

# **Software Requirements Specification**

Distributed Vending Machine

**2021** 객체지향개발방법론 1조

**SRS**

**[Ver 1.0]**

**201914173** 김현웅

**201714164** 박서영

**201711315** 신원세

**201710560** 정의재

## Contents

### **1. Introduction**

- 1.1. Purpose
- 1.2. Scope
- 1.3. Definitions, acronyms, and abbreviations
- 1.4. References
- 1.5. Overview

### **2. Overall description**

- 2.1. Product perspective
- 2.2. Product functions
- 2.3. User characteristics
- 2.4. Constraints
- 2.5. Assumptions and dependencies

### **3. Specific requirements**

- 3.1. External interface requirements
- 3.2. System features
- 3.3. Performance requirements
- 3.4. Design constraints
- 3.5. Software system attributes

## 1. Introduction

### 1.1. Purpose

- 본 문서는 DVM(Distributes Vending Machine)의 Controller Software를 구현하기 위한 요구사항을 명시한다.

### 1.2. Scope

- MS129 DVM은 현재 DVM이 판매하는 음료수를 판매하는 기능 뿐만 아니라 현재 판매하지 않고 있는 음료수에 대해서도 그 음료수를 판매하고 있는 자판기의 위치의 정보를 제공한다. 또한 선결제 기능을 제공하여 사용자는 다른 DVM의 음료수를 먼저 결제한 후 발급받은 인증코드를 이용하여 음료수를 구매할 수도 있다. DVM들 간 네트워크를 통해 Message를 교환함으로써 다른 DVM들과 서로 필요한 정보를 교환한다.

### 1.3. Definitions,acronyms,and abbreviations

| Terms           | Description   |
|-----------------|---|
| prepayment(선결제) | 현재 사용하고 있는 자판기가 아닌 다른 자판기에 대한 결제를 먼저 진행하고 인증코드를 발급받는다 |
| code(인증코드)      | 상품 선결제에 사용될 무작위의 4자리 숫자 코드이며, 선결제를 선택한 사용자에게 제공된다.    |
| stock(재고)       | 자판기에 존재하는 음료 개수                                       |
| drink(음료정보)     | 음료 이름, 가격, 재고   |
| card(카드)        | 결제 진행에 사용되는 모든 결제 카드를 '카드'라고 부른다.                     |

### 1.4. References

- IEEE Std 83-1998
- 건국대학교 DS Lab (<http://dslab.konkuk.ac.kr/>)
- dslab.konkuk.ac.kr/Class/2021/21SMA/LectureNote/OOPT\_All(2020).pdf

### 1.5. Overview

- MS129의 전반적인 기능 설명
- MS129가 사용되기 위한 External Interface Requirements 설명

## 2. Overall Description

### 2.1. Product perspective

- 프로젝트 수행에 있어 구현에 제한이 있기 때문에 사용자에게 키보드 입력을 받는 방법으로 기능을 수행한다.

### 2.2. Product functions

- MS129는 사용자에게 결제를 완료한 음료를 제공한다.
- 사용하고있는 DVM이 판매하지 않는 음료를 선택할 경우, 다른 DVM들과 메시지를 주고받아 해당 음료를 판매하는 DVM들의 위치를 안내해주며, 선결제 기능 또한 제공한다.
- 선결제를 진행할때, 사용자가 성공적으로 결제를 완료하면 인증코드가 발급되며, 다른 DVM에서 인증코드입력시 선결제를 진행했던 음료를 수령할 수 있다.

### 2.3. User characteristics

- 사용 가능한 카드(유효한 카드, 충분한 잔액)를 가지고 있는 사용자
- DVM을 사용해본 경험이 있는 사용자
- 제공되는 위치 정보 텍스트로 다른 자판기의 위치를 찾아갈 수 있는 사용자

### 2.4. Constraints

- 카드 결제는 일시불만 가능
- 교환,환불은 불가능
- DVM 간의 통신시 Message는 항상 성공적으로 전송
- DVM의 전원은 초기 설정 상태를 유지

### 2.5. Assumptions and dependencies

- SW 내부에서 카드의 정보(사용가능 여부)와 카드의 잔액을 저장. 결제는 이 내부 데이터를 이용하여 이루어짐.
- DVM에서 판매하는 음료의 종류와 수량은 초기 설정에서 변경되지 않음
- DVM 별 위치는 초기 설정에서 변경되지 않음

### 3. Specific requirements

#### 3.1. External interface requirements

##### 3.1.1. User interfaces

###### 3.1.1.1. Dial

- 사용자의 입력을 다이얼로(MS129에서는 키보드로) 입력받아 DVM에게 전달하는 Input Display

###### 3.1.1.2. Card Reader

- 사용자가 결제에 사용할 카드의 정보를 읽어들이는 단말기

##### 3.1.2. Hardware interfaces

###### 3.1.2.1. Main Screen

- 사용자에게 안내할 모든 정보와 프로세스의 진행 상황 그리고 그 결과를 출력

###### 3.1.2.2. Out Space

- 사용자가 결제를 성공적으로 마친 음료가 나오는곳

###### 3.1.2.3.

##### 3.1.3. Software interfaces

- MS129 DVM SW는 외부 Software를 사용하지 않음

##### 3.1.4. Communication interfaces

| src_id    | dst_id                      | msg type     | msg description(optional) |
|-----------|-----------------------------|--------------|---------------------------|
| 현재 자판기 ID | 대상 자판기 ID<br>(broadcast는 0) | 1. 재고확인 요청   | 음료이름                      |
|           |                             | 2. 재고확인 응답   | TRUE/FALSE                |
|           |                             | 3. 자판기상태 요청  | NULL                      |
|           |                             | 4. 자판기 상태 응답 | TRUE/FALSE                |

## **3.2. System features**

### **3.2.1. Class/Object (Controller)**

#### **3.2.1.1. Attributes(direct or inherited)**

- 3.2.1.1.1. Attribute 1 - code\_list
  - Type : List<Code>
  - Description:발급된 코드들을 저장한다.
- 3.2.1.1.2. Attribute 2 - currentDVM
  - Type : DVM
  - Description:현재DVM객체를 저장한다.
- 3.2.1.1.3. Attribute 3 - stateBroadCastMessage
  - Type : Message
  - Description:브로드캐스트 메시지의 상태를 저장한다.
- 3.2.1.1.4. Attribute 4 - stateMessage
  - Type : Message
  - Description:상태 메시지를 저장한다.
- 3.2.1.1.5. Attribute 5 - accesible\_DVM\_list
  - Type : List<DVM>
  - Description:사용가능한 DVM리스트를 저장한다.
- 3.2.1.1.6. Attribute 6 - codeList
  - Type : List<Code>
  - Description:코드 리스트를 저장한다.

### **3.2.2. Class/Object(DVM)**

#### **3.2.2.1. Attributes(direct or inherited)**

- 3.2.2.1.1. Attribute 1 - state

- Type : Boolean
- Description : DVM의 ON/OFF 상태를 저장한다.

#### 3.2.2.1.2. Attribute 2 - drink\_list

- Type : List<Drink>
- Description : DVM이 가지고있는 음료의 리스트를 저장한다.

#### 3.2.2.1.3. Attribute 3 - id

- Type : int
- Description : DVM별로 구분하기 위한 값의 역할이다.

#### 3.2.2.1.4. Attribute 4 - address

- Type : int
- Description : DVM의 주소를 저장한다.

#### 3.2.2.1.5. Attribute 5 - stateMessage

- Type : Message
- Description : 상태를 전달한다.

### 3.2.3. Class/Object(Message)

#### 3.2.3.1. Attributes(direct or inherited)

##### 3.2.3.1.1. Attribute 1 - src\_id

- Type : int
- Description : msg를 보내는 곳의 id를 저장한다.

##### 3.2.3.1.2. Attribute 2 - dst\_id

- Type : int
- Description : msg를 받는 곳의 id를 저장한다.

##### 3.2.3.1.3. Attribute 3 - msg\_type

- Type : int
- Description : msg의 종류를 저장한다.

### 3.2.4. Class/Object(CardPayment)

#### 3.2.4.1 Attributes(direct or inherited)

##### 3.2.4.1.1. Attribute 1 - basicCardList

- Type : List<Card>

- **Description** : 결제가 가능한 형태의 카드를 저장한다. 입력받은 카드와 비교하기 위함이다.

#### 3.2.4.1.2. Attribute 2 - drink\_info

- **Type** : Drink
- **Description** : 카드결제할 음료 객체를 저장한다.

#### 3.2.4.1.3. Attribute 3 - price\_info

- **Type** : int
- **Description** : 음료의 가격을 저장한다.

#### 3.2.4.1.4. Attribute 4 - balance\_info

- **Type** : int
- **Description** : 카드의 잔고를 저장한다.

### 3.2.5. Class/Object(Card)

#### 3.2.5.1. Attributes(direct or inherited)

##### 3.2.5.1.1. Attribute 1 - balance

- **Type** : int
- **Description** : 카드의 잔액을 저장한다.

##### 3.2.5.1.2. Attribute 2 - card\_available

- **Type** : boolean
- **Description** : 카드의 유효 여부를 저장한다.

### 3.2.6. Class/Object(Drink)

#### 3.2.6.1. Attributes(direct or inherited)

##### 3.2.6.1.1. Attribute 1 - name

- **Type** : string
- **Description** : 음료수의 이름을 저장한다.

##### 3.2.6.1.2. Attribute 2 - price

- **Type** : int
- **Description** : 음료수의 가격을 저장한다.

##### 3.2.6.1.3. Attribute 3 - stock

- **Type** : int
- **Description** : 음료수의 재고량을 저장한다.

### 3.2.7. Class/Object(Code)

#### 3.2.7.1. Attributes(direct or inherited)



#### 3.2.7.1.1. Attribute 1 - code

- Type : int
- Description : 코드를 저장한다.

#### 3.2.7.1.2. Attribute 2 - drink

- Type : Drink
- Description : 음료 객체를 저장한다.

### 3.2.8. Class/Object(CodePayment)

#### 3.2.8.1. Attributes(direct or inherited)

##### 3.2.8.1.1. Attribute 1 - code\_info

- Type : Code
- Description : 코드 객체 정보를 저장한다.

##### 3.2.8.1.2. Attribute 2 - isCodeAvailable

- Type : Boolean
- Description : 코드 유효성을 저장한다.

### 3.3. Performance requirements

- DVM 간 통신을 위해 사용하는 Message의 전송 속도는 1s 이하여야 함
- 음료 선택시 중복 선택이 불가능하여야 함

### 3.4. Design constraints

- JDK 개발환경을 구축함
- JAVA 언어를 이용하여 구현
- OOAD 개발 방법론을 적용하여 설계

### 3.5. Software system attributes

#### 3.5.1. Reliability

- 사용자가 수행하고자 하는 기능을 성공적으로 구현하며 DVM간의 지연이 작아야 함

#### 3.5.2. Security

- 사용자가 사용하는 카드 정보는 SW 내부에서 관리되기 때문에 외부에서 접근할 수 없음
- DVM 간 교환하는 message는 정보를 필요로 하는 DVM 외에는 접근할 수 없음

#### 3.5.3. Maintainability

- GitHub를 통한 SW의 버전관리. 유지보수 용이
- Zoom을 이용한 프로젝트 팀원들의 주기적인 소통

#### 3.5.4. Availability

- 사용자는 DVM을 통해 능동적으로 자신이 구매하고 싶은 상품을  
고를 수 있고 필요한 접근이 필요한 DVM의 정보 또한 얻을 수  
있음

#### 3.5.5. Portability

- JDK환경을 구축하여 개발하므로 이식 용이