
분산 자판기 시스템(DVM)

Team Information

이 채 성, 고 형 민, 김 모 세

1

1. Selected Architectural Drivers

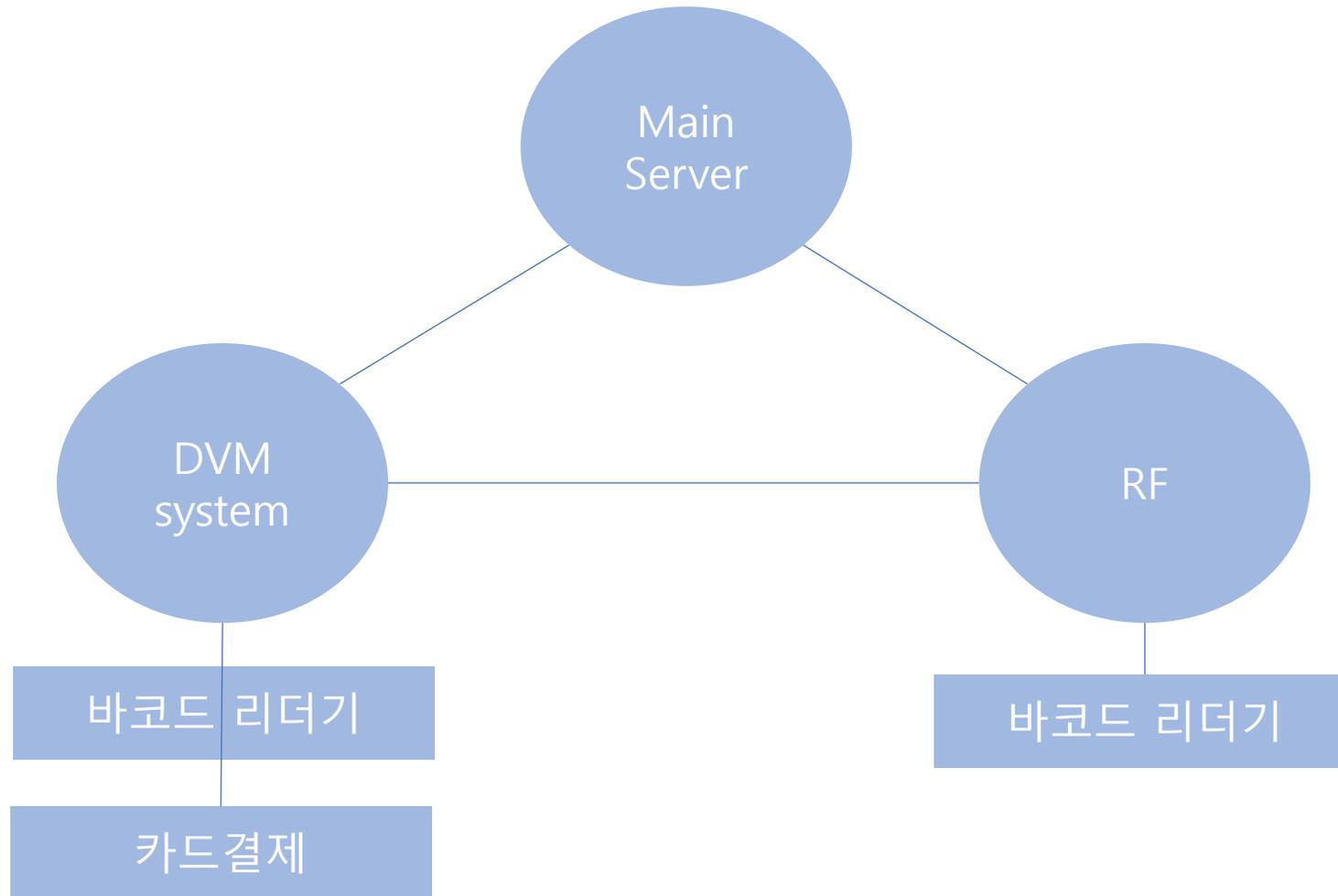
Usecase	Description
UC1	사용자가 재고가 있는 상품을 선택하여 구매하면 상품이 지급되어야 한다.
UC2	사용자가 재고가 없는 상품을 선택할 경우 쿠폰 구매 의사를 물어본 후 결제를 진행한다.
UC3	사용자가 재고가 없는 상품을 선택하고 결제를 진행하지 않았을 경우 해당 상품이 있는 자판기 위치를 안내한다.
UC4	사용자가 쿠폰을 가지고 있는 경우 쿠폰으로 상품을 지급 받는다.
UC5	자판기에 오류가 발생한 경우 사용자는 관리자에게 연락을 취할 수 있다.
UC6	자판기 관리자는 자판기 내의 정보(재고, 자판기 설정, 냉장고 정보)를 관리한다.
UC7	냉장고 관리자는 냉장고 내의 정보(재고, 냉장고 설정, 자판기 정보)를 관리한다.
UC8	자판기 관리자는 메인 서버를 통해 관리자, 자판기, 냉장고, 상품 정보 등을 관리한다.
UC9	물류 관리자는 메인 서버를 통해 냉장고 위치와 재고를 확인할 수 있다.

ID	Concern
CRN1	라이선스 비용이 발생하지 않거나 저렴해야 한다
CRN2	쿠폰 사용성을 위해 쿠폰 번호는 휴대폰 번호 입력시 휴대폰 등으로도 전송되어야 한다
CRN3	자판기 관리자와 물류 관리자가 따로 지정되어 있고, 물류를 제외한 전반적인 시스템은 자판기 관리자가 관리 한다.

ID	Quality Attribute	Scenario	Associated use case
QA1	접근성	자판기의 상품이 품절되면, 빠르게 다시 채워져야 하고, 다른 자판기에 재고가 있다면 안내 해야 한다	Uc3/4
QA2	가용성	가용성 일부 시스템이 사용할 수 없는 상황이 되더라도, 자판기는 사용 가능 해야 한다	all
QA3	성능	자판기의 응답 속도가 보장 되어야 한다	All
QA4	업데이트 용이성	상품, 자판기, 물류 정보 등을 일일이 기기마다 입력하지 않고 업데이트가 용이 해야 한다	uc8
QA5	사용성	새로운 시스템은 사용자와 관리자 모두가 사용하기 편리한 기능이 있어야 한다	all
QA6	확장성	자판기 수/냉장고 수 등이 가변적일수 있으며 많은 자판기/냉장고가 추가 될 수 있어야 한다	Uc8/9

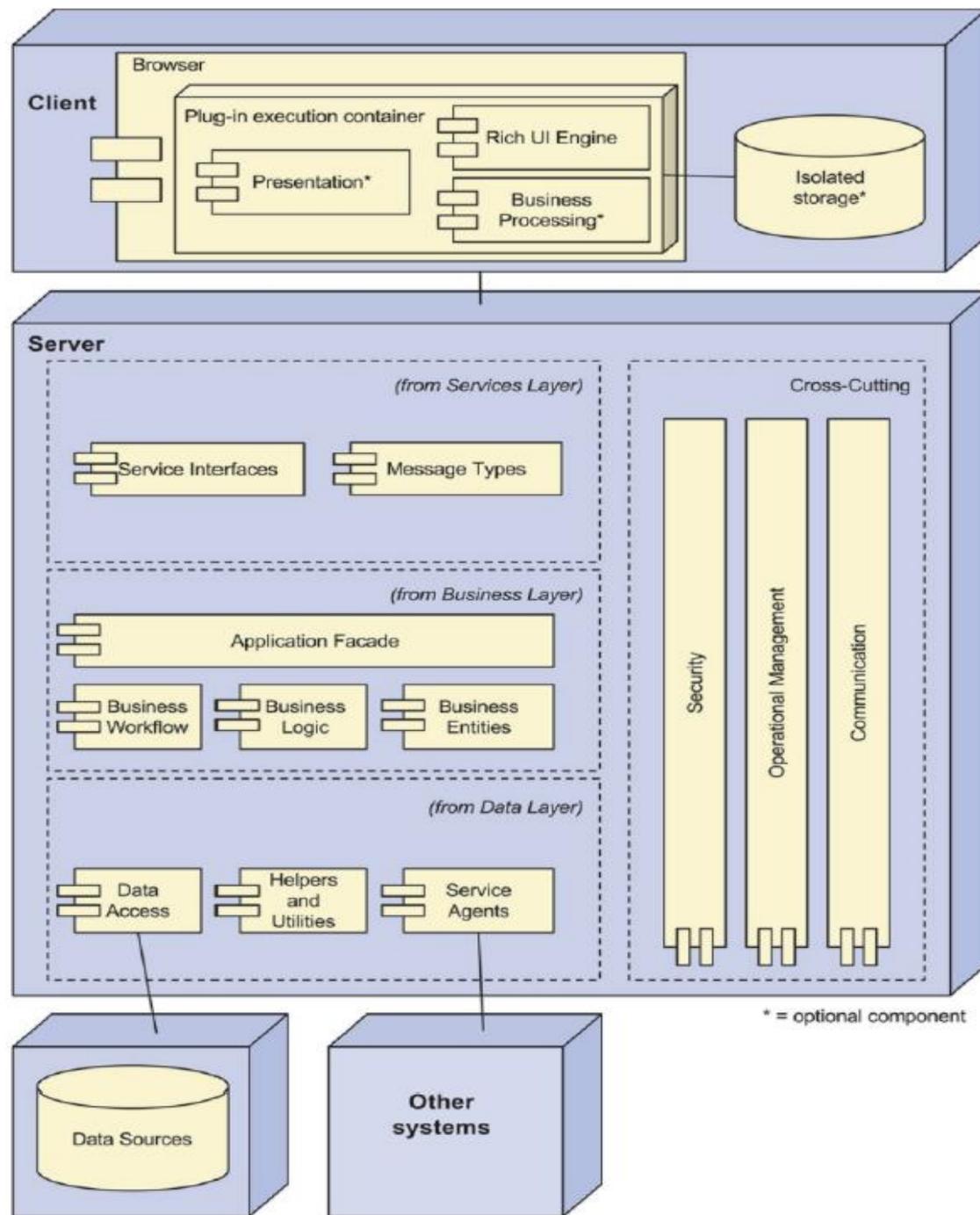
ID	Constraint
CON1	개발 인건비를 줄이기 위하여 각 시스템은 동일한 구조를 가져야 하며, 최소한의 수정만으로 각 시스템에 최적화 되어야 한다.
CON2	네트워크는 불안정할수 있으며, 네트워크 연결이 끊어지더라도 일부 기능(쿠폰 사용/발급 등)을 제외한 자판기 필수 기능은 정상적으로 운영되어야 함
CON3	각 자판기는 작동이 정지될 수 있으며, 작동이 정지되지 않은 자판기는 정상적으로 운영되어야 함
CON4	자판기를 관리하는 인원은 매우 적으며, 물류 비용을 아끼기 위해 물류인력은 긴 시간 간극을 두고 인력을 호출함

3. Choose elements of the system to refine



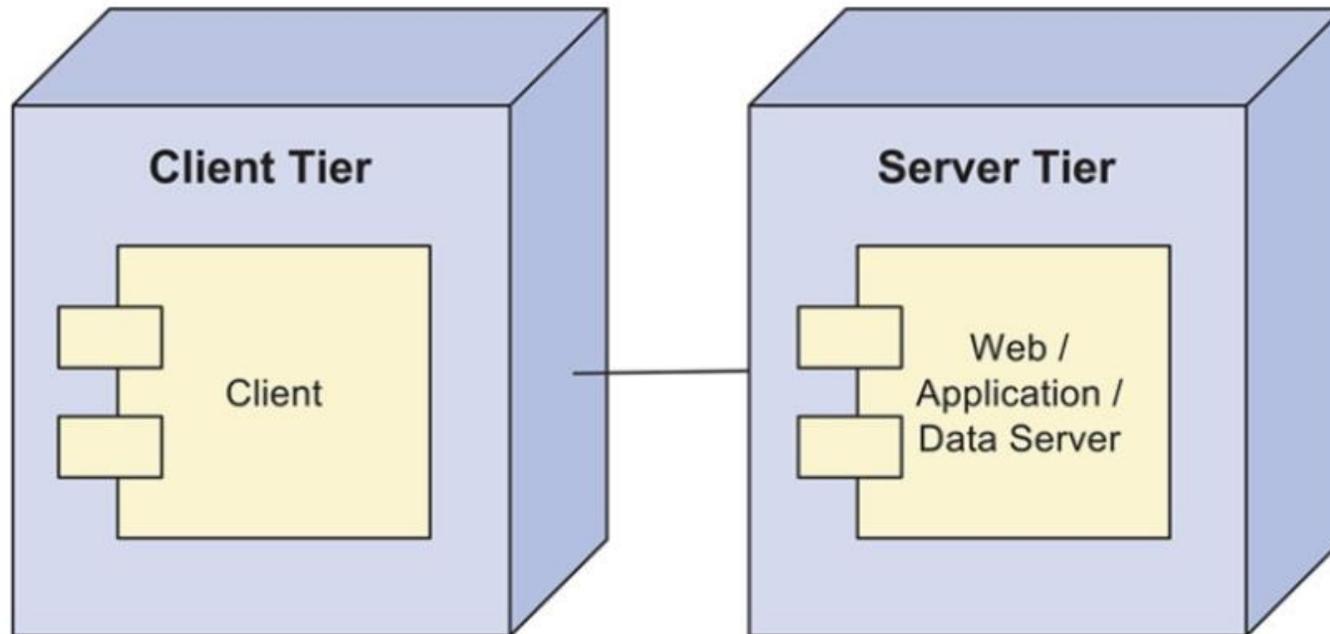
4.Choosing Design Concept

Design Decision	rationale
Internet Application	COND1에서 정의한 것과 같이 개발 인건비를 절감하기 위하여 모든 시스템이 동일한 구조로 되어 있어야 하며, 최소한의 수정만으로 각 기기의 적용될 수 있도록 해야함
	CNR1에서 정의한 것과 같이 라이선스 비용을 절감하기 위하여 공개된 프레임 워크를 사용해야 한다.
	대부분의 기능 요구사항을 해결할 수 있는 능력이 필요하며 높은 유연성과 확장성이 요구되므로, 따라서 수행 능력이 강력하고 확장성이 좋은 인터넷 애플리케이션을 선택
Discarded alternatives	자판기에 리치 클라이언트 애플리케이션, 메인 서버에 인터넷 애플리케이션으로 각 시스템을 구상하는 방식을 고려 하였으나 개발 난이도가 올라가고 인건비가 증가하는 관계로 채택하지 않음



4.Choosing Design Concept

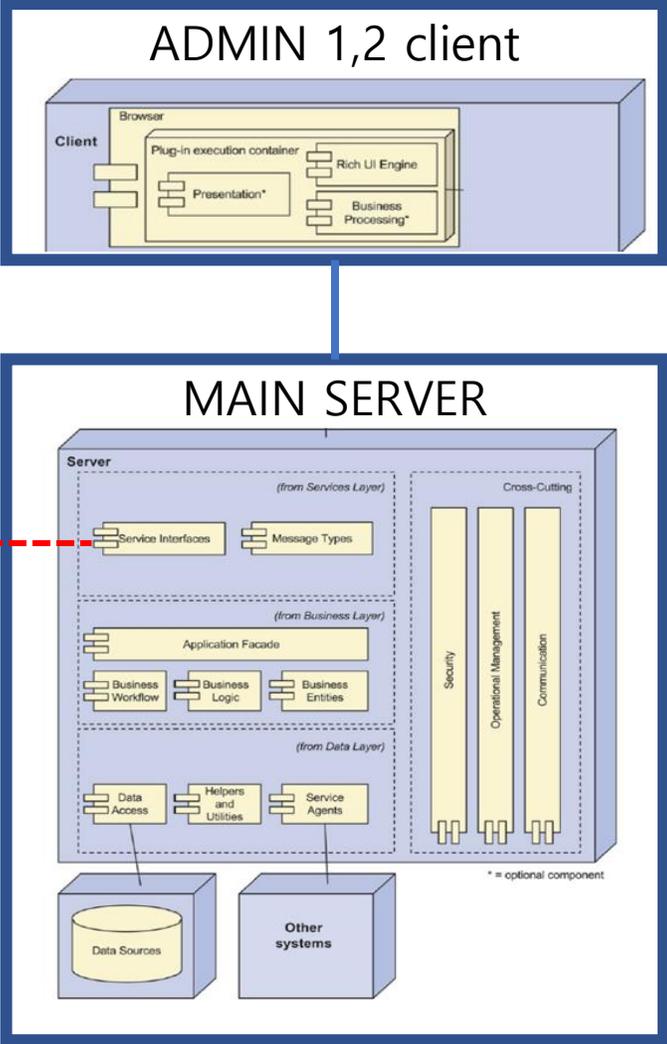
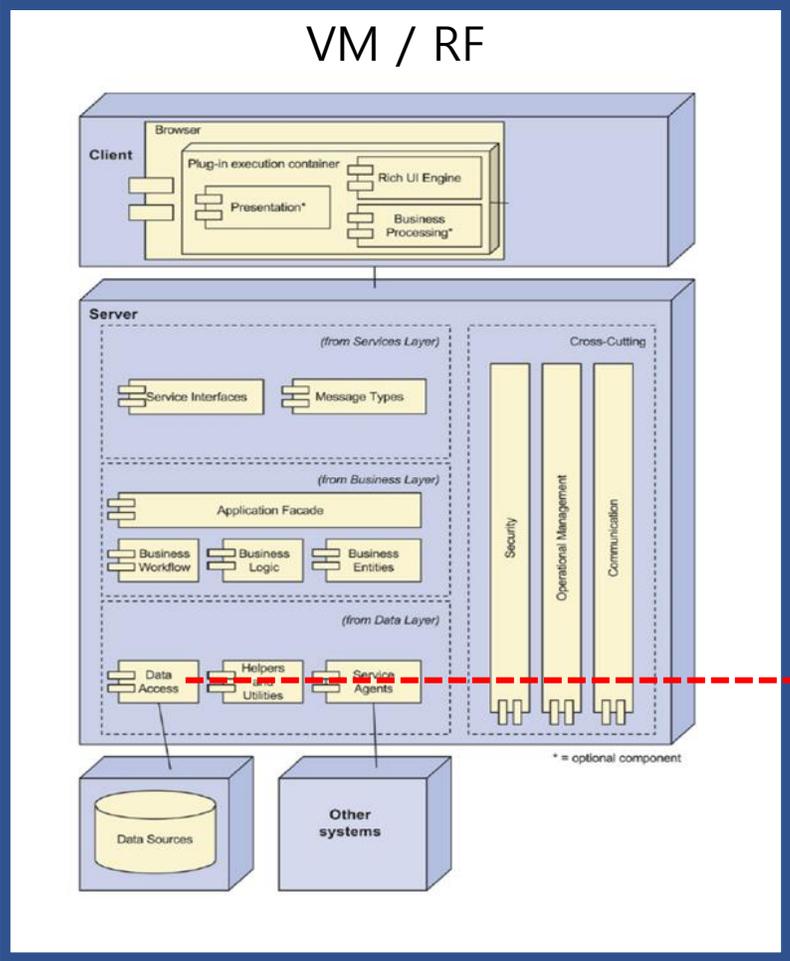
Deployment pattern	rationale
2 tier pattern	QA2, QA6를 고려하여 이를 만족하기 위해서 간단한 구조를 가지면서, 독립성을 보장할 수 있어야 되므로 각 시스템은 2tier 패턴으로 구성 되어야 한다



5.Choosing Design Concept

Design Decision and Location	Rationale
클라이언트 내에 분리된 저장소를 제거함	클라이언트의 상태를 저장할 필요가 없기 때문에 상태 저장에 필요한 분리 저장소가 필요하지 않다.

6. Sketch views and record design decisions



7.Kanban Board

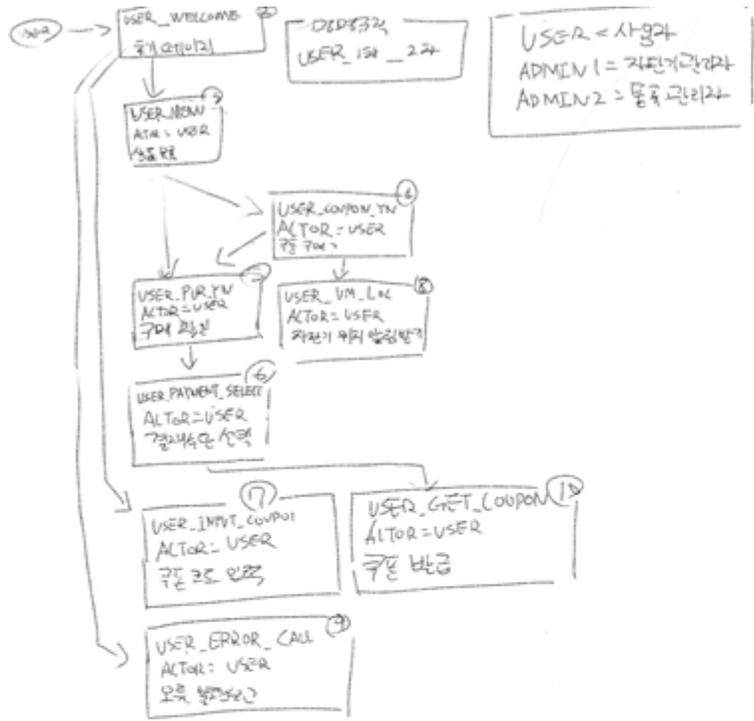
Completely	Partially	Not	Design Decisions
	UC1		전반적인 구조를 설립함으로써 부분적인 해결
	UC2		전반적인 구조를 설립함으로써 부분적인 해결
	UC3		전반적인 구조를 설립함으로써 부분적인 해결
	UC4		전반적인 구조를 설립함으로써 부분적인 해결
	UC5		전반적인 구조를 설립함으로써 부분적인 해결
	UC6		전반적인 구조를 설립함으로써 부분적인 해결
	UC7		전반적인 구조를 설립함으로써 부분적인 해결
	UC8		전반적인 구조를 설립함으로써 부분적인 해결
	UC9		전반적인 구조를 설립함으로써 부분적인 해결
QA1			독립적인 아키텍처 구조를 설립함으로써 접근성에 대한 문제 해결
QA2			독립적인 아키텍처 구조를 설립함으로써 가용성에 대한 문제 해결

7.Kanban Board

Completely	Partially	Not	Design Decisions
QA3			각각 독립성을 가진 시스템을 사용하여 서버 지연시간에 의한 응답속도를 해결
QA4			메인 서버를 갖는 구조를 통해 업데이트 용이성 확보
	QA5		전반적인 구조를 설립함으로써 부분적인 해결
QA6			각각 독립성을 가진 자판기 및 냉장고가 메인 서버에 의해 관리됨으로 해결
		CRN1	
		CRN2	
CRN3			디자인 컨셉에서 해결
CON1			각각의 구성 요소를 리치 인터넷 어플리케이션 구조로 통일하여 해결
CON2			독립적인 아키텍처 구조를 설립함으로써 네트워크 불안정에 대한 문제 해결
CON3			독립적인 아키텍처 구조를 설립함으로써 가용성에 대한 문제 해결
CON4			냉장고가 시스템에 추가됨으로 해결

2

Architecture

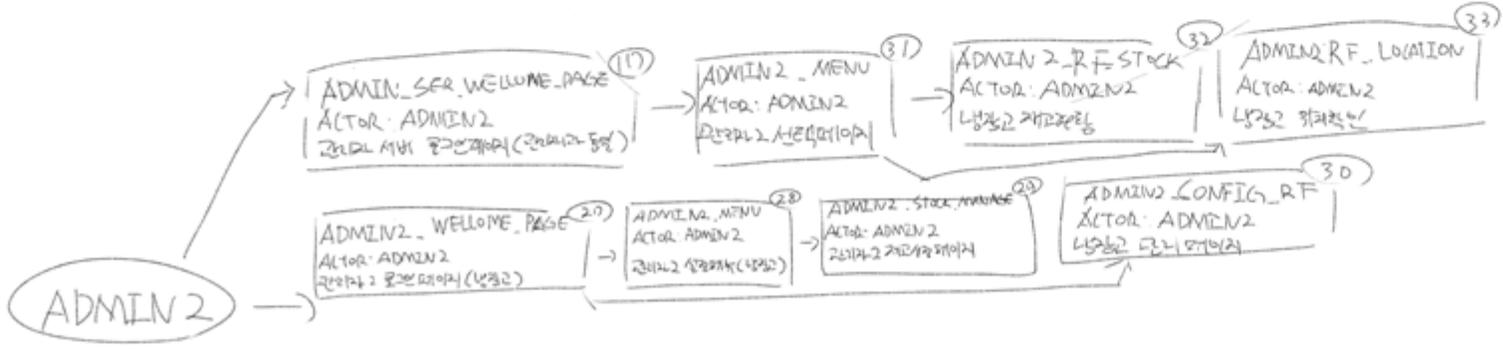
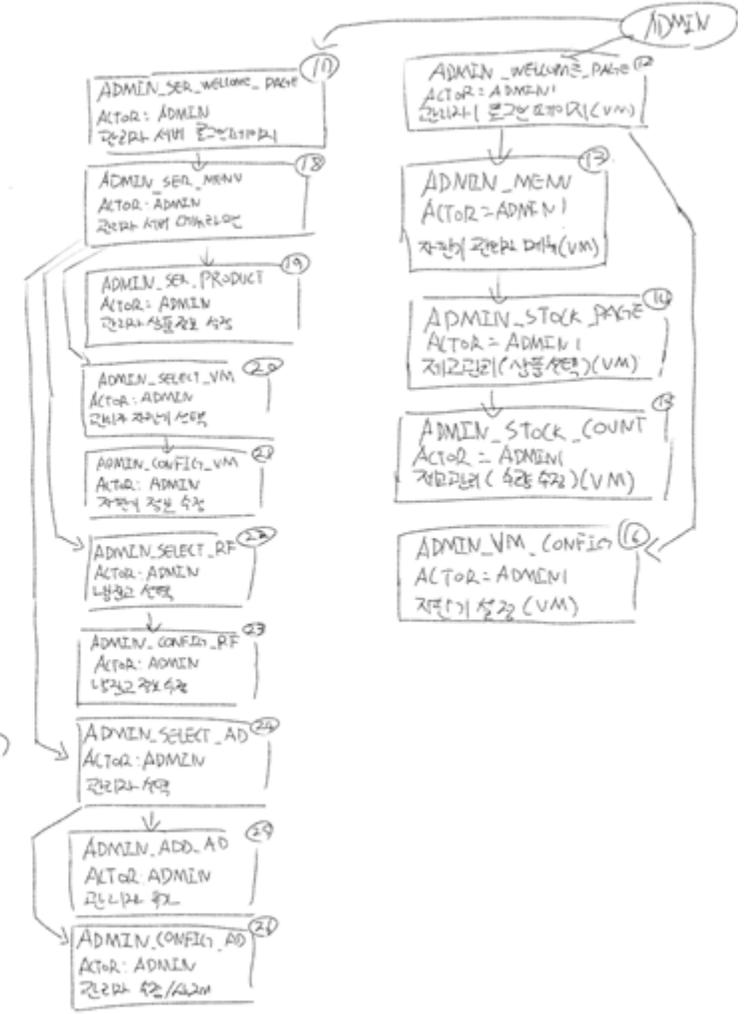


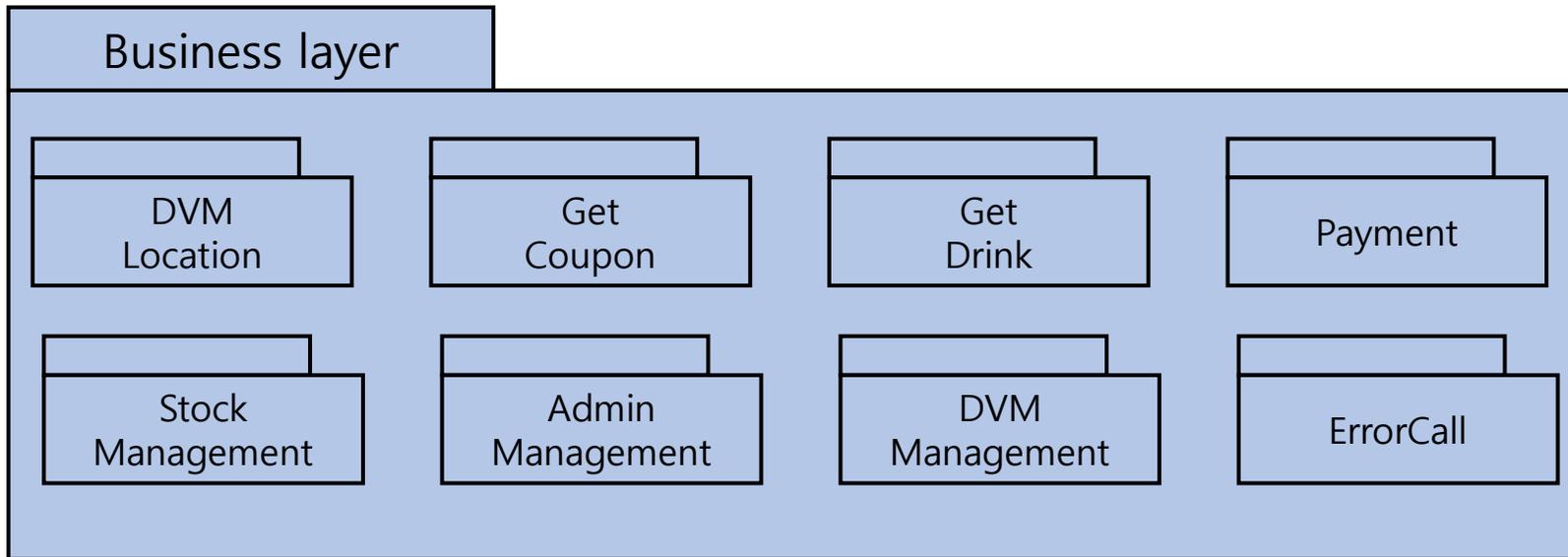
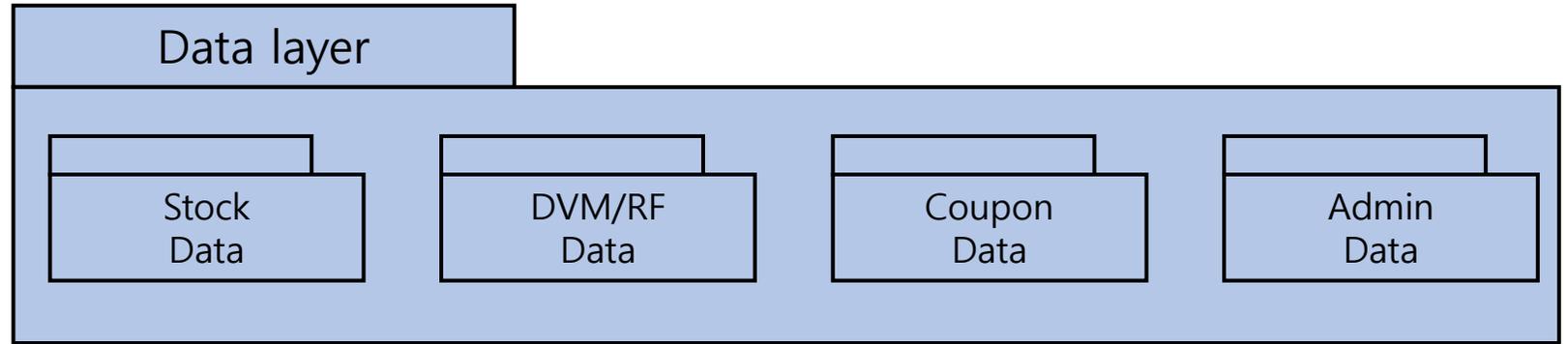
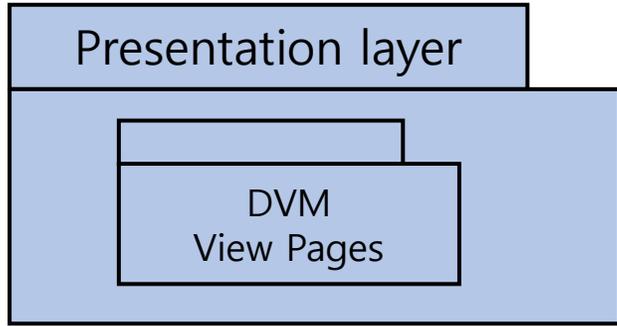
34
 SYS_PULL_DATA
 ACTOR: SYSTEM
 VM/RF데이터 메인서버에서

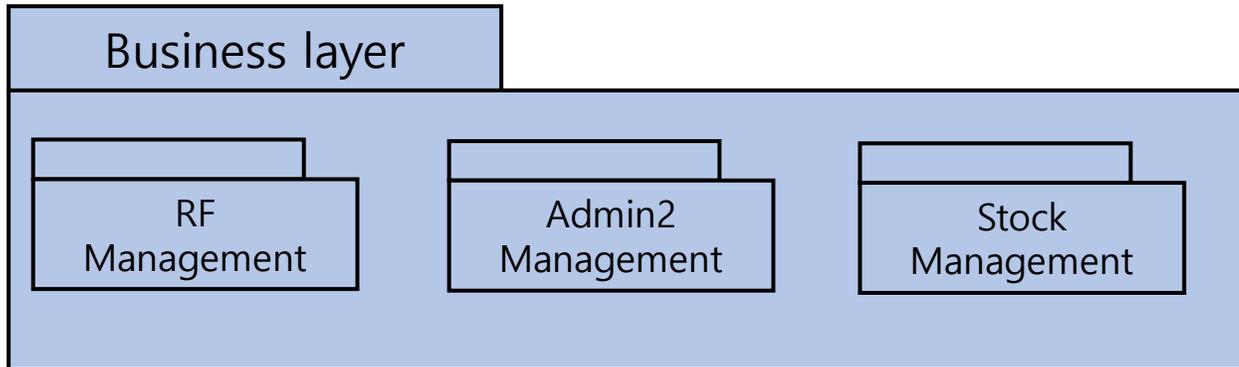
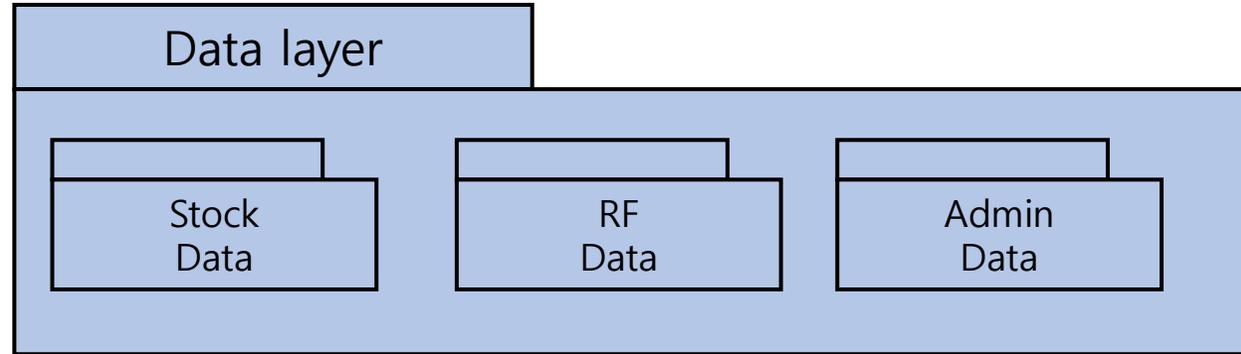
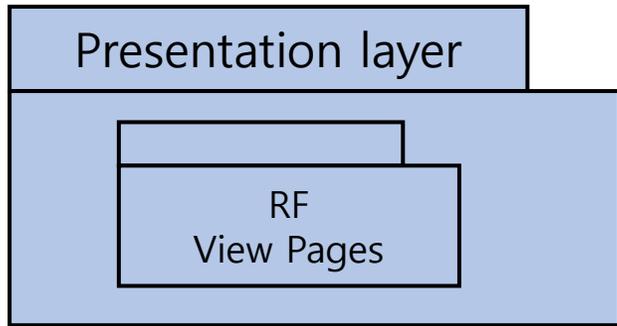
31
 SYS_COUPON
 ACTOR: SYSTEM
 쿠폰 생성, 사용

35
 SYS_PUSH_DATA
 ACTOR: SYSTEM
 VM/RF데이터 메인서버에 푸시

38
 SYS_SEND_SMS
 ACTOR: SYSTEM
 오후 야간 근무자에게 전송







Presentation layer

Server
View Pages

Business layer

Server
Controller

Check
Admin

Check
DVM

Check
RF

Check
Location

ErrorCall

Data layer

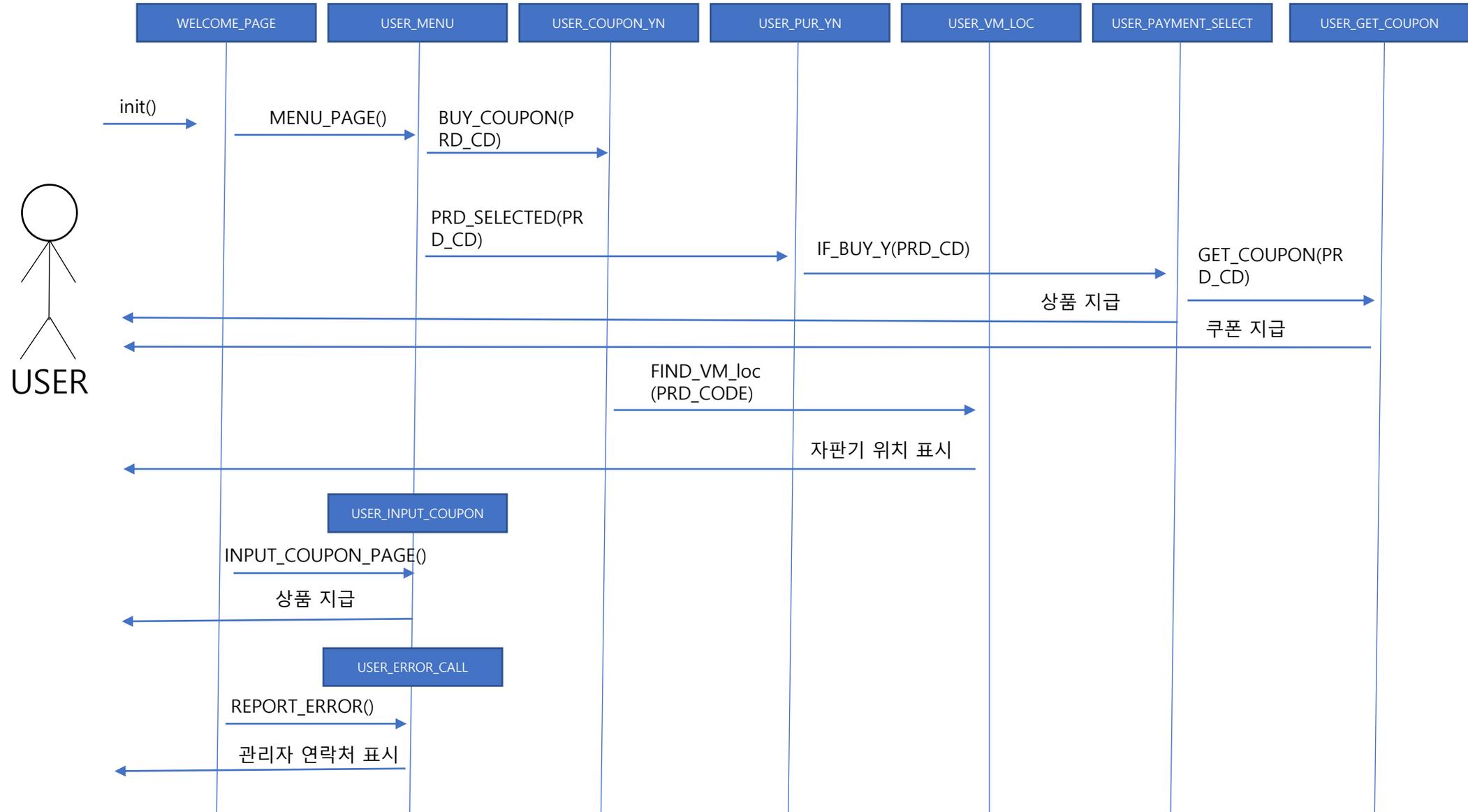
Stock
Data

DVM/RF
Data

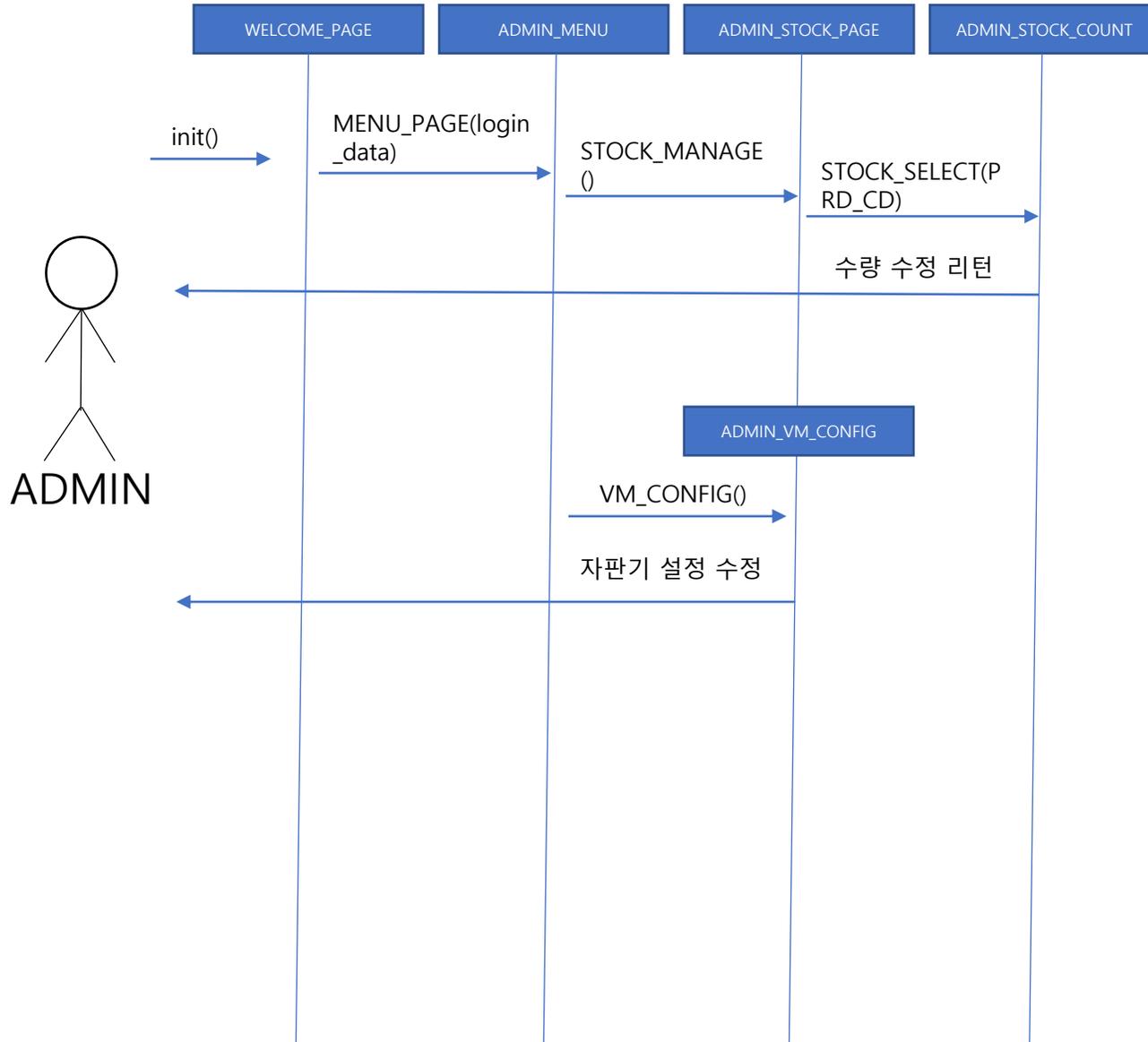
Coupon
Data

Admin
Data

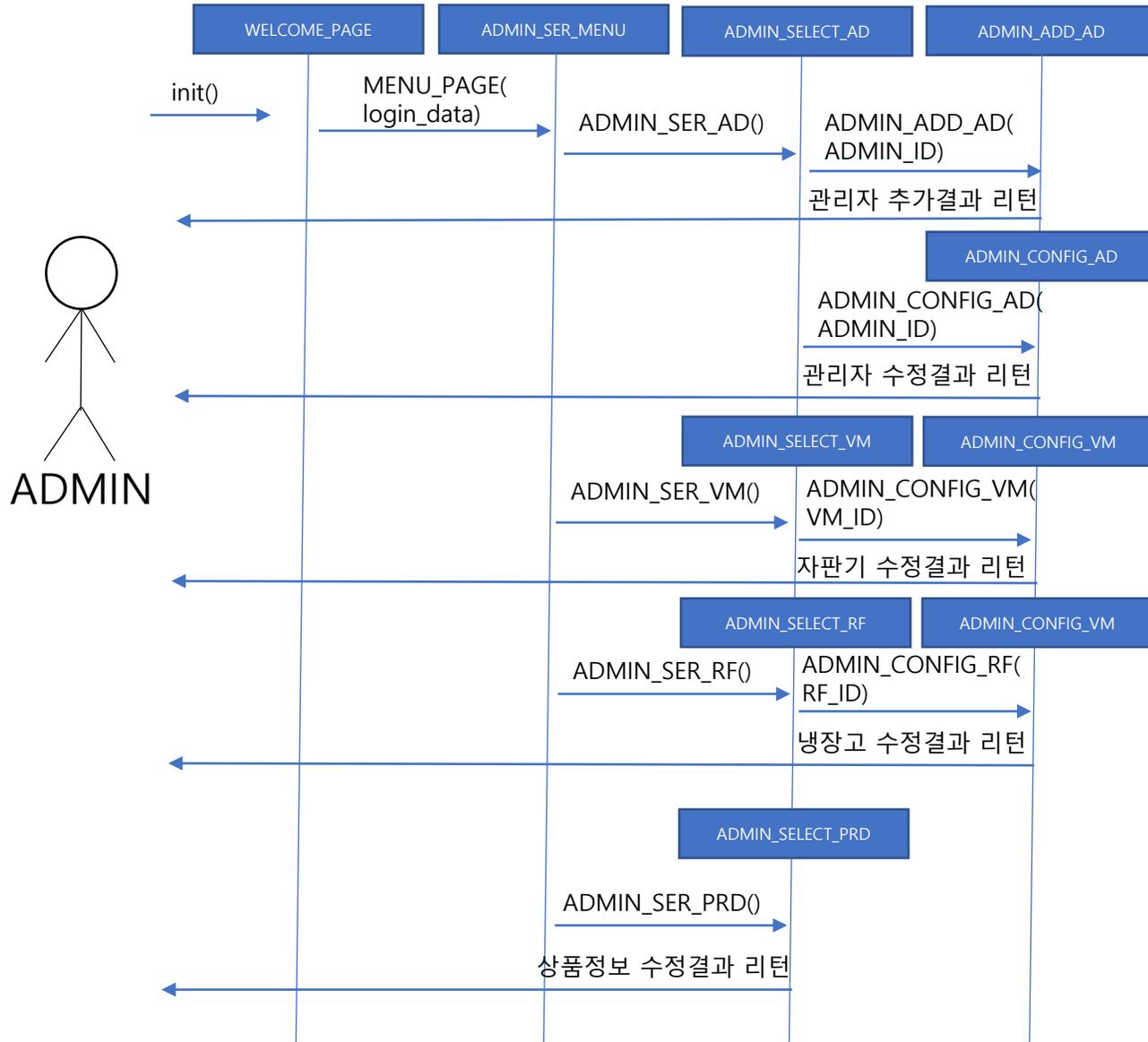
Sequence Diagram(USER USE CASE)



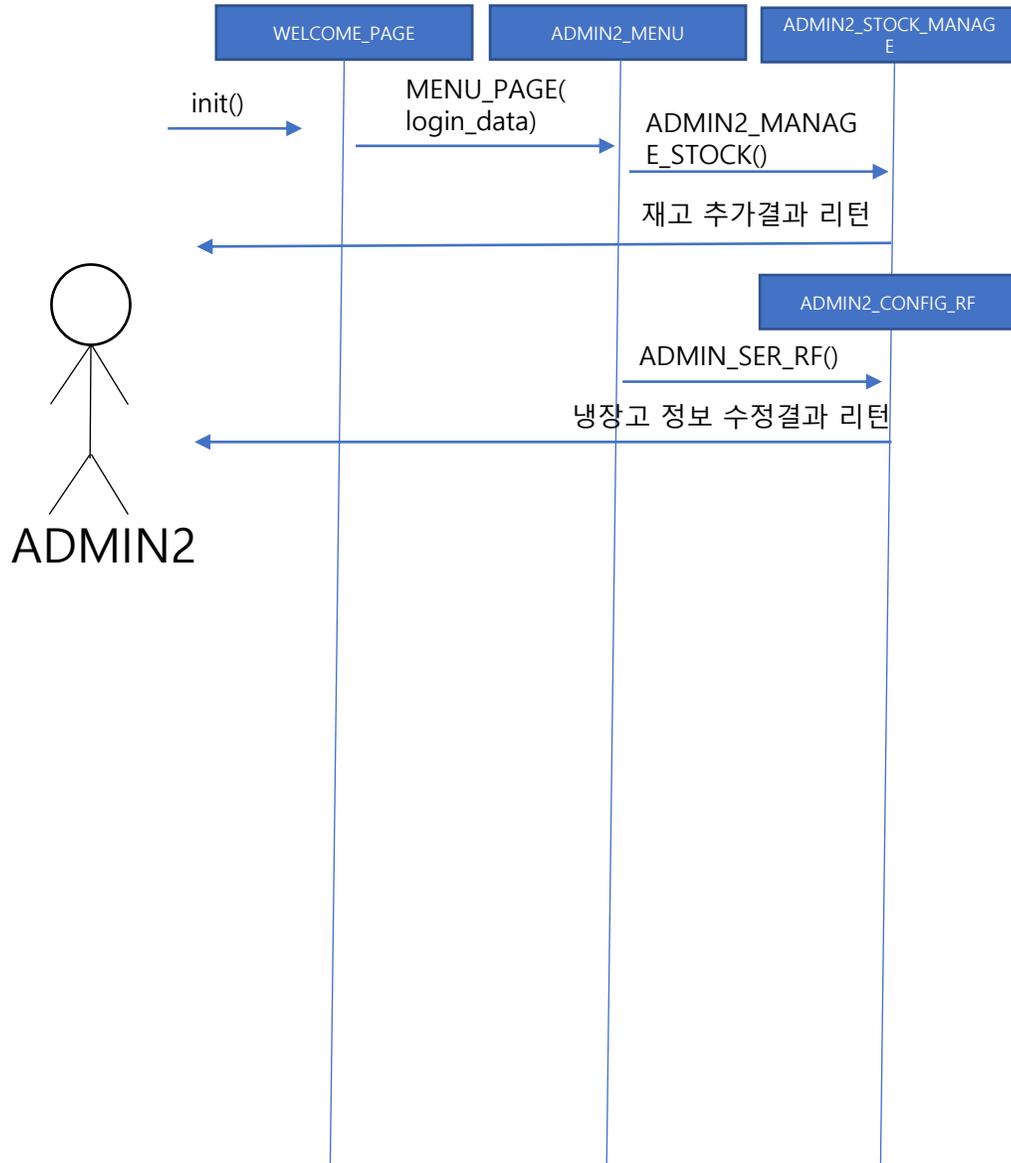
Sequence Diagram(ADMIN VM USE CASE)



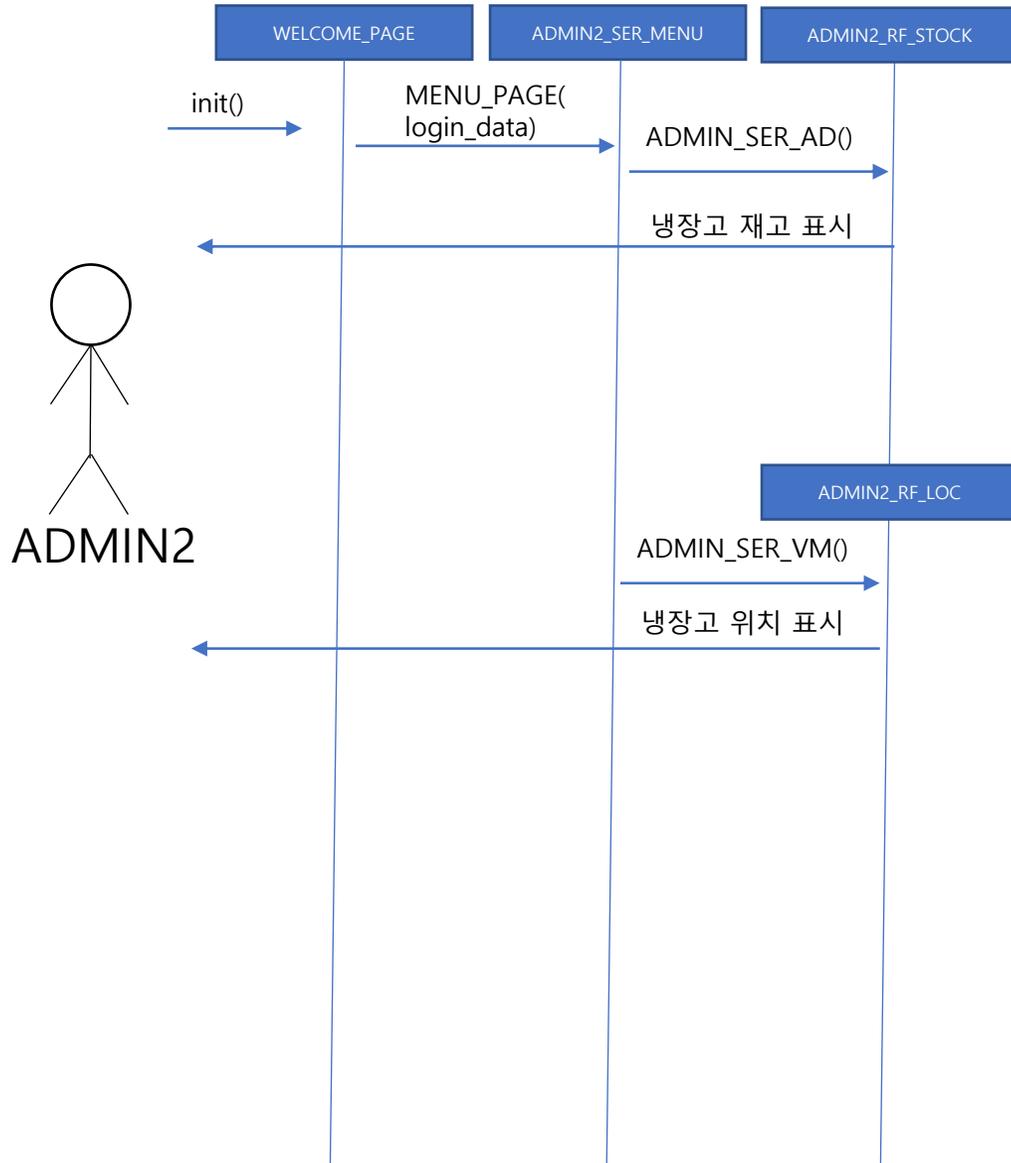
Sequence Diagram(ADMIN SERVER USE CASE)



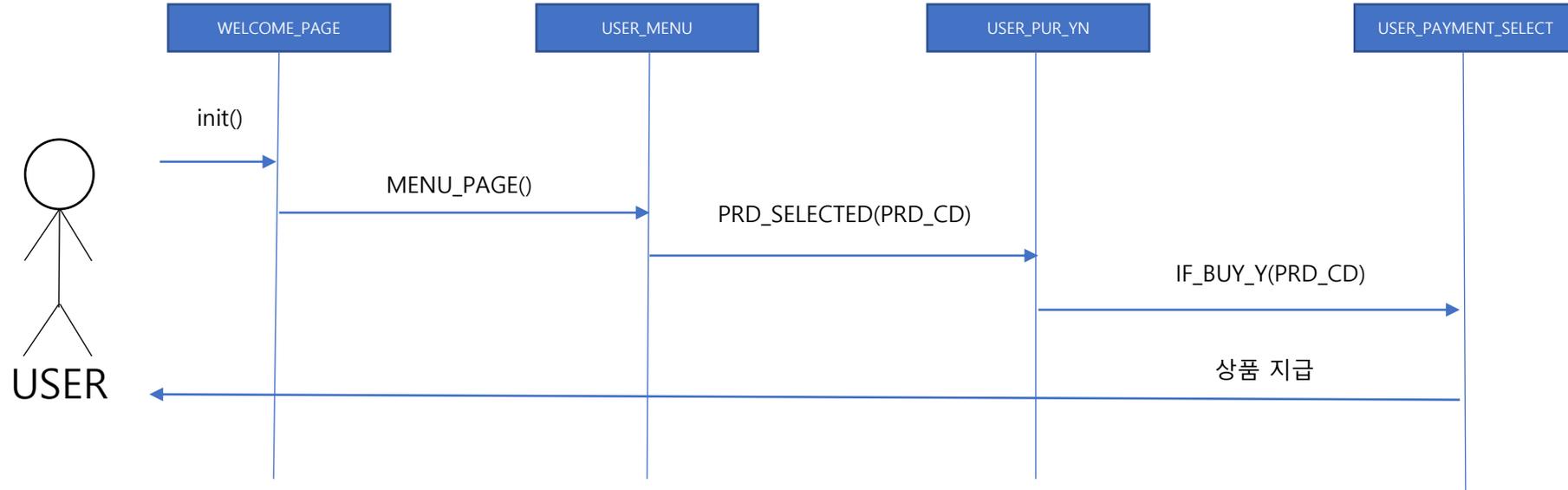
Sequence Diagram(ADMIN2 RF USE CASE)



Sequence Diagram(ADMIN2 SERVER USE CASE)



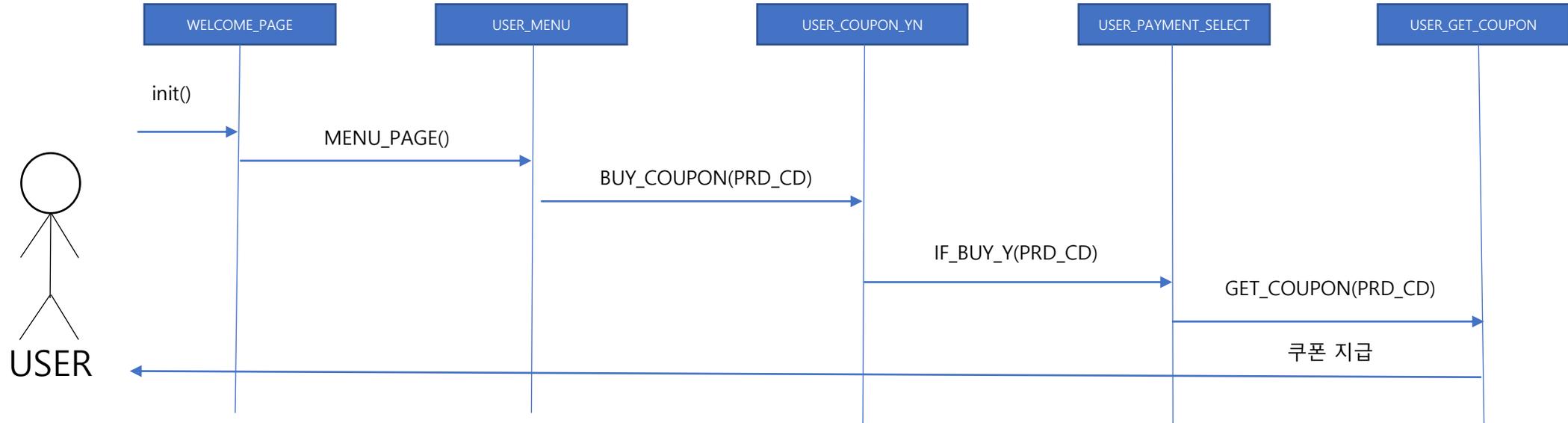
Sequence Diagram(UC1)



UC1	사용자가 재고가 있는 상품을 선택하여 구매하면 상품이 지급되어야 한다.
------------	---

Method Name	Description
init()	시스템 준비
MENU_PAGE	아무 입력시 초기 판촉화면에서 선택페이지로 이동
PRD_SELECTED(PRD_CD)	선택한 품목 코드(PRD_CD) 를 파라미터로 결제 선택 페이지로 이동
IF_BUY_Y(PRD_CD)	구매 결정시 선택한 품목 코드(PRD_CD)를 파라미터로 결제 페이지로 이동

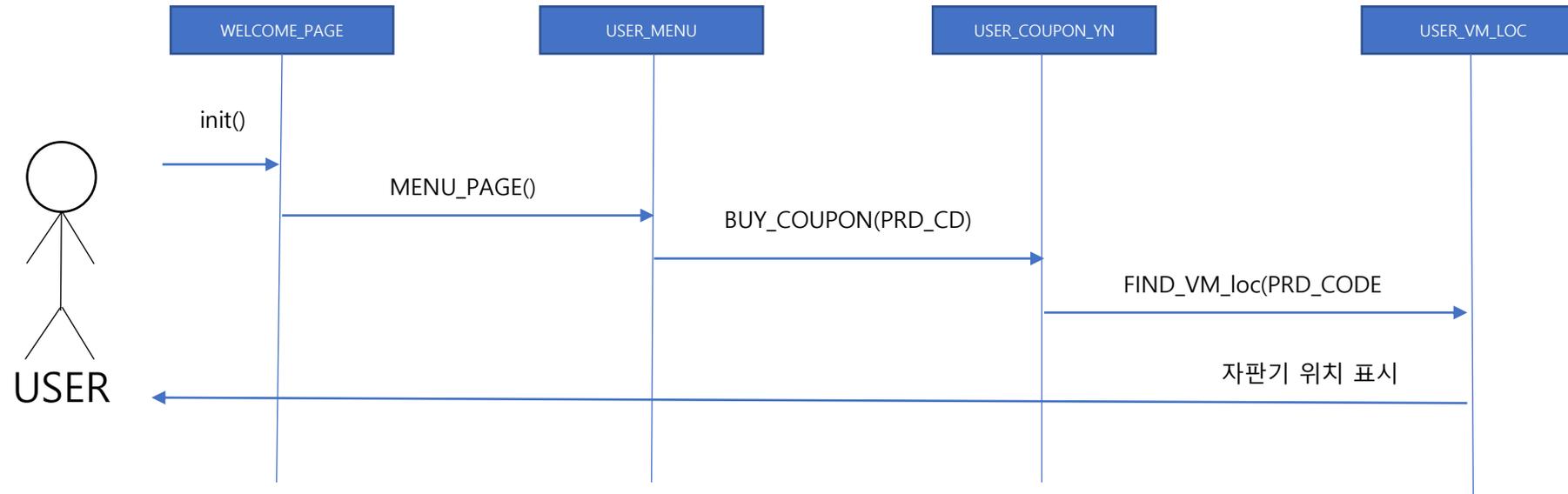
Sequence Diagram(UC2)



UC2 사용자가 재고가 없는 상품을 선택할 경우 쿠폰 구매 의사를 물어본 후 결제를 진행한다.

Method Name	Description
init()	시스템 준비
MENU_PAGE	아무 입력시 초기 판촉화면에서 선택페이지로 이동
BUY_COUPON(PRD_CD)	선택한 품목 코드(PRD_CD) 를 파라미터로 쿠폰 구매 의사
IF_BUY_Y(PRD_CD)	쿠폰 구매 결정시 선택한 품목 코드(PRD_CD)를 파라미터로 결제 페이지로 이동
GET_COUPON(PRD_CD)	메인 서버에 쿠폰 발급 요청

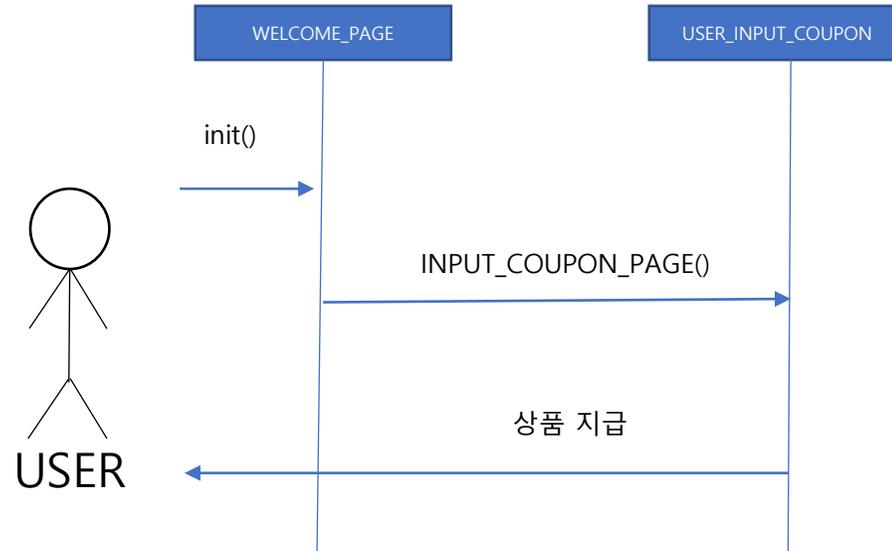
Sequence Diagram(UC3)



UC3	사용자가 재고가 없는 상품을 선택하고 결제를 진행하지 않았을 경우 해당 상품이 있는 자판기 위치를 안내한다.
------------	--

Method Name	Description
init()	시스템 준비
MENU_PAGE	아무 입력시 초기 판촉화면에서 선택페이지로 이동
BUY_COUPON(PRD_CD)	선택한 품목 코드(PRD_CD) 를 파라미터로 쿠폰 구매 의사
FIND_VM_loc	쿠폰 구매 거부 시 자판기 위치 알림

Sequence Diagram(UC4)



UC4	사용자가 쿠폰을 가지고 있는 경우 쿠폰으로 상품을 지급 받는다.
------------	-------------------------------------

Method Name	Description
init()	시스템 준비
INPUT_COUPON_PAGE()	쿠폰 번호 입력 시 상품 지급

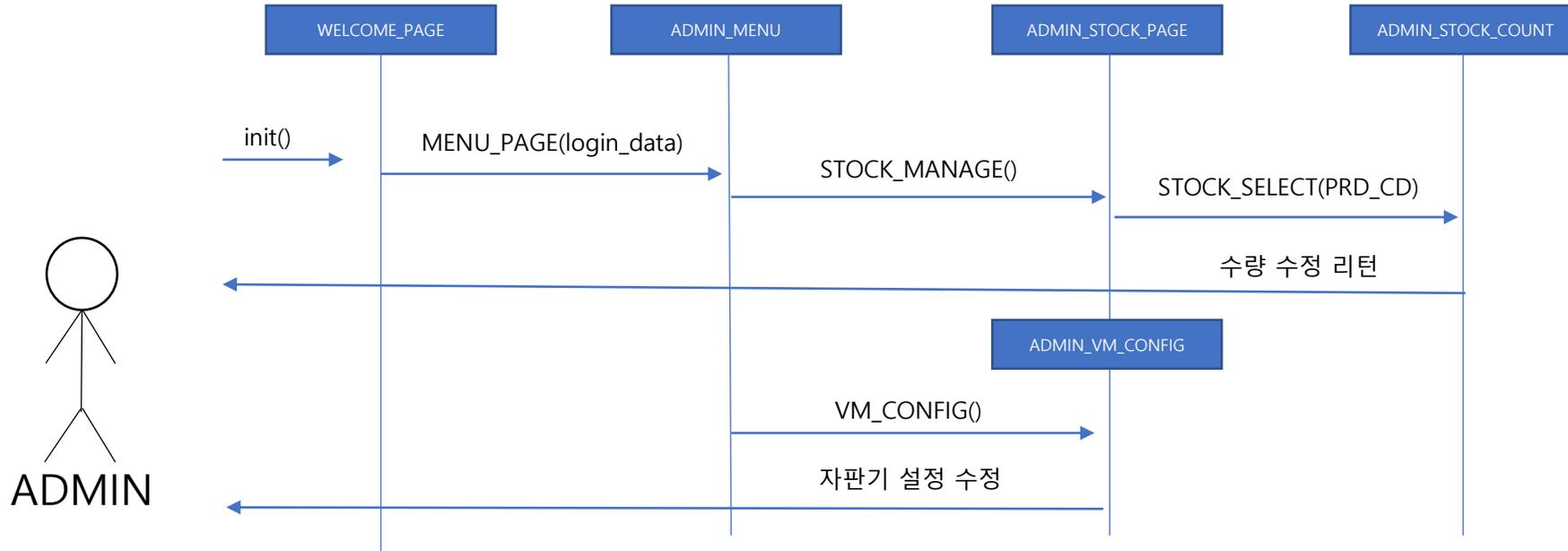
Sequence Diagram(UC5)



UC5	자판기에 오류가 발생한 경우 사용자는 관리자에게 연락을 취할 수 있다.
------------	---

Method Name	Description
init()	시스템 준비
REPORT_ERROR()	에러 보고 시 관리자 정보 반환

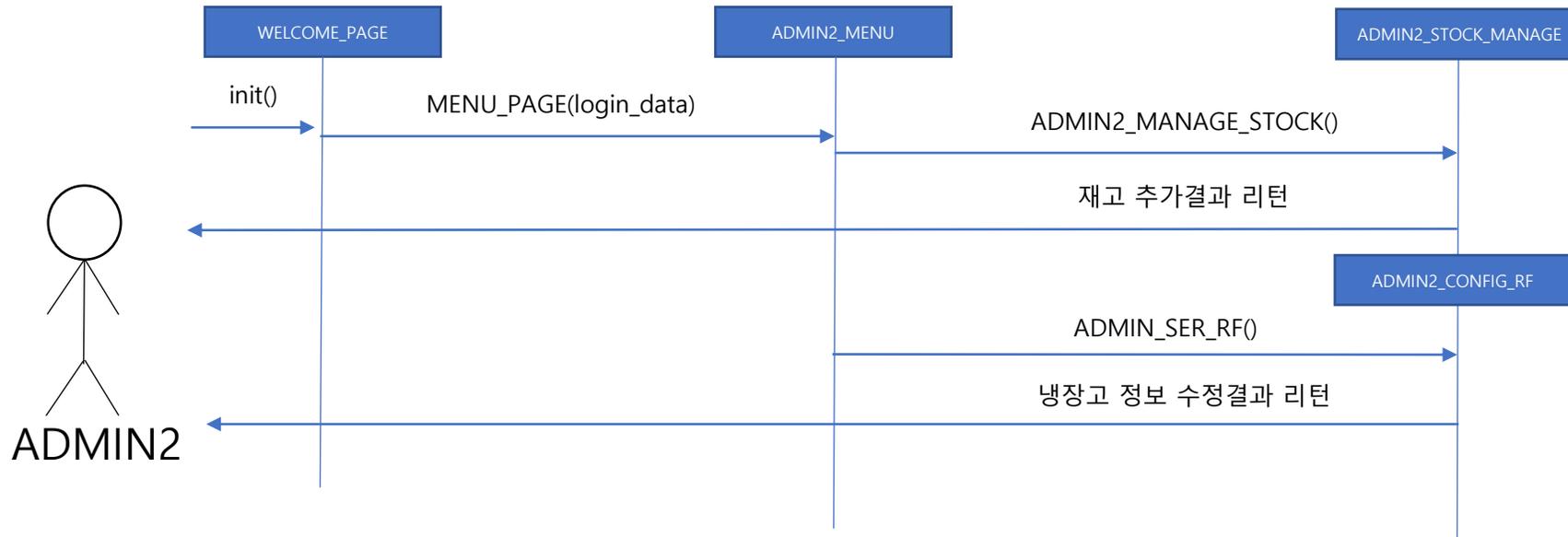
Sequence Diagram(UC6)



UC6	자판기 관리자는 자판기 내의 정보(재고, 자판기 설정, 냉장고 정보)를 관리한다.
------------	---

Method Name	Description
STOCK_MANAGE()	상품 재고 관리
STOCK_SELECT(PRD_CD)	상품 수량 수정
VM_CONFIG()	자판기 설정

Sequence Diagram(UC7)

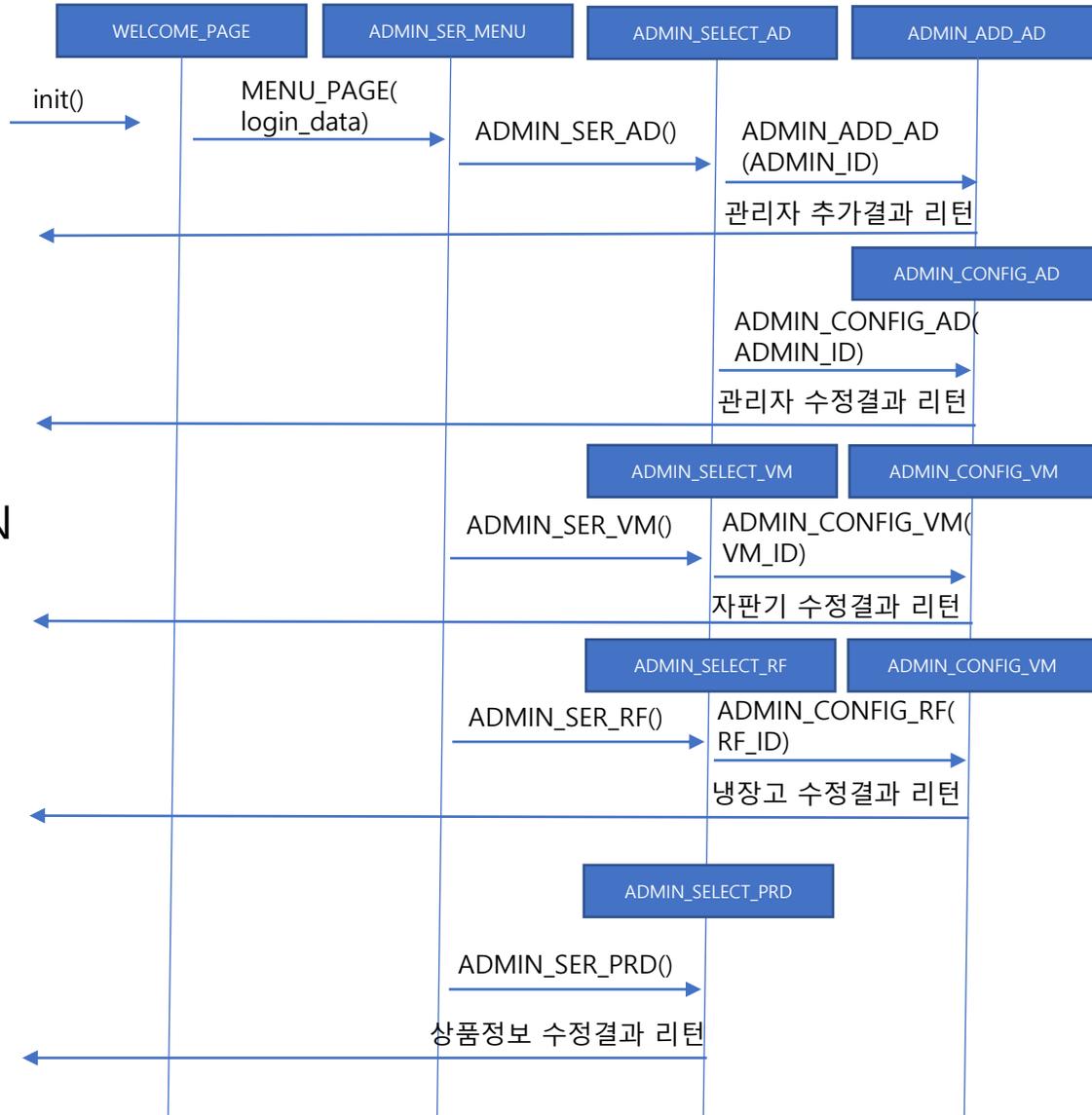
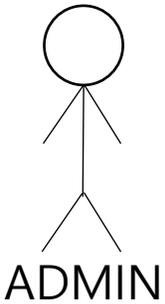


UC7

냉장고 관리자는 냉장고 내의 정보(재고, 냉장고 설정, 자판기 정보)를 관리한다.

Method Name	Description
ADMIN2_MANAGE_STOCK()	냉장고 재고 관리
ADMIN_SER_RF()	냉장고 정보 관리

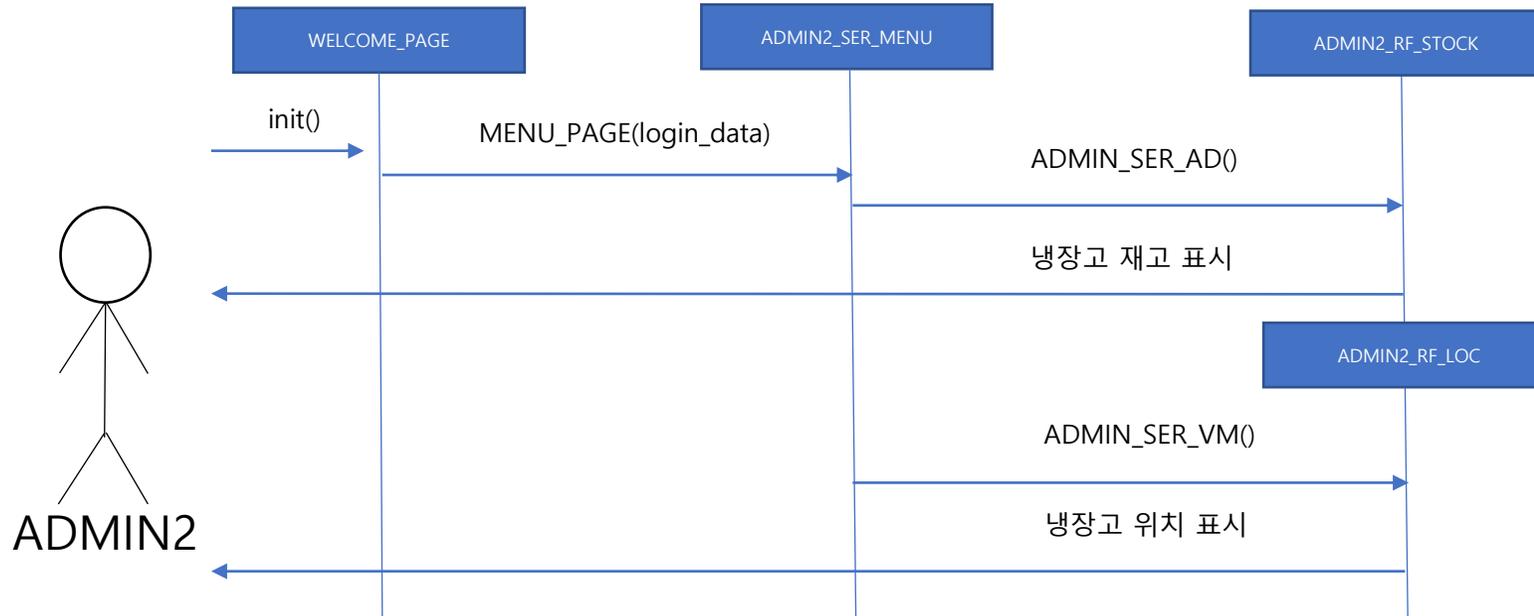
Sequence Diagram(UC8)



UC8	자판기 관리자는 메인 서버를 통해 관리자, 자판기, 냉장고, 상품 정보 등을 관리한다.
------------	--

Method Name	Description
ADMIN_SER_AD()	
ADMIN_ADD_AD(ADMIN_ID)	관리자 추가
ADMIN_CONFIG_AD(ADMIN_ID)	관리자 정보 수정
ADMIN_SER_VM()	
ADMIN_CONFIG_VM(VM_ID)	자판기 정보 수정
ADMIN_SER_RF()	
ADMIN_CONFIG_VM(RF_ID)	냉장고 정보 수정
ADMIN_SER_PRD()	상품 정보 수정

Sequence Diagram(UC9)



UC9 물류 관리자는 메인 서버를 통해 냉장고 위치와 재고를 확인할 수 있다.

Method Name	Description
MENU_PAGE(login_data)	관리자 접속
ADMIN_SER_AD()	냉장고 재고 표시
ADMIN_SER_VM()	냉장고 위치 표시

7.Kanban Board

Completely	Partially	Not	Design Decisions
UC1			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC2			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC3			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC4			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC5			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC6			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC7			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC8			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC9			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
QA1			독립적인 아키텍처 구조를 설립함으로써 접근성에 대한 문제 해결
QA2			독립적인 아키텍처 구조를 설립함으로써 가용성에 대한 문제 해결

7.Kanban Board

Completely	Partially	Not	Design Decisions
QA3			각각 독립성을 가진 시스템을 사용하여 서버 지연시간에 의한 응답속도를 해결
QA4			메인 서버를 갖는 구조를 통해 업데이트 용이성 확보
QA5			구성 요소들을 설립하면서 해결
QA6			각각 독립성을 가진 자판기 및 냉장고가 메인 서버에 의해 관리됨으로 해결
		CRN1	
		CRN2	
CRN3			디자인 컨셉에서 해결
CON1			각각의 구성 요소를 리치 인터넷 어플리케이션 구조로 통일하여 해결
CON2			독립적인 아키텍처 구조를 설립함으로써 네트워크 불안정에 대한 문제 해결
CON3			독립적인 아키텍처 구조를 설립함으로써 가용성에 대한 문제 해결
CON4			냉장고가 시스템에 추가됨으로 해결

3

1. Selected Architectural Drivers

- Iteration Goal 설정
 - 현재까지 만족되지 않은 Architectural Drivers

- | | |
|------|---|
| CRN1 | 라이선스 비용이 발생하지 않거나 저렴해야 한다 |
| CRN2 | 쿠폰 사용성을 위해 쿠폰 번호는 휴대폰 번호 입력시 휴대폰 등으로도 전송 되어야 한다 |

1. Externally Developed Components



목적	설명
언어	Java , JavaScript, XML
홈페이지	https://spring.io/
목적	
장점	1.Lisense = Apache 2.0 dwd w
제한점	

1. Externally Developed Components



목차	설명
언어	Java , JavaScript, XML
홈페이지	https://spring.io/
목적	
장점	1.Lisense = Apache 2.0 dwd w
제한점	

1. Externally Developed Components



목차	설명
언어	Cross platform(API)
홈페이지	https://message.gabia.com/api/
목적	SMS메시지 전송
장점	1.Lisense =기업/개인
제한점	

7.Kanban Board

Completely	Partially	Not	Design Decisions
UC1			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC2			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC3			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC4			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC5			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC6			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC7			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC8			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
UC9			유스케이스를 지원하기 위한 시스템 구성요소를 모두 생성함
QA1			독립적인 아키텍처 구조를 설립함으로써 접근성에 대한 문제 해결
QA2			독립적인 아키텍처 구조를 설립함으로써 가용성에 대한 문제 해결

7.Kanban Board

Completely	Partially	Not	Design Decisions
QA3			각각 독립성을 가진 시스템을 사용하여 서버 지연시간에 의한 응답속도를 해결
QA4			메인 서버를 갖는 구조를 통해 업데이트 용이성 확보
QA5			구성 요소들을 설립하면서 해결
QA6			각각 독립성을 가진 자판기 및 냉장고가 메인 서버에 의해 관리됨으로 해결
CRN1			로열티 없는 Externally Developed Components를 채택하여 해결
CRN2			문자 전송에 사용할 Externally Developed Components를 채택하여 해결
CRN3			디자인 컨셉에서 해결
CON1			각각의 구성 요소를 리치 인터넷 어플리케이션 구조로 통일하여 해결
CON2			독립적인 아키텍처 구조를 설립함으로써 네트워크 불안정에 대한 문제 해결
CON3			독립적인 아키텍처 구조를 설립함으로써 가용성에 대한 문제 해결
CON4			냉장고가 시스템에 추가됨으로 해결