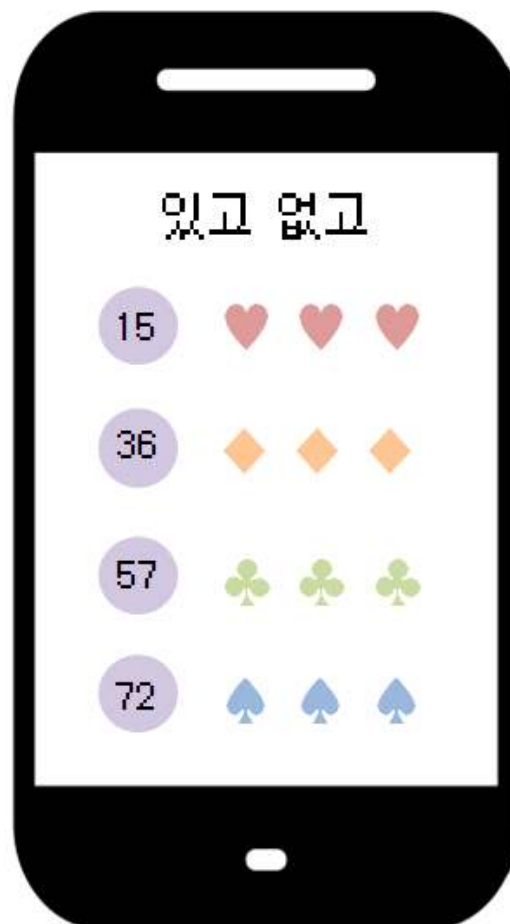


### 3. Design



재고 관리 및 상품 구매, 판매 시스템 [ 있고 없고 ]

Student No.	
Name	
E-mail	

## [ Revision history ]

Revision date	Version #	Description	Author
2018.06.01	1.0.0	First Draft	

## = Contents =

1. Introduction .....	4
2. Class Diagram .....	5
3. Sequence Diagram .....	9
4. State machine Diagram .....	17
5. Implementation Requirements .....	18
6. Glossary .....	18
7. References .....	18

## 1. Introduction

### 1) Summary

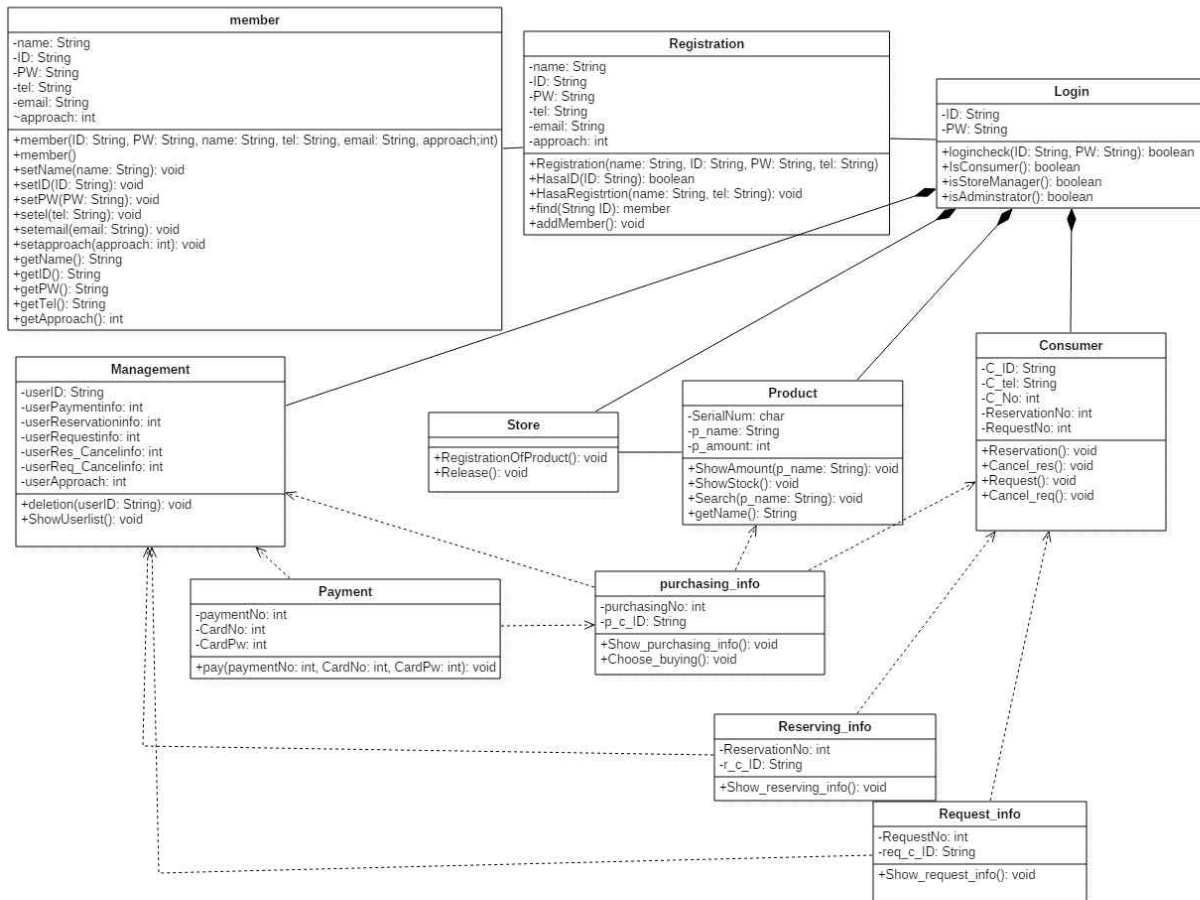
우리는 필요한 물품이 생기면 상점에 구매하러 간다. 하지만 종종 물건을 구매하기 위해 상점을 가면 원하는 상품이 없어 곤란을 겪는 경우가 있다. 일반적으로 상점의 재고 관리는 상점 점장과 पार्ट타이머들의 전유영역이다. 그렇기 때문에 고객들은 상점에 있는 상품들과 상품의 수량이 남아있는 지를 상점에 가지 않는 한 알지 못한다. 하지만 상품의 재고 사항은 소비자에게도 필요한 정보이다. 물건을 사러 상점을 가면 재고가 없거나 원하는 재품이 입고되지 않아 헛걸음을 한 경우가 종종 있을 것이다. 점장과 पार्ट타이머들의 전유영역의 일부를 고객에게 공개한다면 고객들은 헛걸음을 하는 빈도가 줄어들 것이다. 위의 불편함을 해결하기 위해 재고 관리하는 프로그램의 일부를 고객들에게 공개하고, 예약을 할 수 있는 시스템 '있고 없고'를 고안했다. 이를 통해서 고객들이 직접 상점에 가지 않아도 입고된 상품과 상품의 재고사항을 알 수 있을 것이며, 상점에서도 따로 고객을 위한 페이지나 어플리케이션을 만들지 않아도 한 시스템에서 정보를 제공할 수 있을 것이다.

이 문서는 '있고 없고'를 개발하기 위한 세 번째 단계인 Design에 대한 내용으로, 실제 구현에 대한 세부적인 사항을 명시하고, Diagram들을 통해 시스템의 전반적인 모습을 시각화, 순서화하여 보여준다.

### 2) Important Points of Design

- 사용자의 권한에 따라 사용 가능한 operation(method)을 구별해야한다.
- Class간의 정보 공유를 하되 외부로 노출이 되지 않도록 해야 한다.

## 2. Class Diagram



### 1) Registration

#### (1) Attributes

- name:String : 사용자 이름
- ID:String : 사용자 DI
- tel:String : 사용자 전화번호
- email:String : 사용자 E-Mail
- approach:int : 사용자에게 부여되는 권한

#### (2) Methods

- +Registration(name:String, ID:String, PW:String, tel:String) : 회원가입 생성자
- +HasaID(ID:String):boolean : 입력한 ID가 이미 있는 지 알려줌
- +HasaRegistration(name:String, tel:String):void : 이미 가입된 정보인지 알려줌.
- +addMember():void : user member 추가

## 2) Login

### (1) Attributes

- ID:String : 로그인에 필요한 ID
- PW:String : 로그인에 필요한 비밀번호

### (2) Methods

- +logincheck(ID:String, PW:String):boolean : ID와 비밀번호가 맞는지 확인
- +isConsumer():boolean : 로그인하는 ID가 소비자인지 판별
- +isStoreManager():boolean : 로그인하는 ID가 판매자인지 판별
- +isAdministrator():boolean : 로그인하는 ID가 관리자인지 판별

## 3) member

### (1) Attributes

- name:String : 사용자 이름
- ID:String : 사용자 ID
- tel:String : 사용자 전화번호
- email:String : 사용자 E-Mail
- approach:int : 사용자에게 부여되는 권한

### (2) Methods

- +member() : 생성자
- +member(in ID:String, in PW:String, in name:String, in tel:String, in email:String, in approach:int): 생성자
- setName(in name:String): void : name을 설정
- setID(in ID:String): void : ID를 설정
- setPW(in PW:String): void : 비밀번호 설정
- settel(in tel:String): void : 전화번호 설정
- setemail(in email:String): void : 이메일 설정
- setapproach(in approach:int): void : 권한 설정
- getName(): String : 이름 얻음
- getID(): String : 아이디 얻음
- getPW(): String : 비밀번호 얻음
- getTel(): String : 전화번호 얻음
- getApproach(): int : 접근 얻음

#### 4) Management

##### (1) Attributes

- userID:String : 사용자의 ID
- userPaymentinfo:int : 사용자가 결제한 상품 번호
- userReservationinfo:int : 사용자가 예약한 상품 번호
- userRequestinfo:int : 사용자가 요청한 상품 번호
- userRes\_Cancelinfo:int : 사용자가 예약 취소한 상품 번호
- userReq\_Cancelinfo:int : 사용자가 요청 취소한 상품 번호
- userApproach:int : 사용자의 권한(소비자, 판매자, 관리자)

##### (2) Methods

- +deletion(userID:String):void : 사용자 정보 삭제
- +ShowUserlist():void : 사용자 list를 보여줌.

#### 5) Store

##### (1) Methods

- +RegistrationOfProduct():void : 입고하는 메소드
- +Release():void : 오프라인에서 출고된 상품의 수량을 감소시키는 메소드

#### 6) Consumer

##### (1) Attributes

- C\_ID:String : 소비자 ID
- C\_tel:String : 소비자 전화번호
- C\_No:int : 소비자 일련번호
- ReservationNo:int : 예약 상품 번호
- RequestNo:int : 요청 상품 번호

##### (2) Methods

- +Reservation():Void : 상품 예약하기
- +Cancel\_res():void : 상품 예약 취소하기
- +Request():void : 상품 요청하기
- +Cancel\_req:void : 상품 요청 취소하기

#### 7) Product

##### (1) Attributes

- SerialNum:char : 상품의 일련번호
- p\_name:String : 상품의 이름

-p\_amount:int : 상품의 수량

(2) Methods

+ShowAmount(p\_name:String):void : 상품의 수량 보여주기

+ShowStock():void : 재고사항 보여주기

+Search(p\_name:String):void : 상품 검색하기

+getName(): String : 상품명 얻음

8) Payment

(1) Attributes

-paymentNo:int : 결제 상품 번호

-CardNo:int : 카드 번호

-CardPw:int : 카드 비밀번호

(2) Method

+pay(paymentNo:int,CardNo:int,CardPw:int):void : 결제하기

9) purchasing\_info

(1) Attributes

-purchasingNo:int : 결제 상품 번호

-p\_c\_ID:String : 결제하는 사람의 ID

(2) Methods

+Show\_purchasing\_info():void : 구매 내역 보여주기

+Choose\_buying():void : 구매 상품 선택하기

10) Reserving\_info

(1) Attributes

-ReservationNo:int : 예약 상품 번호

-r\_c\_ID:String : 예약한 사람의 ID

(2) Methods

+Show\_reserving\_info():void : 예약 내용 보여주기

11) Request\_info

(1) Attributes

-RequestNo:int : 요청 상품 번호

-req\_c\_ID:String : 요청한 사람의 ID

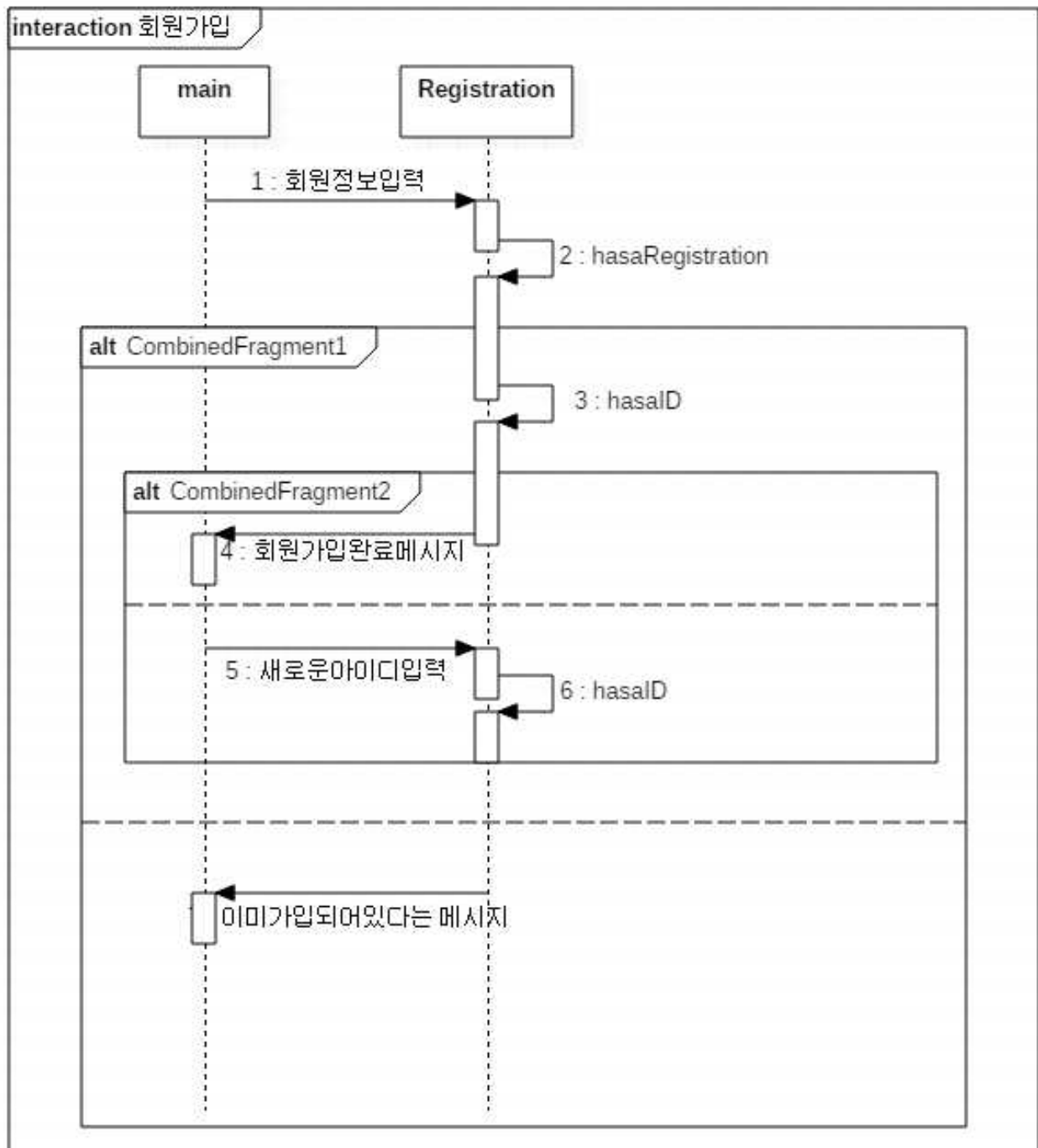
(2) Methods



+Show\_request\_info():void : 요청 내역 보여주기

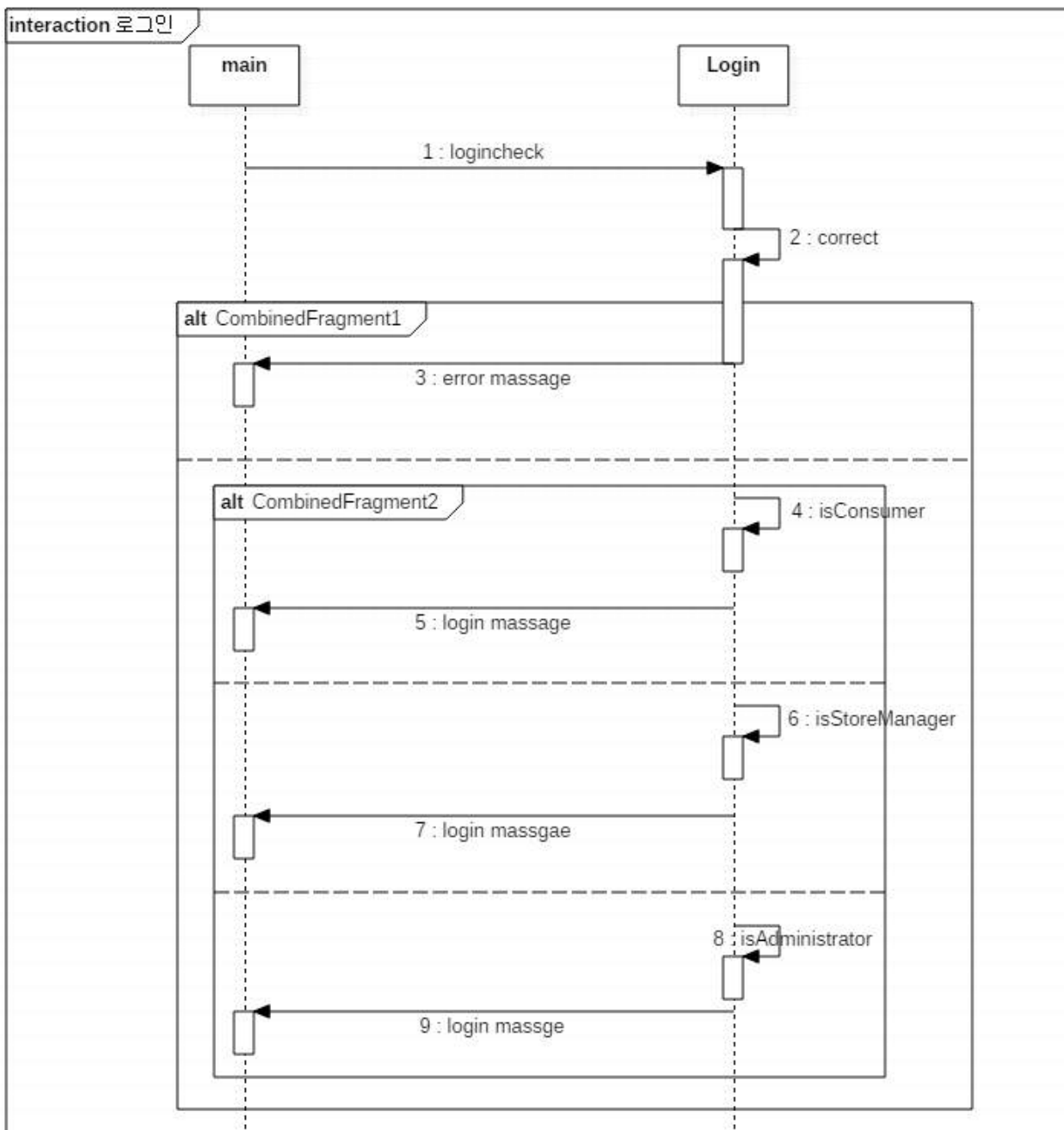
### 3. Sequence Diagram

#### 1) 회원가입



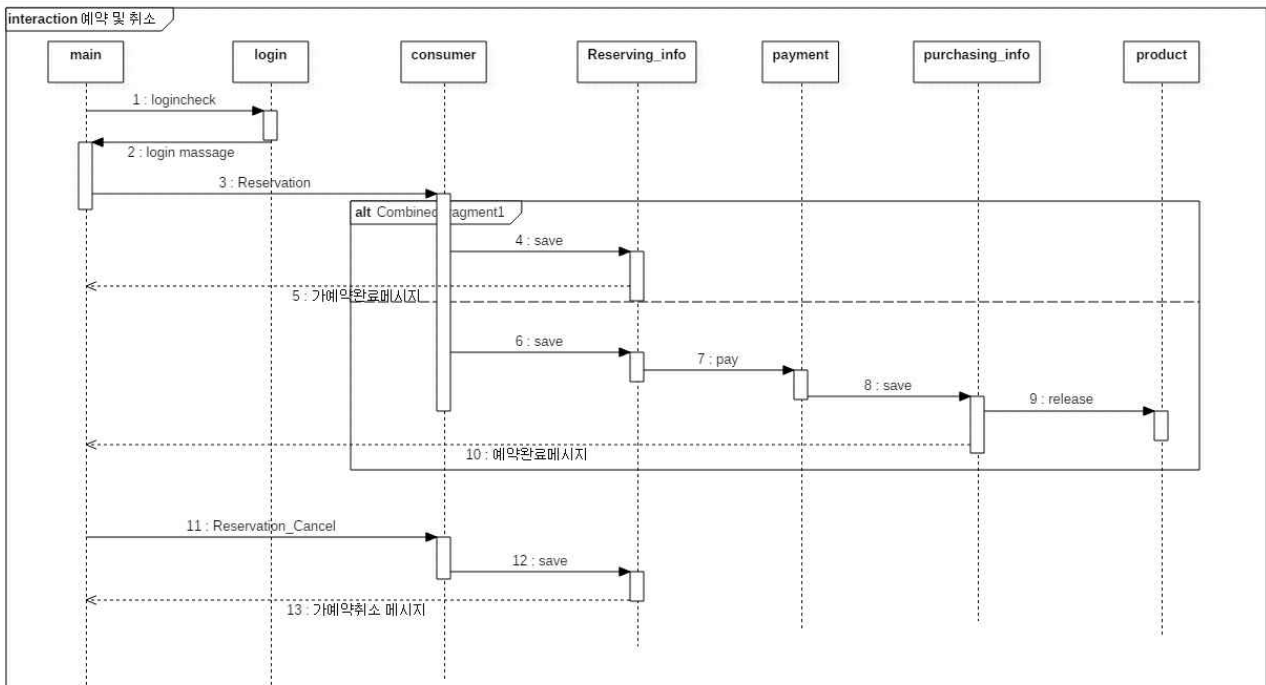
위의 그림은 시스템 실행 후 회원가입 기능을 수행할 때를 표현한 Sequence Diagram이다. main에서 회원정보입력을 선택하면 이미 회원가입이 되어있는지 확인한 후 가입이 되어있다면 이미 가입이 되어있다는 메시지를 출력한다. 그렇지 않으면 아이디를 입력 후 이미 가입된 아이디 있는 지 확인한 다음에 업로드하면 회원가입 완료 메시지를 보내고 있다면 새로운 아이디를 입력한다.

## 2) 로그인



위의 그림은 시스템 실행 후 로그인 기능을 수행할 때를 표현한 Sequence Diagram 이다. main에서 아이디, 비밀번호를 입력하면 logincheck를 통해 로그인이 제대로 되었는지 확인한다. 잘 못되었으면 error 메시지를 보내고, 확인되었으면 소비자인지 판매자인지, 관리자인지를 확인하여 return한다.

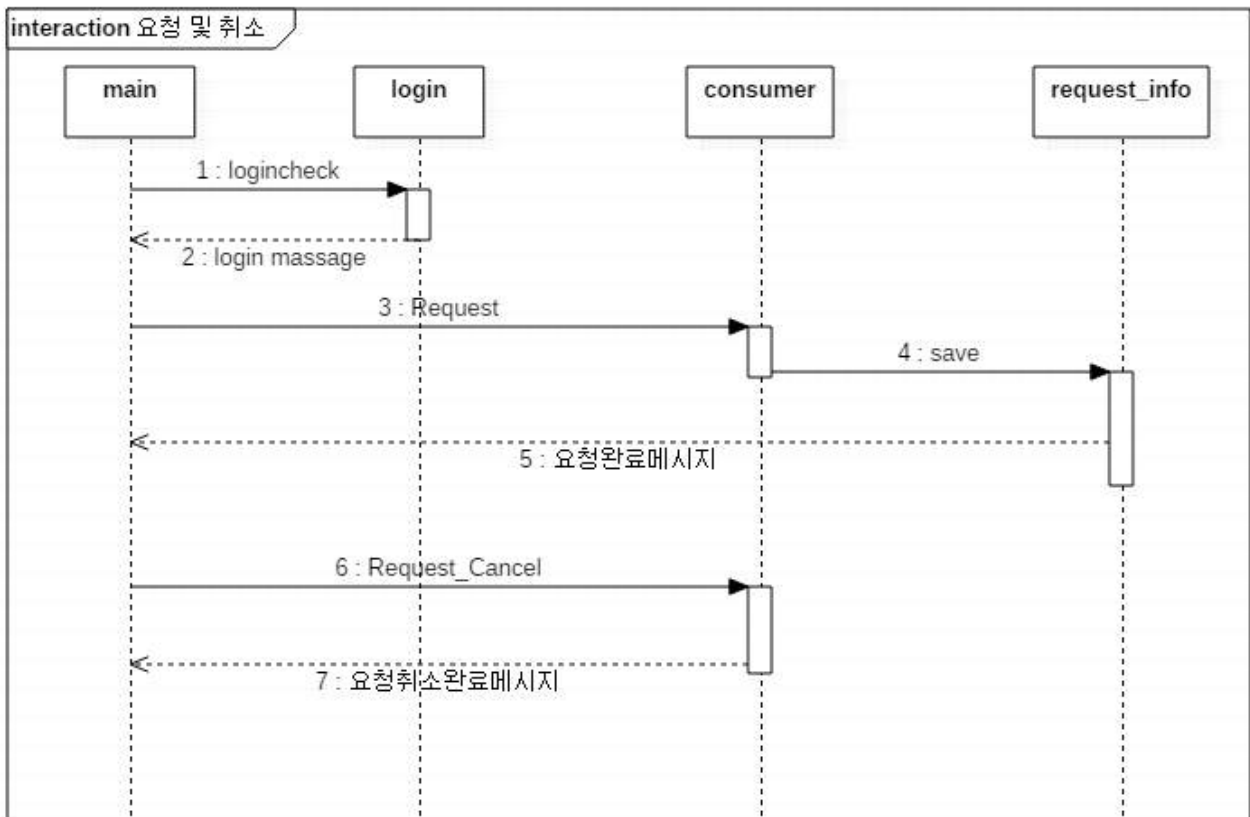
### 3) 예약 및 예약취소



위의 그림은 시스템 시행 후 예약 및 예약 취소 수행을 표현하는 Sequence diagram 이다. 로그인을 한 다음에 예약하기를 진행하면 예약한 사항이 저장이 된다. 만약 결제를 하지 않았다면 가예약 상태로 끝이 나고, 실제 수량 변동에 영향을 주지 않는다. 예약을 한 다음에 결제를 한다면 구매 내역에 저장이 되고 상품의 재고 수량은 감소되고, 예약완료 메시지가 출력된다.

예약을 취소할 때는 가예약 된 사항만 취소가 가능하기 때문에 가예약 사항을 취소한 뒤 예약 취소 내용을 저장하고 가예약 취소 메시지를 출력한다.

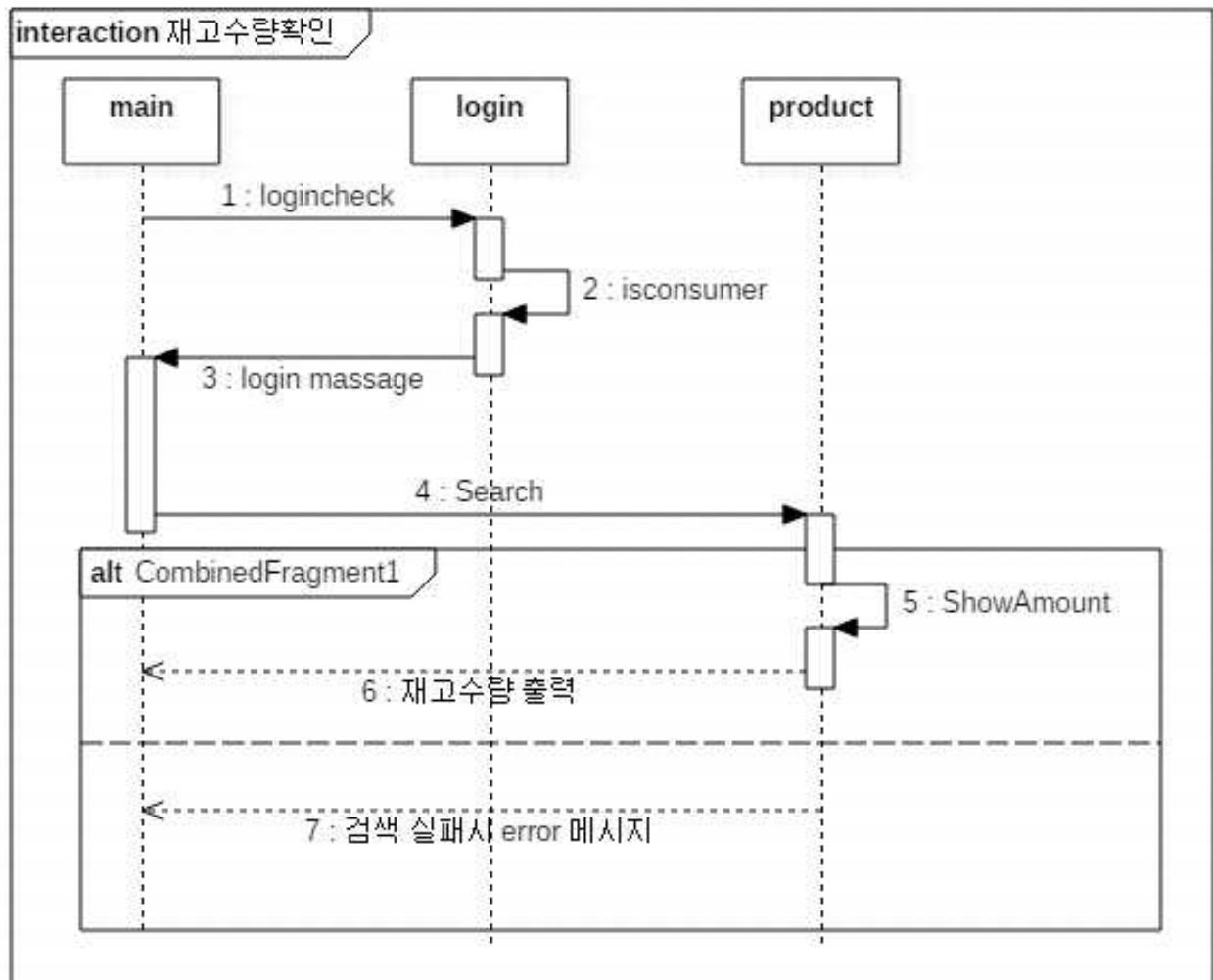
#### 4) 요청 및 요청취소



위의 그림은 시스템 시행 후 상품 요청 및 요청 취소 수행을 표현하는 Sequence diagram이다. 로그인이 성공적으로 끝난 다음에 Request를 실행하면 Request\_info에 요청 사항의 내용이 저장된다. 요청이 완료되면 요청 완료 메시지가 출력된다.

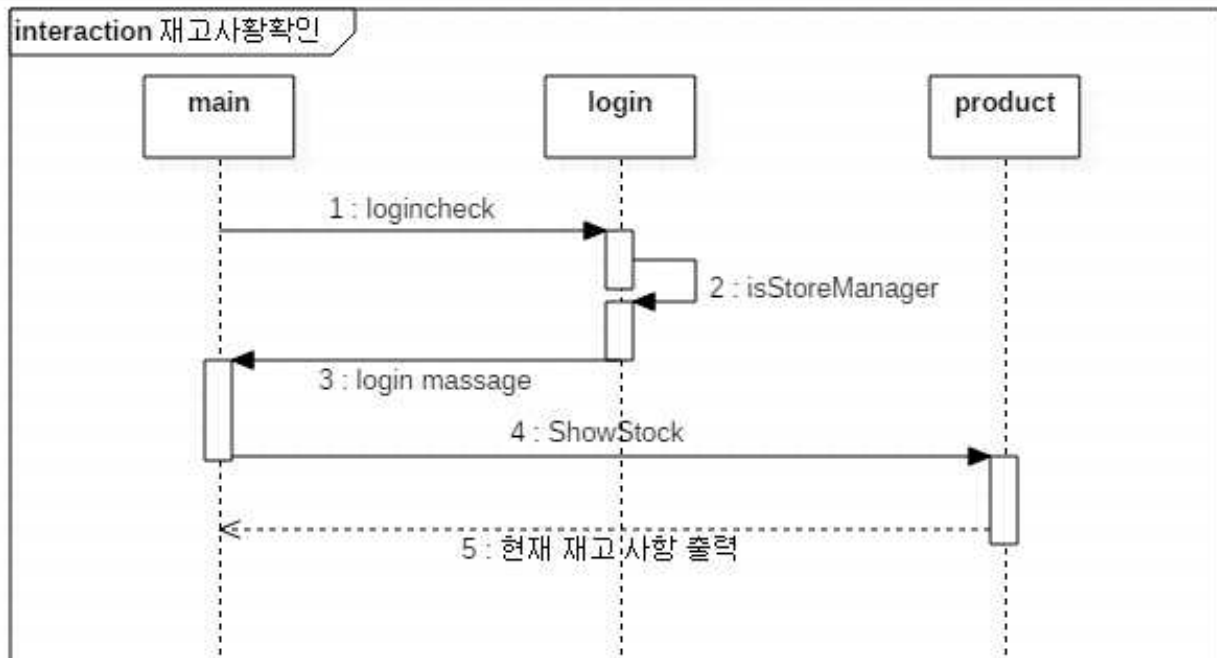
요청 취소는 login이 성공적으로 마친 후에 Request\_Cancel를 통해 요청 취소를 한다. 요청 취소가 되었다면 요청 취소 완료 메시지를 출력한다.

## 5) 재고수량확인(Consumer)



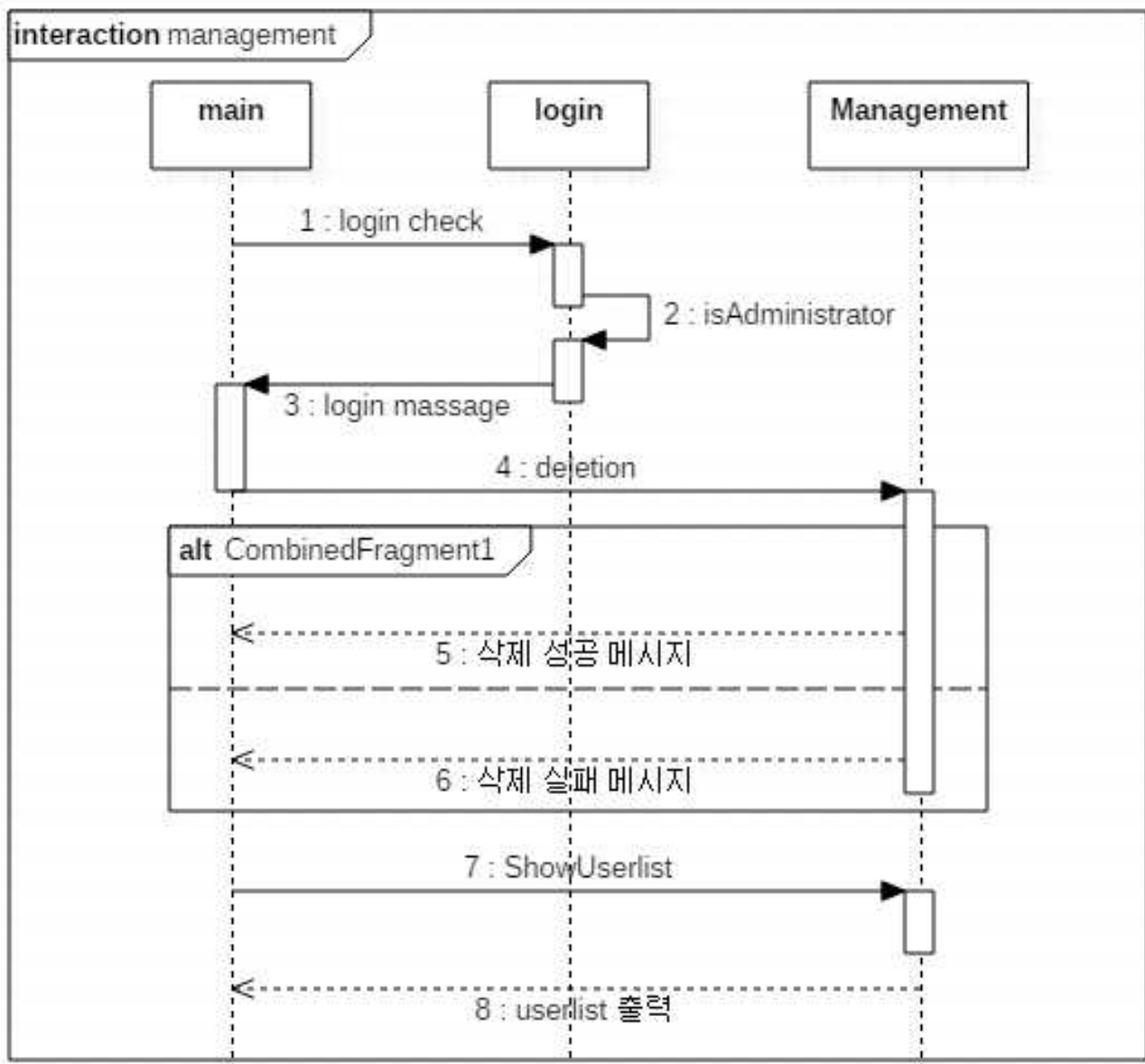
위의 그림은 시스템 시행 후 재고 수량 확인을 표현하는 Sequence diagram이다. 로그인 사용자가 소비자로서 성공적으로 수행되면 login message가 출력되고 Search로 수량이 궁금한 상품을 검색하면 상품이 존재할 시 재고 수량을 출력한다. 검색했을 때 상품이 존재하지 않거나 실패하면 error 메시지를 출력한다.

## 6) 재고사항확인(StoreManager)



위의 그림은 시스템 시행 후 재고사항확인을 표현하는 Sequence diagram이다. 판매자로 로그인이 성공적으로 수행하면 로그인 성공 메시지를 출력한다. ShowStock을 실행하며 현재 재고 사항을 출력한다.

## 7) Management

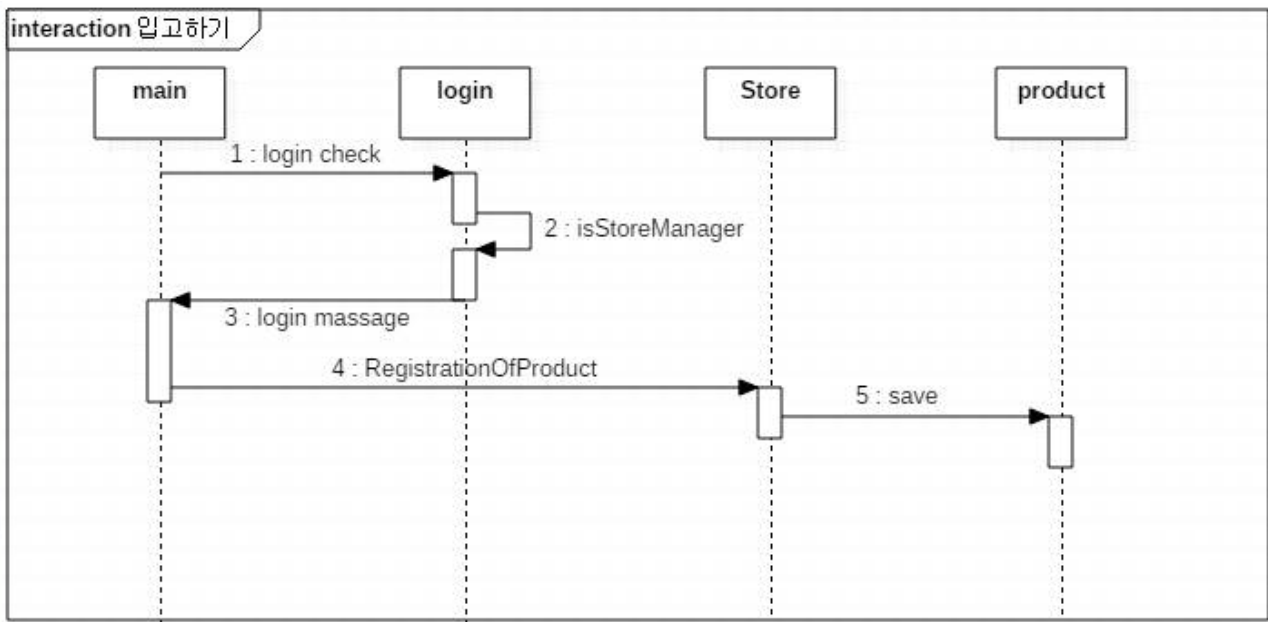


위의 그림은 시스템 시행 후 management 수행을 표현하는 Sequence diagram이다. main에서 login check를 통해 로그인을 진행하고, isAdministrator를 통해 관리자의 아이디로 로그인을 한다. 로그인이 성공되면 성공 메시지를 출력하고 main에서 deletion을 실행한다. 사용자 정보 삭제가 성공적으로 진행되었다면 성공 메시지를 출력하고, 실패시 error 메시지를 출력한다.

관리자가 로그인 했을 때 main에서 ShowUserlist을 실행하면 사용자 내역이 출력된다.

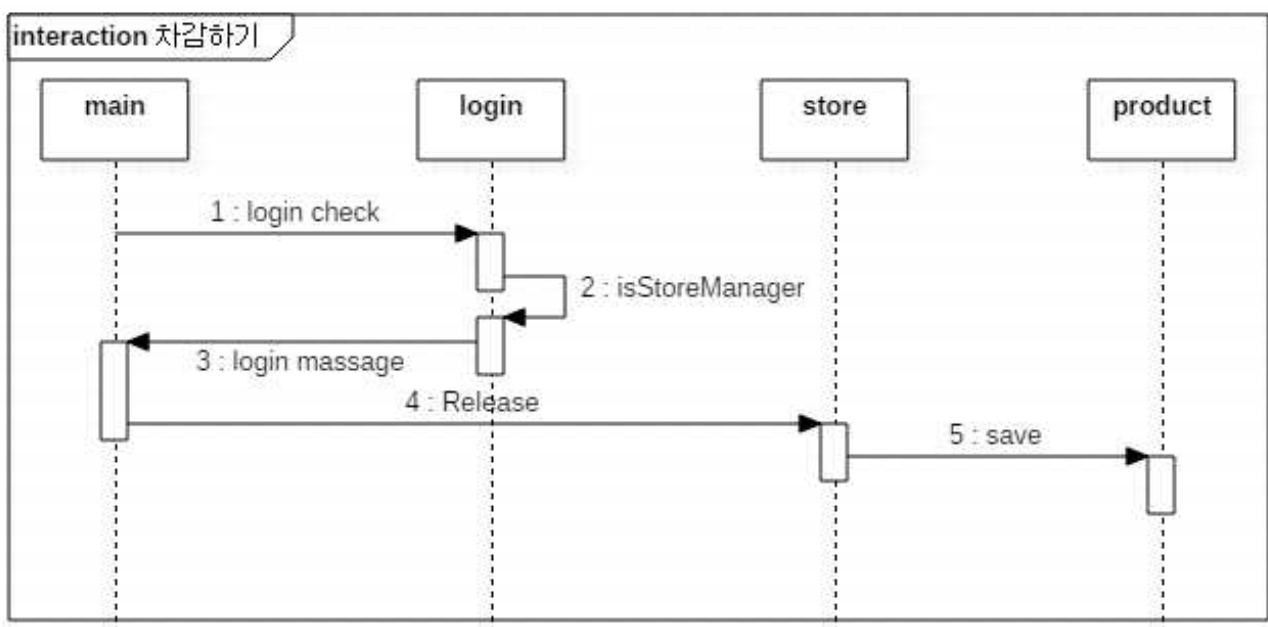


## 8) 상품입고



위의 그림은 시스템 시행 후 상품 입고 수행을 표현하는 Sequence diagram이다. main에서 login check를 시행한뒤 isStoreManager가 Ture이면 RegistrationOf Product를 실행한다. 상품 이름과 수량 등을 입력하고 product에 저장한다.

## 9) 오프라인에서 출고된 상품 수량 차감하기



위의 그림은 시스템 시행 후 오프라인에서 발생한 출고된 상품의 수량을 차감 수행을 표현하는 Sequence diagram이다. main에서 login check를 실행한 뒤 isStoreManager가 ture로 return이 되었으면 Release를 실행한다. 차감된 내용을 product에 저장한다.



## 5. Implementation Requirements

### 1) H/W platform requirements

- (1) CPU : Intel PENTIUM IV 이상
- (2) RAM : 1GB 이상
- (3) HDD / SSD : 10GB 이상

### 2) S/W platform requirements

- (1) OS : Microsoft Window 7이상
- (2) Implemetaion Language : JAVA

## 6. Glossary

TERMS	Description
Attribute	객체지향형 프로그래밍 언어에서 멤버 변수를 가리킴.
Class Diagram	객체지향형 시스템 설계에서, 시스템의 논리 설계를 위한 클래스의 존재와 그들의 관계를 도식으로 정의한 것으로 단일 클래스 다이어그램은 시스템 클래스 구조를 보여줌.
Method	멤버 함수라고도 하며, 객체 지향형 프로그래밍 언어에서 클래스 혹은 객체에 소속된 서브루틴을 가리킴.
Operation	Method와 같은 역할을 하며 객체지향형 프로그래밍 언어에서 클래스 혹은 객체에 소속된 서브루틴을 가리킴
Sate Machine Diagram	시스템의 동작을 설명하는 데 사용되는 상태 다이어그램의 유형으로 설명된 시스템이 한정된 수의 상태로 구성됨.

## 7. References

- 강의자료 : Structural Modeling II, Behavior Modeling I, II, III
- 참고자료 : Real Estate Trade System, Always be there