

**Software Design
Specification
For
Distributed Vending Machine**

Team 4

201711836 송호영

201811217 이해인

201710240 이해림

Table of Contents

1. Introduction.....	3p
2. Purpose	
3. Scope	
4. Definitions, acronyms, and abbreviations	
5. References	
6. Overview	
2. System architecture description.....	4p
3. Real Use Cases and Interaction Diagrams.....	5p
3.1.n Use Cases and Interaction Diagrams	
4. Class Diagram.....	18p
5. Traceability Analysis.....	19p

1. Introduction

1.1 Purpose

본 문서는 Distributed Vending Machine(DVM)의 Controller System를 구현하기 위한 디자인을 명시한 문서이다.

1.2 Scope

DVM은 자동 판매기간 분산 네트워크를 통해 재고가 없는 상황에도 소비자가 원하는 물품을 구매할 수 있도록 재고를 소유한 자판기의 위치 정보를 전달한다. 소비자가 원한다면 선결제를 통해 음료를 미리 구매하여 선점할 수 있으며, 인증코드와 바코드를 받는다. 이후 목표 자판기에 인증코드/바코드를 입력하면 음료를 받을 수 있다. 본 문서에서 기술하는 DVM Controller System은 DVM이 기능대로 동작할 수 있도록 하는 소프트웨어이다.

1.3 Definitions, acronyms, and abbreviations

Num	Term	Description
1	SW	Software의 약자
2	DVM	Distributed Vending Machine의 약자
3	선결제	사용자가 현 DVM에서 구매 불가능한 물품을 타DVM에서 구매하고자 할 때 이용하는 시스템이다. 사용자는 현 DVM에서 미리 결제를 진행한 뒤 인증번호 및 바코드를 출력 받아 해당 자판기에서 결제된 음료를 전달받을 수 있다.
4	C_Number	사용자가 선결제를 한 경우 해당 DVM에서 음료를 전달받기 위한 인증 번호이다. 이외에 관리자는 관리자 메뉴로 접속하기 위해 고유한 인증번호를 갖고있다.
5	Msg	Message의 준말로 DVM이 네트워크에 연결된 다른 DVM과 정보를 주고받는 데이터 단위이다. Message Protocol에 형식이 정의된다.
6	Title	DVM에 등록된 각각의 음료 이름
7	Title ID	DVM에 등록된 각각의 음료 번호
8	item	DVM에 등록된 Title의 재고
9	Address	DVM에 갖고 있는 자신의 위치
10	ID	네트워크 내 DVM의 고유번호
11	Broadcast Msg	DVM네트워크 내 모든 DVM에게 보내는 메시지

1.4 References

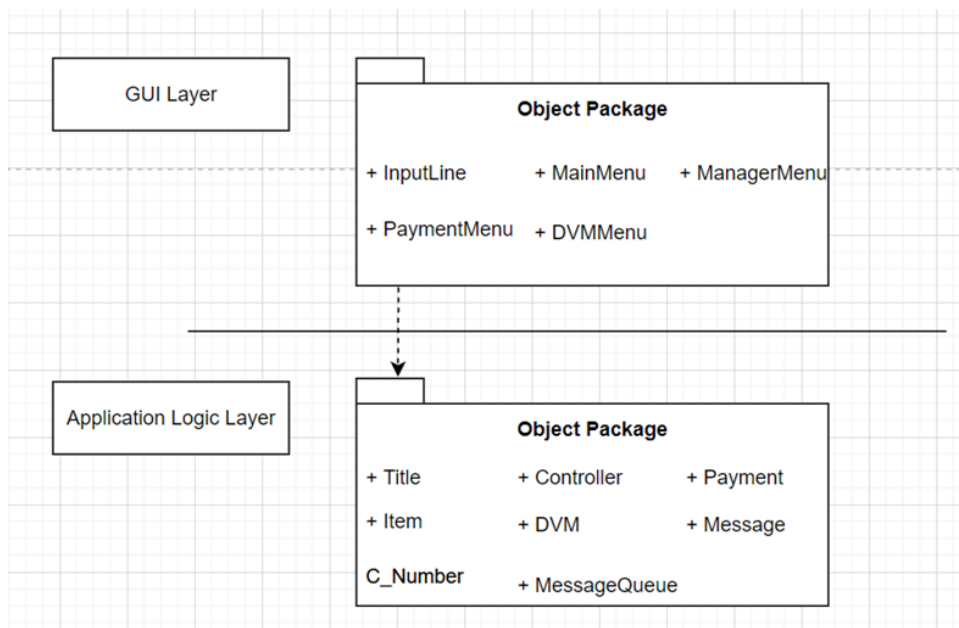
- IEEE Std 1016
- <http://dslab.konkuk.ac.kr/>

1.5 Overview

- 2. System architecture description에서는 구현하는 소프트웨어의 계층을 표현한다
- 3. Design Real Use Cases and Interaction Diagrams에서는 SRS에서 확보한 UseCase의 실행과정을 상세히 묘사하고, 이에따른 Sequence Diagram을 묘사한다.
- 4. Class Diagram을 통해 객체와 메소드를 정의하고 각각의 관계를 정의한다.
- 5. Design Traceability Analysis에서는 오퍼레이션과 메소드간의 관계를 묘사한다.

2. System architecture description

- 2. System architecture는 기본 3계층 Layer를 사용하며, OS Layer를 제외한 GUI, Logic Layer에서 소프트웨어를 구현한다.



3. Real Use Cases and Interaction Diagrams

-Real Use Cases and Interaction Diagrams에서는 SRS에서 확보한 UseCase의 Typical Course를 개발자의 관점으로 재정의하고, 이에 따른 Sequence Diagram을 묘사한다.

3.1 Sequence Diagrams with Use Cases

3.1.1음료 리스트 화면 출력

Use Case	1. 음료 리스트 화면 출력
Actor	Customer
Purpose	사용자에게 메뉴를 표시
Overview	사용자가 자판기에 어떠한 입력을 가했을 경우 디스플레이 화면이 켜지면서 메뉴가 출력됨.
Type	Evident
Cross Reference	N/A
Pre-Requisites	3분간 어떠한 입력도 받지 않아 종료 상태여야 함
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 DVM 스크린을 터치 2. (S) : 시스템이 이를 인식 후 메뉴 화면을 출력함
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagram	<pre> sequenceDiagram actor Customer participant Controller as :Controller Customer->>Controller: ReqMainMenu() Controller-->>Customer: ShowMainMenu() </pre>

3.1.2 구매할 음료 입력

Use Case	
2. 구매할 음료 입력	
Actor	Customer
Purpose	사용자가 구매할 메뉴를 선택할 수 있게 함
Overview	자판기 디스플레이에 출력된 메뉴 중 한가지를 선택한다.
Type	Evident
Cross Reference	Functions: R1.1, R1.2.2, R1.3, R.4.4
Pre-Requisites	자판기에 음료리스트가 출력된 상태여야 함
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 자판기에 출력된 메뉴 중 원하는 품목 입력 2. (S) : 해당 Title의 ref) 20. 재고 정보 확인 3. (S) : 해당 Title의 정보를 저장.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E2.1 DVM에 재고가 0인 경우 Ref) 5.재고가 부족한 제품 안내.Typical Courses of Event
Sequence Diagram	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant Title as :Title Customer->>Controller: InputSelect() Controller->>Title: Ref=CheckStock() Title-->>Controller: alt Ref==false Controller-->>Customer: InfoNoItem() else Ref==true Controller-->>Customer: PrintSelectPay(DVMid) end </pre>

3.1.3 사용자 선택 취소

Use Case	3. 사용자 선택 취소
Actor	Customer
Purpose	사용자가 선택한 메뉴를 취소할 수 있게 함
Overview	사용자가 선택 취소 버튼을 누른다.
Type	Evident
Cross Reference	Functions: R1.1, R1.2.1
Pre-Requisites	사용자가 기존에 선택한 음료가 존재해야 함
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 선택을 취소 2. (S) : 저장된 title을 삭제 3. (S) : 메인 화면으로 복귀
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagram	<pre> sequenceDiagram actor Customer participant Controller as :Controller Controller->>Controller: Init() Customer->>Controller: CancelItem() Controller-->>Customer: ReturnMain() </pre>

3.1.4 사용자 인증번호/바코드 입력

Use Case	4. 사용자 인증번호/바코드 입력
Actor	Customer
Purpose	선결제한 사용자가 제품을 전달받기 위해 인증번호를 입력
Overview	사용자가 적합한 인증번호 입력 시 시스템이 해당 제품 전달
Type	Evident
Cross Reference	Functions: R3.1.1, R3.1.2, R3.2
Pre-Requisites	사용자가 타 DVM에서 인증번호를 출력 받은 상태여야 함
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 DVM에 인증번호 입력버튼을 선택 2. (S) : 시스템이 이를 인식 후 입력 화면을 출력 3. (A) : 사용자가 이전에 출력 받은 인증번호를 입력 4. (S) : C_Number Stack에서 해당 인증번호를 검색 5. (S) : 해당 C_Number에 지정된 Title을 확인 6. (S) : 인증번호에 따른 제품을 전달
Alternative Courses of Events	A3.1 사용자(관리자)가 관리자 메뉴에 접근하려는 경우 Ref)19.관리자 메뉴 출력.Typical Courses of Event
Exceptional Courses of Events	E3.1 사용자가 입력한 인증번호를 찾지 못한 경우 1. 시스템이 오류 메시지 출력 2. 이전화면으로 복귀
Sequence Diagram	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant C_NumberManager as :C_NumberManager Customer->>Controller: InputSelect() activate Controller loop Controller->>Customer: ShowInputLine() deactivate Controller activate Customer Customer->>Controller: InputNumber() deactivate Customer activate Controller Controller->>C_NumberManager: Ref=CheckCnumber(Cnumber) activate C_NumberManager C_NumberManager-->>Controller: deactivate C_NumberManager deactivate Controller opt Ref=-1 Controller->>Customer: InfoCnumberError() deactivate Controller activate Customer deactivate Customer end opt Ref=1 Controller->>C_NumberManager: PopCnumber(Cnumber) activate C_NumberManager C_NumberManager-->>Controller: ReturnItem(Title,ifhold) deactivate C_NumberManager deactivate Controller activate Customer deactivate Customer end end </pre>

3.1.5 재고가 부족한 제품 안내

Use Case	
5. 재고가 부족한 제품 안내	
Actor	None
Purpose	사용자가 재고가 부족한 제품을 고른 경우 이를 안내해 줌
Overview	사용자가 선택한 제품의 재고가 현 DVM에서 0인 경우 이를 안내해 줌
Type	Hidden
Cross Reference	Functions: R1.2.1, R4.4, R1.4
Pre-Requisites	사용자가 선택한 음료의 재고가 DVM에서 0이어야 한다
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 자판기에 출력된 메뉴 중 원하는 품목 입력 2. (S) : 해당 Title의 ref) 20. 재고 정보 확인 3. (S) : 재고가 부족한 제품 안내 화면 출력
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagram	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant Title as :Title Customer->>Controller: InputSelect() activate Controller Controller->>Title: Ref=CheckStock() activate Title Title-->>Controller: deactivate Title Controller-->>Customer: InfoNoItem() deactivate Controller Controller-->>Customer: PrintSelectPay(DVMid) deactivate Controller </pre> <p>The sequence diagram illustrates the interaction between three lifelines: Customer, :Controller, and :Title. The process begins with the Customer sending an 'InputSelect()' message to the Controller. The Controller then sends a 'Ref=CheckStock()' message to the Title. The Title returns an empty message to the Controller. Subsequently, the Controller sends an 'InfoNoItem()' message back to the Customer. Finally, the Controller sends a 'PrintSelectPay(DVMid)' message to the Customer. Two optional regions are shown: one for 'Ref==false' where 'InfoNoItem()' is sent, and another for 'Ref==true' where 'PrintSelectPay(DVMid)' is sent.</p>

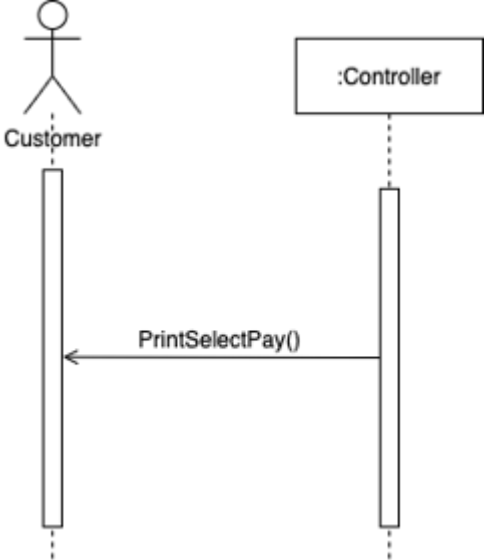
3.1.6 구매 가능한 자판기 안내

Use Case		6. 구매 가능한 자판기 안내
Actor	None	
Purpose	사용자에게 구매 가능한 자판기 안내	
Overview	사용자가 선택한 음료의 재고가 있는 타 DVM을 안내한다.	
Type	Hidden	
Cross Reference	R1.3, R4.5.1	
Pre-Requisites	사용자가 재고가 부족한 제품 안내를 받은 상황	
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 구매 가능한 자판기 안내 버튼을 입력 2. (S) : 저장된 Title을 Msg protocol에 저장. 3. (S) : Ref)21 메시지 발신을 이용해 broadcast Msg 발신 4. (S) : Ref)22 메시지 수신 5. (S) : Ref)21 Msg queue에서 긍정적인 DVM에 주소 요청 Msg 발신 6. (S) : Ref)22 메시지 수신 7. (S) : Message Queue에서 수신된 address를 출력	
Alternative Courses of Events	N/A	
Exceptional Courses of Events	E5.1 선택한 제품을 보유한 자판기가 없음 1. 해당 메시지 출력 후 메인화면으로 복귀	
Sequence Diagram	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant Message as :Message participant MessageQueue as :Message_Queue actor OtherDVM as Other DVM Customer->>Controller: ReqFindDVM() activate Controller Controller->>Message: loop n Message(ID, Type, Data) activate Message Message->>MessageQueue: SendMessage(StkReqMsg) activate MessageQueue MessageQueue->>OtherDVM: SendMessage(StkMsg) activate OtherDVM OtherDVM-->>MessageQueue: Dequeue(type) deactivate OtherDVM MessageQueue-->>Message: Dequeue(type) deactivate MessageQueue Message-->>Controller: Dequeue(type) deactivate Message Controller->>Customer: InfoNoUsableDVM() deactivate Controller Controller->>Message: loop n Message(ID, Type, Data) activate Message Message->>MessageQueue: SendMessage(LocReqMsg) activate MessageQueue MessageQueue->>OtherDVM: SendMessage(LocMsg) activate OtherDVM OtherDVM-->>MessageQueue: Dequeue(type) deactivate OtherDVM MessageQueue-->>Message: Dequeue(type) deactivate MessageQueue Message-->>Controller: Dequeue(type) deactivate Message Controller->>Customer: ShowUsableDVM() deactivate Controller </pre>	

3.1.7 사용자 자판기 선택

Use Case		7. 사용자 자판기 선택
Actor	Customer	
Purpose	사용자가 선결제할 자판기를 선택	
Overview	타 DVM 중 사용자가 선결제할 DVM을 선택한다.	
Type	Evident	
Cross Reference	Functions: R1.4 R2.1	
Pre-Requisites	사용자가 구매 가능한 자판기 안내를 받은 상황	
Typical Courses of Events	(A) : Actor (S) : System 1. (S) : DVM_Stack을 화면에 출력 2. (A) : 사용자가 선결제 할 DVM 선택 3. (S) : DVM_Stack을 비운다. 4. (S) : Ref)8. 결제수단 목록 출력	
Alternative Courses of Events	N/A	
Exceptional Courses of Events	E2.1 사용자가 선결제 할 DVM을 고르지 않고 취소할 경우 1. 메인 화면으로 복귀	
Sequence Diagram	<pre> sequenceDiagram actor Customer participant Controller as :Controller Customer->>Controller: SelectDVM() activate Controller Controller-->>Customer: PrintSelectPay(DVMid) deactivate Controller opt Controller-->>Customer: ReturnMain() end deactivate Controller </pre>	

3.1.8 결제 수단 목록 출력

Use Case	8.결제 수단 목록 출력
Actor	None
Purpose	사용자가 결제할 방법을 고를 수 있도록 목록을 출력
Overview	화면에 간편 결제 / 카드 결제 선택 화면을 출력한다.
Type	Hidden
Cross Reference	Functions: R1.2.1, R1.4, R 2.2
Pre-Requisites	재고가 있는 제품을 선택하였거나 선결제할 자판기를 고른 상태
Typical Courses of Events	(S) : System 1. (S) : 간편결제/카드결제 선택 화면을 출력
Exceptional Courses of Events	N/A
Sequence Diagrams	 <pre> sequenceDiagram actor Customer participant Controller as :Controller Controller->>Customer: PrintSelectPay() </pre> <p>The diagram illustrates the interaction between a Customer and a Controller. The Customer is represented by a stick figure, and the Controller is represented by a rectangular box labeled ':Controller'. Both have vertical lifelines. A solid arrow points from the Controller's lifeline to the Customer's lifeline, labeled 'PrintSelectPay()'. Dashed lines extend from the top and bottom of each lifeline to indicate the start and end of the interaction.</p>

3.1.9 결제 수단 결정

Use Case		9. 결제 수단 결정
Actor	Customer	
Purpose	사용자가 결제 수단 결정	
Overview	사용자가 카드로 결제할 것인지, 간편 결제로 결제할 것인지 선택한다.	
Type	Evident	
Cross Reference	Functions: R2.1, R2.3.1, R2.3.2	
Pre-Requisites	간편결제 / 카드결제 선택 화면이 출력되어 있어야 함	
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 카드결제버튼을 입력 2. (S) : 카드결제 화면을 출력	
Alternative Courses of Events	(A) : Actor (S) : System 1. (A) : 간편결제버튼을 입력 2. (S) : 간편결제 화면을 출력	
Exceptional Courses of Events	N/A	
Sequence Diagrams	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant Payment as :Payment Customer->>Controller: SelectPayment(DVMid) activate Controller Controller->>Payment: Payment(Title_id, DVMid) activate Payment Payment->>Controller: deactivate Payment Controller-->>Customer: ShowCardPay() deactivate Controller opt Controller-->>Customer: ShowSmartPay() end </pre> <p>The sequence diagram illustrates the interaction between the Customer, the Controller, and the Payment object. The Customer initiates the process by sending a 'SelectPayment(DVMid)' message to the Controller. The Controller then sends a 'Payment(Title_id, DVMid)' message to the Payment object. After the Payment object returns, the Controller sends 'ShowCardPay()' back to the Customer. An optional path (marked 'opt') shows the Controller sending 'ShowSmartPay()' back to the Customer.</p>	

3.1.10 간편 결제

Use Case	10. 간편 결제
Actor	Customer
Purpose	사용자가 선택 제품에 대한 금액을 결제
Overview	사용자가 금액을 지불하기 위해 간편 결제를 시도한다.
Type	Evident
Cross Reference	Functions: R1.2.1, R2.1, R2.2, R2.4.1, R2.4.2, R3.1.1, R3.2
Pre-Requisites	사용자가 결제 수단 결정에서 간편 결제를 선택한 상황
Typical Courses of Events	(A) : Actor (S) : System 1. (S) : 시스템이 화면에 QR코드를 출력 2. (A) : 사용자가 QR코드를 이용해 간편 결제를 진행 3. (S) : 결제 대행사로부터 승인 결과를 수신 4. (S) : 결제 완료 후 저장된 Title을 확인 5. (S) : Ref)15. 음료 전달
Alternative Courses of Events	(A) : Actor (S) : System A4.1 사용자가 선결제를 하는 경우 1. (S) : Ref)13. 인증번호/바코드 생성 2. (S) : Ref)14. 인증번호/바코드 출력
Exceptional Courses of Events	E3.1 금액 부족 등 예상치 못한 결제의 종료 시 1. 오류 메시지를 출력한 후 메인 화면으로 복귀
Sequence Diagrams	<pre> sequenceDiagram actor Actor actor Other DVM participant Controller as :Controller participant Payment as :Payment participant C_Number as :C_Number participant Message as :Message Actor->>Controller: SmartPay() activate Controller Controller->>Payment: SmartPay(Pay_confirm) activate Payment Payment->>C_Number: CreateCnumber() activate C_Number C_Number->>Message: Message(ID,Type,Data) activate Message Message->>Other DVM: SendMessage(Msg) deactivate Message C_Number-->>Payment: PrintCnumber() deactivate C_Number Payment-->>Actor: ReturnItem(Title,Ifhold) deactivate Payment deactivate Controller </pre>

3.1.11 카드 결제

Use Case		11. 카드결제
Actor	Customer	
Purpose	사용자가 선택 제품에 대한 금액을 결제	
Overview	사용자가 음료를 선택 후 금액을 지불하기 위해 카드 투입 후 결제를 시도함.	
Type	Evident	
Cross Reference	Functions: R1.2.1, R2.1, R2.4.1, R2.4.2, R3.1.1, R3.2	
Pre-Requisites	사용자가 결제 수단 결정에서 간편 결제를 선택한 상황	
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : Ref)9. 소비자가 카드결제를 선택 2. (S) : 시스템이 결제 화면을 출력 3. (A) : 사용자가 카드를 카드리더기에 입력 4. (S) : 카드리더기를 통해 얻은 카드정보를 카드사에 전달 5. (S) : 카드사로부터 승인결과를 수신 6. (S) : 결제 완료 후 저장된 Title을 확인 7. (S) : Ref)15. 음료 전달	
Alternative Courses of Events	(A) : Actor (S) : System A5.1 사용자가 선결제를 하는 경우(Title.item == 0) 1. (S) : Ref)13. 인증번호/바코드 생성 2. (S) : Ref)14. 인증번호/바코드 출력	
Exceptional Courses of Events	E3.1 카드사 승인거부시 1. 오류 메시지를 출력한 후 메인화면으로 복귀	
Sequence Diagrams		

3.1.12 결제 취소

Use Case	12. 결제 취소
Actor	Customer
Purpose	사용자가 결제를 취소
Overview	사용자가 결제가 시작되기 전 결제를 취소함
Type	Evident
Cross Reference	Functions: R2.2 R2.3.1 R2.3.2
Pre-Requisites	결제가 진행된 상태가 아니여야 함
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 결제 시작 전 취소버튼을 누름 2. (S) : 시스템이 이를 인식 후 메인화면으로 복귀
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant Payment as :Payment Customer->>Controller: CancelPay() activate Controller Controller->>Payment: init() activate Payment deactivate Payment Controller->>Controller: Init() deactivate Controller Controller->>Customer: ReturnMain() deactivate Controller </pre> <p>The sequence diagram illustrates the interaction for canceling a payment. It involves three lifelines: Customer (actor), :Controller, and :Payment. The process begins with the Customer sending a 'CancelPay()' message to the :Controller. The :Controller then sends an 'init()' message to the :Payment object. After the :Payment object is deactivated, the :Controller performs a self-call 'Init()'. Finally, the :Controller sends a 'ReturnMain()' message back to the Customer, and the :Controller's lifeline ends.</p>

3.1.13 인증번호/바코드 생성

Use Case	13. 인증번호/바코드 생성
Actor	None
Purpose	시스템이 특정 자판기에서 입력 가능한 인증번호/바코드를 생성한다.
Overview	사용자가 선결제를 완료한 경우 시스템이 인증번호/바코드를 생성한다.
Type	Hidden
Cross Reference	Functions: R.2.3.1 R.2.3.2 R.3.1.2
Pre-Requisites	사용자가 선결제를 성공적으로 마친 상태여야 한다.
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : Ref)11. 선결제 완료 2. (S) : 시스템이 C_Number 및 바코드를 생성 3. (S) : 생성된 6자리 난수의 정보를 담은 메시지 생성 4. (S) : 메시지를 해당 DVM에 전송
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram actor Actor as Actor participant Payment as :Payment participant C_Number as :C_Number participant Message as :Message participant Other_DVM as :Other DVM Payment->>C_Number: CreateCNumber() activate C_Number C_Number->>Message: Message(ID, Type, Data) activate Message Message->>Other_DVM: SendMessage(Msg) deactivate Message C_Number-->>Payment: deactivate C_Number </pre>

3.1.14 인증번호/바코드 출력

Use Case	14. 인증번호/바코드 출력
Actor	None
Purpose	시스템이 생성된 인증번호/바코드를 사용자에게 출력한다
Overview	사용자의 인증번호/바코드가 생성되면 이후 사용자에게 출력해준다.
Type	Hidden
Cross Reference	R3.1.1
Pre-Requisites	시스템이 정상적으로 인증번호/바코드를 생성한 뒤여야 한다
Typical Courses of Events	(S) : System 1. (S) : Ref)13. 인증번호/바코드 생성 2. (S) : 시스템이 인증번호/바코드를 화면에 출력
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram actor Customer participant Controller as :Controller Controller->>Customer: PrintCnumber() </pre> <p>The diagram shows a sequence of interactions between a Customer (actor) and a Controller (object). The Controller sends a message labeled 'PrintCnumber()' to the Customer. Both lifelines have activation bars indicating the duration of their participation in the process.</p>

3.1.15 음료전달

Use Case	15. 음료 전달
Actor	None
Purpose	사용자가 선택한 음료를 자판기를 통해 전달한다.
Overview	사용자가 선택한 결제 완료된 음료를 DVM이 인식 후 상품을 전달한다.
Type	Hidden
Cross Reference	Functions: R1.1, R1.2.1, R2.3.1, R2.3.2, R3.3
Pre-Requisites	사용자가 제품 결제를 완료한 상태여야 함.
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 DVM에 음료를 결정 후 결제버튼을 선택 2. (S) : 저장된 Title에 해당하는 음료를 전달 3. (S) : Ref)16. 재고 수량 업데이트
Alternative Courses of Events	A.1 사용자가 인증번호를 입력했을 경우 2. (S) : 저장된 Title에 해당하는 음료를 전달 3. (S) : Ref)16. 재고 수량 업데이트
Exceptional Courses of Events	E2.1 DVM의 기계적인 고장으로 음료 전달이 안됨 1. 해당 메시지 출력 후 관리자에게 전달
Sequence Diagrams	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant Title as :Title participant Payment as :Payment Controller->>Customer: ReturnItem(Title, lhold) Controller->>Title: UpdateStock(TitleID, lhold) Controller->>Payment: init() Title->>Controller: init() Controller->>Customer: ReturnMain() </pre>

3.1.16 재고 수량 업데이트

Use Case	16. 재고 수량 업데이트
Actor	None
Purpose	결제가 정상적으로 종료된 경우 이를 재고 목록에 반영함
Overview	결제가 정상적으로 종료되어 음료수 전달 혹은 인증번호가 출력되었을 경우 해당 재고를 제거 혹은 홀드 함
Type	Hidden
Cross Reference	Functions: R3.2, R3.1.1
Pre-Requisites	사용자에게 음료를 전달한 상황
Typical Courses of Events	(S) : System 1. (S) : Ref)15. 음료 전달 2. (S) : 시스템이 해당 title의 item 차감
Alternative Courses of Events	(A) : Actor (S) : System A*.1 시스템이 DVM네트워크로부터 선결제 메시지를 수신한 경우 1. (A) : 타 DVM에서 시스템에 선결제 메시지를 발신 2. (S) : 시스템이 해당 메시지의 Title을 저장 3. (S) : 해당 Title의 Hold 변수 증가, item 차감
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant Title as :Title participant MessageQueue as :Message_Queue actor OtherDVM as Other DVM Customer->>Controller: opt ReturnItem(Title, Ihold) activate Controller Controller->>Title: UpdateStock(TitleID, Ihold) activate Title Title->>MessageQueue: opt UpdateStock(TitleID, Ihold) activate MessageQueue MessageQueue->>OtherDVM: SendMessage(Msg) activate OtherDVM OtherDVM->>MessageQueue: Dequeue(Type) deactivate OtherDVM MessageQueue->>Title: UpdateStock(TitleID, Ihold) deactivate MessageQueue deactivate Title deactivate Controller </pre>

3.1.17 관리자 메뉴 출력

Use Case		17. 관리자 메뉴 출력
Actor	Manager	
Purpose	관리자가 자판기의 재고 현황을 파악할 수 있게 함	
Overview	관리자가 로그인하면, 현재 자판기에 존재하는 재고 수량, 음료의 종류, 관리자 인증번호 수정 등을 보여주는 관리자 메뉴를 출력한다.	
Type	Evident	
Cross Reference	Functions: R3.3, R4.2	
Pre-Requisites	관리자가 자신의 고유 인증번호가 있는 상태여야 함	
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 DVM에 인증번호 입력버튼을 선택 2. (S) : 시스템이 이를 인식 후 입력 화면을 출력 3. (A) : 사용자가 자신의 고유 인증번호를 입력 4. (S) : 시스템이 관리자 메뉴를 출력	
Alternative Courses of Events	N/A	
Exceptional Courses of Events	E3.1 관리자가 잘못된 고유인증번호를 입력 1. 시스템이 인식 후 오류 메시지를 출력	
Sequence Diagrams	<pre> sequenceDiagram actor Customer participant Controller as :Controller participant C_NumberManager as :C_NumberManager loop Customer->>Controller: InputSelect() Controller-->>Customer: ShowInputLine() Customer->>Controller: InputCnumber() Controller->>C_NumberManager: Ref=CheckCnumber(Cnumber) C_NumberManager-->>Controller: end opt Ref==1 Controller-->>Customer: InfoCnumberError() end opt Ref==0 Controller-->>Customer: ManShowTitle() end </pre>	

3.1.18 재고 변경/관리

Use Case	18. 재고 변경/관리
Actor	Manager
Purpose	관리자가 해당 재고의 보충 등을 실행한 후 이를 시스템에 알림
Overview	관리자가 재고를 보충한 뒤 해당 재고의 수량을 기입함.
Type	Evident
Cross Reference	Functions: R1.2.3, R4.1
Pre-Requisites	관리자가 자신의 고유 인증번호를 통해 관리자메뉴를 출력한 상태여야 함
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 관리자가 재고를 변경할 Title 선택 2. (S) : 시스템이 해당 Title 의 item 목록 출력 3. (A) : 관리자가 item 을 추가하거나 삭제 4. (S) : 시스템이 Title 의 item 을 변경 5. (S) : 시스템이 변경 결과 출력
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram actor Manager participant Controller as :Controller participant Title as :Title loop Manager->>Controller: ManSelectTitle() loop Controller->>Manager: ManShowItem(TitleID) Manager->>Controller: ManEditItem() opt add item Controller->>Title: AddItem(item) Title-->>Controller: end opt delete item Controller->>Title: DeleteItem(itemID_list) Title-->>Controller: end end Controller-->>Manager: ManShowTitle() end </pre>

3.1.19 위치 정보 확인

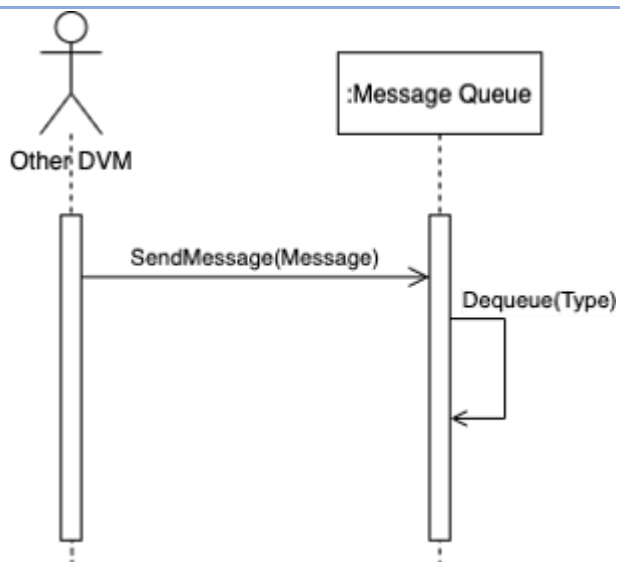
Use Case	19. 위치 정보 확인
Actor	None
Purpose	시스템이 저장된 자신의 위치정보를 확인함.
Overview	DVM네트워크에서 자신의 위치정보를 요구 시 저장소에서 위치정보를 가져옴
Type	Hidden
Cross Reference	R4.5.1, R4.5.2
Pre-Requisites	DVM네트워크로부터 요구 메시지를 수신 받은 상태여야 함
Typical Courses of Events	(S) : System 1. (S) : 위치정보를 요구하는 메시지를 수신 2. (S) : 저장소에 저장된 자신의 Address를 반환
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram actor Other DVM participant MQ as :Message Queue participant DVM as :DVM participant Message as :Message Other DVM->>MQ: SendMessage(Message) activate MQ MQ->>MQ: Dequeue(Type) MQ->>DVM: getX(),getY() DVM-->>MQ: deactivate DVM MQ->>Message: Message(ID,Type,Data) activate Message Message->>Other DVM: SendMessage(Msg) deactivate Message deactivate MQ </pre>

Use Case	20. 재고 정보 확인
Actor	None
Purpose	요청받은 품목에 대한 재고 확인
Overview	재고 확인을 요청 받은 품목에 대해 재고를 확인함.
Type	Hidden
Cross Reference	Functions: R.1.1.3, R.4.5.1, R.4.5.2
Pre-Requisites	어떠한 재고 확인 요청이 있어야 함
Typical Courses of Events	(S) : System 1. (S) : 재고 확인 요청을 수신 2. (S) : 요청 받은 Title에 대한 item 이 있는지 반환
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram actor Actor as Other DVM participant MQ as :Message Queue participant Title as :Title participant Message as :Message Actor->>MQ: SendMessage(Message) activate MQ MQ->>MQ: Dequeue(Type) MQ->>Title: CheckStock() activate Title Title-->>MQ: deactivate Title MQ->>Message: Message(ID, Type, Data) activate Message Message-->>Actor: SendMessage(Msg) deactivate Message deactivate MQ </pre>

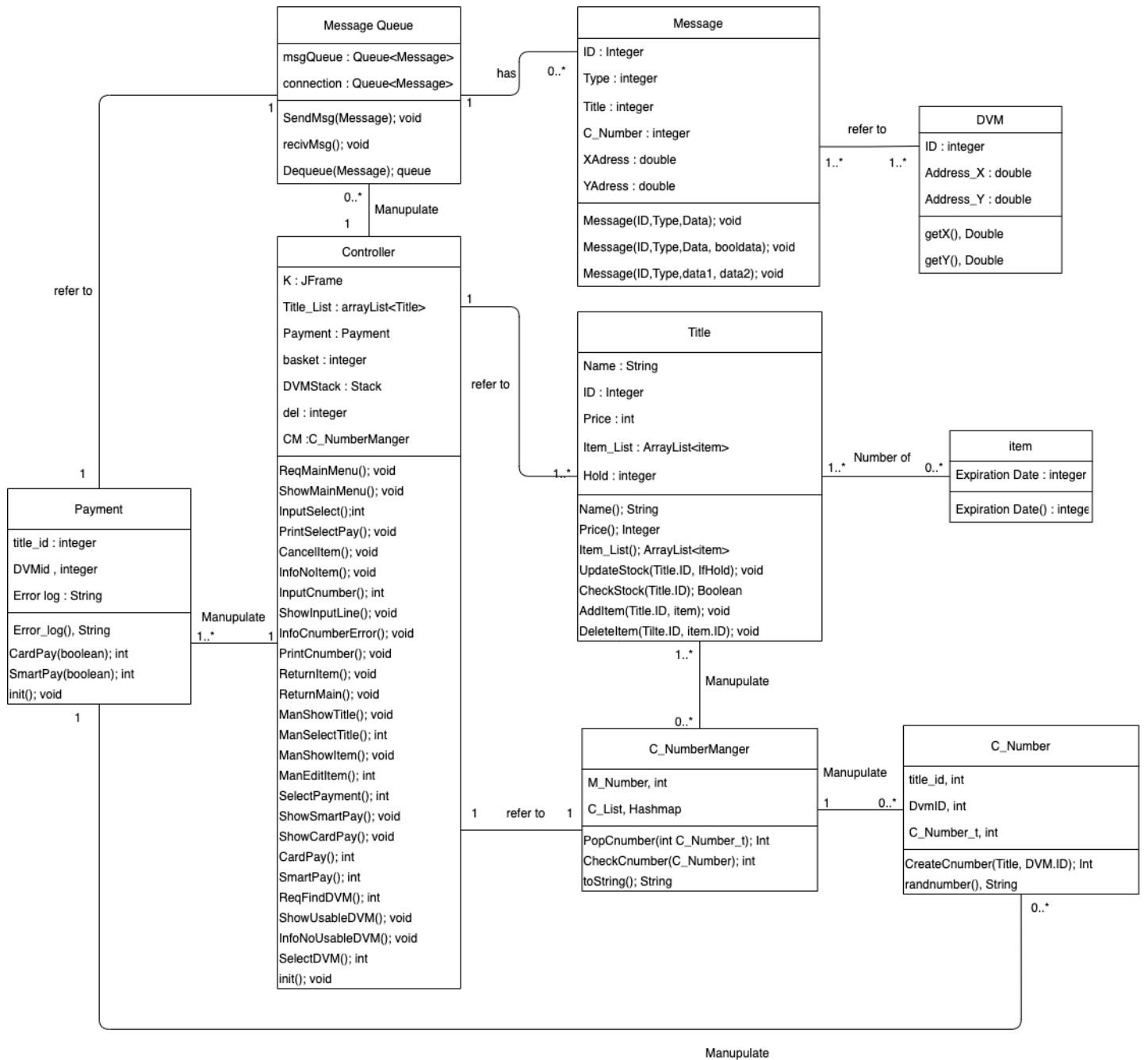
3.1.21 메시지 발신

Use Case		21. 메시지 발신
Actor	None	
Purpose	시스템이 타 DVM에 메시지를 발신	
Overview	필요에 따라 시스템이 DVM네트워크에 메시지를 발신하여 정보를 전달하거나 요청한다.	
Type	Hidden	
Cross Reference	Functions: R1.3, R3.1.1, R4.5.2	
Pre-Requisites	사용자가 구매 가능한 자판기 안내를 요청한 상황 사용자가 선결제를 완료하여 인증번호/바코드가 생성된 상황	
Typical Courses of Events	(A) : Actor (S) : System 1. (S) : 메시지 발신 요청 수신 2. (S) : 해당 Msg 프로토콜 발신	
Alternative Courses of Events	N/A	
Exceptional Courses of Events	N/A	
Sequence Diagrams	<pre> sequenceDiagram participant Message as Message participant OtherDVM as Other DVM Message->>OtherDVM: SendMessage(Message) </pre> <p>The diagram illustrates the interaction between a Message object and an Other DVM actor. The Message object is represented by a rectangular box at the top left. A vertical dashed line extends downwards from it to a solid vertical bar representing its activation period. The Other DVM actor is represented by a stick figure at the top right, with a vertical dashed line extending downwards to a smaller solid vertical bar representing its activation period. A horizontal arrow points from the Message object's activation bar to the Other DVM actor's activation bar, labeled 'SendMessage(Message)'.</p>	

3.1.22 메시지 수신

Use Case	22. 메시지 수신
Actor	None
Purpose	다른 DVM으로부터 전송된 메시지를 수신
Overview	DVM 네트워크로부터 정보 요청 / 전달 메시지를 받는다.
Type	Hidden
Cross Reference	R1.4, R4.4, R4.5.1
Pre-Requisites	사용자가 타 DVM에서 메시지를 발신한 상황
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 타 DVM이 메시지 발신 2. (S) : 시스템이 해당 메시지를 Msg queue에 저장
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Sequence Diagrams	 <pre> sequenceDiagram actor Actor as Other DVM participant System as :Message Queue Actor->>System: SendMessage(Message) System->>System: Dequeue(Type) </pre> <p>The diagram shows two lifelines: 'Other DVM' (represented by a stick figure) and ':Message Queue' (represented by a rectangular box). The 'Other DVM' lifeline has an activation bar. A message arrow labeled 'SendMessage(Message)' points from the 'Other DVM' lifeline to the ':Message Queue' lifeline. The ':Message Queue' lifeline has an activation bar and a self-message arrow labeled 'Dequeue(Type)'.</p>

4. Class Diagram



5. Traceability Analysis

