

배민 통합 서비스

T09

202011252 박유나

202211328 윤찬규

202211329 이 건

목 차

1. 팀원 구성	2
2. 프로젝트 소개	2
2.1. 배민 통합 서비스	2
3. 최종 산출물	2
3.1. HW Platform	2
3.2. SW Platform	2
3.3. 개발 내용	2
3.3.1. 배민 서비스	2
3.3.2. 배달 대행 서비스	3
3.3.3. 결제 서비스	3
4. 핵심 문제 정의 및 해결 방안	3
4.1. AS-IS 분석	3
4.1.1. AS-IS 모델	3
4.1.2. 비즈니스 문제점	3
4.2. TO-BE 분석	3
4.2.1. 비즈니스 비전	3
4.2.2. TO-BE 모델	4
4.3. 기술적 문제 정의	4
4.3.1. 서비스 내 아키텍처 설계	4
4.3.2. 서비스 간 통신	5
4.3.3. 데이터의 동기화	5
4.3.4. 결제 방법의 다양성	5
5. 요구사항	6
5.1. 기능요구사항	6
5.1.1. 배민 서비스	6
5.1.2. 배달 대행 서비스	10
5.1.3. 결제 서비스	13
5.2. 비기능요구사항	15
6. SW Architecture	15
6.1. Initial Architecture	15
6.2. Conceptual Model	16
6.3. Skeleton Architecture	16
6.4. 적용한 아키텍처 및 디자인 패턴	18
6.4.1. Event Driven Architecture	18
6.4.2. Layered Architecture	18
6.4.3. Observer Design pattern	18
6.4.4. Proxy Design pattern	18
6.5. Sequence Diagram	19
6.6. Class Diagram	23
6.7. Deployment Diagram	24
7. Alternative Solutions + Project Justification	24
8. Risk Analysis + Risk Reduction Plan	24
8.1. Spring 사용	24
8.2. Kafka 사용	24
9. Success Criteria	24
9.1. 전체 서비스	24
9.2. 배민 서비스	24
9.3. 배달 대행 서비스	24
9.4. 결제 서비스	24

10. 역할 분담	25
11. 스케줄	25
12. 참고자료 리스트	25

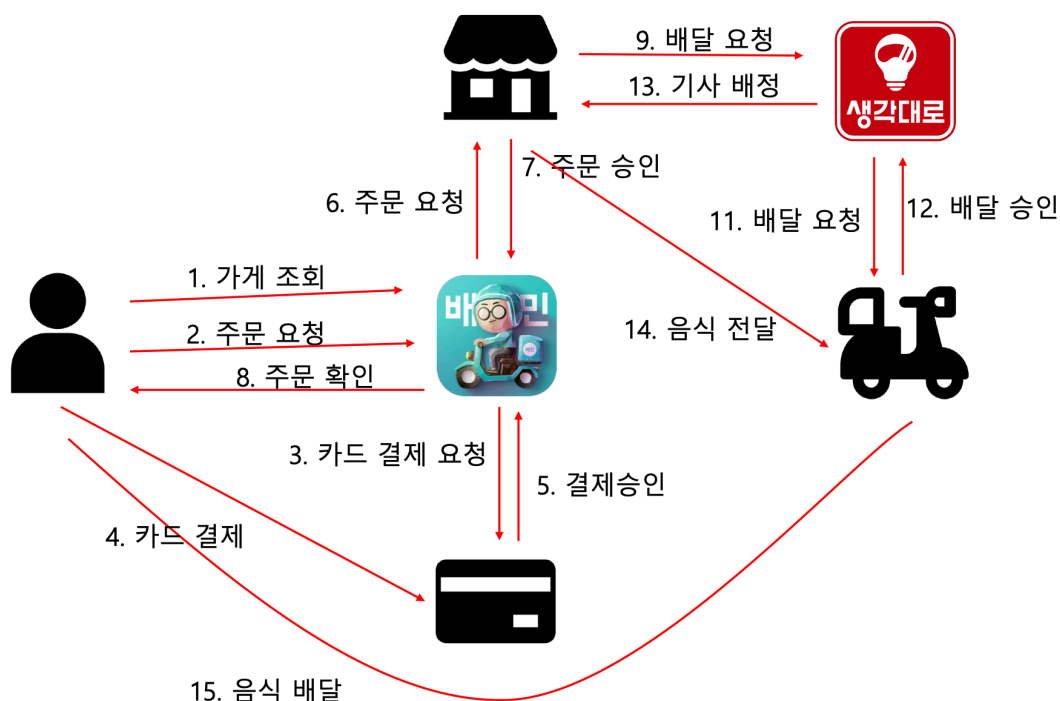
1. 팀원 구성

이름	학번
곽유나	202011252
윤찬규	202211328
이건	202211329

2. 프로젝트 소개

2.1. 배민 통합 서비스

배민 통합 서비스는 고객이 음식점을 조회하고 주문하는 '배민 서비스', 주문을 결제하는 '결제 서비스', 주문한 음식의 배달을 담당하는 '배달 대행 서비스'로 구성되어 있다. 고객이 음식을 주문하고 배달 받기까지의 간단한 프로세스는 아래와 같다.



3. 최종 산출물

3.1. HW Platform

Jar 파일로 제공되므로 하드웨어 플랫폼 영향을 받지 않는다.

3.2. SW Platform

Language	Java
Runtime	JDK 17

3.3. 개발 내용

3.3.1. 배민 서비스

- 고객은 가게를 조회할 수 있다.
- 고객은 주문을 요청, 취소, 가게는 주문을 수락, 거절할 수 있다.
- 고객은 주문 내역을 확인할 수 있다.

3.3.2. 배달 대행 서비스

- 가게는 주문을 받아 배달 대행 서비스에 배달을 요청할 수 있다.
- 배달 대행 서비스는 기사에게 배달을 요청할 수 있다.
- 기사는 배달을 수락, 거절할 수 있다.
- 가게는 배정된 기사의 정보를 확인할 수 있다.

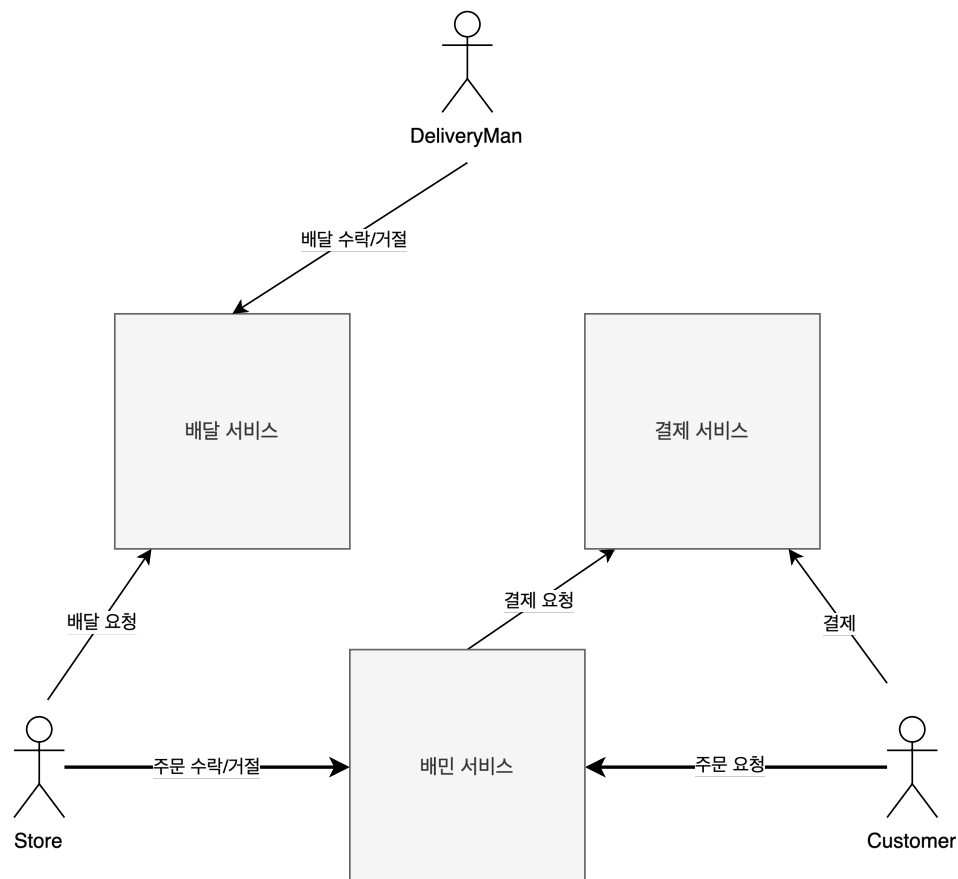
3.3.3. 결제 서비스

- 결제 요청을 승인, 취소할 수 있다.
- 다양한 결제 수단에 대한 결제를 처리한다.
- 결제 상태를 관리한다.
- 결제 정보를 저장하고, 결제 내역을 관리한다.

4. 핵심 문제 정의 및 해결 방안

4.1. AS-IS 분석

4.1.1. AS-IS 모델



4.1.2. 비즈니스 문제점

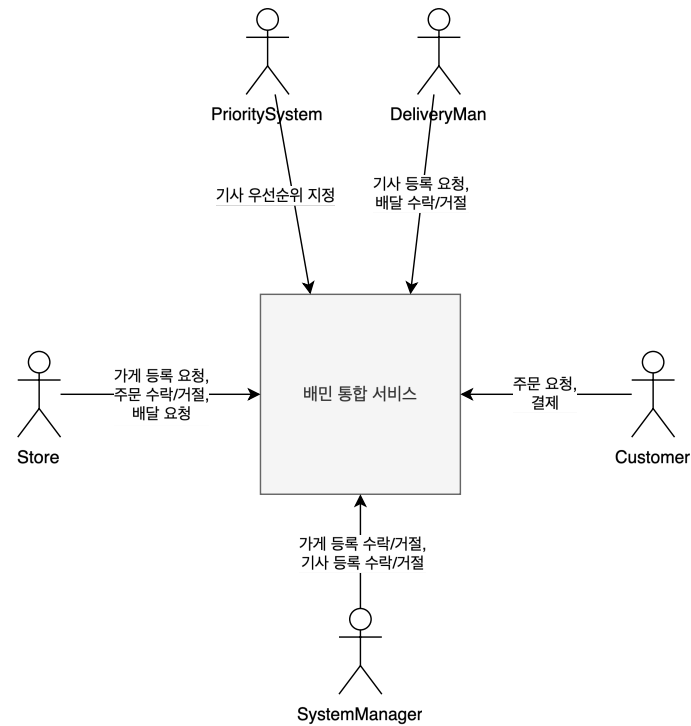
- 주문에 대한 배달을 수행할 때 주문, 배달, 결제 등의 기능을 맡는 주체가 달라서 서로에 대한 정보를 파악하기 힘들다.
- 기사에게 무작위적인 배달 요청 시에 배달 요청의 지연이 발생한다.

4.2. TO-BE 분석

4.2.1. 비즈니스 비전

- 가게는 배민 통합 서비스를 통해 기사에게 배달을 요청하고, 기사는 수락할 수 있으며 고객은 배민 통합 서비스에서 배달 상황을 확인할 수 있다.
- 가게의 배달 요청이 지연되지 않도록 서비스에서 기사에게 우선 순위에 따라 배달을 요청한다.

4.2.2. TO-BE 모델



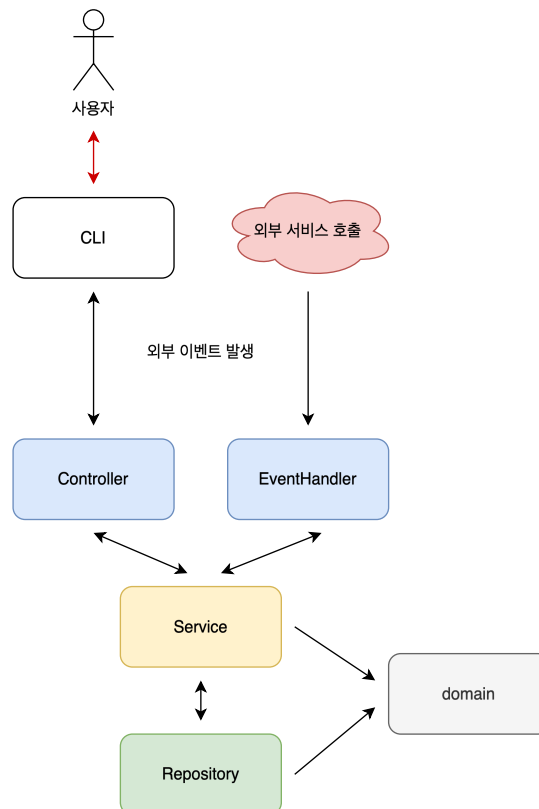
4.3. 기술적 문제 정의

4.3.1. 서비스 내 아키텍처 설계

문제 정의

각 서비스 '배민 서비스', '결제 서비스', '배달 대행 서비스'를 재사용성과 확장성을 고려하여 좋은 아키텍처를 설계할 수 있도록 한다.

해결 방안



각 서비스는 계층 구조를 기반으로 한 3개의 계층 즉, 1계층(Controller, EventHandler), 2계층(Service), 3계층(Repository)으로 구성한다.

- 1계층: 외부 이벤트에 반응하여 필요한 기능 로직을 호출하는 계층이다. '배민 통합 서비스'에서 외부 이벤트는 사용자가 서비스를 이용할 수 있는 인터페이스인 CLI의 요청과 다른 서비스에서 publish 한 이벤트가 있다.

- 2계층: 서비스 계층은 기능을 실행하기 위해 필요한 로직을 수행하는 계층이다.
- 3계층: 레파지토리 계층에서는 데이터를 접근하고, 저장, 관리하는 계층이다.

4.3.2. 서비스 간 통신

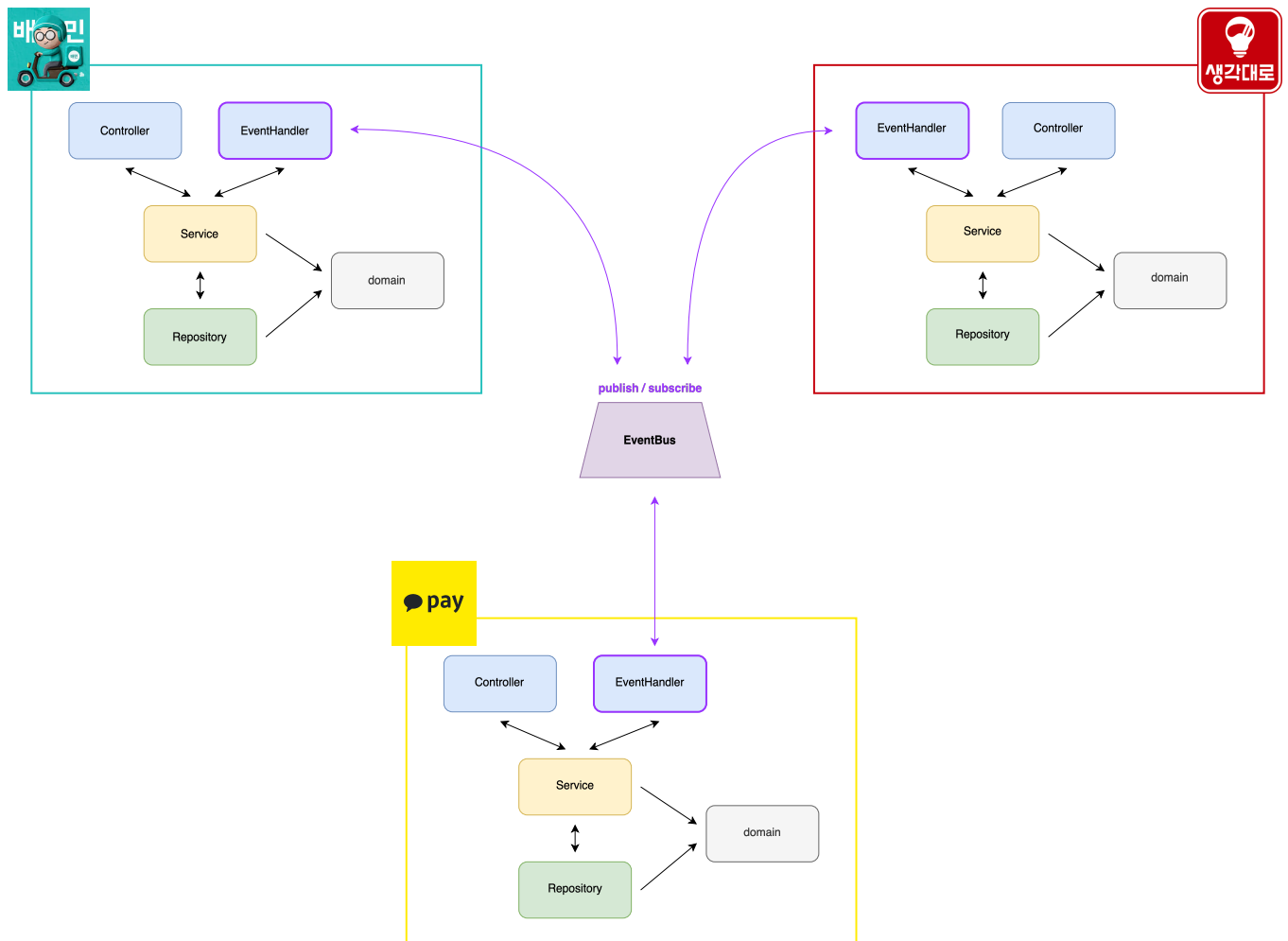
문제 정의

배민 통합 서비스는 '배민 서비스', '결제 서비스', '배달 대행 서비스'로 구성되어 있다. 따라서 각 서비스 작업의 결과가 다른 서비스를 호출하는 이벤트가 될 수 있다. 이를 위해 서비스에서 다른 서비스를 호출할 수 있는 방법이 필요하다.

해결 방안

서비스에서 서비스를 호출하기 위해 타 서비스의 메서드를 직접 호출할 수 있을 것이다. 하지만 이는 서비스에서 다른 서비스를 의존하고, 서비스간 강결합이 발생한다. 좋은 객체 지향 설계를 위해서는 서비스간 loose-coupling을 지향할 필요가 있다.

따라서 스프링 이벤트(Application Events)를 활용한다. 기본적으로 publisher와 listener가 동작하여 이벤트를 발행하는 메서드를 호출하면 이벤트 리스너가 처리하는 형태이다. 이러한 구조를 통해 서비스간 호출 로직을 제공한다.



4.3.3. 데이터의 동기화

문제 정의

각 서비스에는 여러 액터가 존재하기 때문에 같은 데이터를 접근, 수정할 수 있다. 해당 데이터에 대한 일관성을 유지하는 방법이 필요하다.

해결 방안

Kafka를 사용하여 마이크로서비스 간에 발생하는 데이터 동기화 문제를 해결한다.

4.3.4. 결제 방법의 다양성

문제 정의

사용자가 다양한 결제 수단(신용카드, 간편 결제 등)을 통해 결제를 진행할 수 있어야 한다.

해결 방안

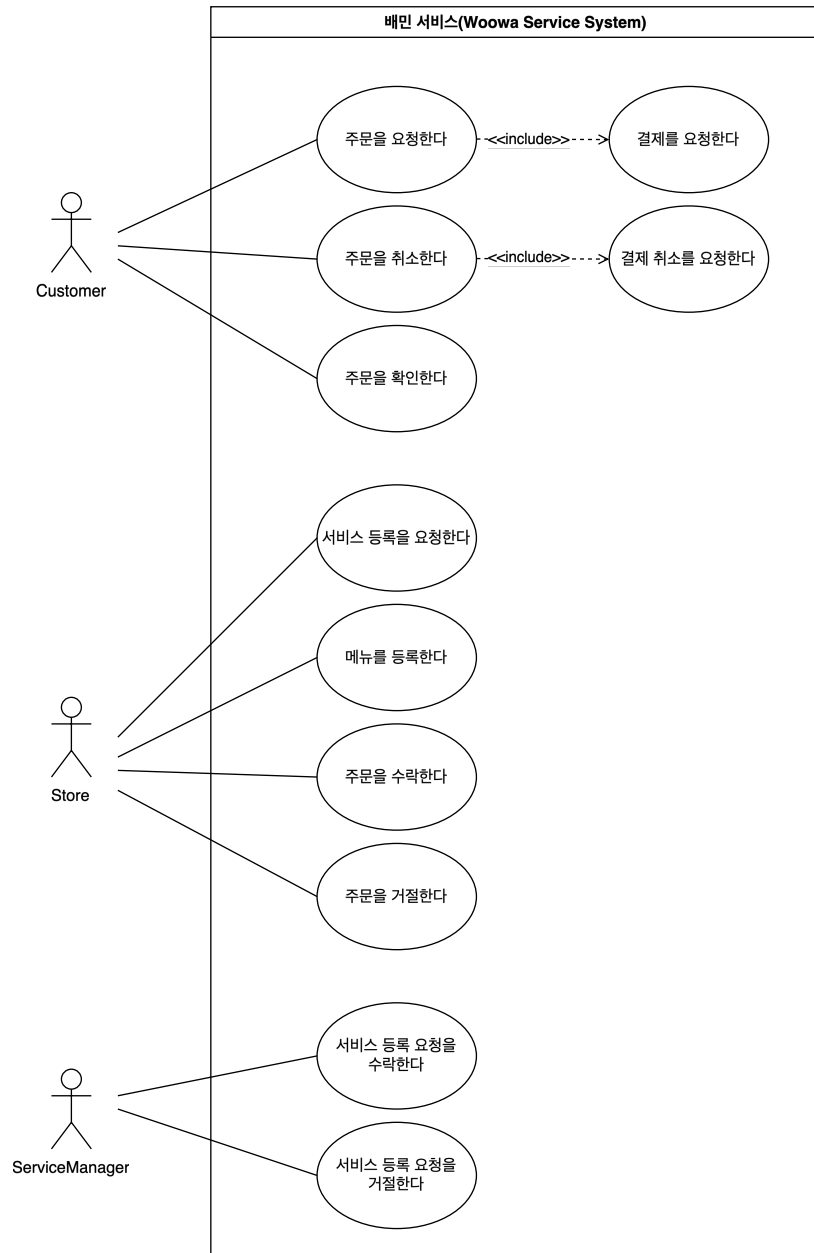
인터페이스를 통해 공통 기능들을 추상화하여, 다양한 결제 수단들에 대해 기능을 확장할 수 있도록 설계한다.

5. 요구사항

5.1. 기능요구사항

5.1.1. 배민 서비스

5.1.1.1. UseCase Diagram



5.1.1.2. Actor

- 가게 (Store)
 - 서비스 등록 요청하기, 메뉴 등록하기, 주문 수락&거절하기
- 서비스 관리자 (Service Manager)
 - 가게 관리
- 고객 (Customer)
 - 주문 요청하기

5.1.1.3. UseCase Brief Description

- **UC1: 주문 요청하기 (Request Order)**
 - 고객이 메뉴를 선택하여 가게에 주문을 요청한다
- **UC2: 결제 요청하기 (Request Payment)**
 - 결제 서비스에 결제를 위임한다
- **UC3: 주문 취소하기 (Cancel Order)**

- ▶ 고객이 주문 목록 중 아직 배송이 취소되지 않은 주문을 취소한다
- **UC4: 결제 취소를 요청하기 (Request Cancel Payment)**
 - ▶ 결제 서비스에 결제를 취소를 요청한다
- **UC5: 주문 확인하기 (Check Order Status)**
 - ▶ 고객이 현재 주문의 상태를 확인한다
- **UC6: 서비스 등록 요청하기 (Request Service Registration)**
 - ▶ 가게가 서비스에 가게 정보 등록을 요청한다
- **UC7: 메뉴 등록하기 (Register Menu)**
 - ▶ 가게가 서비스에 메뉴를 등록한다
- **UC8: 주문 수락하기 (Accept Order)**
 - ▶ 가게가 고객의 주문을 수락한다
- **UC9: 주문 거절하기 (Reject Order)**
 - ▶ 가게가 고객의 주문을 거절한다
- **UC10: 서비스 등록 요청 수락하기 (Accept Service Registration Request)**
 - ▶ 서비스 관리자가 가게의 등록 요청을 수락한다
- **UC11: 서비스 등록 요청 거절하기 (Reject Service Registration Request)**
 - ▶ 서비스 관리자가 가게의 등록 요청을 거절한다

5.1.1.4. UseCase Description

- **UC1: 주문 요청하기 (Place Order)**

UseCase	Description
이름	주문 요청하기 (Place Order)
범위	배달의 민족 서비스
수준	사용자 수준
주요 액터	Customer (고객)
사전 조건	고객이 서비스에 로그인되어 있다
사후 조건	주문 요청이 생성되고, 주문 번호가 반환된다. 주문에 대한 결제가 완료되면, 가게에 성공적으로 요청이 된다.
기본 흐름	1. 메뉴와 수량을 선택한다. 2. 주문자 성함, 번호, 주소를 입력한다. 2. 고객이 주문을 요청한다. 3. 주문 번호가 고객에게 반환된다. 4. 결제를 요청한다(UC2). 5. 결제가 완료되면, 가게에 주문을 요청한다.
대체 흐름	1a. 메뉴의 총 가격이 가게의 최소 주문 금액 미달이면 주문이 거부된다. 4a. 결제가 진행되지 않으면 주문 요청은 생성되지만 가게에 주문이 요청되지 않는다.

- **UC2: 결제 요청하기 (Request Payment)**

UseCase	Description
이름	결제 요청하기 (Request Payment)
범위	배달의 민족 서비스
수준	시스템 수준
주요 액터	Woowa Service (배민 서비스)
사전 조건	고객이 주문을 요청한 후 결제 요청 정보가 생성된다.
사후 조건	결제 서비스에 결제 요청 정보가 생성된다.
기본 흐름	1. 주문 서비스가 고객의 이름, 번호, 결제 금액 등의 정보를 기재하여 결제 요청 정보를 생성한다. 2. 결제 서비스에 결제 요청 정보를 전송한다.
대체 흐름	

- **UC3: 주문 취소하기 (Cancel Order)**

UseCase	Description
이름	주문 취소하기 (Cancel Order)
범위	배달의 민족 서비스

수준	사용자 수준
주요 액터	Customer (고객)
사전 조건	주문 요청이 생성된 상태여야 한다
사후 조건	주문이 취소된다
기본 흐름	1. 고객이 주문 내역에서 주문을 선택한다. 2. 주문 취소를 요청한다. 3. 주문 메뉴, 환불 금액 등의 취소 정보가 반환된다. 4. 결제 서비스에 환불을 요청한다(UC4).
대체 흐름	1a. 주문의 배송이 시작된 상태라면, 주문을 취소할 수 없다.

• UC4: 결제 취소를 요청하기 (Request Cancel Payment)

UseCase	Description
이름	주문 취소하기 (Cancel Order)
범위	배달의 민족 서비스
수준	사용자 수준
주요 액터	Woowa Service (배민 서비스)
사전 조건	주문 요청이 취소된 상태여야 한다
사후 조건	결제 서비스에 결제 취소가 요청된다.
기본 흐름	1. 고객의 이름, 번호, 결제 금액 등의 정보로 결제 취소 요청을 생성한다. 2. 결제 취소를 요청한다.
대체 흐름	1a. 주문의 배송이 시작된 상태라면, 주문을 취소할 수 없다.

• UC5: 주문 확인하기 (Check Order Status)

UseCase	Description
이름	주문 확인하기 (Check Order Status)
범위	배달의 민족 서비스
수준	사용자 수준
주요 액터	Customer (고객)
사전 조건	주문 요청이 생성된 상태여야 한다
사후 조건	주문 상태가 고객에게 표시된다
기본 흐름	1. 고객이 주문 내역을 조회한다. 2. 시스템이 현재 주문 상태(결제 전, 요청 중, 조리 중, 배송 중)를 표시한다.
대체 흐름	

• UC6: 서비스 등록 요청하기 (Request Service Registration)

UseCase	Description
이름	서비스 등록 요청하기 (Request Service Registration)
범위	배달의 민족 서비스
수준	가게 수준
주요 액터	Store (가게)
사전 조건	가게가 시스템에 로그인한 상태여야 한다
사후 조건	서비스 관리자에게 등록 요청이 전송된다
기본 흐름	1. 가게가 가게 이름, 배달비, 최소 주문 금액을 입력한다. 2. 서비스 등록을 요청한다.
대체 흐름	

• UC7: 메뉴 등록하기 (Register Menu)

UseCase	Description
---------	-------------

이름	메뉴 등록하기 (Register Menu)
범위	배달의 민족 서비스
수준	가게 수준
주요 액터	Store (가게)
사전 조건	가게가 시스템에 로그인한 상태여야 한다
사후 조건	메뉴가 등록되고 고객에게 표시된다
기본 흐름	1. 가게가 메뉴 등록 페이지를 연다. 2. 가게가 메뉴 정보를 입력하고 등록을 완료한다.
대체 흐름	

• UC8: 주문 수락하기 (Accept Order)

UseCase	Description
이름	주문 수락하기 (Accept Order)
범위	배달의 민족 서비스
수준	가게 수준
주요 액터	Store (가게)
사전 조건	가게가 주문 요청을 받은 상태여야 한다
사후 조건	주문이 수락되어 고객에게 알림이 전송된다
기본 흐름	1. 가게가 수신한 주문 요청을 검토한다. 2. 가게가 주문을 수락한다.
대체 흐름	

• UC9: 주문 거절하기 (Reject Order)

UseCase	Description
이름	주문 거절하기 (Reject Order)
범위	배달의 민족 서비스
수준	가게 수준
주요 액터	Store (가게)
사전 조건	가게가 주문 요청을 받은 상태여야 한다
사후 조건	주문이 거절되어 고객에게 알림이 전송된다
기본 흐름	1. 가게가 수신한 주문 요청을 검토한다. 2. 가게가 주문을 거절하고, 거절 사유를 입력한다.
대체 흐름	

• UC10: 서비스 등록 요청 수락하기 (Accept Service Registration Request)

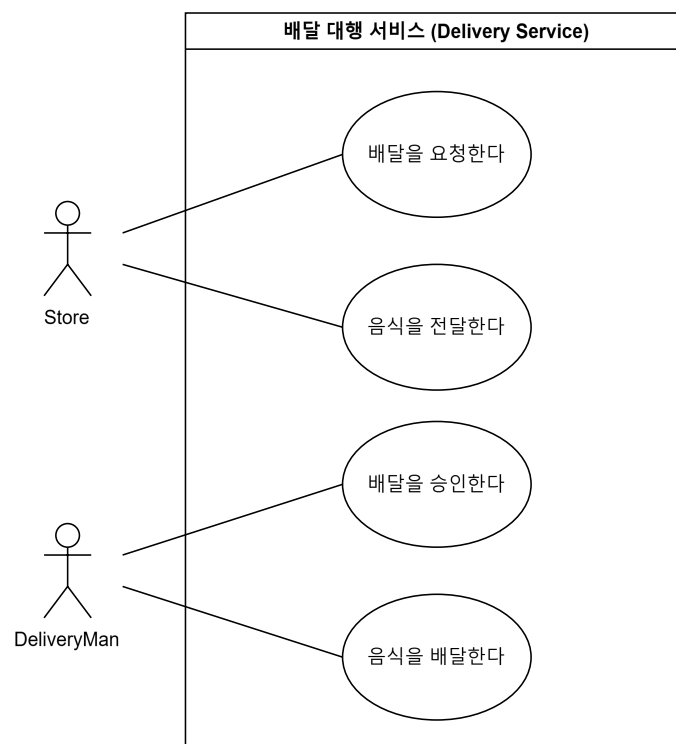
UseCase	Description
이름	서비스 등록 요청 수락하기 (Accept Service Registration Request)
범위	배달의 민족 서비스
수준	관리자 수준
주요 액터	Service Manager (서비스 관리자)
사전 조건	가게의 서비스 등록 요청이 접수된 상태여야 한다
사후 조건	가게의 서비스가 등록되고, 가게에게 알림이 전송된다
기본 흐름	1. 서비스 관리자가 등록 요청 목록에서 가게 이름, 배달비, 최소 주문 금액 등의 정보를 검토한다. 2. 승인 사유를 명시한다. 3. 관리자가 등록 요청을 수락한다.
대체 흐름	

• UC11: 서비스 등록 요청 거절하기 (Reject Service Registration Request)

UseCase	Description
이름	서비스 등록 요청 거절하기 (Reject Service Registration Request)
범위	배달의 민족 서비스
수준	관리자 수준
주요 액터	Service Manager (서비스 관리자)
사전 조건	가게의 서비스 등록 요청이 접수된 상태여야 한다
사후 조건	가게의 서비스 등록 요청이 거절되고, 가게에게 알림이 전송된다
기본 흐름	1. 서비스 관리자가 등록 요청 목록에서 가게 이름, 배달비, 최소 주문 금액 등의 정보를 검토한다. 2. 거절 사유를 명시한다. 3. 관리자가 등록 요청을 거절한다.
대체 흐름	

5.1.2. 배달 대행 서비스

5.1.2.1. Use Case Diagram



5.1.2.2. Actor

- 가게 (Store)
 - ▶ 배달 요청하기, 음식 전달하기, 서비스 등록하기
- 기사 (Delivery Man)
 - ▶ 배달 승인하기, 음식 배달하기, 서비스 등록하기
- 배달 대행 서비스 (Delivery Service)
 - ▶ 배달 요청하기, 기사 배정하기
- 서비스 관리자 (Service Manager)
 - ▶ 가게 관리, 기사 관리

5.1.2.3. UseCase Brief Description

- **UC1: 배달 요청하기 (Request Delivery)**
 - ▶ 가게가 주문 승인 이후 배달 대행 서비스에 특정 고객에게로의 배달을 요청한다.
- **UC2: 음식 전달하기 (Offer Food)**
 - ▶ 가게가 배정받은 기사에게 음식을 전달한다.
- **UC3: 배달 승인하기 (Approve Delivery)**
 - ▶ 배달 대행 서비스에서 받은 배달 요청에 대한 승인 여부를 결정한다.
- **UC4: 음식 배달하기 (Deliver Food)**
 - ▶ 배달 대행 서비스에서 받은 배달 요청 정보에 따라 가게에서 음식을 전달받고 해당 음식을 고객에게 배달한다.

- UC5: 기사 요청하기 (Request Delivery Man)

- 가게에서 받은 배달 요청에 따라 우선순위가 높은 기사부터 배달 요청을 보낸다.

• UC6: 기사 배정하기 (Assign Delivery Man)

- 배달 요청을 승인한 기사에 대한 정보를 가게에 전달한다.

5.1.2.4. UseCase Description

• UC1: 배달 요청하기 (Request Delivery by Store)

UseCase	Description
이름	배달 요청하기 (Request Delivery by Store)
범위	배달 대행 서비스
수준	가게 수준
주요 액터	Store (가게), Delivery Service (배달 대행 서비스)
사전 조건	배민 서비스에게 주문을 요청받고 승인한 상태여야 한다.
사후 조건	배달을 배정받은 기사에 대한 정보가 가게에 전달된다. 주문 취소가 불가능해진다.
기본 흐름	1. 가게가 고객의 위치를 기반으로 배달 대행 서비스에 배달 요청을 보낸다.
대체 흐름	1a. 주문 취소가 들어온 경우 배달 요청을 중단한다.

• UC2: 음식 전달하기 (Offer Food)

UseCase	Description
이름	음식 전달하기 (Offer Food)
범위	배달 대행 서비스
수준	가게 수준
주요 액터	가게
사전 조건	배달 요청을 승인한 기사가 존재하는 상태여야 한다.
사후 조건	음식이 기사에게 전달되었는지 여부를 반환한다.
기본 흐름	1. 기사가 가게에 들어온다. 2. 배정받은 기사와 가게에 들어온 기사의 ID가 일치한지 확인한다. 3. 일치하다면 음식을 기사에게 전달한다. 4. 음식 전달 완료 메시지를 배달의 민족 서비스로 전송한다.
대체 흐름	2a. 기사의 정보가 일치하지 않는다면 다른 기사가 가게에 들어온 이후 다시 2. 를 실행한다.

• UC3: 배달 승인하기 (Approve Delivery)

UseCase	Description
이름	배달 승인하기 (Approve Delivery)
범위	배달 대행 서비스
수준	기사 수준
주요 액터	기사
사전 조건	배달 대행 서비스로부터 배달 요청을 받은 상태여야 한다.
사후 조건	배달 요청에 대한 승인 여부가 배달 대행 서비스에게 전달된다.
기본 흐름	1. 기사가 배달 요청을 확인한다. 2. 배달 요청에 대한 승인 여부를 결정한다. 3. 승인 여부를 배달 대행 서비스에 전달한다.
대체 흐름	

• UC4: 음식 배달하기 (Deliver Food)

UseCase	Description
이름	음식 배달하기 (Deliver Food)
범위	배달 대행 서비스
수준	기사 수준

주요 액터	기사
사전 조건	가게로부터 음식을 전달받은 상태여야 한다.
사후 조건	배달 상태가 종료된다.
기본 흐름	1. 주문 정보에 따라 고객의 위치로 이동한다. 2. 고객에게 음식을 전달한다. 3. 배달 상태 종료 메시지를 배달의 민족 서비스로 전송한다.
대체 흐름	

• UC5: 기사 요청하기 (Request Delivery Man)

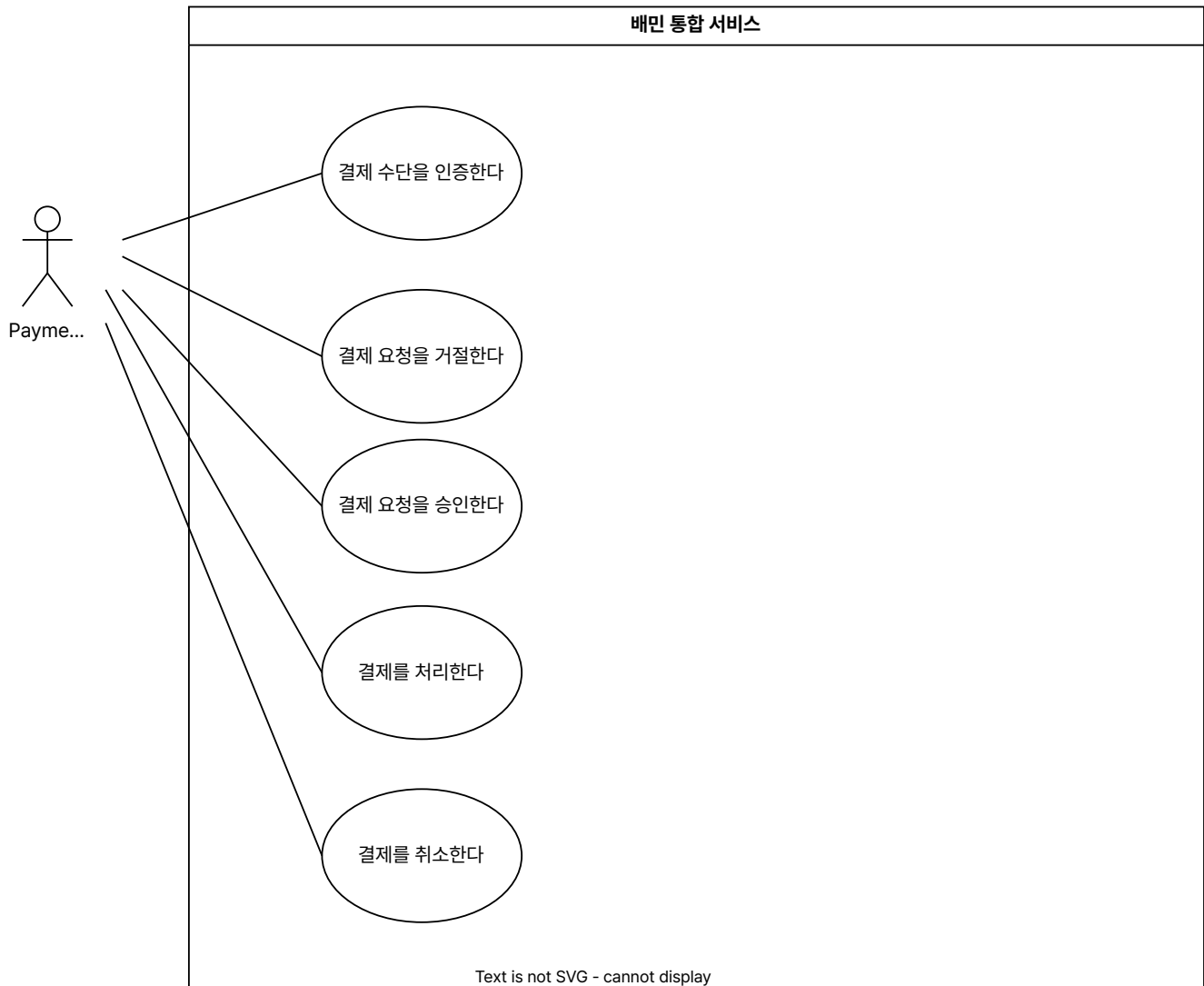
UseCase	Description
이름	기사 요청하기 (Request Delivery Man)
범위	배달 대행 서비스
수준	시스템 수준
주요 액터	Delivery Service (배달 대행 서비스)
사전 조건	가게로부터 배달 요청을 받은 상태여야 한다.
사후 조건	기사에게 배달 요청에 대한 승인 여부를 전달받는다.
기본 흐름	1. 우선순위 결정 규칙에 따라 기사의 우선순위를 결정한다. 2. 우선순위가 높은 기사부터 요청을 보낸다.
대체 흐름	2a. 기사가 요청을 거절한 경우 다음 우선순위의 기사에게 요청을 보낸다.

• UC6: 기사 배정하기 (Assign Delivery Man)

UseCase	Description
이름	기사 배정하기 (Assign Delivery Man)
범위	배달 대행 서비스
수준	시스템 수준
주요 액터	Delivery Service (배달 대행 서비스)
사전 조건	요청을 승인한 기사가 존재하는 상태여야 한다.
사후 조건	요청을 승인한 기사의 정보가 가게에 전달된다.
기본 흐름	1. 가게에게 요청을 승인한 기사의 정보를 전달한다.
대체 흐름	

5.1.3. 결제 서비스

5.1.3.1. Use Case Diagram



5.1.3.2. Actor

- 배민 서비스 (WoowaService)
 - 결제 요청, 결제 취소 요청
- 결제 서비스 (PaymentService)
 - 결제 수단 인증, 결제 요청 승인, 결제 요청 거절, 결제 처리, 결제 취소

5.1.3.3. UseCase Brief Description

- UC1: 결제 수단 인증하기 (Verify Payment Method)**
 - 결제 서비스가 결제 수단을 확인하고 인증한다.
- UC2: 결제 요청 승인하기 (Accept Payment Request)**
 - 결제 서비스가 결제 요청을 승인한다.
- UC3: 결제 요청 거절하기 (Reject Payment Request)**
 - 결제 서비스가 결제 요청을 거절한다.
- UC4: 결제 처리하기 (Process Payment)**
 - 결제 서비스가 결제 수단에 따라 결제를 진행한다.
- UC5: 결제 취소하기 (Cancel Payment)**
 - 결제 서비스는 결제 정보를 통해 결제를 취소하고 결제 정보의 상태를 '취소'로 변경한다.

5.1.3.4. UseCase Description

- UC1: 결제 수단 인증하기 (Verify Payment Method)**

UseCase	Description
이름	결제 수단 인증하기 (Verify Payment Method)

범위	결제 서비스
수준	결제 서비스 수준
주요 액터	결제 서비스
사전 조건	배민 서비스로부터 결제 요청을 받는다.
사후 조건	배민 서비스에 결제 수단 인증 메시지가 전달된다.
기본 흐름	1. 결제 서비스는 전달 받은 결제 수단이 고객의 명의로 등록되어 있는지 확인한다. 2. 결제 서비스는 확인한 결제 수단 정보가 기재된 결제 수단 인증 완료 메시지를 배민 서비스에 전달한다.
대체 흐름	1a. 결제 수단이 등록되어 있지 않다면 결제 수단 인증이 거부된다.

• UC2: 결제 요청 승인하기 (Accept Payment Request)

UseCase	Description
이름	결제 요청 승인하기 (Accept Payment Request)
범위	결제 서비스
수준	결제 서비스 수준
주요 액터	결제 서비스
사전 조건	결제 수단 인증이 완료되었다. 배민 서비스로부터 결제 요청을 받는다.
사후 조건	
기본 흐름	1. 결제 정보의 현재 상태를 '승인됨'으로 변경한다. 2. 결제 요청 승인 메시지를 배민 서비스에 전달한다.
대체 흐름	

• UC3: 결제 요청 거절하기 (Reject Payment Request)

UseCase	Description
이름	결제 요청 거절하기 (Reject Payment Request)
범위	결제 서비스
수준	결제 서비스 수준
주요 액터	결제 서비스
사전 조건	결제 수단 인증이 완료되었다. 배민 서비스로부터 결제 요청을 받는다.
사후 조건	
기본 흐름	1. 결제 정보의 현재 상태를 '거부됨'으로 변경한다. 2. 결제 요청 거부 메시지를 배민 서비스에 전달한다.
대체 흐름	

• UC4: 결제 처리하기 (Process Payment)

UseCase	Description
이름	결제 처리하기 (Process Payment)
범위	결제 서비스
수준	결제 서비스 수준
주요 액터	결제 서비스
사전 조건	결제 수단 인증이 완료되었다. 배민 서비스로부터 결제 요청을 받는다. 결제 요청 승인이 완료되었다.
사후 조건	결제 정보가 저장된다.
기본 흐름	1. 결제 수단에 따라 결제를 진행한다. 2. 결제 정보의 현재 상태를 '완료됨'으로 변경한다. 2. 결제 정보를 저장하고 결제 금액 등의 정보가 기재된 결제 완료 메시지를 배민 서비스에 전달한다.
대체 흐름	1a. 결제가 실패하면 배민 서비스에 실패 사유 정보가 기재된 결제 실패 메시지를 전달한다.

• UC5: 결제 취소하기 (Cancel Payment)

UseCase	Description
이름	결제 취소하기 (Cancel Payment)
범위	결제 서비스
수준	결제 서비스 수준
주요 액터	결제 서비스
사전 조건	배민 서비스로부터 결제 취소 요청을 받는다.
사후 조건	주문에 대한 결제 금액이 환불된다.
기본 흐름	<ol style="list-style-type: none"> 1. 결제 서비스는 주문에 대한 결제 정보가 저장되어 있는지 확인한다. 2. 결제 정보가 있다면 결제 서비스는 지불된 금액을 환불하고 결제 정보의 현재 상태를 '취소됨'로 변경한다. 3. 결제 취소가 성공적으로 완료되면 결제 서비스는 결제 취소 금액 등의 정보가 기재된 결제 취소 완료 메시지를 배민 서비스에 전달한다.
대체 흐름	1a. 결제 정보가 없는 경우, 결제 서비스는 결제 정보 없음 메시지를 배민 서비스에 전달한다.

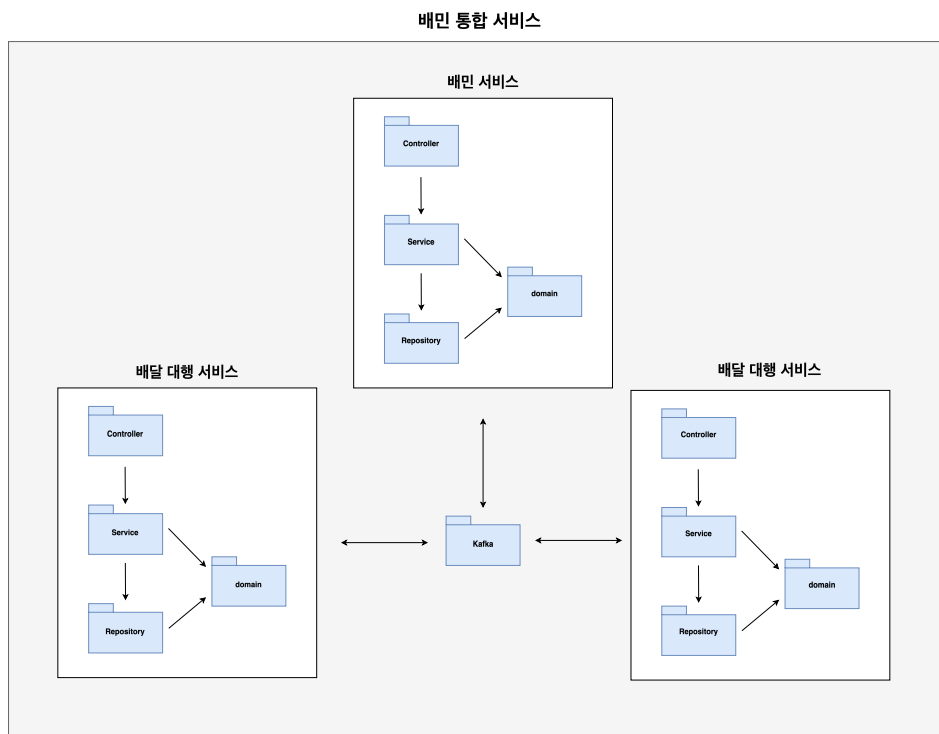
5.2. 비기능요구사항

1. 사용자는 배민 통합서비스가 배민/배달 대행/결제 서비스로 분산되어있다는 것을 몰아야한다. 각 분산 서비스는 다른 서비스의 결과에 따라 연쇄적으로 서비스가 실행되고 하나의 통합 서비스처럼 보여야한다.
2. 고객의 주문 요청이 완료된 후 가게로부터 3분 이내에 수락이 이루어 지지 않으면 주문 요청이 취소된다.
3. 고객이 결제하는 수단은 3가지 이상이 지원된다
4. 고객의 결제 정보가 주문 요청 정보와 일치하는 유효한 정보일 경우에만 결제에 성공해야 한다.
5. 가게가 배달을 요청했을 때, 우선 순위에 따라 배달을 요청할 기사를 1초 이내에 판별해야 한다.
6. 기사가 배달 요청을 1분 이내에 수락하지 않을 시 요청을 취소하고, 다음 우선 순위의 기사에게 요청이 생성된다.
7. 모든 기사가 배달 요청을 거절했을 때, 배달 비용을 높여 처음부터 다시 배탈 요청을 진행한다.

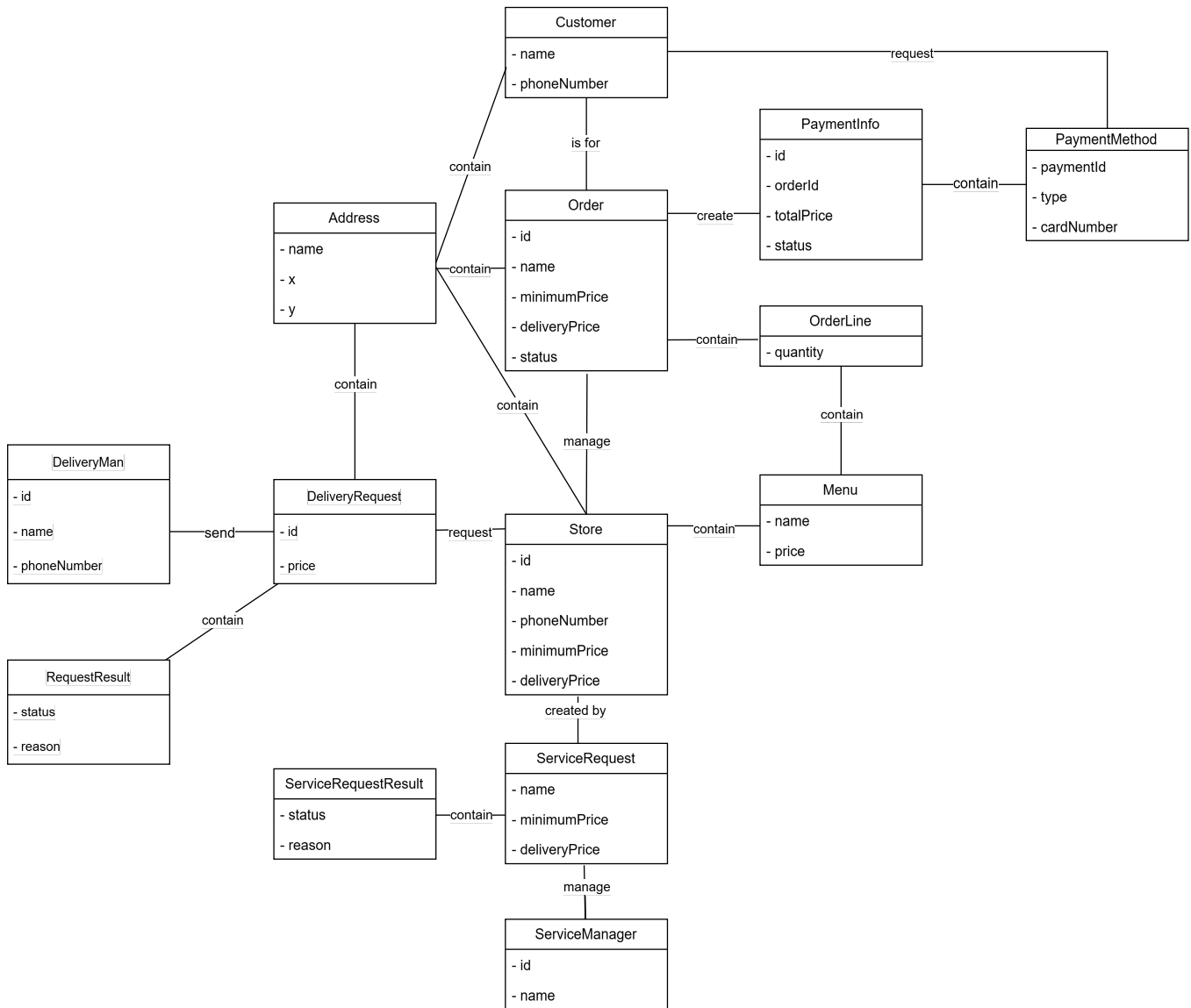
6. SW Architecture

6.1. Initial Architecture

고객, 가게, 기사가 통합된 서비스를 이용할 수 있도록 한다.

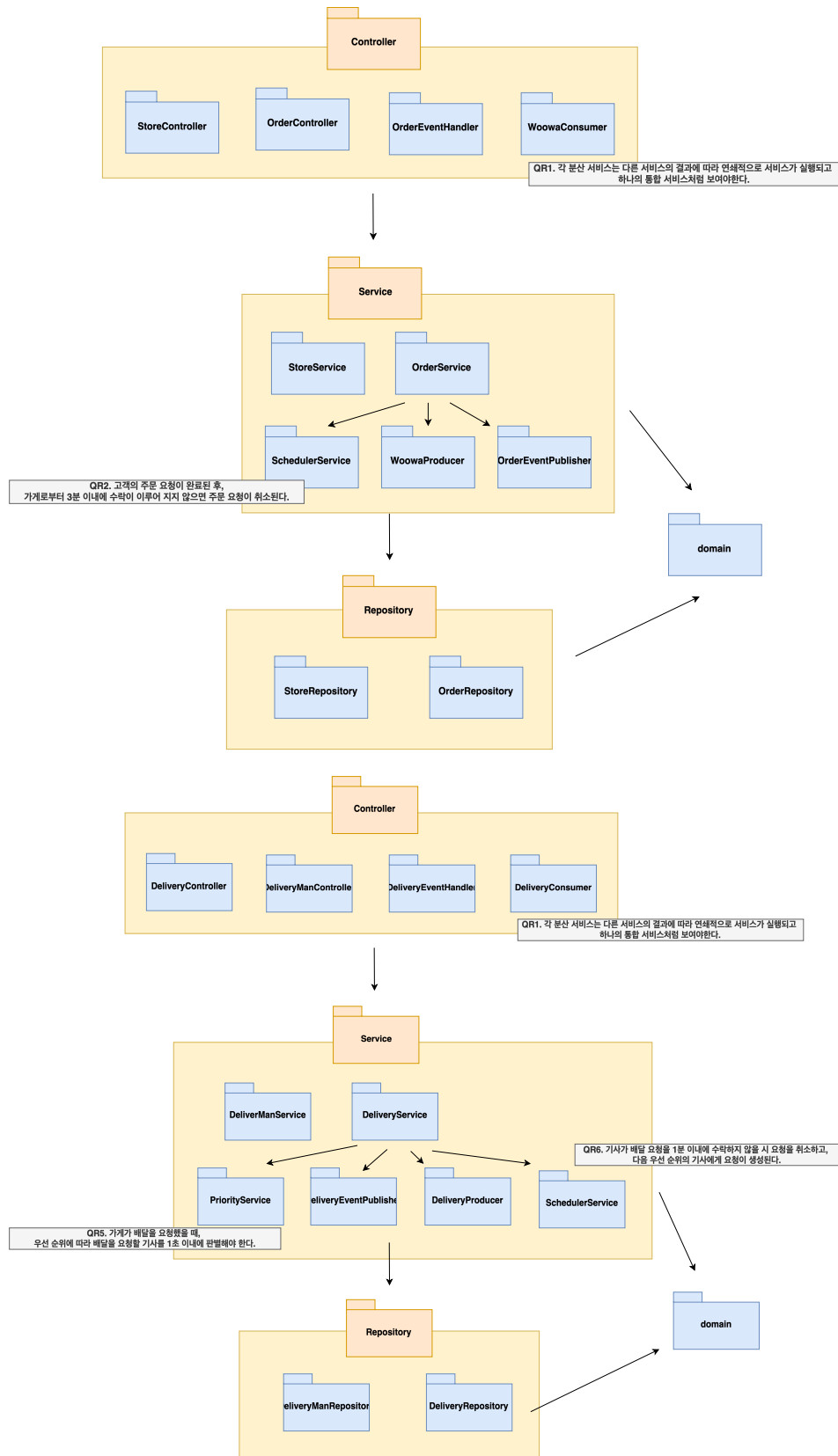


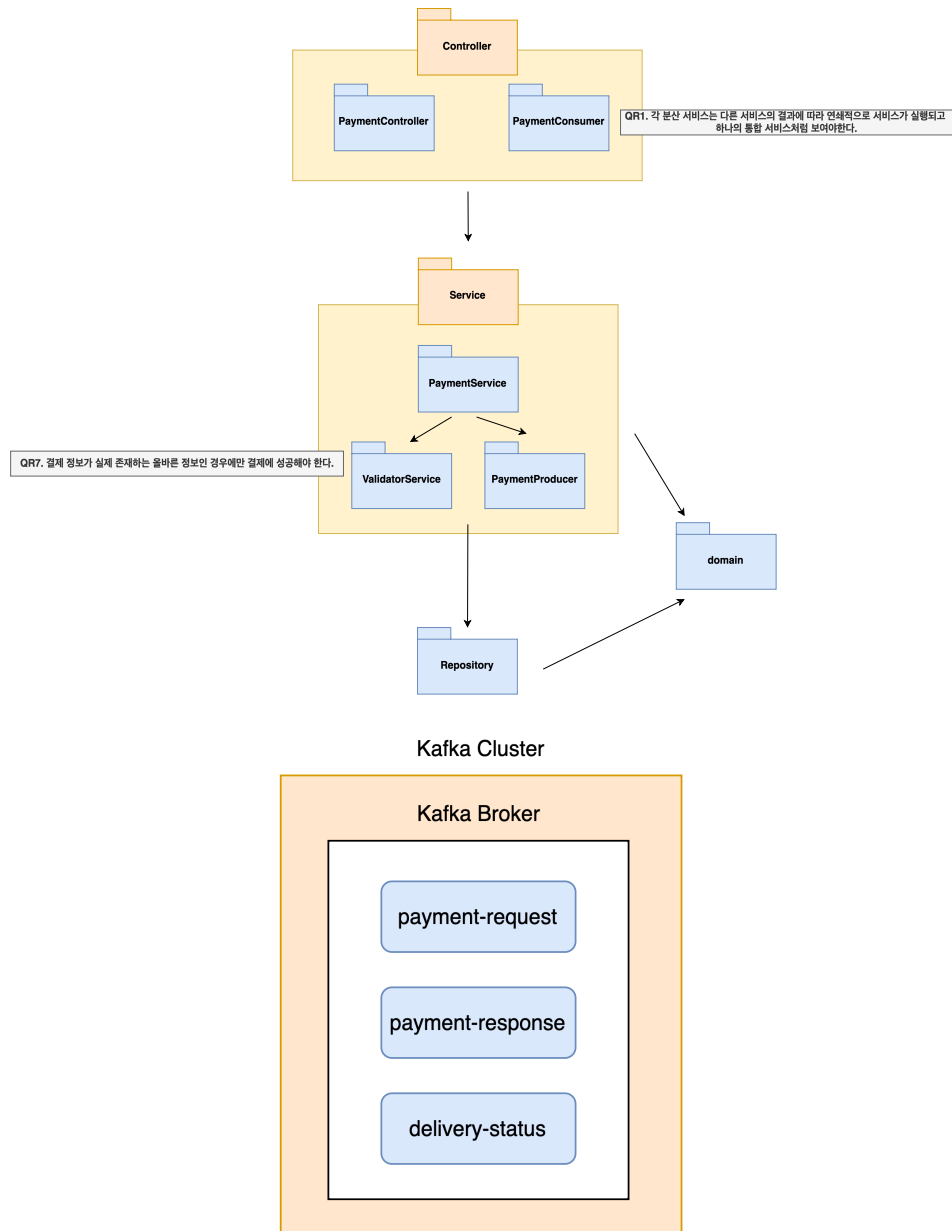
6.2. Conceptual Model



6.3. Skeleton Architecture

Initial Architecture를 Quality Requirement에 따라 정제하였다.





6.4. 적용한 아키텍처 및 디자인 패턴

6.4.1. Event Driven Architecture

Kafka와 마이크로서비스 통신을 위한 이벤트 기반 아키텍처를 적용하였다.

6.4.2. Layered Architecture

각 마이크로서비스 구조에 대해 계층 설계를 적용하여 코드의 재사용성과 유지보수성을 높였다.

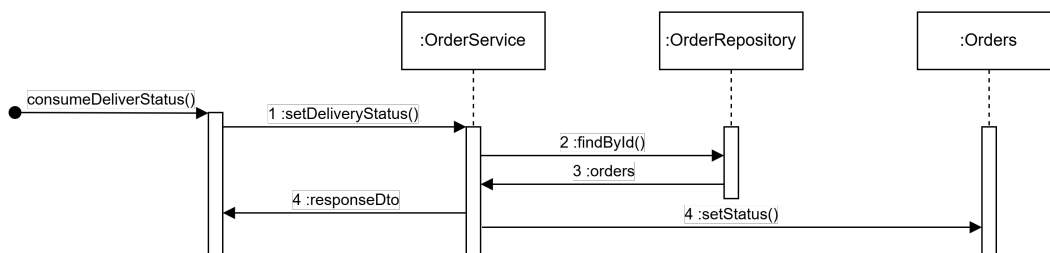
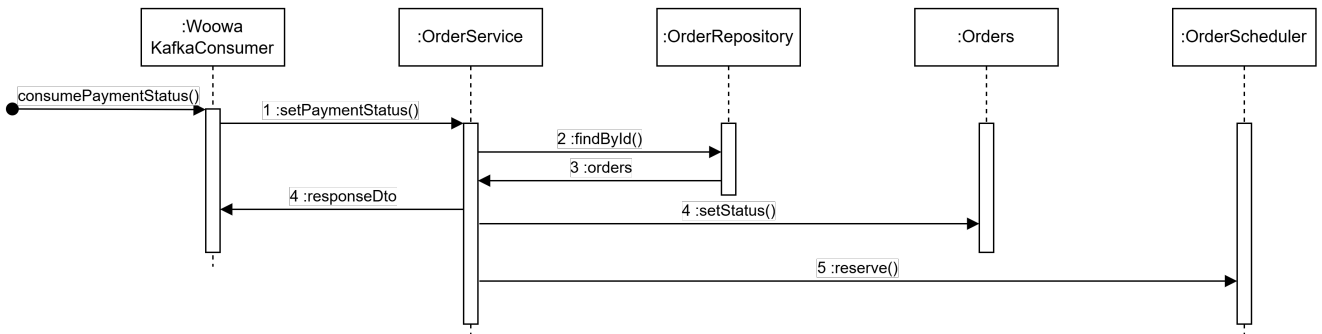
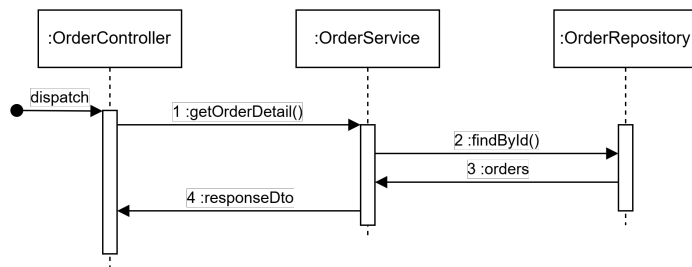
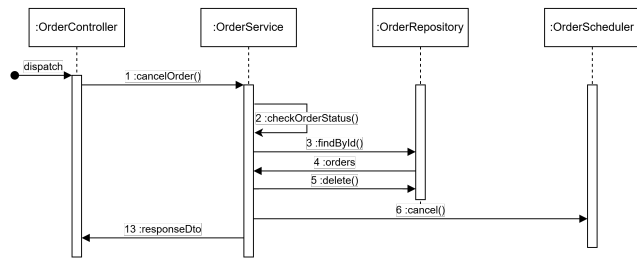
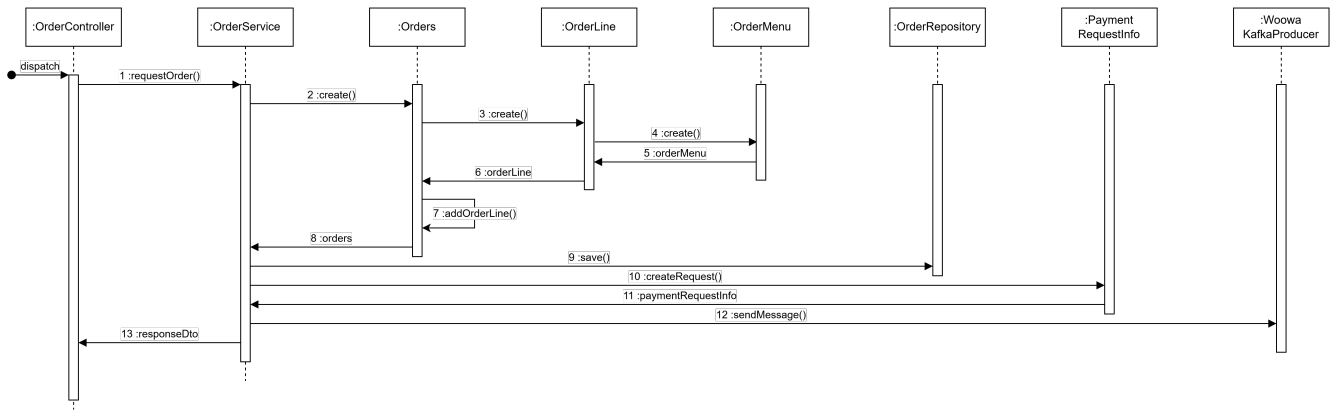
6.4.3. Observer Design pattern

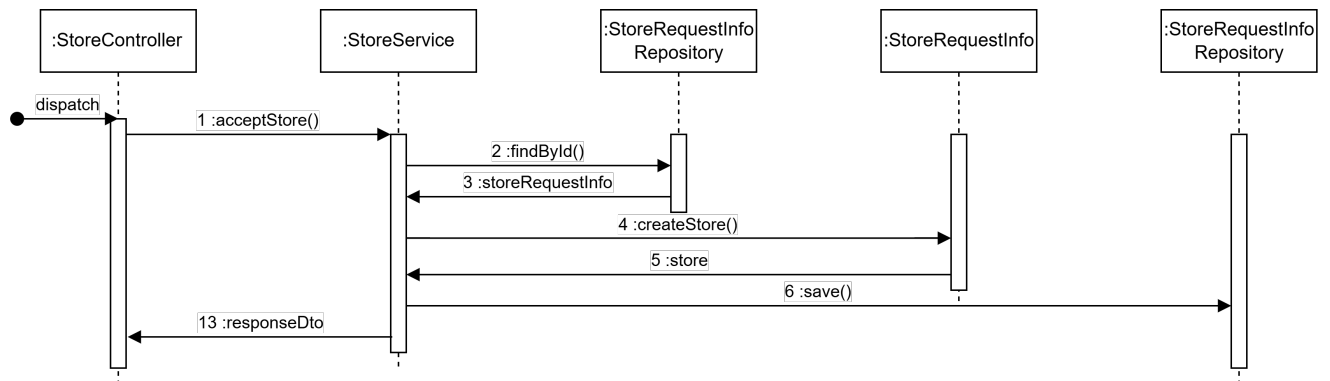
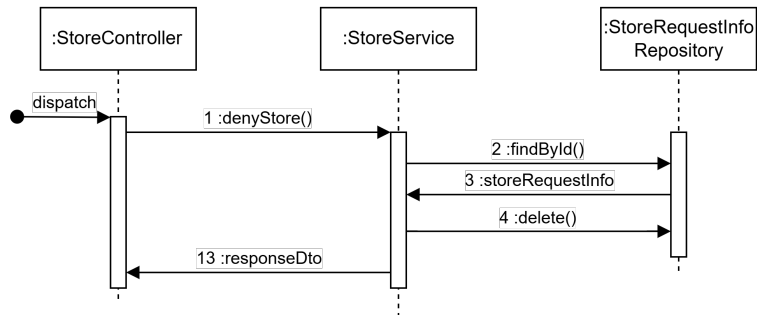
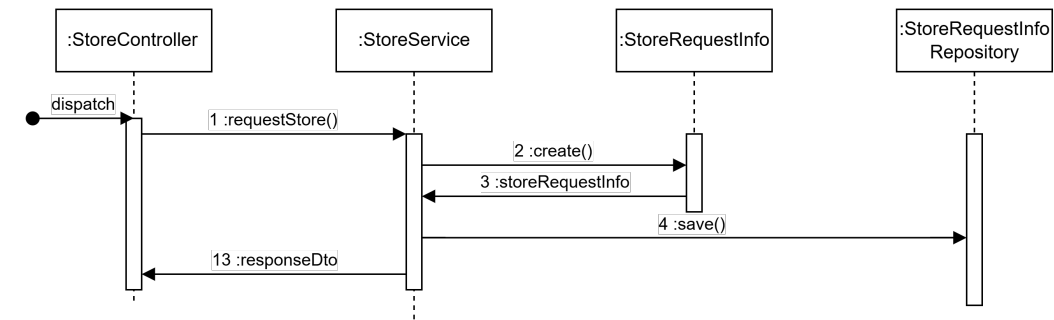
각 마이크로 서비스 내의 이벤트 publish, subscribe 구조에 적용하여 상호작용에 유연하게 대응할 수 있도록 하였다.

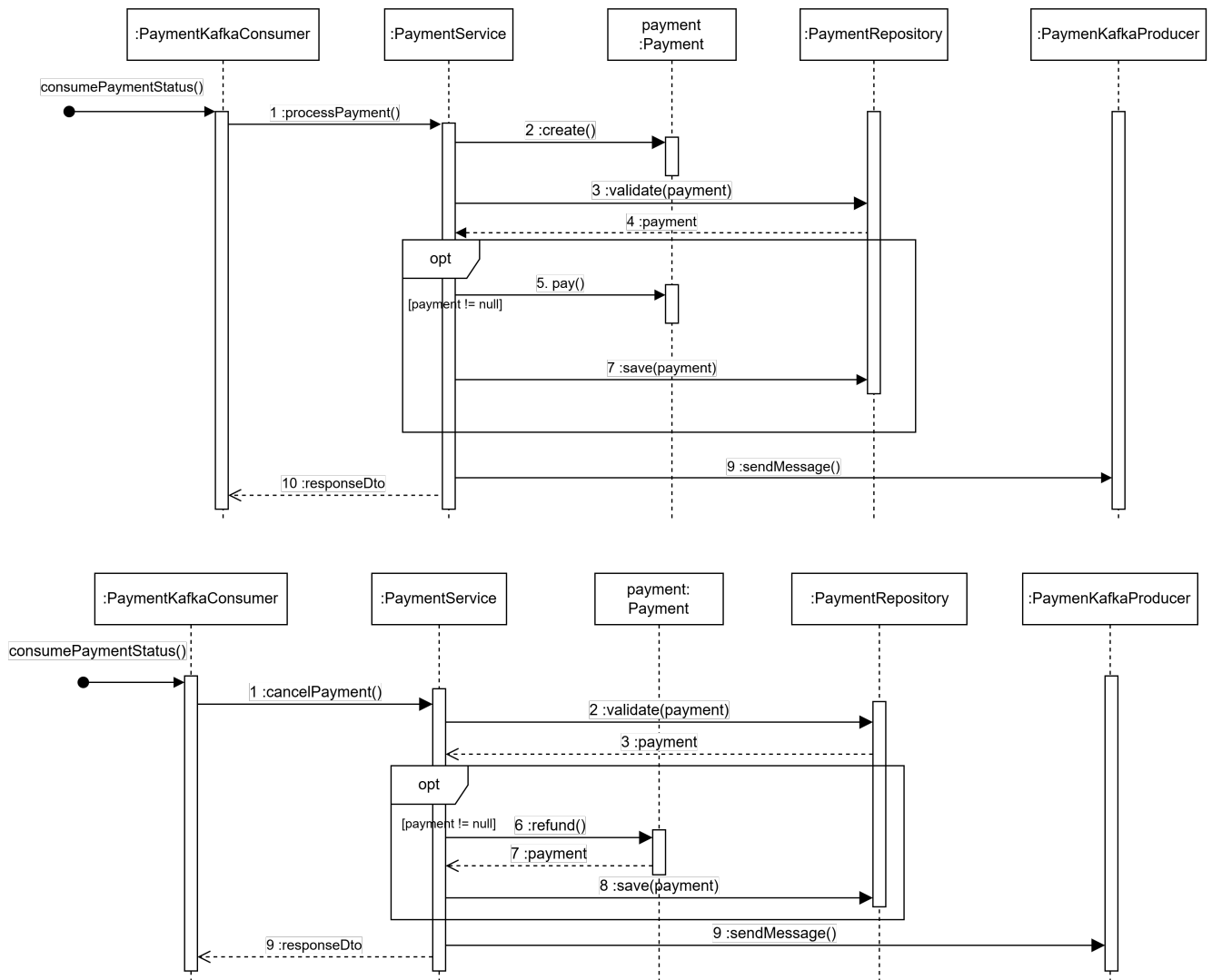
6.4.4. Proxy Design pattern

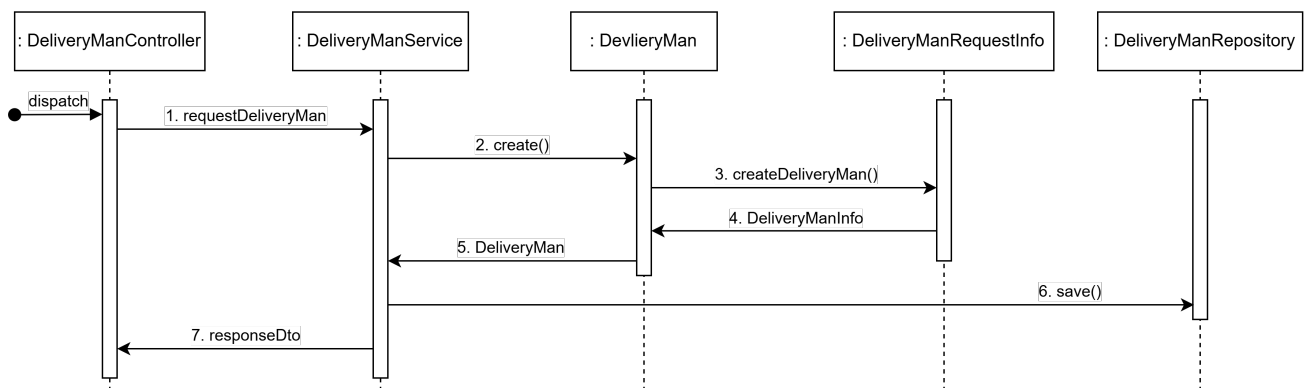
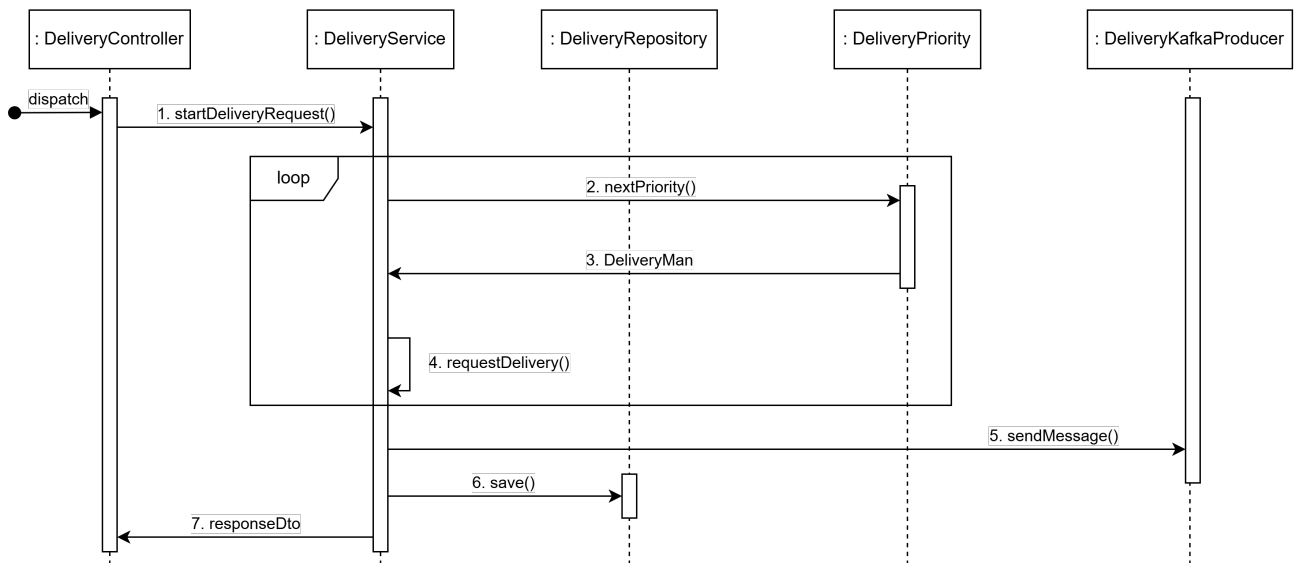
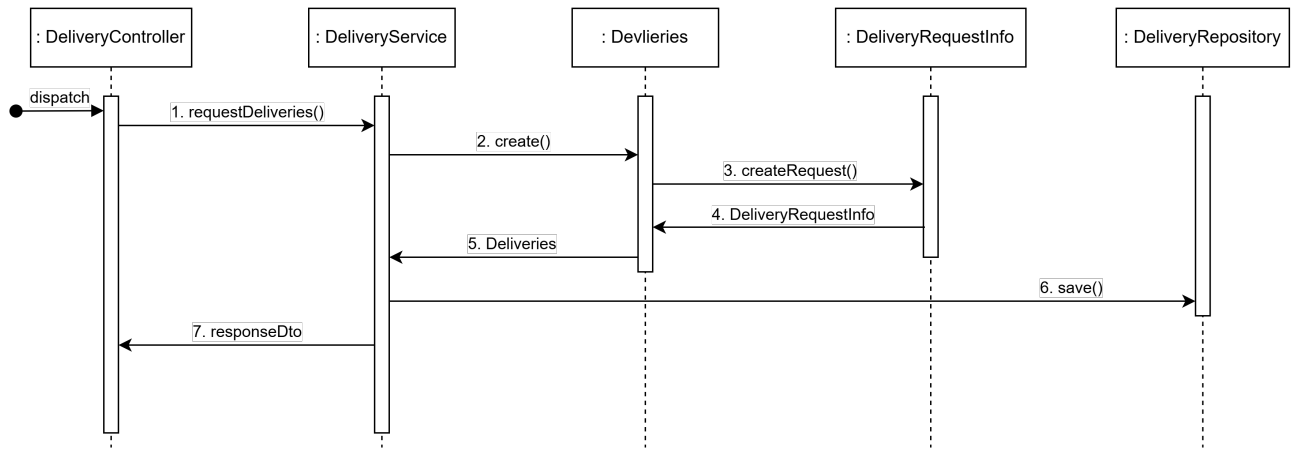
SQL을 직접 작성하지 않고 Spring Data JPA를 활용하여 자동으로 생성되는 레파지토리를 사용하였다.

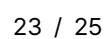
6.5. Sequence Diagram



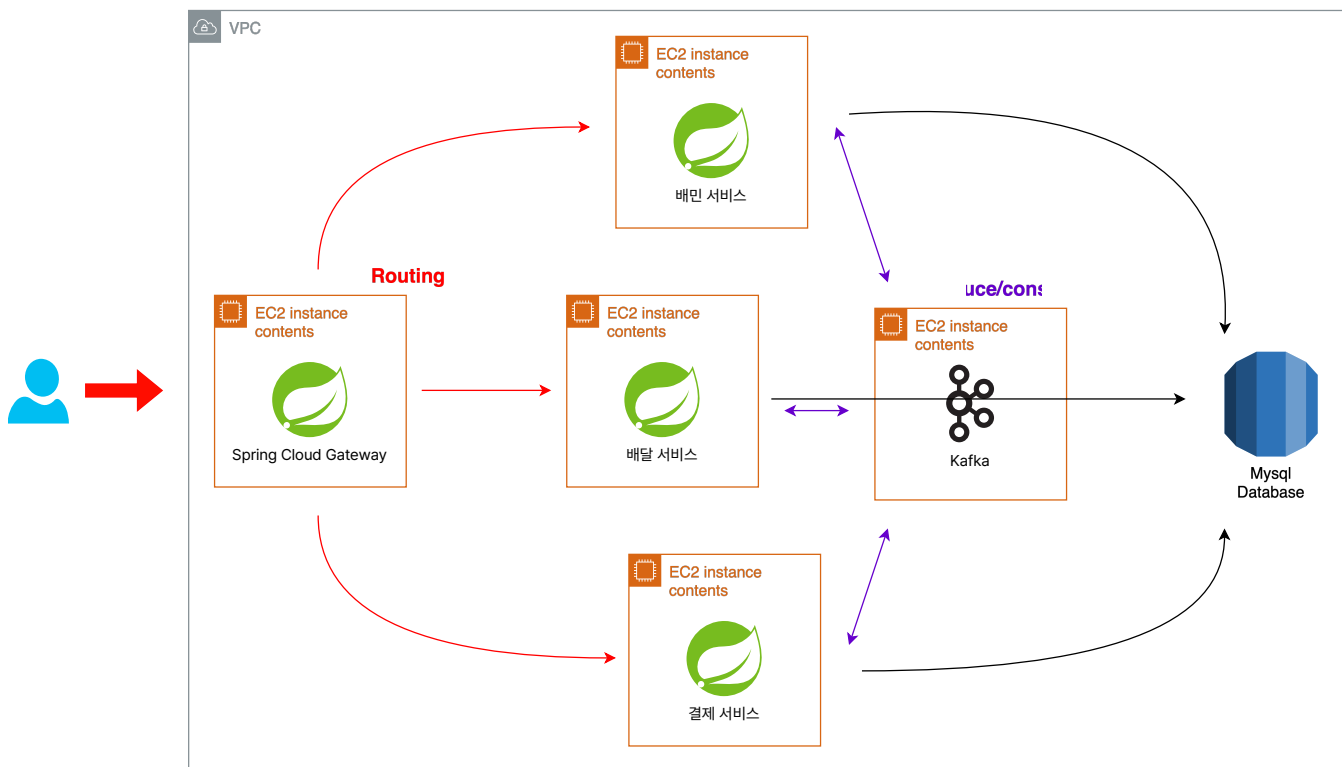








6.7. Deployment Diagram



7. Alternative Solutions + Project Justification

- <https://www.youtube.com/watch?v=704qQs6KoUk> (대규모 트랜잭션을 처리하는 배민 주문시스템 규모에 따른 진화)
- <https://techblog.woowahan.com/7835/> (이벤트 기반 아키텍처 설계)

8. Risk Analysis + Risk Reduction Plan

8.1. Spring 사용

Spring에 대한 사용 경험이 부족하여 디테일한 측면에서의 구현이 어려울 수 있다. 다른 여러 Spring 프로젝트의 구조를 참고하여 보고, 관련 공부를 진행하여 부족한 부분을 채우고자 한다.

8.2. Kafka 사용

Kafka에 대한 사용 경험이 없어 사용에 있어 어려움을 겪을 수 있다. 공식 문서인 <https://kafka.apache.org/20/documentation.html>를 활용하여 프로젝트에 적용해보고자 한다.

9. Success Criteria

9.1. 전체 서비스

- 사용자의 조회 및 요청을 시작으로 각 객체의 승인 여부에 따른 시행 통과

해당 시행 과정에서 예외처리가 모두 진행되어야 하며 올바른 결과를 출력해야 한다. 즉, JUnit 프레임워크를 사용하여 제작한 단위 테스트를 모두 통과해야 한다.

9.2. 배민 서비스

1. 고객이 '배민 서비스'를 통해 주문을 요청했을 경우 결과를 조회할 수 있어야 한다.
2. 가게는 '배민 서비스'를 통해 고객이 요청한 주문을 조회하고 수락, 거절할 수 있어야 한다.

9.3. 배달 대행 서비스

1. 가게 측에서 배달 요청을 받은 경우, 해당 서비스와 연결된 배달 기사 중 우선순위가 높은 배달 기사부터 차례대로 배달 요청을 전송해야 한다.
2. 배달 기사가 배달 요청을 승인하면, 해당 배달 기사에 대한 정보를 가게에 전송해주어야 한다.

9.4. 결제 서비스

1. 결제 요청을 받았을 때 해당 결제 수단에 대한 결제 방법으로 결제를 완료해야 한다.
2. 결제 완료 시에 성공 여부에 대한 정보를 전송해야 한다.

10. 역할 분담

이름	역할
곽유나	배민 서비스 구현, 아키텍처 설계
윤찬규	배달 대행 서비스 구현, 아키텍처 설계
이건	결제 서비스 구현, 아키텍처 설계

11. 스케줄

11.25 11.30: 중간 발표 11.28 12.08: 구현 12.16 12.21: 최종 발표

12. 참고자료 리스트

카프카 공식 문서: <https://kafka.apache.org/documentation/>