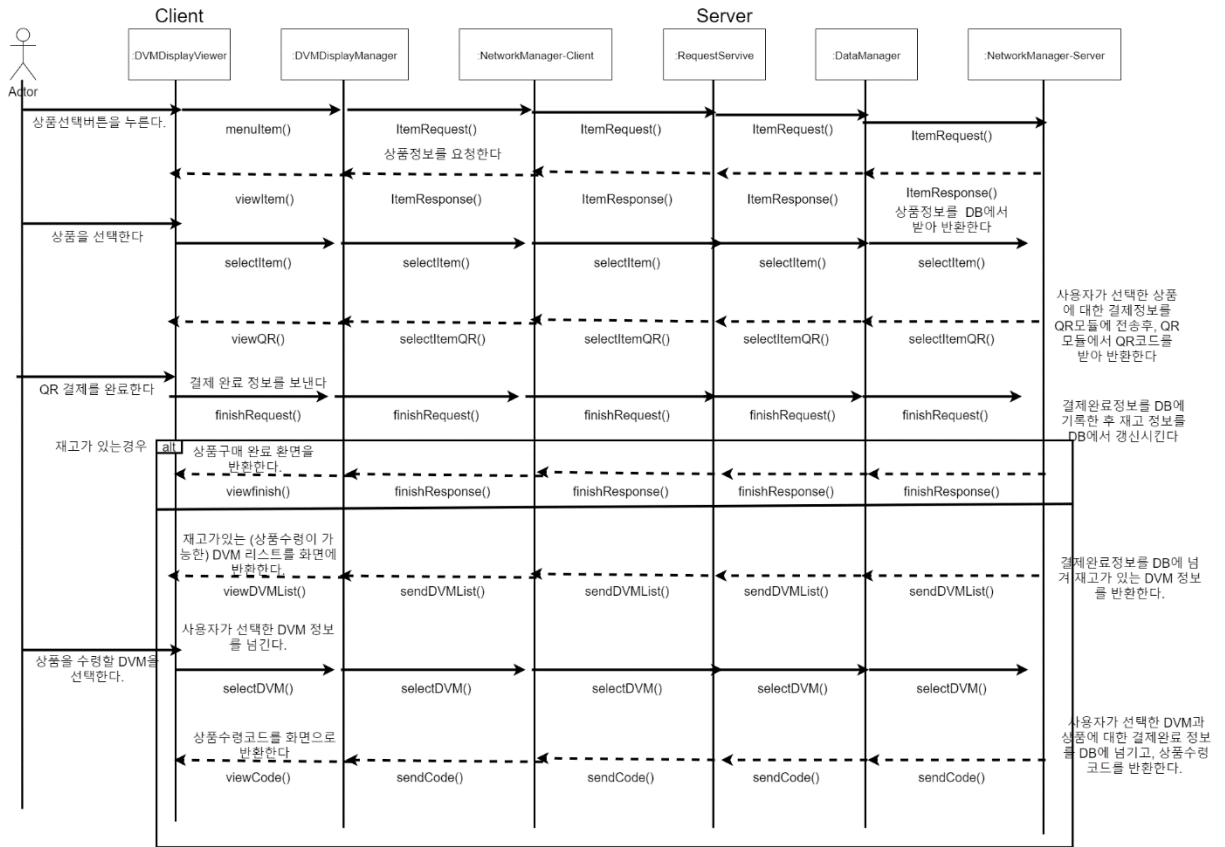
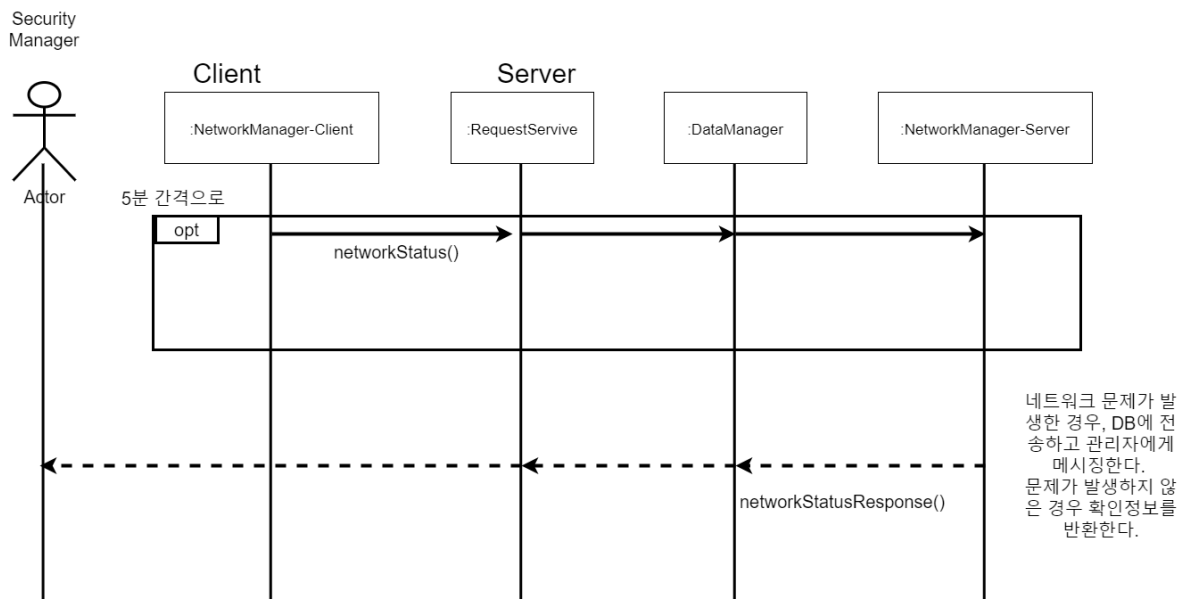


Figure 7. Use Case 3 Sequence Diagram

UC-3 사용자가 코드 환불을 선택하여 진행, QR 모듈의 반환 값에 따라 성공, 실패 화면에 반환



UC-5,6,7,8,10 사용자가 상품을 선택하고 해당 DVM에 재고가 있는지 확인 후 결제를 진행, 재고가 없을 경우 다른 DVM의 재고를 조회하고 선결제를 진행한다.



UC-9 네트워크 상태정보를 5분마다 전송, 문제가 발생할 경우 보안관리자에게 메일 전송

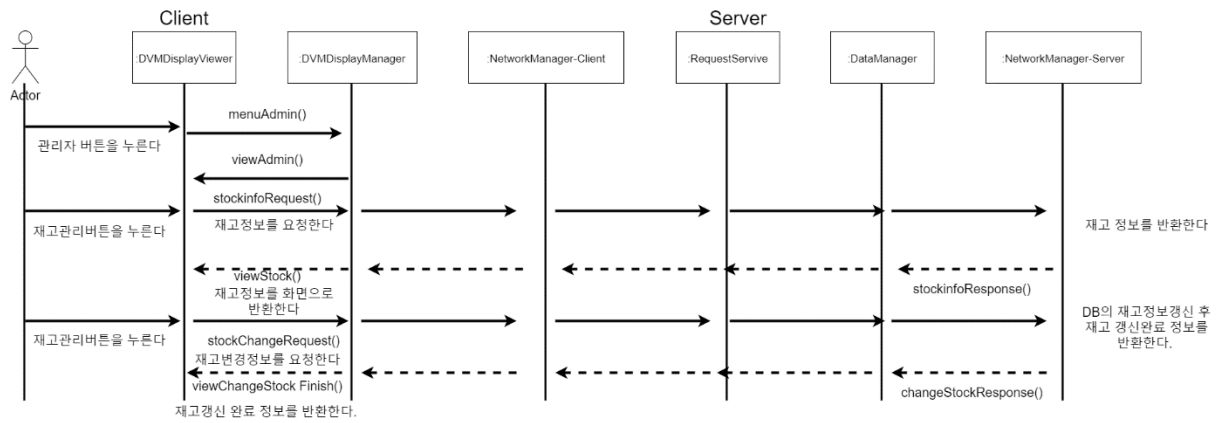


Figure 10. Use Case 11 Sequence Diagram

UC-11 사용자가 관리자 로그인을 진행하고 DVM의 재고를 조정

4-2. Mechanisms

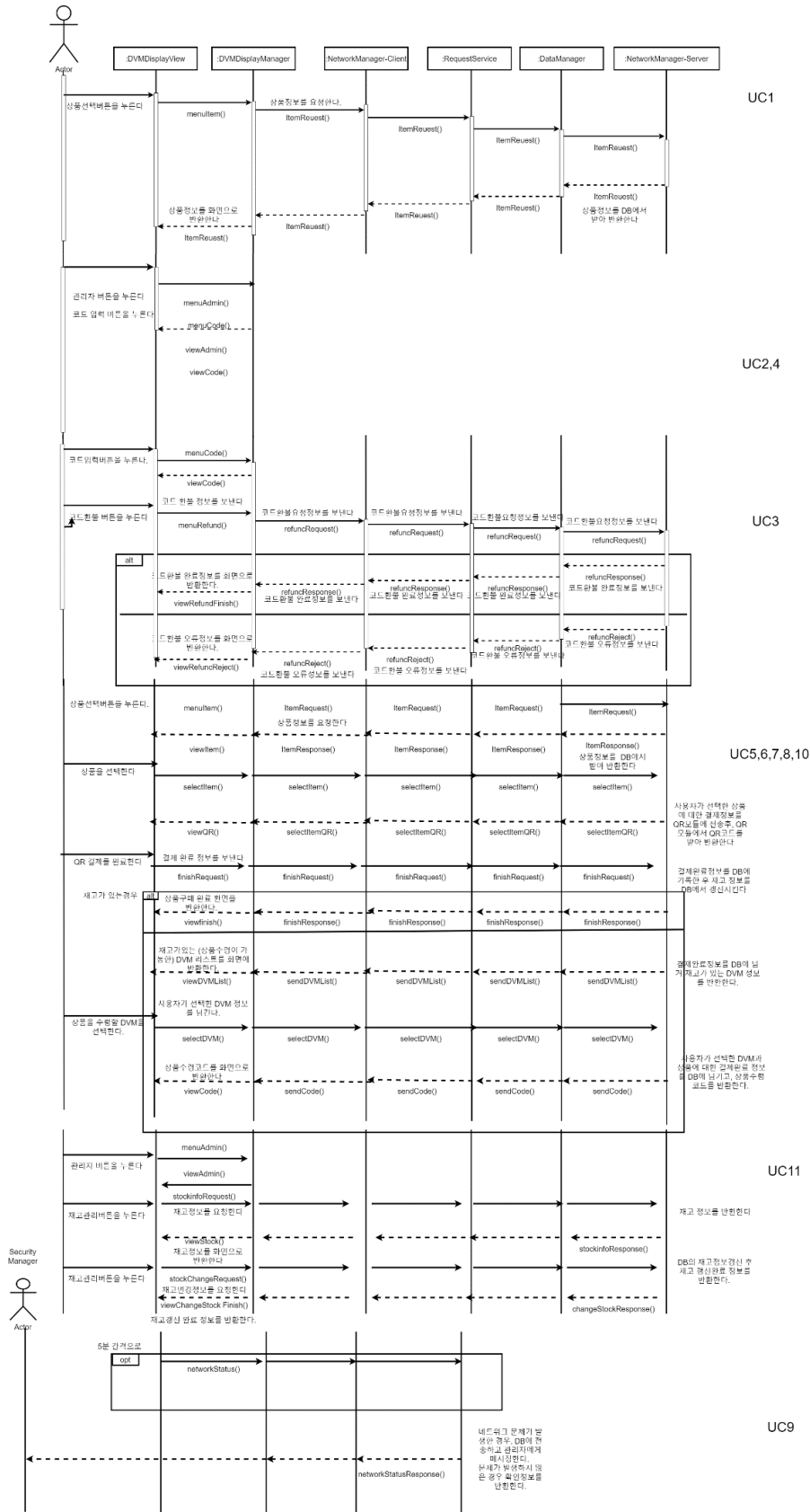


Figure 11. Mechanism of DVM System

5. Other Views Section

5-1. Process View

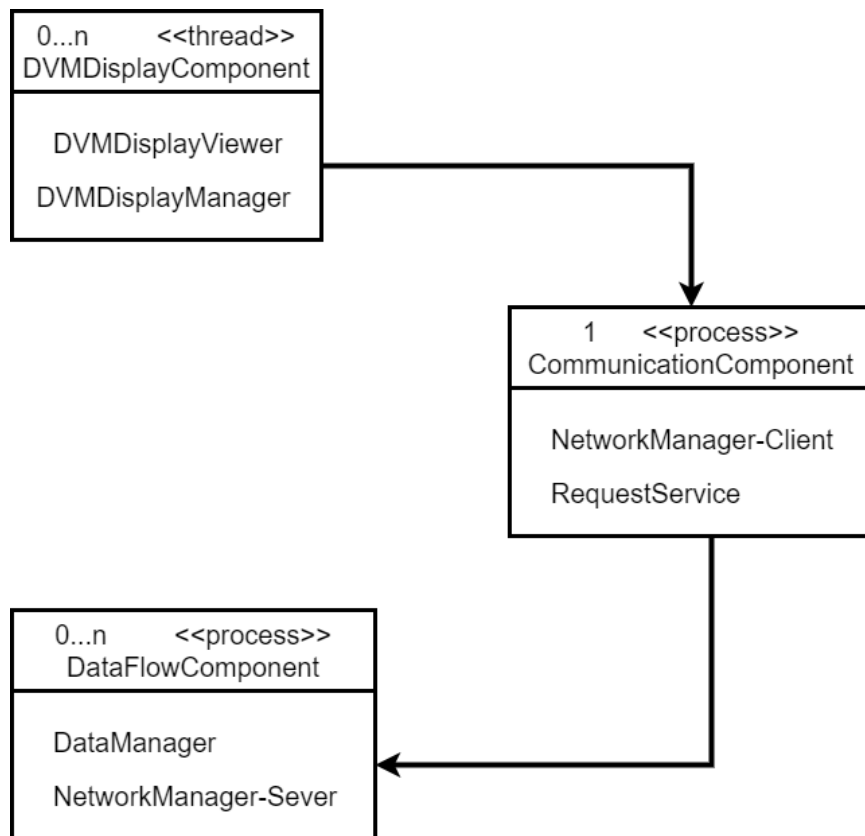


Figure 12. DVM Process View

5-2. Development View

Implementation mapping
<ul style="list-style-type: none">- For each logical component a directory is created that contains all code files belonging to that component.- For each class a separate file is created.

5-3. Physical View

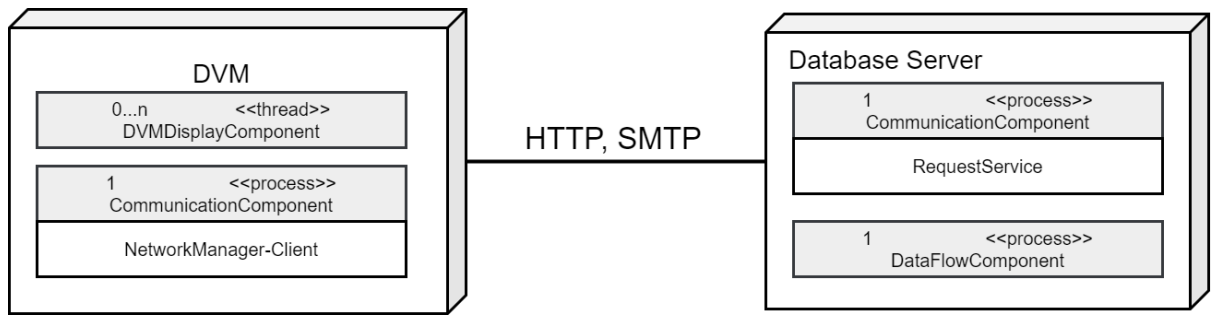


Figure 12. DVM Physical View

6. Conceptual Framework

6-1. Domain Lexicon

해당 없음.

6-2. Lexicon Diagram

해당 없음.

7. Conclusion

- Advantages and known limitations of the chosen solution:

선택된 솔루션의 장점은 아키텍처의 Context나 문제점 해결 과정에서 서술되었으나, 한계점에 대해서 문서에 작성되지 않음

- Well non-functional requirements are met:

2.3절에 설명됨.

- Inconsistencies:

해당 없음