

**Software Design  
Specification  
For  
Distributed Vending Machine**

**Team 4**

**201711836 송호영**

**201811217 이해인**

**201710240 이해림**

**201511212 이도경**

## Table of Contents

<b>1. Introduction.....</b>	<b>3p</b>
1.1 Purpose	
1.2 Scope	
1.3 Definitions, acronyms, and abbreviations	
1.4 References	
1.5 Overview	
<b>2. System architecture description.....</b>	<b>4p</b>
<b>3. Real Use Cases and Interaction Diagrams.....</b>	<b>5p</b>
3.1 Use Cases and Interaction Diagrams	
<b>4. Class Diagram.....</b>	<b>18p</b>
<b>5. Traceability Analysis.....</b>	<b>19p</b>

# 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 접속하기위해 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 수량
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

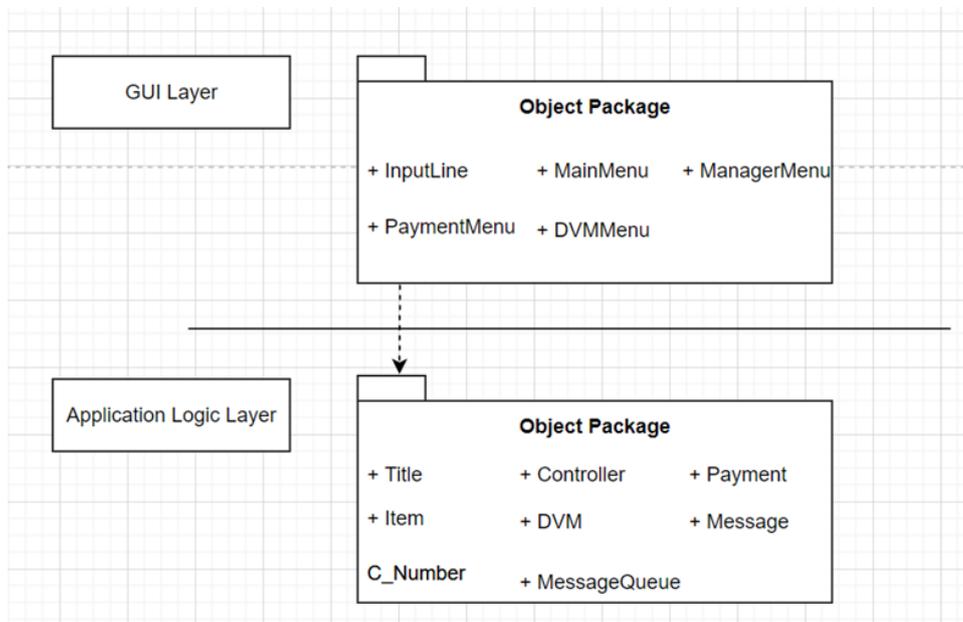
- IEEE Std 1016
- <http://dslab.konkuk.ac.kr/>

## 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

##### 3.1.1 음료 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 스크린에 어떠한 입력을 줌 (ex. 화면 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 Actor     participant Controller as :Controller     Actor-&gt;&gt;Controller: ReqMainMenu()     Controller--&gt;&gt;Actor: ShowMainMenu()           </pre>

### 3.1.2 구매할 음료 입력

Use Case	2. 구매할 음료 입력
<b>Actor</b>	Customer
<b>Purpose</b>	사용자가 구매할 메뉴를 선택할 수 있게 함
<b>Overview</b>	자판기 디스플레이에 출력된 메뉴 중 한가지를 선택한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R1.1, R1.2.2, R1.3, R.4.4
<b>Pre-Requisites</b>	자판기에 음료리스트가 출력된 상태여야 함
<b>Typical Courses of Events</b>	(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.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	E2.1 DVM에 재고가 0인 경우 Ref) 5.재고가 부족한 제품 안내.Typical Courses of Event
<b>Sequence Diagram</b>	<pre> sequenceDiagram     actor Customer     participant Controller as :Controller     participant Title as :Title     Customer-&gt;&gt;Controller: InputSelect(TitleID)     activate Controller     Controller-&gt;&gt;Title: CheckStock(TitleID)     activate Title     Title--&gt;&gt;Controller:      deactivate Title     Controller--&gt;&gt;Customer: [[if(CheckStock(TitleID))]] PrintSelectPay()     Controller--&gt;&gt;Customer: [[if(!CheckStock(TitleID))]] InfoNoItem()     deactivate Controller                     </pre>

### 3.1.3 사용자 선택 취소

Use Case	3. 사용자 선택 취소
<b>Actor</b>	Customer
<b>Purpose</b>	사용자가 선택한 메뉴를 취소할 수 있게 함
<b>Overview</b>	사용자가 선택 취소 버튼을 누른다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R1.1, R1.2.1

<b>Pre-Requisites</b>	사용자가 기존에 선택한 음료가 존재해야 함
<b>Typical Courses of Events</b>	(A) : Actor (S) : System 1. (A) : 사용자가 선택을 취소 2. (S) : 저장된 title을 삭제
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A
<b>Sequence Diagram</b>	<pre> sequenceDiagram     actor Customer     participant Controller as :Controller     Customer-&gt;&gt;Controller: CancelItem()     activate Controller     Controller-&gt;&gt;Controller: Init()     deactivate Controller     Controller--&gt;&gt;Customer: ShowMainMenu()     deactivate Controller </pre>

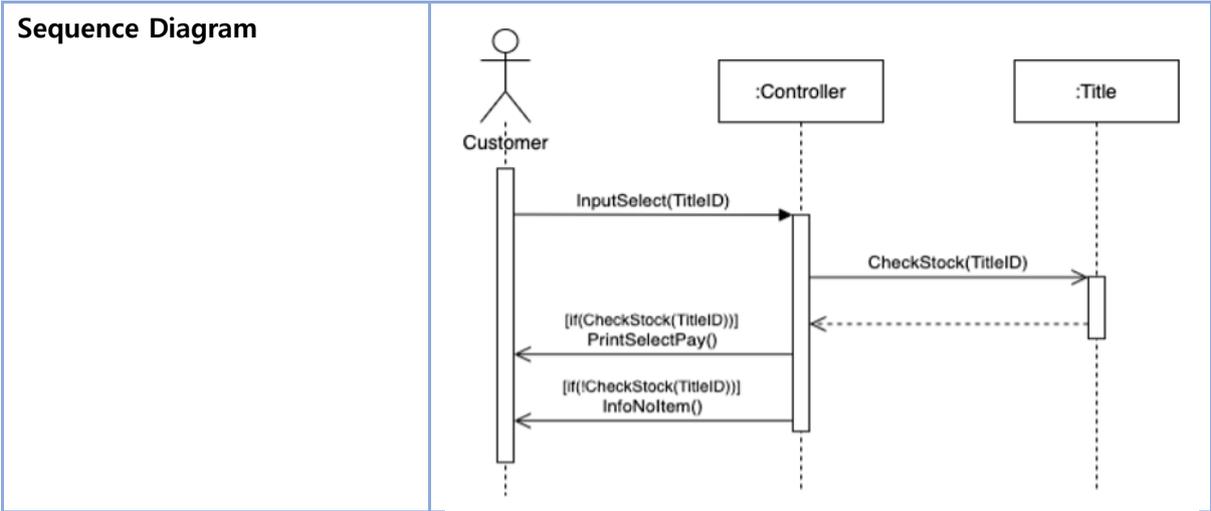
### 3.1.4 사용자 인증번호/바코드 입력

Use Case 4. 사용자 인증번호/바코드 입력	
<b>Actor</b>	Customer
<b>Purpose</b>	선결제한 사용자가 제품을 전달받기 위해 인증번호를 입력
<b>Overview</b>	사용자가 적합한 인증번호 입력 시 시스템이 해당 제품 전달
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R3.1.1, R3.1.2, R3.2
<b>Pre-Requisites</b>	사용자가 타 DVM에서 인증번호를 출력 받은 상태여야 함
<b>Typical Courses of Events</b>	(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

<b>Alternative Courses of Events</b>	A3.1 사용자(관리자)가 관리자 메뉴에 접근하려는 경우 Ref)19.관리자 메뉴 출력.Typical Courses of Event
<b>Exceptional Courses of Events</b>	E3.1 사용자가 입력한 인증번호를 찾지 못한 경우 1. 시스템이 오류 메시지 출력 2. 이전화면으로 복귀
<b>Sequence Diagram</b>	<pre> sequenceDiagram     actor Customer     participant Controller as :Controller     participant C_Number as :C_Number     participant Title as :Title      Customer-&gt;&gt;Controller: ReqInputNumber()     Controller--&gt;&gt;Customer: ShowInputLine()     Customer-&gt;&gt;Controller: InputCnumber()     Controller-&gt;&gt;C_Number: CheckCnumber(Cnumber)     C_Number--&gt;&gt;Controller: PopCnumber(Cnumber)     Controller--&gt;&gt;Customer: InfoError()     Note over Controller: [!(CheckCnumber==1)]     Controller--&gt;&gt;Customer: ReturnItem()     Note over Controller: [!(CheckCnumber==0)]     Controller-&gt;&gt;Title: UpdateStock(TitleID, False)     </pre>

### 3.1.5 재고가 부족한 제품 안내

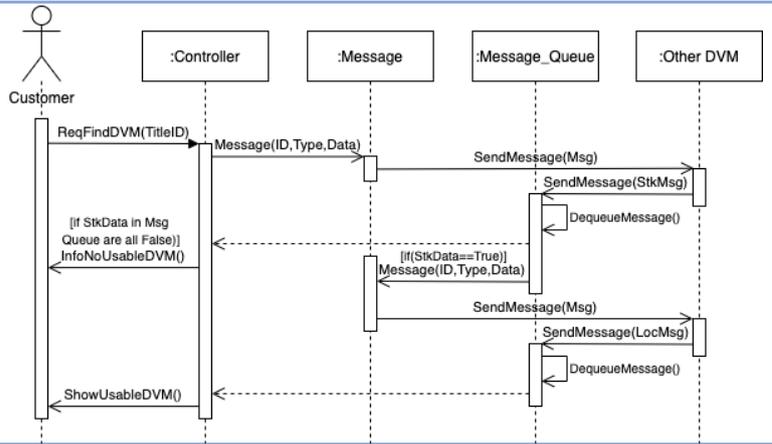
Use Case	
<b>5. 재고가 부족한 제품 안내</b>	
<b>Actor</b>	None
<b>Purpose</b>	사용자가 재고가 부족한 제품을 고른 경우 이를 안내해 줌
<b>Overview</b>	사용자가 선택한 제품의 재고가 현 DVM에서 0인 경우 이를 안내해줌
<b>Type</b>	Hidden
<b>Cross Reference</b>	Functions: R1.2.1, R4.4, R1.4
<b>Pre-Requisites</b>	사용자가 선택한 음료의 재고가 DVM에서 0이어야 한다
<b>Typical Courses of Events</b>	(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
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A



3.1.6 구매 가능한 자판기 안내

Use Case	6. 구매 가능한 자판기 안내
<b>Actor</b>	None
<b>Purpose</b>	사용자에게 구매 가능한 자판기 안내
<b>Overview</b>	사용자가 선택한 음료의 재고가 있는 타 DVM을 안내한다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	R1.3, R4.5.1
<b>Pre-Requisites</b>	사용자가 재고가 부족한 제품 안내를 받은 상황
<b>Typical Courses of Events</b>	(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 긍정인 DVM에 주소 요청 Msg 발신 6. (S) : Ref)22 메시지 수신 7. (S) : Message Queue에서 수신된 address를 출력
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	E5.1 선택한 제품을 보유한 자판기가 없음 1. 해당 메시지 출력 후 이전화면으로 복귀

**Sequence Diagram**



3.1.7 사용자 자판기 선택

Use Case		7. 사용자 자판기 선택
<b>Actor</b>	Customer	
<b>Purpose</b>	사용자가 선결제할 자판기를 선택	
<b>Overview</b>	타 DVM 중 사용자가 선결제할 DVM을 선택한다.	
<b>Type</b>	Evident	
<b>Cross Reference</b>	Functions: R1.4 R2.1	
<b>Pre-Requisites</b>	사용자가 구매 가능한 자판기 안내를 받은 상황	
<b>Typical Courses of Events</b>	(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	
<b>Alternative Courses of Events</b>	N/A	
<b>Exceptional Courses of Events</b>	E2.1 사용자가 선결제 할 DVM을 고르지 않고 취소할 경우 1. 해당 메시지 출력 후 메인 화면으로 복귀	
<b>Sequence Diagram</b>	<pre> sequenceDiagram     actor Customer     participant Controller as :Controller      Customer-&gt;&gt;Controller: ReqfindDVM(TitleID)     activate Controller     Controller-&gt;&gt;Customer: ShowDVMStack()     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	선택된 제품이 재고가 존재해야 함
Typical Courses of Events	(A) : Actor (S) : System 1. (S) : 간편결제/카드결제 선택 화면을 출력
Alternative Courses of Events	(A) : Actor (S) : System A*.1 구매가능한 자판기에 대한 안내를 받은 상황 1. (S) : 간편결제/카드결제 선택 화면을 출력
Exceptional Courses of Events	N/A
Sequence Diagrams	<pre> sequenceDiagram     actor Customer     participant Controller as :Controller     Note over Customer, Controller: [DVM에 선택된 제품의 재고가 존재하는 상황]     Customer-&gt;&gt;Controller: InputSelect(TitleID)     Controller--&gt;&gt;Customer: PrintSelectPay()     </pre>

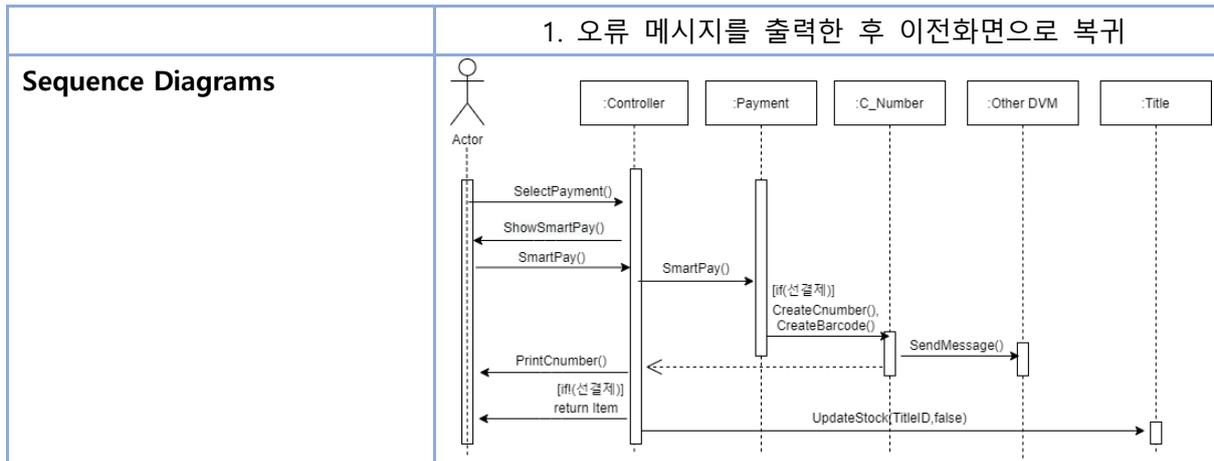
### 3.1.9 결제 수단 결정

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

	<ol style="list-style-type: none"> <li>(A) : 카드결제버튼을 입력</li> <li>(S) : Ref)11. 카드결제를 실행</li> </ol>
<b>Alternative Courses of Events</b>	(A) : Actor (S) : System <ol style="list-style-type: none"> <li>(A) : 간편결제버튼을 입력</li> <li>(S) : Ref)10. 간편결제를 실행</li> </ol>
<b>Exceptional Courses of Events</b>	N/A
<b>Sequence Diagrams</b>	<pre> sequenceDiagram     actor Actor     participant Controller as :Controller     participant Payment as :Payment      Actor-&gt;&gt;Controller: SelectPayment()     activate Controller     Controller-&gt;&gt;Payment: ReqSmartPay()     activate Payment     Payment--&gt;&gt;Controller:      deactivate Payment     Controller-&gt;&gt;Actor: ShowSmartPay()     deactivate Controller     Controller-&gt;&gt;Payment: ReqCardPay()     activate Payment     Payment--&gt;&gt;Controller:      deactivate Payment     Controller-&gt;&gt;Actor: ShowCardPay()     deactivate Controller   </pre>

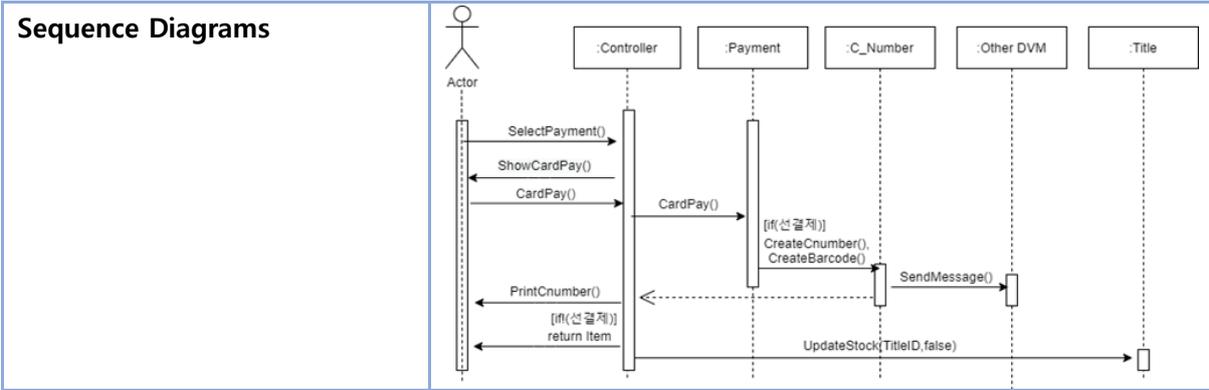
### 3.1.10 간편 결제

Use Case	10. 간편 결제
<b>Actor</b>	Customer
<b>Purpose</b>	사용자가 선택 제품에 대한 금액을 결제
<b>Overview</b>	사용자가 금액을 지불하기 위해 간편 결제를 시도한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R1.2.1, R2.1, R2.2, R2.4.1, R2.4.2, R3.1.1, R3.2
<b>Pre-Requisites</b>	사용자가 결제 수단 결정에서 간편 결제를 선택한 상황
<b>Typical Courses of Events</b>	(A) : Actor (S) : System <ol style="list-style-type: none"> <li>(S) : 시스템이 화면에 QR코드를 출력</li> <li>(A) : 사용자가 QR코드를 이용해 간편 결제를 진행</li> <li>(S) : 결제 대행사로부터 승인 결과를 수신</li> <li>(S) : 결제 완료 후 저장된 Title을 확인</li> <li>(S) : Ref)15. 음료 전달</li> </ol>
<b>Alternative Courses of Events</b>	(A) : Actor (S) : System A4.1 사용자가 선결제를 하는 경우 <ol style="list-style-type: none"> <li>(S) : Ref)13. 인증번호/바코드 생성</li> <li>(S) : Ref)14. 인증번호/바코드 출력</li> </ol>
<b>Exceptional Courses of Events</b>	E3.1 금액 부족 등 예상치 못한 결제의 종료 시



### 3.1.11 카드 결제

Use Case	11. 11. 카드결제
<b>Actor</b>	Customer
<b>Purpose</b>	사용자가 선택 제품에 대한 금액을 결제
<b>Overview</b>	사용자가 음료를 선택 후 금액을 지불하기 위해 카드 투입 후 결제를 시도함.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R1.2.1, R2.1, R2.4.1, R2.4.2, R3.1.1, R3.2
<b>Pre-Requisites</b>	사용자가 결제 수단 결정에서 간편 결제를 선택한 상황
<b>Typical Courses of Events</b>	(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
<b>Alternative Courses of Events</b>	(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
<b>Exceptional Courses of Events</b>	E3.1 카드사 승인거부시 1. 오류 메시지를 출력한 후 이전화면으로 복귀



3.1.12 결제 취소

Use Case	12. 결제 취소
<b>Actor</b>	Customer
<b>Purpose</b>	사용자가 결제를 취소
<b>Overview</b>	사용자가 결제가 시작되기 전 결제를 취소함
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R2.2 R2.3.1 R2.3.2
<b>Pre-Requisites</b>	결제가 진행된 상태가 아니여야 함
<b>Typical Courses of Events</b>	(A) : Actor (S) : System 1. (A) : 사용자가 결제 시작 전 취소버튼을 누름 2. (S) : 시스템이 이를 인식 후 메인화면으로 복귀
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A
<b>Sequence Diagrams</b>	<pre> sequenceDiagram     actor Actor     participant Controller as :Controller     participant Payment as :Payment      Actor-&gt;&gt;Controller: CancelPay()     activate Controller     Controller-&gt;&gt;Payment: init()     activate Payment     Payment-&gt;&gt;Payment: init()     deactivate Payment     Controller--&gt;&gt;Actor: ReqMainMenu()     deactivate Controller   </pre>

### 3.1.13 관리자 메뉴 출력

Use Case	17. 관리자 메뉴 출력
<b>Actor</b>	Manager
<b>Purpose</b>	관리자가 자판기의 재고 현황을 파악할 수 있게 함
<b>Overview</b>	관리자가 로그인하면, 현재 자판기에 존재하는 재고 수량, 음료의 종류, 관리자 인증번호 수정 등을 보여주는 관리자 메뉴를 출력한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R3.3, R4.2
<b>Pre-Requisites</b>	관리자가 자신의 고유 인증번호가 있는 상태여야 함
<b>Typical Courses of Events</b>	(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
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	E3.1 관리자가 잘못된 고유인증번호를 입력 1. 시스템이 인식 후 오류 메시지 출력
<b>Sequence Diagrams</b>	<pre> sequenceDiagram     actor Customer     participant Controller as :Controller     participant C_Number as :C_Number      Customer-&gt;&gt;Controller: ReqInputNumber()     Controller--&gt;&gt;Customer: ShowInputLine()     Customer-&gt;&gt;Controller: InputNumber()     Controller-&gt;&gt;C_Number: CheckNumber(Cnumber)     C_Number--&gt;&gt;Controller:      Controller--&gt;&gt;Customer: [[if(CheckNumber==-1)]] InfoError()     Controller--&gt;&gt;Customer: [[if(CheckNumber==0)]] ShowManagerMenu()           </pre>

### 3.1.14 재고 변경/관리

Use Case	18. 재고 변경/관리
<b>Actor</b>	Manager
<b>Purpose</b>	관리자가 해당 재고의 보충 등을 실행한 후 이를 시스템에 알림

<b>Overview</b>	관리자가 재고를 보충한 뒤 해당 재고의 수량을 기입함.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functions: R1.2.3, R4.1
<b>Pre-Requisites</b>	관리자가 자신의 고유 인증번호를 통해 관리자메뉴를 출력한 상태여야 함
<b>Typical Courses of Events</b>	(A) : Actor (S) : System 1. (A) : 관리자가 재고의 변화율 기입 2. (S) : 시스템이 보충된 Title의 item을 변경 3. (S) : 시스템이 변경 결과 출력
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A
<b>Sequence Diagrams</b>	<pre> sequenceDiagram     actor Manager     participant Controller as :Controller     participant Title as :Title     Manager-&gt;&gt;Controller: ManageStock()     activate Controller     Controller-&gt;&gt;Title: AddItem(TitleID,item)     activate Title     Title-&gt;&gt;Controller: DeleteItem(TitleID,itemID)     deactivate Title     Controller-&gt;&gt;Manager: ShowManagerMenu()     deactivate Controller   </pre>

### 3.1.15 메시지 발신

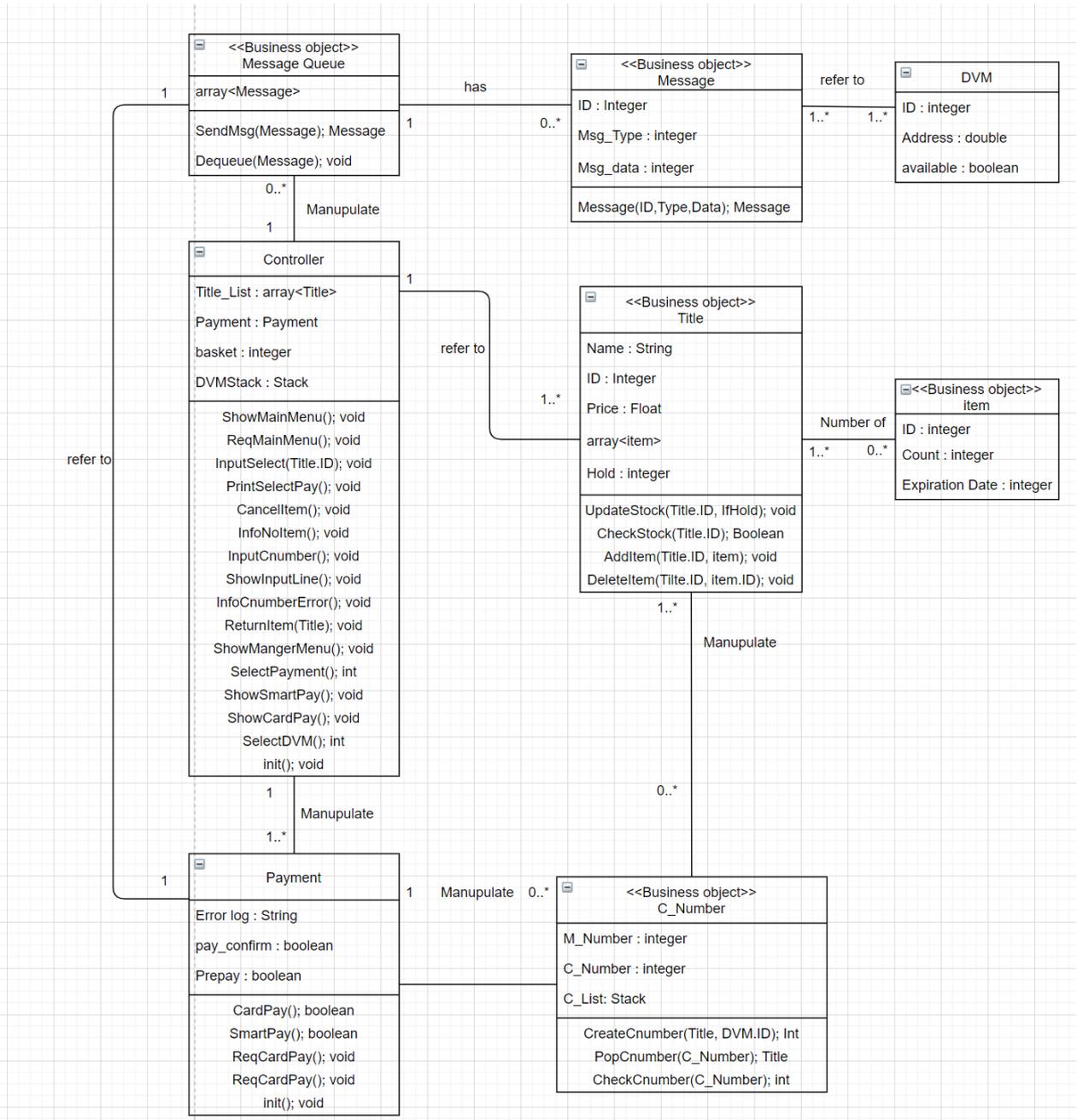
Use Case	21. 메시지 발신
<b>Actor</b>	None
<b>Purpose</b>	시스템이 타 DVM에 메시지를 발신
<b>Overview</b>	필요에 따라 시스템이 DVM네트워크에 메시지를 발신하여 정보를 전달하거나 요청한다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	Functions: R1.3, R3.1.1, R4.5.2
<b>Pre-Requisites</b>	사용자가 구매 가능한 자판기 안내를 요청한 상황 사용자가 선결제를 완료하여 인증번호/바코드가 생성된 상황
<b>Typical Courses of Events</b>	(A) : Actor (S) : System 1. (S) : 메시지 발신 요청 수신 2. (S) : 해당 Msg 프로토콜 발신

<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A
<b>Sequence Diagrams</b>	<pre> sequenceDiagram     participant Controller as :Controller     participant Message as :Message     participant OtherDVM as :Other DVM     Controller-&gt;&gt;Message: Message(ID, Type, Data)     Message-&gt;&gt;OtherDVM: SendMessage(Data)     </pre>

### 3.1.16 메세지 수신

Use Case	
	<b>22. 메시지 수신</b>
<b>Actor</b>	None
<b>Purpose</b>	다른 DVM으로부터 전송된 메시지를 수신
<b>Overview</b>	DVM 네트워크로부터 정보 요청 / 전달 메시지를 받는다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	R1.4, R4.4, R4.5.1
<b>Pre-Requisites</b>	사용자가 타 DVM에서 메시지를 발신한 상황
<b>Typical Courses of Events</b>	(A) : Actor (S) : System 1. (A) : 타 DVM이 메시지 발신 2. (S) : 시스템이 해당 메시지를 Msg queue에 저장
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A
<b>Sequence Diagrams</b>	<pre> sequenceDiagram     participant OtherDVM as :Other DVM     participant MessageQueue as :Message Queue     OtherDVM-&gt;&gt;MessageQueue: SendMessage(StkData)     MessageQueue-&gt;&gt;MessageQueue: DequeueMessage()     </pre>

## 4. Class Diagram



## 5. Traceability Analysis

