

# DISTRIBUTED VENDING MACHINE (DVM)

## Software Requirements Specification (SRS)



Project Team  
Team 2

Date  
2021.04.22

Team Members

201310513	황인우
201311255	최우석
201512265	박인우
201711306	박정현

# INDEX

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
  - 3.6 Other requirements

## 1. Introduction

### 1.1. Purpose

- 본 문서는 분산 자판기 시스템의 자판기 **Controller system SW**를 개발하기 위한 문서이다.

### 1.2. Scope

- 사용자가 원하는 상품을 구입할 수 있도록 한다.
- 결제는 카드결제와 현금결제가 가능하다.
- 재고가 존재하지 않는 상품에 한하여 다른 자판기와 **Broadcast** 통신을 진행하여 해당 상품이 존재하는 다른 자판기를 탐색하여 사용자에게 알려준다.
- 그렇게 탐색된 상품들에 대한 선결제도 가능하다.

### 1.3. Definitions, Acronyms, and Abbreviations

Num	Term	Description
1	Items	프로젝트가 판매하는 상품들 입니다.
2	Code	다른 상품과 교환가능한 특수한 코드(숫자) 입니다.
3	Location	DVM의 위치를 나타냅니다. (2차원 좌표로 나타냅니다.)
4	Cash	사용자와 DVM이 사용하는 현금입니다.(원화)
5	Card	사용자와 DVM이 사용하는 카드입니다.
6	Code Table	사용 가능한 코드인지 아닌지에 대해 참, 거짓값을 가지고있는 테이블 입니다.
7	User	DVM 을 이용하는 사용자.
8	Admin	DVM 의 상태를 관리하는 관리자.

### 1.4. References

- IEEE Std 830-1998
- Konkuk Univ. DS Lab (<http://dslab.konkuk.ac.kr/>)
- SRS Example - Michigan State University CSE (<https://www.cse.msu.edu/~cse435/Handouts/SRSEExample-webapp.doc>)

### 1.5 Overview

- 시스템에 필요한 **External interface requirements** 구성
- 시스템 **System feature** 설명

## 2. Overall description

### 2.1 Product perspective

- 실제 하드웨어 구축은 불가능하기 때문에 윈도우 시스템 상에서 구축하는 것을 목표로 한다.
- 키보드와 마우스(터치 패드)로 시스템을 조작한다.

## 2.2 Product functions

- 재고가 있는 상품에 대해 상품 구매가 가능하다.
- 재고가 없는 상품이 있는 다른 자판기의 위치를 알 수 있다.
- 재고가 없는 상품에 대한 선결제를 통한 코드 발급이 가능하다.
- 결제는 카드 결제와 현금 결제 두가지가 가능하다.
- 결제를 완료하거나 유효한 코드를 제공하면 상품을 제공한다.
- 다른 시스템과 재고 보유량, 코드 사용가능 여부, 위치 정보를 공유한다.
- 로그인으로 관리자 권한을 얻을 수 있다.
- 관리자는 DVM의 상품 종류, 상품 재고, 현금 보유량을 관리할 수 있다.

## 2.3 User characteristics

- 현금, 카드, 선결제 코드의 지불 방법을 갖춘 사용자
- 관리자 로그인이 가능한 사용자
- 좌표로 위치를 찾을 수 있는 사용자

## 2.4 Constraints

- DVM의 전원은 항상 켜져있는 것으로 간주한다.
- 상품에 대한 환불은 불가능하다.
- 네트워크 통신에서 발생하는 에러는 없다고 가정한다.
- 코드 생성에서 발생하는 에러는 없다고 가정한다.

## 2.5 Assumptions and dependencies

- 생성된 코드는 **code table**에 저장되며, 수정사항(추가, 제거)이 생길 경우 다른 시스템과 공유된다.
- 한 번의 사용에서 하나의 상품만 구입이 가능하다.

## 3. Specific requirements

### 3.1 External interface requirements

#### 3.1.1 User Interfaces

- 키보드
- 마우스
- (터치패드)

#### 3.1.2 Hardware Interfaces

- 스크린

#### 3.1.3 System Interfaces

- 사용자의 입출력에 대한 함수 호출

#### 3.1.4 Software Interfaces

- 카드 결제 시스템
- 타 DVM과의 통신모듈

### 3.2 Functional requirements

#### 3.2.1 주요기능1 : 상품구매

- Use case 1~11
- 상품 선택

- 카드 혹은 현금으로 결제
- 결제 결과로 상품 혹은 그에 준하는 코드 반환

<b>Use Case</b>	1. Show Menu
<b>Actor</b>	System
<b>Purpose</b>	사용자에게 초기 선택 화면을 보여줍니다.
<b>Overview</b>	사용자에게 상품 구매, 쿠폰 사용, 관리자 로그인 기능 중 하나를 선택할 수 있는 초기 선택 메뉴를 보여줍니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose Item 3. Pay by Cash 4. Pay by Card
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (S) System이 사용자가 상품 구매, 쿠폰 사용, 관리자 로그인 기능 중 하나를 선택할 수 있는 화면을 출력합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	2.Choose Item
<b>Actor</b>	User
<b>Purpose</b>	사용자가 시스템에 등록된 상품 중 하나를 선택합니다.
<b>Overview</b>	사용자가 구입하고자 하는 시스템에 등록된 상품 중 하나를 선택합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	1. Show Menu 3. Pay by Cash 4. Pay by Card
<b>Pre-Requisites</b>	시스템에 상품이 하나 이상 등록되어 있어야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) User가 상품 선택을 활성화시킵니다. 2. (U) User가 현재 자판기에 등록되어 있는 상품 중 원하는 상품 하나를 선택합니다. 3. (S) System이 User가 선택한 상품의 재고를 확인합니다. User가 선택한 상품의 재고가 있을 경우, 결제 방식 선택으로 넘어갑니다 4. (S) System이 구매하려는 목록을 출력합니다.
<b>Alternative Courses of</b>	A 2.1: User가 선택한 상품의 재고가 없을 경우, 사용자에게 해당

<b>Events</b>	상품은 선결제를 통한 코드 발급만 가능함을 알리고 선결제할 것인지에 대한 여부를 사용자에게 선택하도록 합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	3. Pay by Cash
<b>Actor</b>	User
<b>Purpose</b>	User가 현금으로 결제하는 경우입니다
<b>Overview</b>	결제를 현금으로 진행하게 해줍니다
<b>Type</b>	Evident
<b>Cross Reference</b>	2. Choose Item 6. Calculate Price(Cash)
<b>Pre-Requisites</b>	System에 구매하려는 목록이 등록되어 있고 상품이 추가되어 있어야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) User가 현금 결제 버튼을 선택합니다. 2. (S) System이 현금을 받을 준비를 합니다. 3. (U) User가 현금을 지급합니다.
<b>Alternative Courses of Events</b>	A 1.1. Calculate Price(Cash)의 Alternative Course 1에 의해 반환된 경우, 알림 메시지와 함께 2 를 실행합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	4. Pay by Card
<b>Actor</b>	User
<b>Purpose</b>	User 가 카드로 결제하려는 경우입니다.
<b>Overview</b>	결제를 카드로 진행해주게 합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	2. Choose Item 7. Calculate Price(Card)
<b>Pre-Requisites</b>	System에 구매하려는 목록이 등록되어 있고 상품이 추가되어 있어야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) User가 카드 결제 버튼을 선택합니다 2. (S) System이 카드를 받을 준비를 합니다 3. (U) User가 카드를 투입합니다

<b>Alternative Courses of Events</b>	A 1.1. Calculate Price(Card)의 Alternative Course 1에 의해 반환된 경우, 알림 메시지와 함께 2 를 실행합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	5.Use code
<b>Actor</b>	User
<b>Purpose</b>	User가 코드를 사용하려는 경우입니다.
<b>Overview</b>	사용자가 코드를 입력하여 사용할 수 있도록 합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	8.Code validation 10.Give code
<b>Pre-Requisites</b>	사용자는 이전에 Give Code로부터 제공받은 코드가 존재해야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) User가 코드 입력 버튼을 선택합니다 2. (S) System이 코드를 입력받을 준비를 합니다 3. (U) User가 코드를 입력합니다
<b>Alternative Courses of Events</b>	A 1.1. Code validation의 Alternative Course 1에 의해 반환된 경우, 알림 메시지와 함께 2 를 실행합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	6.Calculate Price (Cash)
<b>Actor</b>	System
<b>Purpose</b>	User가 현금으로 결제할시 결제를 진행하는 과정입니다
<b>Overview</b>	투입된 현금과 합산된 상품 값을 비교하여 결제를 진행합니다
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose item 4. Pay by Cash 9. Choose item or code
<b>Pre-Requisites</b>	Pay by Cash에서 현금을 지급받은 상태여야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 투입된 현금의 합 과 상품의 값을 비교합니다 2. (S) 현금의 합이 지불값보다 높거나 같을시, 결제를 진행합니다 3. (S) 결제가 성공적이면, Choose item or code으로 넘어갑니다
<b>Alternative Courses of</b>	A1.1: 현금의 합이 더 적을시, Pay by Cash로 돌아가 추가 현금을

<b>Events</b>	받습니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	7. Calculate Price (Card)
<b>Actor</b>	System
<b>Purpose</b>	User가 카드로 결제할시 결제를 진행하는 과정입니다
<b>Overview</b>	카드의 지불능력과 합산된 상품 값을 비교하여 결제를 진행합니다
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose item 4. Pay by Card 9. Choose item or code
<b>Pre-Requisites</b>	Pay by Card에서 카드가 투입되어 있어야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 카드의 잔액/한도와 상품의 값을 비교합니다 2. (S) 카드의 잔액/한도가 지불값보다 높거나 같을시, 결제를 진행합니다 3. (S) 결제가 성공적이면, Choose item or code로 넘어갑니다
<b>Alternative Courses of Events</b>	A1.1: 2가 아닌 결과가 나올 시, Pay by Card로 돌아가 다른 카드를 투입 받습니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	8. Code Validation
<b>Actor</b>	System
<b>Purpose</b>	코드의 유효여부를 확인합니다.
<b>Overview</b>	사용자가 입력한 코드의 유효성 및 해당 자판기의 재고 여부를 확인합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose item 5. Use code 11. Give item
<b>Pre-Requisites</b>	Use code에서 사용자가 코드를 입력한 상태여야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 코드의 유효성을 체크합니다. 2. (S) 유효한 코드일 경우 코드로부터 확인된 상품들의 재고가 해당 시스템에 존재하는지 확인합니다.



	<p>3. (S) 상품의 재고가 있을 경우, 해당 코드를 사용한 것으로 처리하고 해당 정보를 다른 시스템에게도 알립니다.</p> <p>4. (S) Give item으로 넘어갑니다</p>
<b>Alternative Courses of Events</b>	A1.1: 유효하지 않은 코드일 경우, Use code로 돌아가 다시 코드를 입력하도록 합니다.
<b>Exceptional Courses of Events</b>	E2.1: 재고가 존재하지 않을 경우 해당 내용을 사용자에게 알리고 과정을 종료합니다.

<b>Use Case</b>	9. Choose Item or Code
<b>Actor</b>	System
<b>Purpose</b>	시스템이 사용자가 재고가 있는 상품을 선택했는지 재고가 없는 상품을 선택했는지 확인합니다.
<b>Overview</b>	시스템이 사용자가 재고가 있는 상품을 선택했는지 재고가 없는 상품을 선택했는지 확인하고 이에 따라 Give item 혹은 Give code를 진행합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	3. Pay By Cash 4. Pay By Card
<b>Pre-Requisites</b>	Pay By Cash, Pay By Card 중 하나의 과정에서 결제가 마무리된 상황이어야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (S) 시스템이 사용자가 재고가 있는 상품을 선택했는지, 재고가 없는 상품을 선택했는지 확인합니다. 2. (S) 재고가 있는 상품을 선택했을 경우 Give item 과정을 진행하고, 재고가 없는 상품을 선택했을 경우 Give Code 과정을 진행합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	10. Give Code
<b>Actor</b>	System
<b>Purpose</b>	사용자가 선택한 상품에 대해 코드를 제공합니다.
<b>Overview</b>	사용자가 선택한 상품을 구입할 수 있는 코드를 제공합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose item 9. Choose item or code

<b>Pre-Requisites</b>	Choose item or code에서 code을 선택해야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 시스템이 사용자가 선택한 상품에 대한 코드를 생성합니다. 2. (S) 시스템이 다른 시스템에 해당 코드 정보를 공유합니다. 3. (S) 코드가 반환됩니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	11. Give Item
<b>Actor</b>	System
<b>Purpose</b>	사용자가 선택한 상품을 제공합니다.
<b>Overview</b>	사용자가 선택한 상품을 제공하고, 제공한 수량 만큼을 시스템 내부 재고량 에서 뺍니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose item 8. Code validation 9. Choose item or code
<b>Pre-Requisites</b>	Choose item or code에서 item을 선택하거나 사용자가 코드를 사용하고 해당 코드가 Code validation 과정을 통과해야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 사용자가 선택한 상품이 반환됩니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

### 3.2.2 주요기능2 : DVM 정보 파악

- Use case 12~15
- 아이템에 대한 다른 DVM의 재고정보 파악
- 현재 DVM으로부터의 거리 파악
- 사용자에게 다른 DVM의 위치정보와 재고정보 제공

<b>Use Case</b>	12. Get None Item Location
<b>Actor</b>	User
<b>Purpose</b>	사용자가 원하는 재고가 없는 상품을 선택합니다.

<b>Overview</b>	사용자가 위치정보를 원하는, 재고가 없는 상품을 선택하도록 합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	2. Choose Item 13. Get Another DVM Info
<b>Pre-Requisites</b>	재고가 없는 상품이 존재해야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) : 주변 재고가 있는 DVM 위치를 보는 탭을 활성화합니다. 2. (S) Get Another DVM Info를 실행합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	13. Get Another DVM Info
<b>Actor</b>	System
<b>Purpose</b>	시스템이 다른 시스템에 정보(재고현황, 위치)를 요청합니다.
<b>Overview</b>	사용자가 요청한 재고가 없는 상품에 대한 정보를 얻기 위해 다른 시스템에 정보(재고현황, 위치)를 요청합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	12. Get None Item Location 14. Give DVM Info 15. Give Item Location
<b>Pre-Requisites</b>	User에게 Get None Item Location으로 요청을 받아야합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 다른 시스템 전체에게 요청 신호를 보냅니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	14. Give DVM Info
<b>Actor</b>	Another DVM
<b>Purpose</b>	타 DVM이 정보를 요청할시 현재 DVM의 정보(재고, 위치)를 알립니다
<b>Overview</b>	다른 DVM 에게 현재 DVM의 정보(재고, 위치)를 알립니다
<b>Type</b>	Hidden

<b>Cross Reference</b>	13. Get Another DVM Info 15. Give Item Location
<b>Pre-Requisites</b>	타 DVM에게 Get Another DVM Info로 요청을 받아야 합니다
<b>Typical Courses of Events</b>	(D): Another DVM 1. (D) 본 기기의 해당 재고 유무를 파악합니다. 2. (D) 기기의 재고와 위치를 System에게 반환합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	15. Give Item Location
<b>Actor</b>	System
<b>Purpose</b>	사용자에게 상품이 존재하는 다른 DVM의 정보(위치, 재고현황)을 알려줍니다.
<b>Overview</b>	사용자가 요청한 상품이 존재하는 다른 DVM의 정보(위치, 재고현황)을 알려줍니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	12. Get None Item Location 13. Get Another DVM Info 14. Give DVM Info
<b>Pre-Requisites</b>	Get None Item Location으로 요청된 상품이 존재해야 합니다. Give DVM Info로 재고정보 및 위치정보를 받아야합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) Give DVM Info로 전달받은 정보를 기반으로 재고가 존재하는 DVM을 구분합니다. 2. (S) 구분된 DVM들을 현재 시스템에서 가까운 순서대로 정렬합니다. 3. (S) 정렬된 DVM 목록을 출력합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	E1.1: 재고가 있는 DVM이 없을 경우 재고가 존재하는 자판기가 없음을 반환합니다.

### 3.2.3 주요기능3 : 관리자모드

- Use case 16-18
- 관리자 로그인 기능
- 재고 관리 및 현금 관리

<b>Use Case</b>	16. Login
<b>Actor</b>	Administrator
<b>Purpose</b>	시스템 관리자가 관리자임을 확인해 관련 작업을 수행할 수 있도록 합니다.
<b>Overview</b>	시스템 관리자가 아이디와 비밀번호를 입력하고 이를 확인하여 관리자 권한을 부여합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	17. Manage Amount 18. Manage Amount Cash
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A): Administrator, (S) : System 1. (A) 로그인 입력부를 활성화합니다. 2. (A) 자판기의 로그인 부에 아이디와 비밀번호를 입력합니다. 3. (S) 등록된 아이디와 비밀번호와 비교하고 일치할 경우 관리자 권한을 부여합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	E 3.1: 아이디와 비밀번호가 일치하지 않을 경우 아이디 혹은 비밀번호가 일치하지 않음을 반환합니다.

<b>Use Case</b>	17. Manage Amount
<b>Actor</b>	Administrator
<b>Purpose</b>	관리자가 상품의 재고를 관리합니다.
<b>Overview</b>	관리자가 상품의 재고를 수정합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	16. Login
<b>Pre-Requisites</b>	Login을 통해 관리자 권한을 얻은 상태여야 합니다. 재고를 추가하려는 상품이 시스템에 등록되어 있어야 합니다.
<b>Typical Courses of Events</b>	(A): Administrator, (S) : System 1. (A) 상품재고 관리 기능을 활성화합니다. 2. (A) 상품의 재고의 양을 추가하거나 제거하고 시스템에 해당 사항을 입력합니다. 3. (S) 부여된 정보를 기반으로 상품 재고량을 수정합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	18. Manage Amount Cash
<b>Actor</b>	Administrator
<b>Purpose</b>	관리자가 현금의 재고를 관리합니다.
<b>Overview</b>	관리자가 현금의 재고를 수정합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	16. Login
<b>Pre-Requisites</b>	Login을 통해 관리자 권한을 얻은 상태여야 합니다.
<b>Typical Courses of Events</b>	(A): Administrator, (S): System 1. (A) 현금 관리 기능을 활성화합니다. 2. (A) 현금의 양을 추가하거나 제거하고 시스템에 해당 사항을 입력합니다. 3. (S) 부여된 정보를 기반으로 현금 재고량을 수정합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

### 3.3 Performance requirements

- 현재위치에서 가장 가까운 DVM 을 반환 할때에는 최대 1초가 넘지 않도록 합니다.
- 상품을 결제완료하면 0.01초안으로 상품이 반환됩니다.
- 코드가 생성되기까지 0.5초가 넘지 않도록 합니다.

### 3.4 Design constraints

- OOAD 개발 방법론을 기반으로 진행한다.
- JAVA 언어를 이용해 개발한다.
- Windows 10 시스템 상에서 개발한다.

### 3.5 Software system attributes

#### 3.5.1 Reliability

- 윈도우 10 이상 권장
- DVM전원이 켜져 있어야 한다

#### 3.5.2 Maintainability

- 관리자 권한으로 물품목록과 재고를 관리할수 있다

#### 3.5.3 Portability

- JAVA를 사용하여 JAVA를 사용 가능한 플랫폼에는 모두 이식이 가능하다

### 3.6 Other requirements

- 자판기가 보유한 상품의 상태 (유통기한, 파손 등) 여부와 사용자의 상품 수령 후 상품 상태에 따른 후속 처리와 같은 판매 이외의 과정에 대해서는 고려하지 않는다.
- **DVM** 내부 모든 음료의 상태는 판매 가능한 상태이다