Preliminary Functional Requirements

Distributed Vending Machine

Project Team

OOO Team

Date

0000-00-00

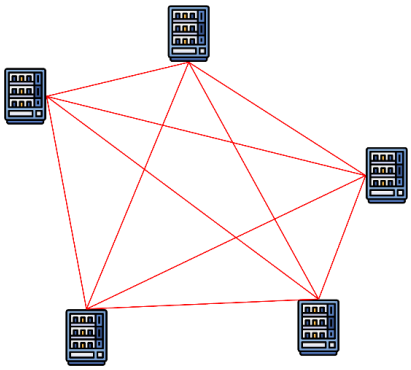
Team Information

1 Overall Description

1.1 Product Perspective

분산형 자판기 시스템을 위한 controller SW를 개발한다.

분산 시스템이란 **서버 없이** 각 시스템이 서로 통신하여 동작하는 시스템이다. 분산형 자판기 시스템에서 사용자는 현재 자판기에서 판매하지 않는 제품이 어느 자판기에서 판매되고 있는지 파악하거나 현재 자판기에서 다른 자판기의 상품을 구매할 수 있다.



1.2 Requirements

- 총 음료의 개수는 20 종류이다.
- 하나의 자판기는 그 중 7 종류의 음료를 취급한다.
  - 취급하지 않는 음료도 선택할 수 있도록 메뉴는 제공한다.
- 결제는 카드로 한다.
  - 잔액 혹은 한도가 부족한 경우 결제되지 않는다.
- 사용자가 선택한 자판기에서 취급하고 재고가 있는 음료를 선택 후 결제하면 즉시 음료가 제공된다.
- 현재 자판기에서 취급하지만 재고가 없는 음료 또는 현재 자판기에서 취급하지 않는 음료를 구매하고자 하는 경우 다른 자판기에 해당 음료의 재고를 확인한 후 위치를 안내한다.
  - broadcast msg를 통해 판매 여부와 재고를 요청하여 확인한다.
  - 응답 msg를 통해 대상 자판기의 id와 위치를 확인하여 안내한다.
  - 사용자가 위치한 곳의 자판기에서 가장 가깝고 구매하고자 하는 음료의 재고가 있는 자판기의 위치를 안내한다.
- 현재 자판기에서 취급하지 않거나 재고가 없는 음료에 한하여 선결제를 할 수 있다.
  - 선결제를 요청하는 경우 현재 자판기에서 결제 후 인증코드를 발급한다.
  - 인증코드는 알파벳 대소문자와 숫자를 포함하는 랜덤한 10자리의 문자열이다. 매번 선결제를 요청할 때마다 시스템은 새로운 인증코드를 생성해야 한다.

  - ■ 사용자가 선택한 자판기에서 가장 가깝고 선결제한 음료의 재고를 가지고 있는 자판기를 선택해야 한다. 사용자가 해당 자판기로 가서 발급받은 인증코드를 입력하면 음료가 나온다.
  - ■ 선결제가 완료된 음료의 경우, 재고가 소진되지 않도록 음료를 지급할 자판기에서는 이미 판매된 것으로 처리해야 한다.
  - ■ 선결제를 마친 후 사용자에게 인증코드와 선결제를 마친 자판기의 위치 정보(좌표) 등 필요한 정보를 안내해야 한다.
- 매번 통신을 주고받을 때마다 자판기의 좌표를 받아와 직선 거리를 계산한다.
  - ■ 자판기의 좌표 정보는 x 좌표의 정보와 y 좌표의 정보로 이루어지며, x와 y의 범위는 0~99이다.
  - ■ 만일 직선 거리가 같은 자판기에 동일한 음료가 있고, 두 자판기에 모두 재고가 있다면 id의 숫자가 작은 자판기로 안내한다.
  - ■ 완벽하게 같은 위치에 두 개 이상의 자판기가 존재할 수 없다. 즉, 각 자판기의 좌표 정보는 서로 동일할 수 없다.

1.3 User characteristics

- 해당 자판기 시스템을 사용하는 사용자는 어느 정도 자판기 사용법에 대한 지식이 있다고 가정한다.

1.4 Constraints

- 해당 SW는 java를 통해 구현하며, java application, 즉 executable (runnable) jar 파일이 최종 결과물로서 개발되어야 한다. UI (user interface)를 제공해야 하며, 사용자와의 상호작용은 모두 UI를 통해 이루어져야 한다.
- 자판기 사이의 msg 전달은 소켓 통신을 사용하여 구현한다.
- 각 자판기는 메시지를 주고받기 위해 server와 client를 모두 가지고 있어야 한다.
- 자판기에서 자체적으로 카드 정보를 가지고 있을 수 없다. 즉, 카드 정보는 외부의 다른 시스템 (카드사, 은행 등)에서 가지고 있어야 하며, 자판기 내부에 저장될 수 없고, 필요 시마다 호출하여 사용하는 휘발성 정보이다.
- 선결제를 마친 후 사용자에게 제공하는 정보는 화면 상에 새로운 팝업 창을 출력하는 것으로 대체한다.

1.5 Assumptions and dependencies

- 각 자판기는 모두 네트워크에 연결되어 있고 각자의 네트워크 연결 정보 (ip 주소, 포트 번호)는 미리 알고 있다고 가정한다.
- 각 자판기의 판매 음료 종류는 사전에 결정된다.
- 각 자판기에서 가지고 있는 각 음료의 최대 재고는 99개로 한정한다.
- 자판기 사이에 주고받는 메시지는 json 형식을 따르며, msg format은 다음의 **표 1~표 4**와 같다.
  - src_id와 dst_id의 value는 각 조를 의미한다. 각 조의 번호는 과목 홈페이지의 Team Projects 페이지를 참조한다.
  - Broadcast msg의 경우 메시지를 전송할 때 dst_id의 value를 0으로 한다.
  - 주고받는 msg에서 음료 개수는 10진수의 숫자로 표기한다. (0~99)
  - 음료의 코드는 최하단의 **표 5**를 참조한다.

표 1 재고 확인 요청 시의 msg format

key	**Value**	
msg_type	보내는 msg의 종류. {req_stock}	
src_id	현재 자판기의 id. {Team1, Team2, …, Team9}	
dst_id	통신 대상 자판기의 id. {0}	
msg_content	**key**	**Value**
	item_code	음료의 코드. {00~20}
	item_num	음료의 개수. {0~99}

표 2 재고 확인 응답 시의 msg format

key	**Value**	
msg_type	보내는 msg의 종류. {resp_stock}	
src_id	현재 자판기의 id. {Team1, Team2, …, Team9}	
dst_id	통신 대상 자판기의 id. {Team1, Team2, …, Team9}	
msg_content	**key**	**Value**
	item_code	음료의 코드. {00, 01, …, 20}
	item_num	음료의 개수. {0, 1, …, 99}
	coor_x	x축 상의 좌표. {0, 1, …, 99}
	coor_y	y축 상의 좌표. {0, 1, …, 99}

표 3 선결제 요청 시의 msg format

key	Value	
msg_type	보내는 msg의 종류. {req_prepay}	
src_id	현재 자판기의 id. {Team1, Team2, …, Team9}	
dst_id	통신 대상 자판기의 id. {Team1, Team2, …, Team9}	
msg_content	**key**	**Value**
	item_code	음료의 코드. {00, 01, …, 20}
	item_num	음료의 개수. {0, 1, …, 99}
	cert_code	인증코드. {[a-zA-Z0-9]*}

표 4 선결제 가능 여부 응답 시의 msg format

key	Value	
msg_type	보내는 msg의 종류. {resp_prepay}	
src_id	현재 자판기의 id. {Team1, Team2, …, Team9}	
dst_id	통신 대상 자판기의 id. {Team1, Team2, …, Team9}	
msg_content	**key**	**Value**
	item_code	음료의 코드. {00, 01, …, 20}
	item_num	음료의 개수. {0, 1, …, 99}
	availability	선결제 가능 여부. {T/F}

- 각 자판기에서 판매할 수 있는 음료의 종류는 다음과 같다.

표 5 음료의 종류 및 코드

음료 종류 (음료 코드)			
콜라 (01)	사이다 (02)	녹차 (03)	홍차 (04)
밀크티 (05)	탄산수 (06)	보리차 (07)	캔커피 (08)
물 (09)	에너지드링크 (10)	유자차 (11)	식혜 (12)
아이스티 (13)	딸기주스 (14)	오렌지주스 (15)	포도주스 (16)
이온음료 (17)	아메리카노 (18)	핫초코 (19)	카페라떼 (20)