

Software Requirement Specification

For the DVM Controller

3조

2021-05-02

201511238 허준호

201911147 강속영

201711320 오준엽

201511218 이지영

201710397 이지훈

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

3.1.1. User Interfaces

3.1.1.1 터치 스크린

3.1.1.2 카드 리더기

3.1.2 Hardware Interfaces

3.1.2.1 터치 스크린

3.1.2.2 취출구

3.1.3. Software Interfaces

3.1.3.1 카드 결제 시스템 소프트웨어

3.1.4. Communication Interfaces

3.1.4.1 message

3.2. Functional requirements

3.2.1 Direct Payment

3.2.1.1 Introduction / Purpose of feature

3.2.1.2 Stimulus / Response sequence

3.2.1.2.1 sequence

3.2.1.2.2 sequence 2

3.2.1.2.3 sequence 3

3.2.1.3 Associated functional requirements

3.2.1.3.1 Functional requirement 1

3.2.1.3.2 Functional requirement 2

3.2.1.3.3 Functional requirement 3

3.2.2 Prepayment

3.2.2.1 Introduction / Purpose of feature

3.2.2.2 Stimulus / Response sequence

3.2.2.2.1 sequence 1

3.2.2.2.2 sequence 2

3.2.2.2.3 sequence 3

3.2.2.2.4 sequence 4

3.2.2.3 Associated functional requirements

3.2.2.3.1 Functional requirement 1

3.2.2.3.2 Functional requirement 2

3.2.2.3.3 Functional requirement 3

3.2.3 Payment with Precode

3.2.3.1 Introduction / Purpose of feature

3.2.3.2 Stimulus / Response sequence

3.2.3.2.1 sequence 1

3.2.3.2.2 sequence 2

3.2.3.3 Associated functional requirements

3.2.3.3.1 Functional requirement 1

3.2.3.3.2 Functional requirement 2

3.2.4 Payment Fail

3.2.4.1 Introduction / Purpose of feature

3.2.4.2 Stimulus / Response sequence

3.2.4.2.1 sequence 1

3.2.4.2.2 sequence 2

3.2.4.3 Associated functional requirements

3.2.4.3.1 Functional requirement 1

3.2.4.3.1 Functional requirement 2

3.2.5 Manager Mode

3.2.5.1 Introduction / Purpose of feature

3.2.5.2 Stimulus / Response sequence

3.2.5.2.1 sequence 1

3.2.5.2.2 sequence 2

3.2.5.2.3 sequence 3

3.2.5.2.4 sequence 4

3.2.5.2.5 sequence 5

3.2.5.2.6 sequence 6

3.2.5.3 Associated functional requirements

3.2.5.3.1 Functional requirement 1

3.2.5.3.2 Functional requirement 2

3.2.5.3.3 Functional requirement 3

3.2.5.3.4 Functional requirement 4

3.2.5.3.5 Functional requirement 5

3.2.5.3.6 Functional requirement 6

3.3 Performance requirements

3.4 Design constraints

3.5 Software system attributes

Change Log

2.5	6	아이템 -> 상품
	10	추가
2.5.7		'카드 결제시 잔고를 확인하는 ... 대체한다.' 추가
2.5.8		'선결제코드는 선결제 ... 사용가능하다.' 추가
2.5.9		'판매하는 음료의 총 개수는 최대 20개이다.' 추가
3.1.1	2	'기본적으로... 음료 정보를 출력' -> '각각의 DVM은 ... 정보를 출력'
3.2.1.2.1	seq 1	'카드가 정상적으로 투입되었으면('c'입력이 확인 되었으면)' 삭제
3.2.2.2.1	resp	'카드가 정상적으로 투입 되었으면('c'입력이 확인되었으면)' 삭제
3.2.4.2.1	seq 1	'카드가 정상적으로 투입되었으면('c'입력이 확인 되었으면)' 삭제
3.2.4.2.2	seq 2	'사용자가 터치... 통하여 선택' -> '사용자가 터치... 통하여 선택'
3.2.5.1.4		'DVM 종료' 삭제
3.2.5.2.2	resp	'상품 추가시, 전체 상품의 ... 벗어나지 않으면' 삭제
3.2.5.2	resp	'인증에 성공한다면(비밀번호가 일치한다면)' 삭제
3.2.5.3.1		'인증에 성공한 경우에만 관리자모드 화면을 출력해야한다.' 추가
3.3	1	'디스플레이 장치의 밝기를 280nt를 넘지 않도록 한다.' 삭제
3.3	5	'DVM 내부메모리에 ... 재고의 차이가 없어야 한다.' -> 삭제
3.4	2	'JDK와 JRE를 사용하여 Java 환경을 구축한다.' 삭제
3.4	3	'tomcat과 java-spring을 사용하여 다른기기와 통신을 준비한다.' 삭제

2.5-6 아이템-> 상품

2.5.7 카드 결제시 잔고를 확인하는 절차가 존재하지 않으므로 일정한

확률(10%)로 결제를 실패시키는 것으로 대체한다. -> 추가

2.5.8 선결제코드는 선결제 진행 시, 안내받은 자판기에서만 사용가능하다. -> 추가

2.5.9 판매하는 음료의 총 개수는 최대 20개이다. -> 추가

3.1.1-2 기본적으로 DVM에서 판매하는 모든 음료 정보를 출력

-> 각각의 DVM은 판매하고 있는 모든 음료의 정보를 출력

3.2.1.2.1 sequence 1

카드가 정상적으로 투입되었으면('c'입력이 확인 되었으면) -> 삭제

3.2.2.2.1 response 카드가 정상적으로 투입 되었으면('c'입력이 확인되었으면) -> 삭제

3.2.4.2.1 sequence 1

카드가 정상적으로 투입되었으면('c'입력이 확인 되었으면) -> 삭제

3.2.4.2.2 sequence 2 - stimulus 사용자가 터치스크린 상의 원하는 상품의 이미지를 터치를 통하여 선택

-> 사용자가 터치스크린 상의 원하는 상품 버튼을 터치를 통하여 선택

3.2.5.1 4.DVM 종료 -> 삭제

3.2.5.2.2 response 상품 추가시, 전체 상품의 가격이 지정된 범위를 벗어나지 않으면 ->삭제

3.2.5.2 response 인증에 성공한다면(비밀번호가 일치한다면) ->삭제

3.2.5.3.1 인증에 성공한 경우에만 관리자모드 화면을 출력해야한다. -> 추가

3.3 -1 디스플레이 장치의 밝기를 280nt를 넘지 않도록 한다.-> 삭제

3.3-5 DVM 내부메모리에 들고 있는 재고와 화면에 출력되는 재고의 차이가 없어야 한다. -> 삭제

3.4 -2 JDK와 JRE를 사용하여 Java 환경을 구축한다. -> 삭제

3.4 -3 tomcat과 java-spring을 사용하여 다른기기와 통신을 준비한다. -> 삭제

2.5 - 10. 각 자판기들에 들어있는 각 상품의 재고는 0개이상 100개이하이다. -> 추가

2.5 - 11. 자판기에 같은 이름의 상품은 등록할 수 없다. -> 추가

3.2.5.1 Introduction / Purpose of feature 상품의 재고 수량 -> 상품의 재고 -> 수정

3.2.5.2.4 & 3.2.5.2.5 수정

3.2.5.2.6 -> 3.2.5.2.6 수정

3.2.5.3.4 & 3.2.5.3.5 통합

3.2.5.3.6 -> 3.2.5.3.5 수정

1.Introduction

1.1. Purpose

본 문서의 목적은 분산자판기(DVM : Distributed Vending Machine) 시스템의 자판기 controller software를 개발하기 위함이다.

1.2. Scope

사용자가 요구하는 음료가 현재 DVM에 존재하지 않을 경우 다른 DVM과의 통신을 통해 해당 음료의 재고를 확인하고, 사용자가 다른 기기에서 해당 음료를 제공받을 수 있게 해주는 인증코드를 발급한 후, 인증코드를 재고가 있는 DVM에 입력시 그 음료를 수령할 수 있게 하는 소프트웨어이다.

1.3. Definitions,Acronyms, and Abbreviations

DVM: Distributed vending machine의 약자이며 한글로 분산 자판기 라고 한다. 이는 추후 설계 및 조립으로 구현된 구체적인 제품을 가리키는 말이기도 하다.

상품: DVM에서 판매하는 20가지 음료

prepayment: 선결제

precode: 선결제시 사용자에게 제공되는 코드

stock : DVM에 존재하는 상품(음료)의 재고

1.4. References

IEEE Std 830의 서술 방식을 참조하여 본 문서를 작성하였다.

Preliminary functional requirements Distributed vending machine(유준범 2021.03)

OOPT_All(2020)(유준범 2021.03)

1.5. Overview

2장은 DVM 제품들의 관계, 그리고 DVM 소프트웨어와 하드웨어의 기능을 간략하게 제시하는 문단이다.

3장은 2장에서 제시한 DVM의 기능 및 인터페이스를 상세하게 설명하는 문단이다.

2. Overall description

2.1 Product perspective

DVM은 자판기들이 네트워크를 통해 연결되어 있는 구조이다.

2.2 Product functions

1. 카드 결제를 제공한다.
2. DVM은 결제된 상품을 제공해준다.
3. 현 DVM에 존재하지 않는 상품을 선택할 시 선결제 방식을 사용할 수 있다.
4. 선결제 방식을 사용할 경우 거리가 가장 가까운 위치의 DVM을 안내해준다.
5. 선결제 방식을 사용할 경우 결제 후 인증코드가 발급 된다.
6. 다른 자판기에서 인증코드를 입력하면 상품이 나온다.
7. 관리자는 인증수단을 통해 인증을 한 후 새로운 상품을 추가하고, 재고를 관리, 상품의 가격을 변경, 판매 내역을 확인 할 수 있다.

2.3 User characteristics

1. 사용자는 DVM의 터치 스크린을 상의 넘버패드를 통해 선결제코드를 입력할 수 있다.
2. 사용자는 DVM의 터치 스크린 상의 상품의 아이콘을 터치하여 상품을 선택할 수 있다.
3. 사용자는 DVM에서 구매한 상품이 취출구를 통해 제공되는 것을 알고 있다.

2.4 Constraints

1. 일시불 카드 결제만 할 수 있다.
2. 환불은 불가능하다.
3. 터치패드에서 한번에 두개 이상의 입력이 들어올 수 없다.

2.5 Assumptions and dependencies

1. 네트워크 통신 에러는 없다고 가정한다.
2. 각 자판기들의 서로간의 거리는 모두 다르다고 가정한다.
3. 각 자판기의 하드웨어 상의 오류는 없다고 가정한다.
4. DVM은 최대 10대까지 가능하다.
5. 하나의 DVM에서 7가지 음료만 판매한다.

6. 메뉴에서 삭제된 상품의 재고는 관리자가 회수한다.
7. 카드 결제시 잔고를 확인하는 절차가 존재하지 않으므로 일정한 확률(10%)로 결제를 실패시키는 것으로 대체한다.
8. 선결제코드는 선결제 진행 시, 안내받은 자판기에서만 사용가능하다.
9. 판매하는 음료의 총 개수는 최대 20개이다.
10. 각 자판기들에 들어있는 각 상품의 재고는 0개이상 100개이하이다.
11. 자판기에 같은 이름의 상품은 등록할 수 없다.

3. specific requirements

3.1. External interface requirements

3.1.1. User Interfaces

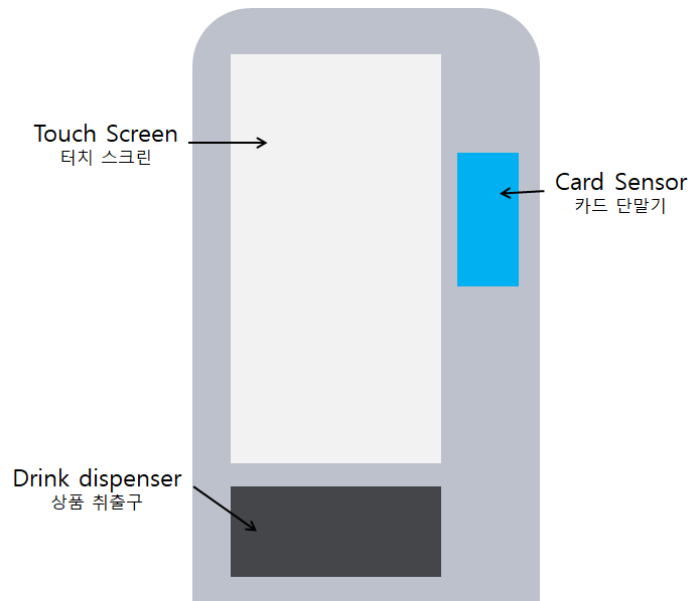
3.1.1.1 터치 스크린

1. 사용자가 상황에 따라 의사를 표현할 수 있는 터치 스크린
2. 각각의 DVM은 판매하고 있는 모든 음료의 정보를 출력
3. 선결제 시 선결제 코드와 수령할 자판기의 위치를 출력
4. 각 상황에 따른 결과를 출력
5. 터치 스크린 상의 넘버패드를 통해 선 결제 코드를 입력할 수 있음
6. 관리자는 터치스크린 상의 넘버패드를 통해 비밀번호를 입력하여 관리자모드에 진입할 수 있음

3.1.1.2 카드 리더기

1. 사용자의 카드 정보를 읽을 단말기
2. 카드의 정보를 리딩하여 사용자가 선택한 상품의 금액을 결제 해주는 시스템
3. 위의 행위를 실제로 구현할 수 없으므로 키보드를 통한 알파벳 'c'입력으로 대체

3.1.2 Hardware Interfaces



3.1.2.1 터치 스크린

1. 사용자 입력을 SW로 전달하거나 시스템 오퍼레이션 결과를 출력

3.1.2.2 취출구

1. 결제 완료 시 해당 음료를 배출
2. 위의 행위를 실제로 구현할 수 없으므로 결제 완료를 알리는 화면으로 대체

3.1.3. Software Interfaces

3.1.3.1 카드 결제 시스템 소프트웨어

3.1.4. Communication Interfaces

3.1.4.1 message

1. DVM간의 의사소통 기능
2. Input : msg
3. Output: msg

3.2. Functional requirements

3.2.1 Direct Payment

3.2.1.1 Introduction / Purpose of feature

현재 자판기에 재고가 있는 상품을 구매

3.2.1.2 Stimulus / Response sequence

3.2.1.2.1 sequence 1

- stimulus: 카드를 투입(키보드 알파벳 'c' 입력)
- response: 상품 목록 화면 출력

3.2.1.2.2 sequence 2

- stimulus: 사용자가 터치스크린 상의 원하는 상품 버튼을 터치로 통하여 선택
- response: 현 자판기에 선택한 상품의 재고가 존재함을 확인하고 결제를 진행한다.

3.2.1.2.3 sequence 3

- stimulus: 결제여부 확인
- response: 상품을 취출구를 통해 사용자에게 제공(결제 완료 화면을 출력)

3.2.1.3 Associated functional requirements

3.2.1.3.1 Functional requirement 1

- 대기 화면이 출력되어야 한다.
- 터치 스크린에 등록되어있는 모든 상품이 출력되어야 한다.

3.2.1.3.2 Functional requirement 2

- 사용자가 선택한 상품의 재고가 현재 DVM에 존재해야한다.

3.2.1.3.3 Functional requirement 3

- 카드 결제 요청이 승인 되어야한다.
- 상품 배출한 뒤에 해당 상품의 재고 정보를 업데이트 해야한다.

3.2.2 Prepayment

3.2.2.1 Introduction / Purpose of feature

현재 자판기에 재고가 존재하지 않은 상품을 선결제

3.2.2.2 Stimulus / Response sequence

3.2.2.2.1 sequence 1

- stimulus: 카드를 투입(키보드 알파벳 'c' 입력)
- response: 상품 목록 제공

3.2.2.2.2 sequence 2

- stimulus: 사용자가 터치스크린 상의 원하는 상품의 이미지를 터치로 통하여 선택
- response: 현 자판기에 선택한 상품의 재고가 존재하지 않음을 확인하고 제품의 재고가 존재하는 현 자판기로부터 가장 가까운 위치의 자판기의 위치를 안내하고 선 결제 선택 화면을 출력

3.2.2.2.3 sequence 3

- stimulus: 사용자가 선결제를 선택
- response: 사용자가 선택한 상품에 대한 결제를 진행한다.

3.2.2.2.4 sequence 4

- stimulus: 결제 여부 확인

- **response:** 선결제 코드와 재고가 존재하는 자판기의 위치를 화면에 출력한다.

3.2.2.3 Associated functional requirements

3.2.2.3.1 Functional requirement 1

- 대기 화면이 출력되어야 한다.
- 터치 스크린에 등록되어있는 모든 상품이 출력되어야 한다.

3.2.2.3.2 Functional requirement 2

- 사용자가 선택한 상품의 재고가 현재 자판기에 존재하지 않아야 한다.

3.2.2.3.3 Functional requirement 3

- 카드 결제 요청이 승인 되어야한다.
- 선결제 코드를 출력하고 해당 상품의 재고 정보를 업데이트 해야한다.

3.2.3 Payment with Precode

3.2.3.1 Introduction / Purpose of feature

선결제시 안내받은 다른 자판기에서 선결제 코드를 통해 상품을 수령

3.2.3.2 Stimulus / Response sequence

3.2.3.2.1 sequence 1

- **stimulus:** 선결제 상품 입력을 선택
- **response:** 선결제 코드를 입력하는 화면 출력

3.2.3.2.2 sequence 2

- **stimulus:** 선결제 코드 입력
- **response:** 음료 배출

3.2.3.3 Associated functional requirements

3.2.3.3.1 Functional requirement 1

- 대기 화면이 출력되어야 한다.
- 선결제 코드 입력화면이 출력되어야한다.

3.2.3.3.2 Functional requirement 2

- 입력한 선결제 코드가 발급된 것이 맞는 지 확인해야한다.
- 상품 배출하고 해당 상품의 재고 정보를 업데이트 해야한다.

3.2.4 Payment Fail

3.2.4.1 Introduction / Purpose of feature

모든 자판기에 선택한 제품의 재고가 존재하지 않아 상품 구매를 실패

3.2.4.2 Stimulus / Response sequence

3.2.4.2.1 sequence 1

- **stimulus:** 카드를 투입(키보드 알파벳 'c' 입력)
- **response:** 상품 목록 화면을 출력

3.2.4.2.2 sequence 2

- **stimulus:** 사용자가 터치스크린 상의 원하는 상품 버튼을 터치를 통하여 선택
- **response:** 현 자판기와 다른 모든 자판기들에 선택한 제품의 재고가 존재하지 않음을 확인하고 재고가 없음을 알리는 화면을 출력

3.2.4.3 Associated functional requirements

3.2.4.3.1 Functional requirement 1

- 대기 화면이 출력되어야 한다.
- 터치 스크린에 등록 되어있는 모든 상품이 출력되어야 한다.

3.2.4.3.1 Functional requirement 2

- 사용자가 선택한 상품의 재고가 모든 자판기에 존재하지 않아야 한다.

3.2.5 Manager Mode

3.2.5.1 Introduction / Purpose of feature

관리자가 DVM의 관리를 위해 인증 수단을 통해서 관리자모드로 진입
관리자 모드는 (1. 상품 추가/삭제, 2. 상품의 재고& 가격 변경 3.
판매내역 확인)으로 이루어진다.

3.2.5.2 Stimulus / Response sequence

3.2.5.2.1 sequence 1

- stimulus: 관리자가 인증수단을 통해 관리자모드 진입 요청(넘버패드를 통해 비밀번호를 입력)
- response: 관리자모드 화면 출력

3.2.5.2.2 sequence 2

- stimulus: 관리자가 상품명과 상품 가격을 설정하여 상품을 추가
- response: 새로운 상품을 목록에 추가

3.2.5.2.3 sequence 3

- stimulus: 관리자가 상품 목록에서 상품을 선택하여 삭제
- response: 선택된 상품을 목록에서 제거

3.2.5.2.4 sequence 4

- stimulus: 관리자가 선택한 상품의 재고& 가격을 지정된 범위 내에서 변경
- response: 상품의 재고, 가격을 갱신

3.2.5.2.6 sequence 5

- stimulus: 관리자가 현 자판기의 판매 내역 확인
- response: 상품별 판매수량, 총 판매금액을 화면에 출력

3.2.5.3 Associated functional requirements

3.2.5.3.1 Functional requirement 1

- 카드 입력 대기 및 선결제 코드 입력 대기 화면이 출력되어야 한다.
- 관리자 모드의 인증수단을 인식할 수 있어야한다.
- 인증에 성공한 경우에만 관리자모드 화면을 출력해야한다.
- 관리자 모드 화면이 출력되어야 한다.

3.2.5.3.2 Functional requirement 2

- 상품의 추가 요청을 인식해야한다.
- 상품 추가시, 상품의 수량이 자판기가 수용할 수 있는 범위 내 인지 확인해야한다.
- 변경사항을 반영해야한다.

3.2.5.3.3 Functional requirement 3

- 상품의 삭제 요청을 인식해야한다.

- 변경사항을 반영해야한다.

3.2.5.3.4 Functional requirement 4

- 재고& 가격 변경 반영 요청을 인식해야한다.
- 재고& 가격 변경 화면을 출력되어야 한다.
- 재고와 가격의 변경 사항이 지정된 범위 내인지 확인해야한다.
- 변경사항을 반영해야한다.

3.2.5.3.5 Functional requirement 5

- 판매 내역 요청을 인식해야한다.
- 상품별 판매수량, 총 판매금액을 화면에 출력해야한다.

3.3 Performance requirements

1. 터치스크린의 반응속도가 0.1초 이내여야 한다.
2. 선결제시 재고가 존재하는 자판기들과 현 자판기의 거리를 1초 이내에 계산하여 재고가 존재하는 가장 가까운 자판기의 위치를 안내해야 한다.
3. 화면 이동 애니메이션에 버벅임이 없어야된다.

3.4 Design constraints

1. OOAD 개발 방법론을 사용한다.

3.5 Software system attributes

1. Security
관리자가 아닌 외부인이 자판기 내부 정보를 수정 할 수 없게 제작한다.
2. Maintainability
유지보수가 가능하게 제작한다.
3. Accessibility
DVM은 누구나 쉽게 이용할 수 있게 간단한 조작방식을 사용한다.

3.6 Other requirements

1. DVM의 내부의 소프트웨어가 정상 작동 하기 위해서는 전기 에너지인 전원을 공급받아야 한다.

2. DVM 관리자는 DVM이 정상적으로 동작이 가능하도록 전원을 공급해줄 수 있는 전원 연결선이나 내부에 내장된 배터리가 일정수준 이상 충전되어있는지에 대해 관리해야한다.
3. DVM 관리자는 매주 각 자판기의 선택 가능한 음료의 재고를 확인하고 채워, 사용자가 불필요하게 다른 자판기로 이동하는 일이 없도록 관리해야한다.
4. DVM의 신뢰성을 보장하기 위해선 매번 동일한 환경에서 DVM이 같은 동작이 일어날 수 있도록 소프트웨어를 설계해야한다.
5. DVM의 안정성을 보장하기 위해선 주위 상황에 따라서 센서가 이를 잘 인식하는지, 인식한다면 올바른 디지털 신호로 변환하는지, 이러한 신호를 올바르게 해석하여 DVM의 의도대로 잘 움직이게 명령을 내리는지 등을 확인해야 한다.
6. DVM의 재사용성을 보장하기 위해선 DVM에 새 기능을 추가할때 기존 기능과 충돌되는 지점이 적어야 한다. 그리고 네트워크에 연결되는 자판기를 추가하거나, 기존 기능을 유지보수할 때 소프트웨어의 큰 수정이 없어야 한다.