

<SE-후디>

<미스터 대박 디너 서비스>

설계 산출물

문서버전	1.4
문서 ID	SE-2022-002
최종변경일	2022-10-16
문서상태	릴리즈

요 약

미스터 대박 디너 서비스의 설계 산출물을 기술.
서브 시스템의 구성과 각 서브 시스템의 구조를 기술.
주요 산출물

- ✓ 아키텍처도
- ✓ 클래스도
- ✓ 교류도
- ✓ 상태도

표 1 문서 변경 기록

문서 이름		<SE 후디> <미스터 대박 디너 서비스> 설계 산출물	
문서 ID		SE-2022-002	
버전		변경일	설명
1	0	2022-10-12	7 개의 유즈케이스에 대한 교류도를 작성하였다. <ul style="list-style-type: none"> ✓ SignUp ✓ Login ✓ Order ✓ Modify Info ✓ Modify Order ✓ Process Order ✓ Process Items
	1	2022-10-13	Order 의 객체 상태도를 추가하였다.
	2	2022-10-14	5 개의 패키지에 대한 아키텍처도를 작성하였다. <ul style="list-style-type: none"> ✓ Interface ✓ Controller ✓ Service ✓ DAO ✓ DTO
	3	2022-10-15	패키지에 따른 클래스도를 작성하였다.
	4	2022-11-26	아키텍처도, 클래스도, 교류도, 객체상태도를 수정하였다.

1 개 요

1.1 목 적

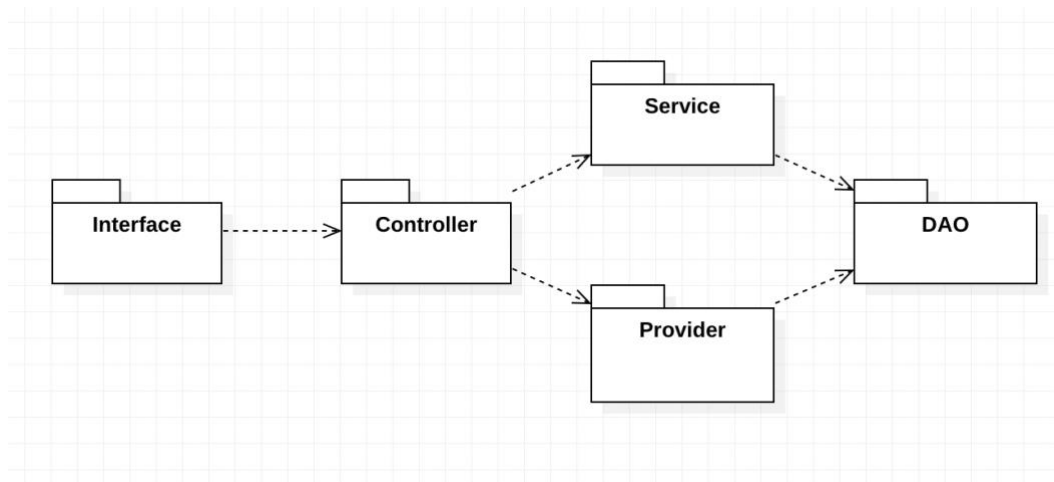
본 문서는 미스터 대박 디너 서비스의 설계 산출물을 기술한다.

- ✓ 전체 시스템이 어떤 서브 시스템으로 구성되어 있는지 “아키텍처도”를 통해 보여준다.
- ✓ 서브 시스템이 어떤 클래스로 구성되어 있고 클래스들간에 어떤 관계가 있는지를 “클래스도”를 통해서 보여준다.
- ✓ 객체간에 발생하는 동적인 행위를 “교류도”를 통해서 보여준다.
- ✓ 한 객체의 상태 변화를 “상태도”를 통해서 보여준다.

1.2 참고 문헌

없음.

2. 아키텍처도



전체 시스템은 아래의 서브 시스템으로 구성된다.

- Interface 시스템
- Controller 시스템
- Service 시스템
- Provider 시스템
- DAO 시스템

Interface 시스템

- 사용자와 상호작용을 하는 시스템으로, 사용자의 입력을 시스템에 전달한다. 시스템의 출력을 사용자 인터페이스를 통해 보여준다.
- Html, css, javascript 언어를 사용하여 작성된다.

Controller 시스템

- Interface 시스템으로부터 데이터를 받아 Service에게 전달한다. Service로부터 처리된 내용을 Interface에게 전달한다.
- Java 언어를 사용하여 작성된다.

Service 시스템

- Controller 시스템에서 보내온 요청을 처리하는 로직을 수행한다.
- Java 언어를 사용하여 작성된다.

DAO 시스템

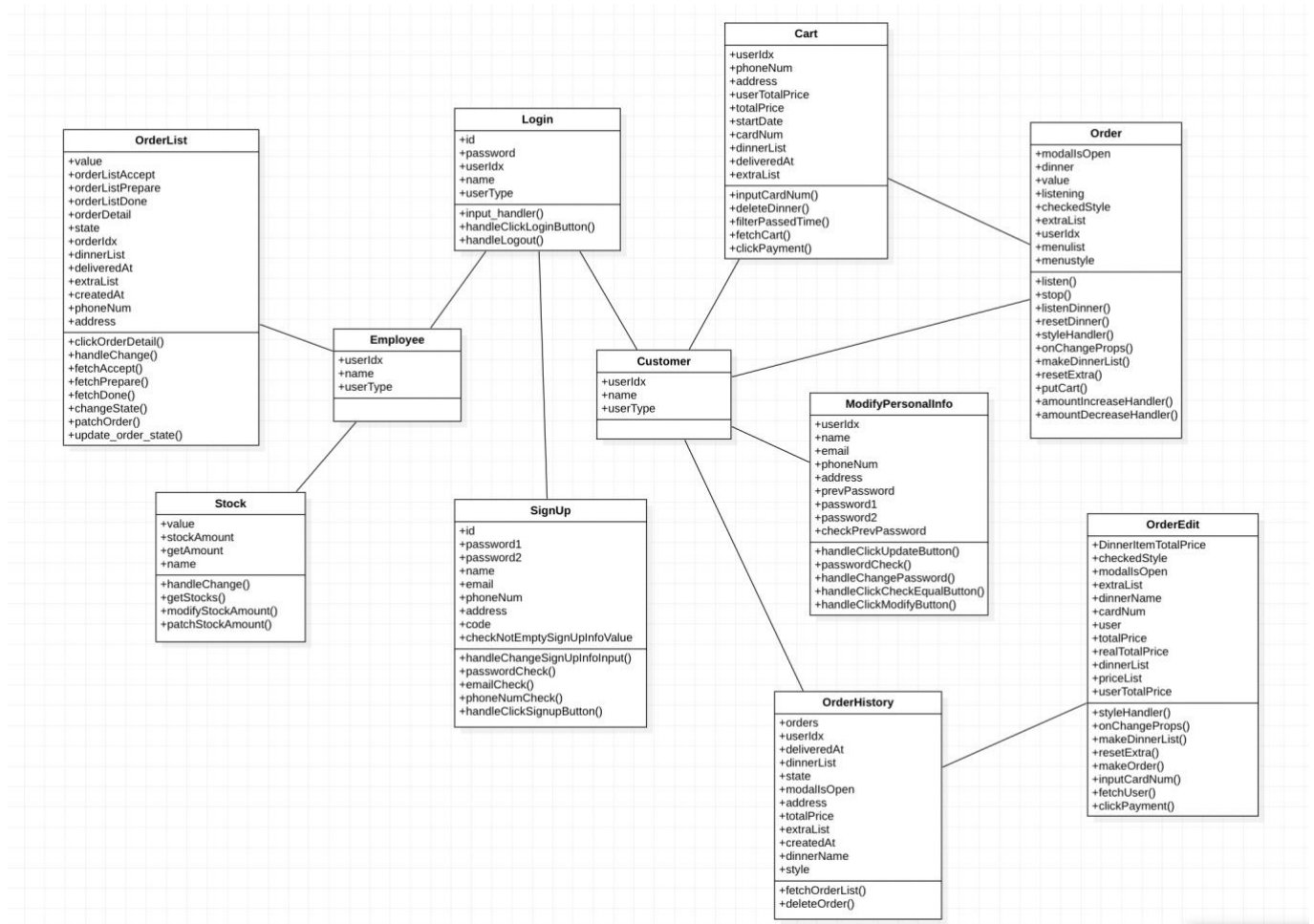
- 실제 데이터베이스에 접근하여 Service 시스템과 연결해주는 역할을 한다.
- Java, SQL 언어를 사용하여 작성된다.

Provider 시스템

- Controller 시스템에서 보내온 GET 요청을 처리하는 로직을 수행한다.
- Java 언어를 사용하여 작성된다.

3. 메인 클래스도

3.1 Interface 클래스도



SignUp

- customer 의 경우 id, password1, password2, email, phoneNum, name, address 을 입력하여 회원가입을 진행한다.
- employee 의 경우 id, password1, password2, name 을 입력하고, 직원코드를 입력하면 회원가입이 완료된다.

Login

- customer 로 로그인하면 main 페이지로 이동한다.
- employee 로 로그인하면 employee 페이지로 이동한다.

Order

- customer 는 메뉴, 옵션 종류와 수량, 스타일을 선택하고 장바구니 담기를 선택한다.
- 음성인식을 통해 메뉴를 선택하여 해당 메뉴 선택 모달창을 띄워준다.

Cart

- order 페이지에서 장바구니 담기를 선택한 메뉴들을 볼 수 있다. 메뉴 삭제가 가능하다.
- user 정보에 저장된 주소와 번호를 불러오고, 날짜와 카드번호를 입력한 후 결제를 진행한다.

ModifyPersonalInfo

- 이름, 아이디, 비밀번호, 이메일, 전화번호, 회원 등급, 주소를 보여준다.
- 현재 비밀번호가 맞는지 확인한 후 비밀번호를 수정할 수 있다.
- 이메일, 전화번호, 주소를 수정할 수 있다.

OrderHistory

- 지금까지의 주문 내역을 볼 수 있다. 주문 상태, 주문 번호, 메뉴, 배달 시간 등 상세 정보를 확인할 수 있다.

OrderEdit

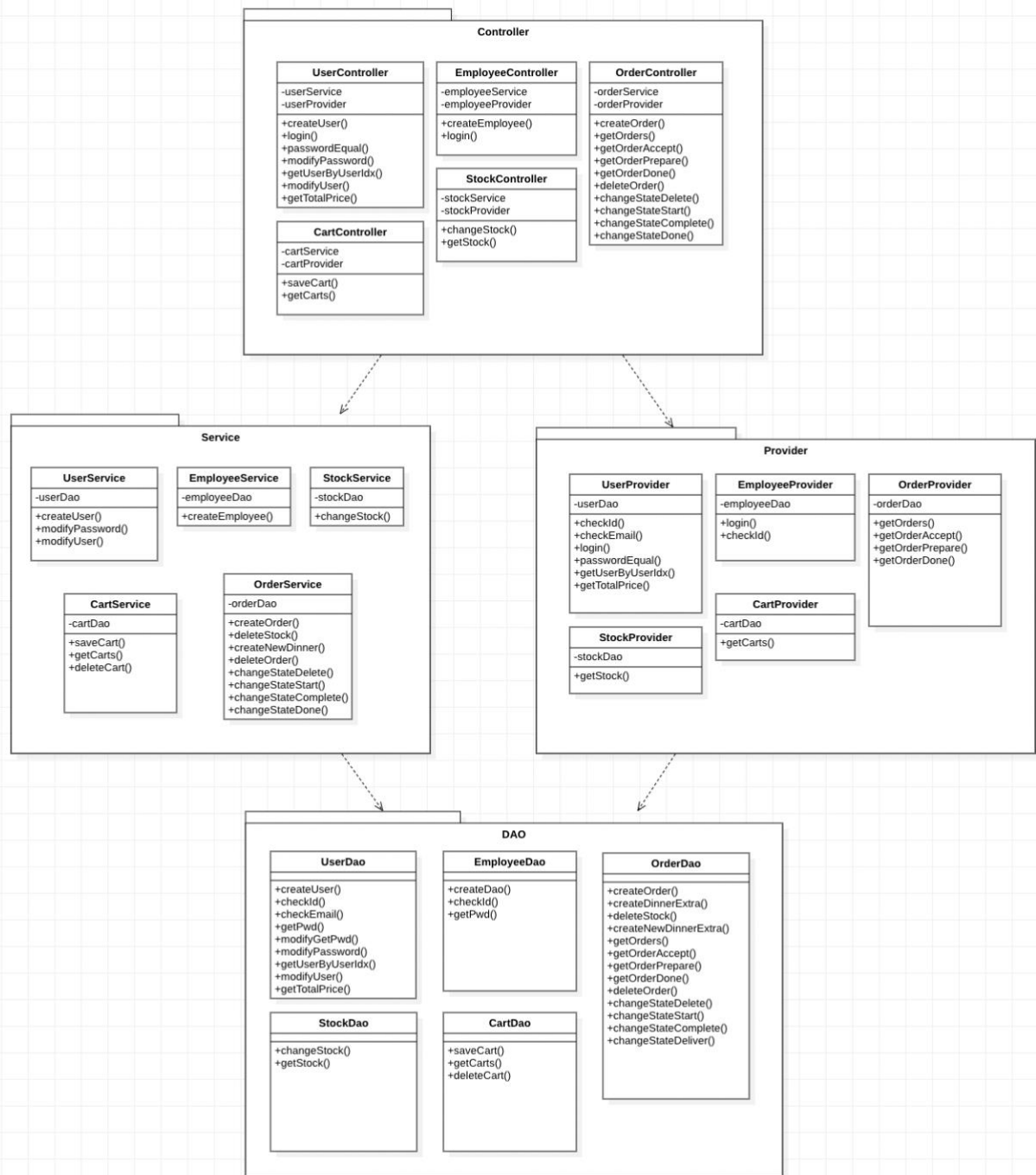
- ‘주문 대기’ 상태의 주문에 대한 메뉴 옵션 정보를 수정할 수 있다. 옵션을 추가하거나 삭제를 할 수 있다.
- 신용카드번호를 입력받아 수정된 주문내역에 대해 결제를 다시 진행한다.

OrderList

- 고객으로부터 들어온 주문 정보를 확인하는 화면을 보여준다.
- 주문 내역, 배송 정보, 예약 시간, 요청 사항 등 주문 관련 정보를 가져와 보여주고 버튼을 클릭해 주문의 state (승인, 조리, 픽업, 배달)를 업데이트한다.

Stock

- 현재 재고 수량을 확인하고 추가 수량을 입력하여 재고 수량을 수정한다.



3.2 Controller 클래스도

UserController

- customer 의 마이페이지의 회원정보 수정, 로그인, 회원가입과 같은 유저 정보와 관련된 요청을 처리한다.
- userService 를 속성으로 가지고, service 에서 처리한 내용을 인터페이스에 넘겨준다.

EmployeeController

- employee 의 로그인, 회원가입 유저 정보와 관련된 요청을 처리한다.
- employeeService 를 속성으로 가지고, service 에서 처리한 내용을 인터페이스에 넘겨준다.

OrderController

- 주문 생성, 주문 조회, 주문 변경 등 주문과 관련된 정보에 대한 요청을 처리한다.
- `orderService` 를 속성으로 가지고, 각각 `service` 에서 처리한 내용을 인터페이스에 넘겨준다.

StockController

- 재고 수량을 불러오고 수정하는 요청을 처리한다.
- `StockService` 를 속성으로 가지고, `service` 에서 처리한 내용을 인터페이스에 넘겨준다.

CartController

- 장바구니 추가 및 삭제 요청을 처리한다.
- `CartService` 를 속성으로 가지고, `service` 에서 처리한 내용을 인터페이스에 넘겨준다.

3.3 Service 클래스도

UserService

- 회원가입과 로그인 등 `user` 와 관련된 서비스를 제공하는 객체이다.
- `UserController` 로부터 메시지를 받아서 `DAO` 를 호출하여 회원가입, 로그인, 등급 업데이트 등을 수행한다.

EmployeeService

- 직원 계정을 관리하는 객체이다.
- `EmployeeController` 로부터 메시지를 받아 `DAO` 를 호출하며 직원 계정 생성, 불러오기, 유효성 검사를 수행한다.

OrderService

- 주문과 관련된 서비스를 제공하는 객체이다.
- `OrderController` 로부터 메시지를 받아 `DAO` 를 호출하며 주문 정보를 불러오거나 주문 상태를 변경하는 등의 기능을 수행한다.

StockService

- 재고확인과 변경을 담당하는 객체이다.
- `StockController` 로부터 메시지를 받아 `DAO` 를 호출하는 방식으로 기능을 수행하며 재고 변경과 불러오기 기능을 수행한다.

CartService

- 장바구니 추가와 삭제를 담당하는 객체이다.
- `CartController` 로부터 메시지를 받아 `DAO` 를 호출하는 방식으로 기능을 수행하면 장바구니 추가와 삭제 기능을 수행한다.

3.4 Provider 클래스도

UserProvider

- 로그인, 회원 정보 조회 등 user 와 관련된 서비스를 제공하는 객체이다.
- UserController로부터 메시지를 받아서 DAO 를 호출하여 회원가입, 로그인, 등급 업데이트 등을 수행한다.

EmployeeProvider

- 직원 계정을 관리하는 객체이다.
- EmployeeController로부터 메시지를 받아 DAO 를 호출하며 직원 계정 생성, 불러오기, 유효성 검사를 수행한다.

OrderProvider

- 주문과 관련된 서비스를 제공하는 객체이다.
- OrderController로부터 메시지를 받아 DAO 를 호출하며 주문 정보를 불러오거나 주문 상태를 변경하는 등의 기능을 수행한다.

StockProvider

- 재고확인을 담당하는 객체이다.
- StockController로부터 메시지를 받아 DAO 를 호출하는 방식으로 기능을 수행하며 재고 변경과 불러오기 기능을 수행한다.

CartProvider

- 장바구니 조회를 담당하는 객체이다.
- CartController로부터 메시지를 받아 DAO 를 호출하는 방식으로 기능을 수행하면 장바구니 추가와 삭제 기능을 수행한다.

3.5 DAO 클래스도

UserDao

- 고객과 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- 데이터베이스 연결을 위한 attribute 들이 존재하고 user 클래스를 이용한다.
- insert, select, update, delete 의 메소드를 통해 삽입, 조회, 수정, 삭제의 기능을 수행한다

EmployeeDao

- 직원과 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- 데이터베이스 연결을 위한 attribute 들이 존재하고 employee 클래스를 이용한다.
- insert, select, update, delete 의 메소드를 통해 삽입, 조회, 수정, 삭제의 기능을 수행한다

OrderDao

- 주문과 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- 데이터베이스 연결을 위한 attribute 들이 존재하고, order, dinner, dinnerList, extra, extraList, cart 클래스를 활용한다.

- insert, select, update, delete 의 메소드를 통해 삽입, 조회, 수정, 삭제의 기능을 수행한다.

StockDao

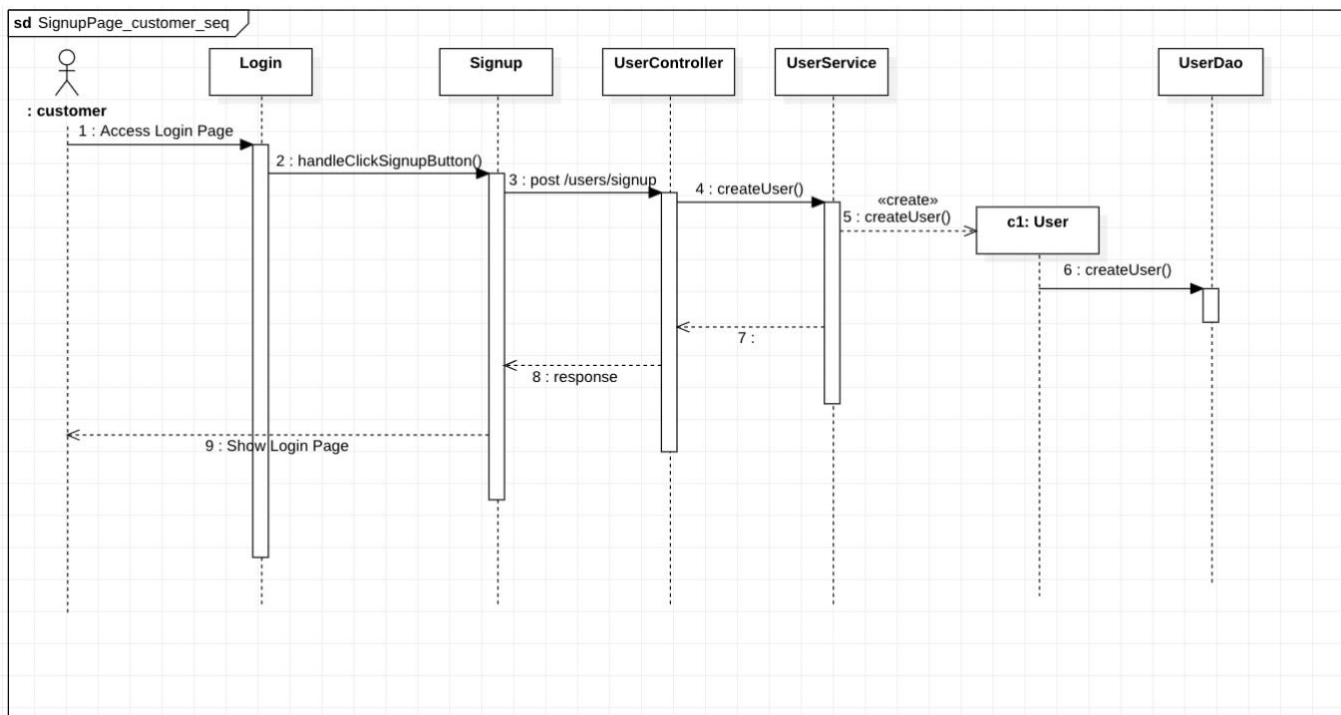
- 재고와 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- 데이터베이스 연결을 위한 attribute 들이 존재하고 패키지의 stock, user 클래스를 이용한다.
- insert, select, update, delete 의 메소드를 통해 삽입, 조회, 수정, 삭제의 기능을 수행한다

CartDao

- 장바구니와 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- 데이터베이스 연결을 위한 attribute 들이 존재하고, cart, user 클래스를 활용한다.
- insert, select, update, delete 의 메소드를 통해 삽입, 조회, 수정, 삭제 기능을 수행한다.

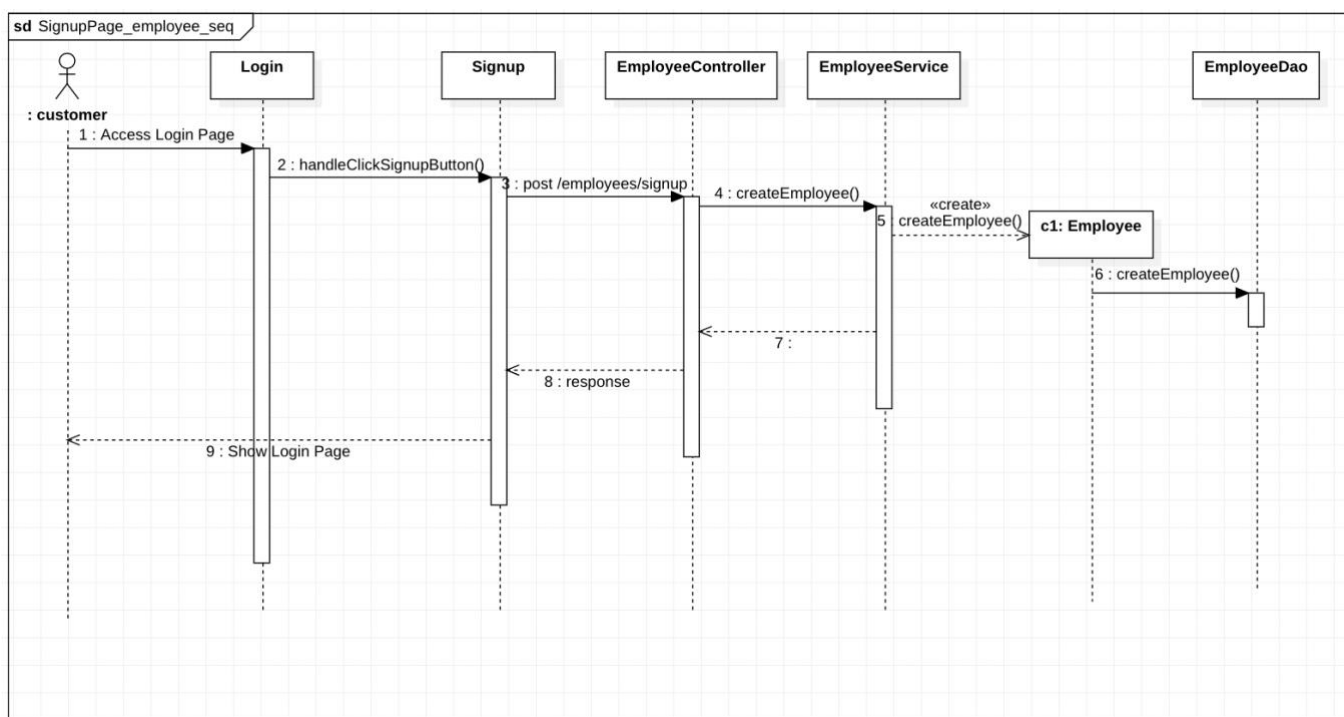
4.1 교류도(usecase : SignUp 회원가입)

고객 회원가입



1. 직원이 로그인페이지에 접근한다.
2. 로그인 페이지에서 회원가입 버튼을 눌러 회원가입 페이지로 접근한다.
3. 회원정보를 입력한 후 회원가입 버튼을 누른다.
4. post 메소드를 호출하여 회원가입 api 를 호출한다.
5. createUser 메소드를 호출한다.
6. 데이터 베이스에서 고객 객체를 생성한다.
7. 회원가입 후 로그인 페이지로 이동한다.

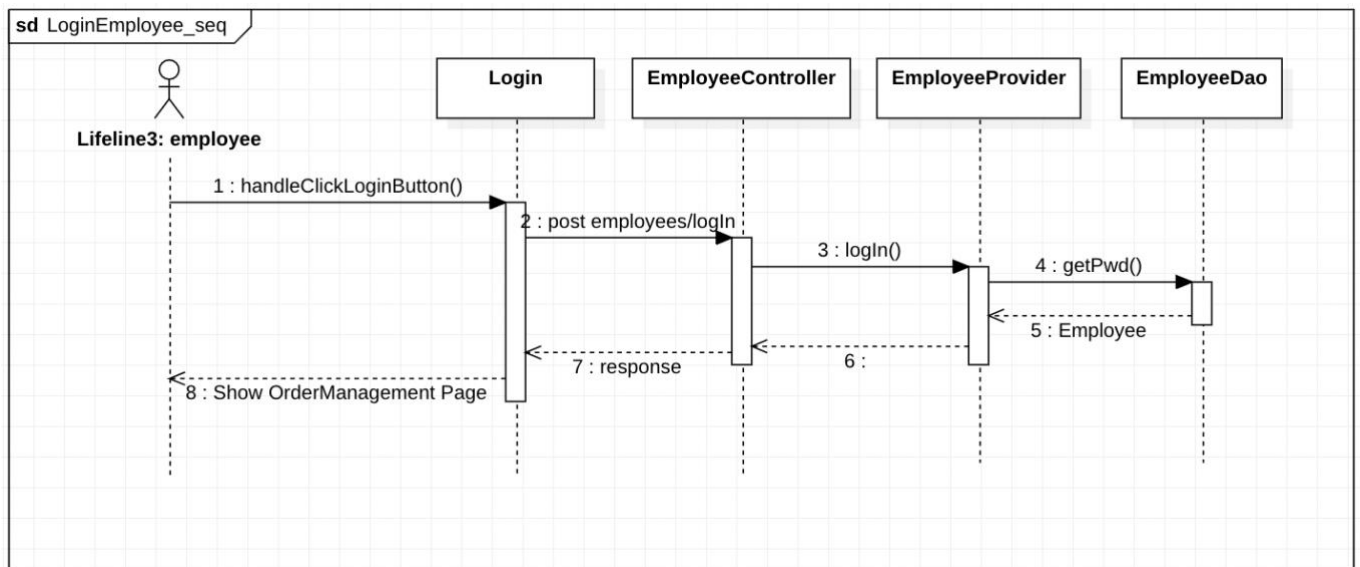
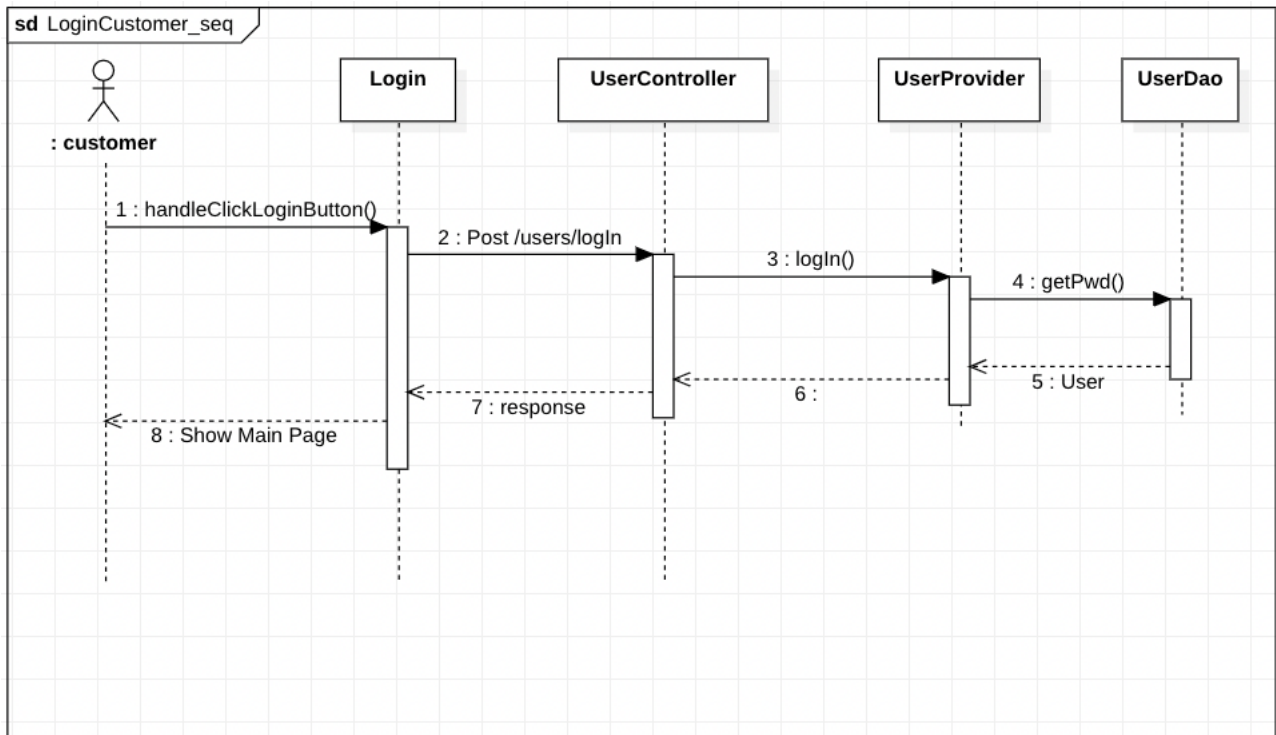
직원 회원가입



1. 직원이 로그인페이지에 접근한다.
2. 로그인 페이지에서 회원가입 버튼을 눌러 회원가입 페이지로 접근한다.
3. 회원정보를 입력한 후 회원가입 버튼을 누른다.
4. post 메소드를 호출하여 회원가입 api 를 호출한다.
5. createEmployee 메소드를 호출한다.
6. 데이터 베이스에서 직원 객체를 생성한다.
7. 회원가입 후 로그인 페이지로 이동한다.

4.2 교류도(usecase : Login 로그인)

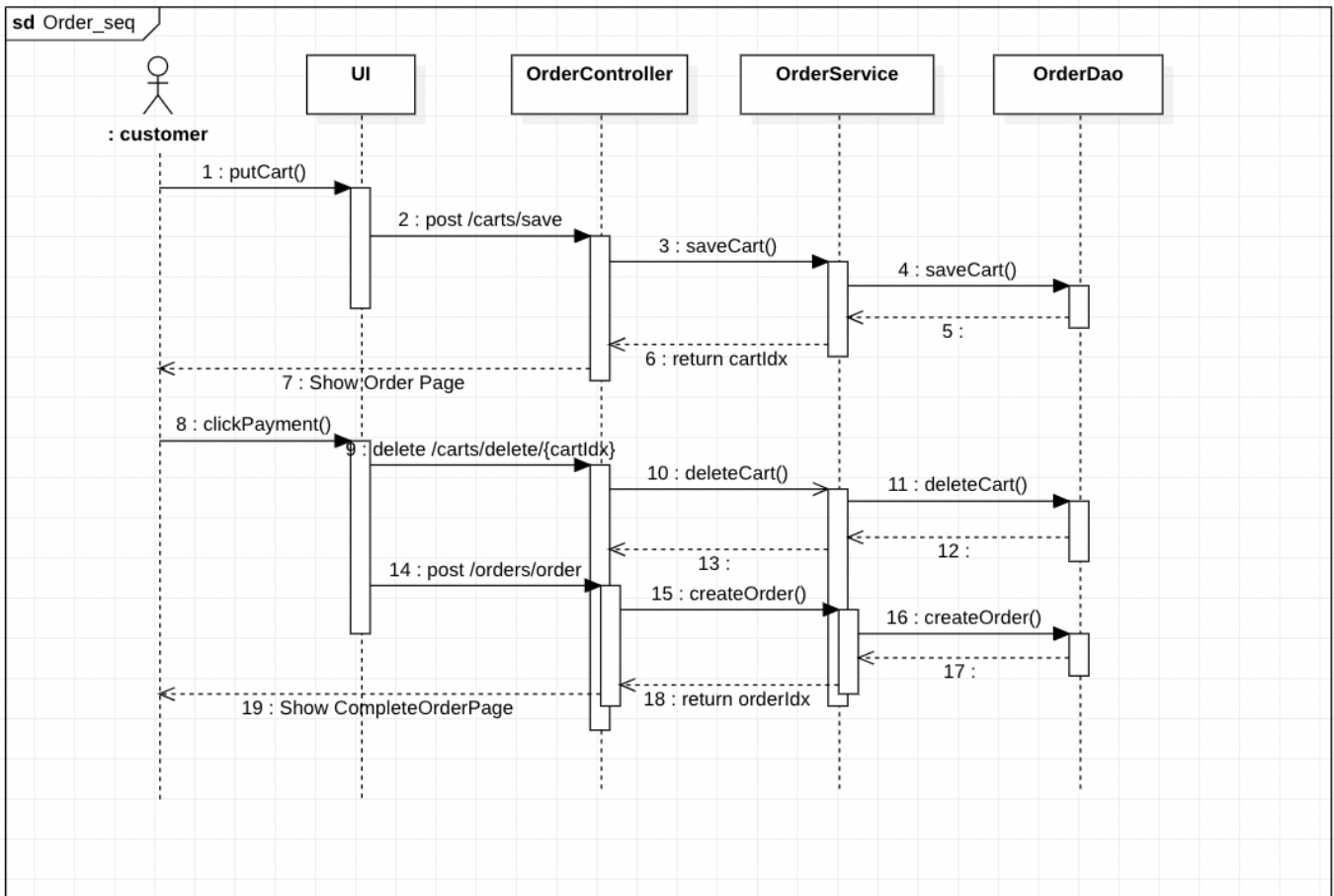
로그인



1. 유저가 로그인 페이지에 접근한다.
2. 로그인 버튼을 눌러 Post 메소드를 호출하여 로그인 Api 를 호출한다.
3. 로그인 함수를 호출한다.
4. 데이터 베이스에서 비밀번호를 확인하고 고객 or 직원 객체를 생성한다.
5. 생성된 객체를 반환한다.

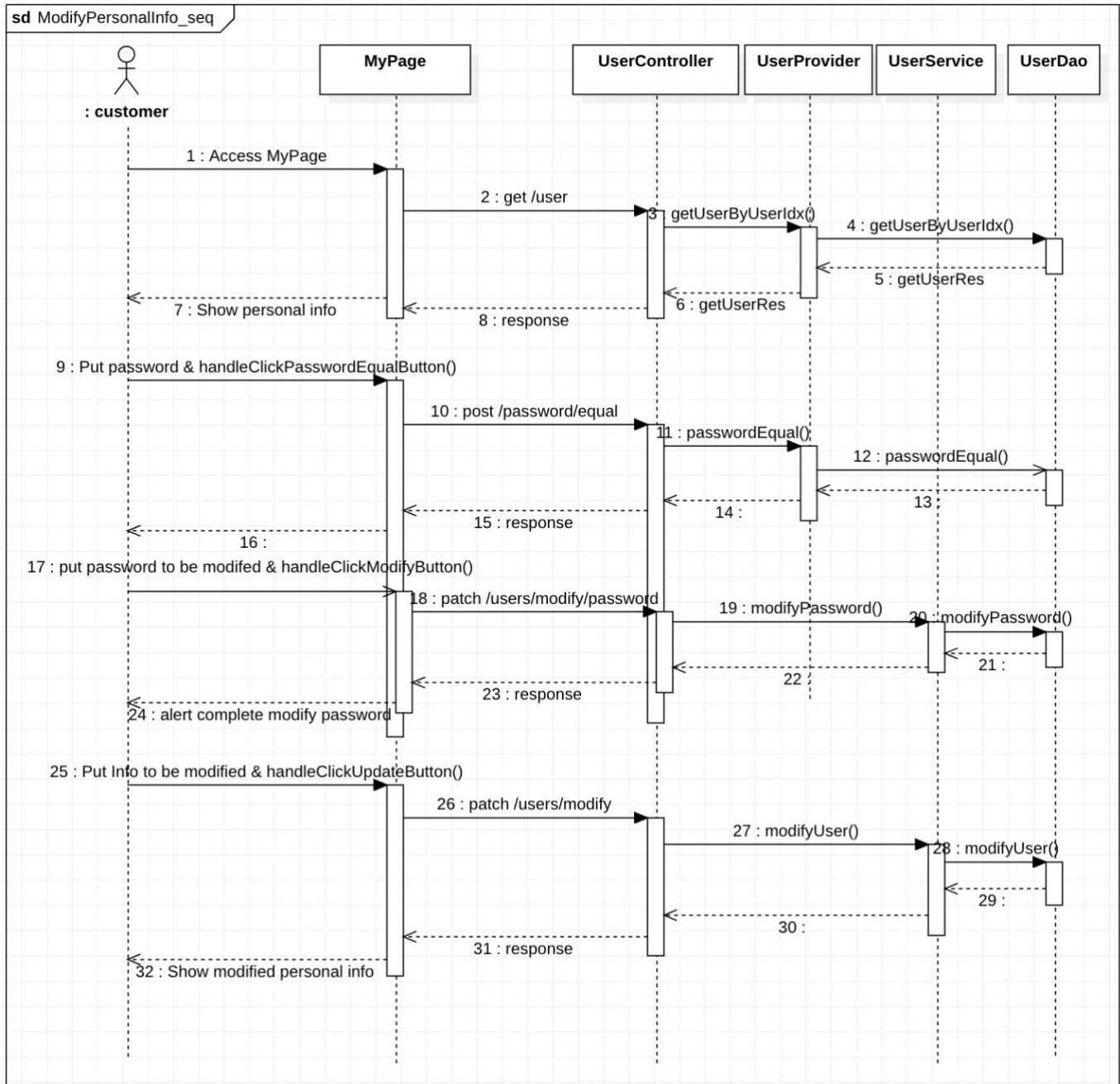
6. 응답
7. 로그인 페이지를 보여준다.

4.3 교류도(usecase : Order 주문하기)



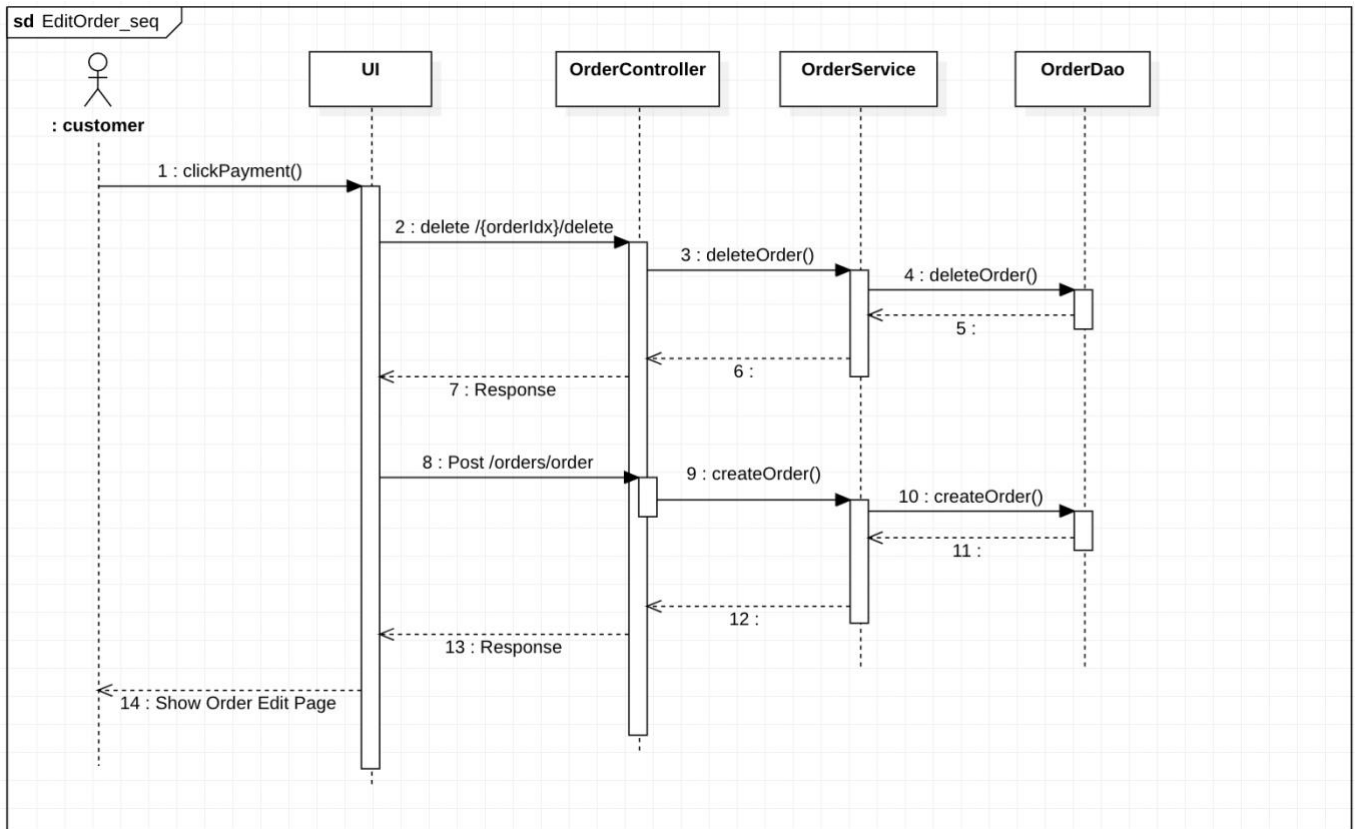
1. 고객이 디너 종류, 옵션, 스타일을 선택하고 “장바구니 담기” 버튼을 누른다.
2. 고객이 입력한 정보를 가지고, UI가 OrderController에게 post 요청을 보낸다.
3. Controller가 Service에게 장바구니에 주문 정보를 저장하는 saveCart 메소드를 호출한다
4. Service가 Dao에게 데이터베이스에서 정보 저장을 요청하는 메소드를 호출한다.
5. cartIdx를 반환한다
6. 고객이 장바구니 페이지에서 정보를 입력하고 “결제하기” 버튼을 누른다.
7. OrderController에게 delete 요청을 보낸다.
8. Controller가 Service에게 현재 장바구니에 담긴 주문 정보를 삭제하는 deleteCart 메소드를 호출한다.
9. Service가 Dao에게 데이터베이스에서 정보 삭제를 요청하는 메소드를 호출한다.
10. 고객이 입력한 정보를 가지고, UI가 OrderController에게 post 요청을 보낸다.
11. Controller가 Service에게 주문을 처리하는 createOrder 메소드를 호출한다.
12. Service가 Dao에게 데이터베이스에서 정보 저장을 요청하는 메소드를 호출한다.
13. orderIdX를 반환한다.
14. 고객에게 주문 완료 페이지를 보여준다.

4.4 교류도(usecase : Modify Info 개인정보 수정)



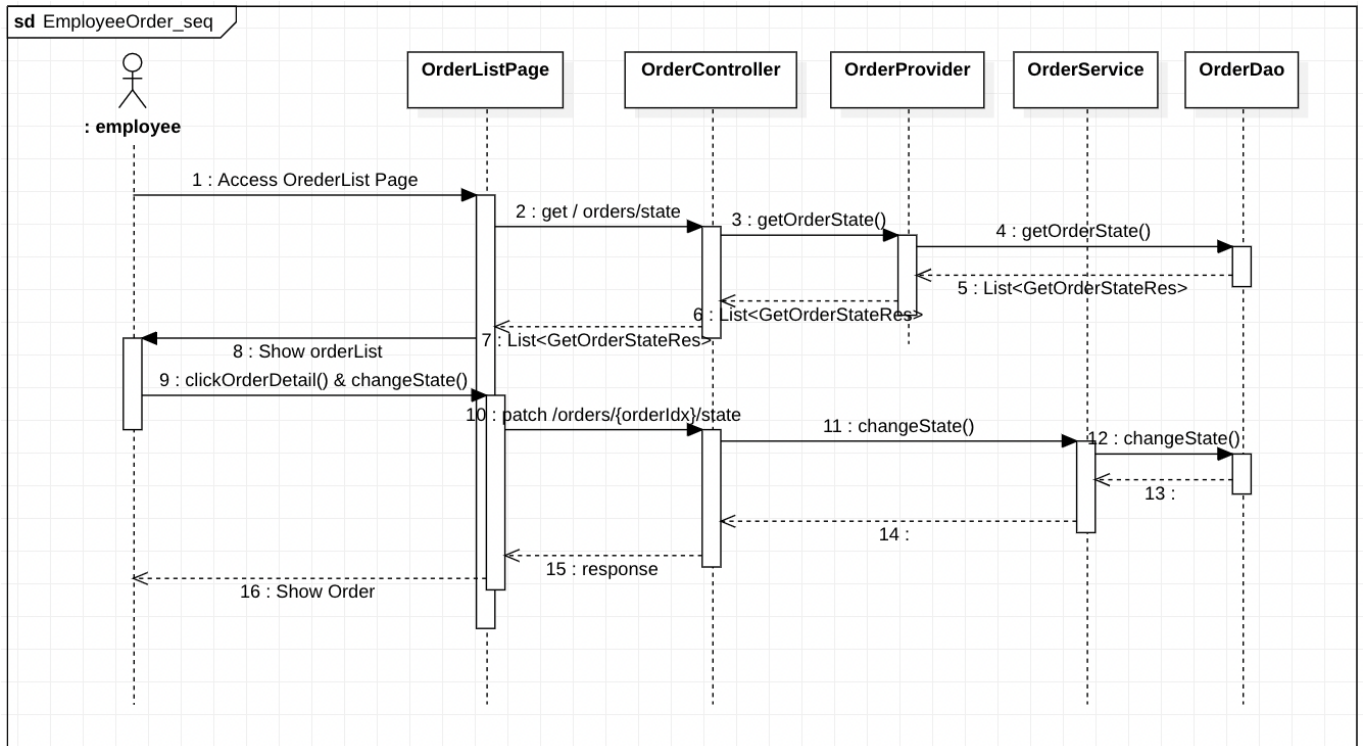
1. 고객이 마이페이지에 접근한다.
2. userIdx 를 이용하여 get 메소드를 호출해 user 정보를 받아온다.
3. 고객의 정보를 보여준다.
4. 현재 비밀번호를 입력하고 비밀번호 확인 버튼을 누른다.
5. post 메소드를 호출해 입력한 비밀번호가 맞는지 확인한다.
6. 수정하고자 하는 비밀번호를 입력하고 수정하기 버튼을 누른다.
7. 비밀번호를 수정하는 patch 메소드를 호출한다.
8. 비밀번호 수정 완료를 알려준다.
9. 수정하고자 하는 고객의 정보를 입력하고 수정하기 버튼을 누른다.
10. 고객의 정보를 수정하는 patch 메소드를 호출한다.
11. 유저 정보를 보여준다.

4.5 교류도(usecase : Modify Order 주문변경)



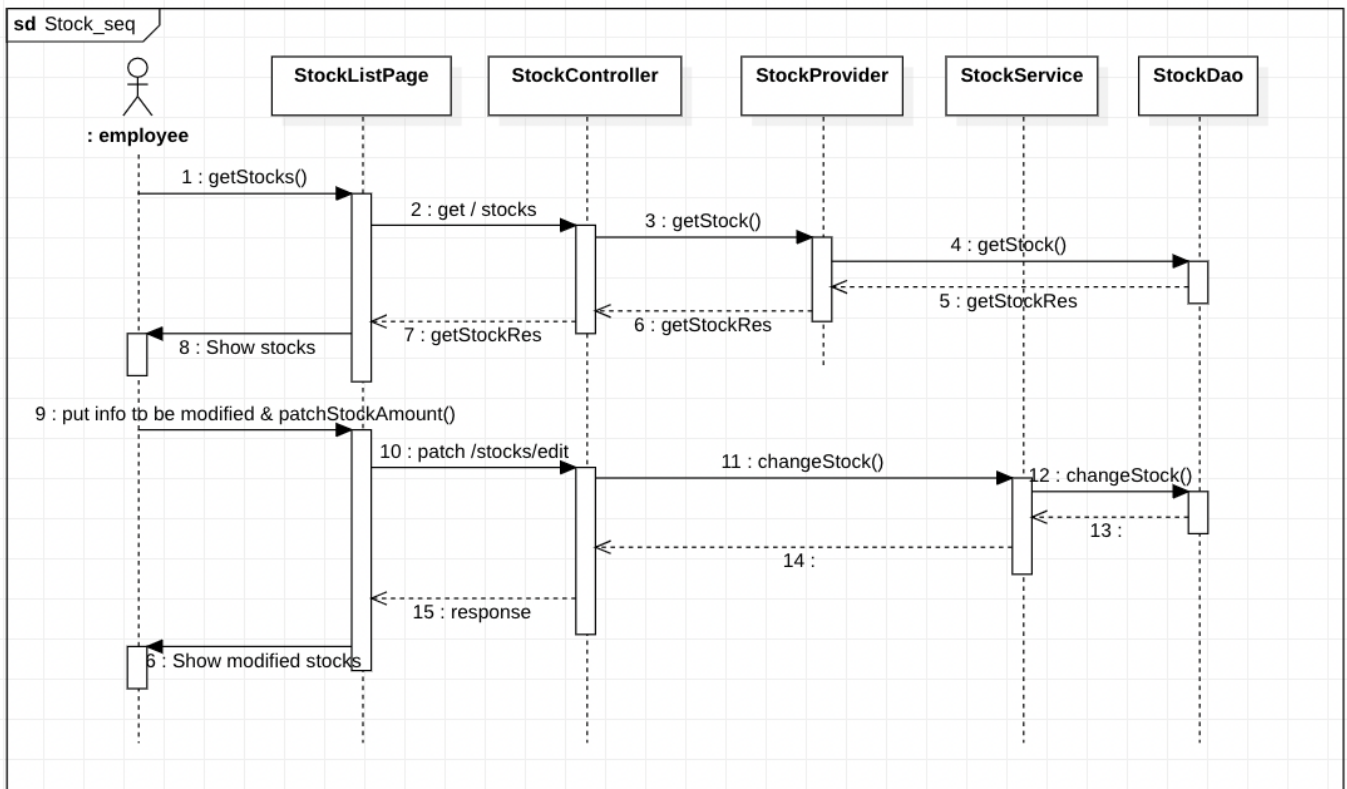
1. 고객이 옵션을 변경한 후 결제하기 버튼을 누른다.
2. orderId 를 이용하여 기존 주문을 삭제하는 delete 메소드를 호출한다.
3. Controller가 Service 에게 deleteOrder 메소드를 호출한다.
4. 데이터 베이스에서 주문 정보를 삭제한다.
5. 옵션을 변경한 주문 정보를 post 메소드를 호출하여 보낸다.
6. Controller가 Service 에게 createOrder 메소드를 호출한다.
7. 데이터 베이스에서 주문 정보를 저장한다.
8. 주문 변경 완료 페이지를 보여준다.

4.6 교류도(usecase : Process Order 주문처리)



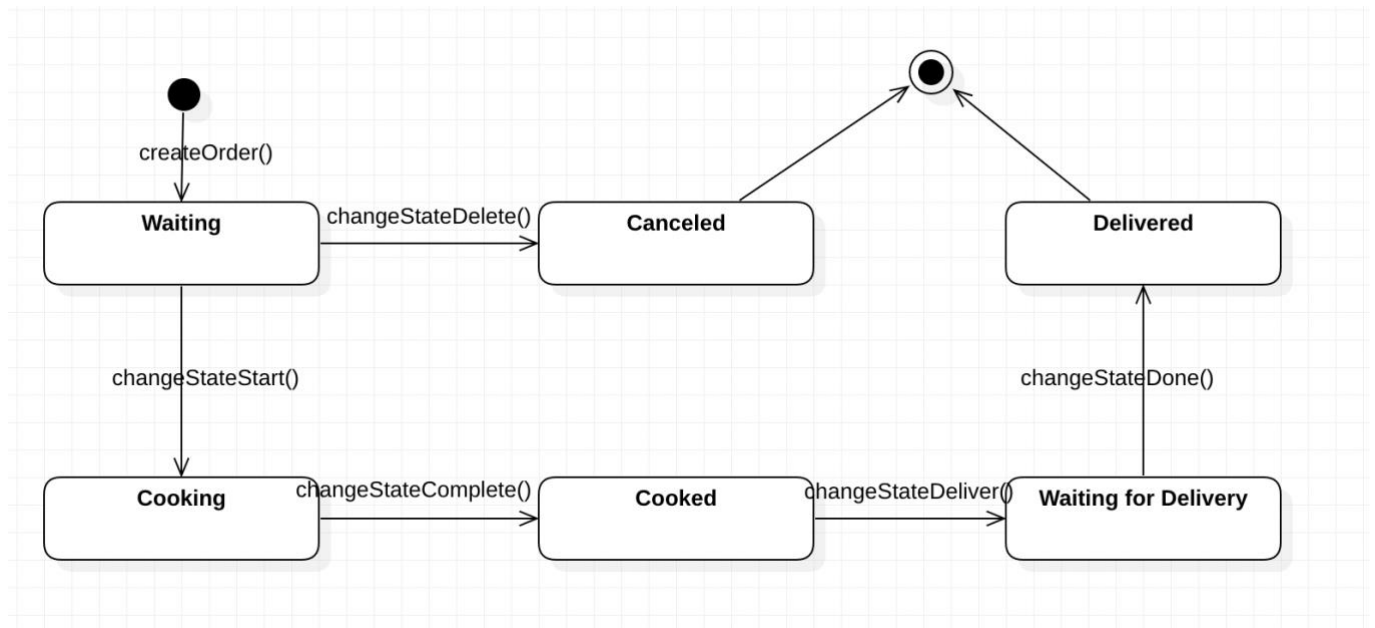
1. 직원이 주문목록 페이지에 접근한다.
2. 주문 상태별로 주문 목록을 불러오는 get 메소드를 호출한다.
3. Controller가 Provider에게 getOrderState 메소드를 호출한다.
4. 상태에 따른 주문 목록 List를 반환한다.
5. 주문 목록을 보여준다.
6. 각 상태에 해당하는 주문 상태변경 버튼을 누른다.
7. orderId를 이용하여 주문 상태를 업데이트하는 patch 메소드를 호출한다.
8. Controller가 Service에게 changeState 메소드를 호출한다.
9. 데이터베이스에 수정된 정보를 반영한다.
10. 주문 목록을 보여준다.

4.7 교류도(usecase : Process Items 재고처리)



1. 직원이 재고 확인 페이지에 접근한다.
2. 재고 수량을 불러오는 get 메소드를 호출한다.
3. Controller가 Provider에게 getStock 메소드를 호출한다.
4. 재고 수량 객체를 반환한다.
5. 재고 수량을 보여준다.
6. 직원이 추가하고자 하는 재고의 값을 입력하고 수정하기 버튼을 누른다.
7. 재고 수량을 추가해주는 patch 메소드를 호출한다.
8. Controller가 Service에게 changeStock 메소드를 호출한다.
9. 데이터베이스에 수정된 재고의 수량을 반영한다.
10. 수정된 재고 수량을 보여준다.

5. Order 객체 상태도



Order는 주문의 상태를 나타내는 클래스이다. 주문은 다음 여덟 개의 상태 중 하나를 가진다.

- **Waiting**
 - 접수가 완료된 상태이다. 예약 시간이 되어 직원이 “조리 시작” 버튼을 누르기 전까지 **Accepted** 상태가 유지된다. “요리 시작” 버튼을 누르면 **Cooking** 상태로 바뀐다.
- **Canceled**
 - **Waiting** 상태에서 직원이 “주문 취소” 버튼을 눌러, 주문이 취소된 상태이다. 이 상태에서 주문은 취소되고 Order 객체는 주문 취소 내역으로 이동한다.
- **Cooking**
 - 요리사가 요리를 진행 중인 상태이다. 직원이 “조리 완료” 버튼을 누르면 **Cooked** 상태로 바뀐다.
- **Cooked**
 - 요리사가 요리를 진행 중인 상태이다. 직원이 “조리 완료” 버튼을 누르면 **Cooked** 상태로 바뀐다.
- **Pick-Up**
 - 배달원이 픽업을 완료한 상태이다. 배달원이 “배달 완료” 버튼을 누르면 **Delivered** 상태로 바뀐다.
- **Delivered**
 - 배달원이 배달을 완료한 상태이다.
 - 배달이 완료되어 주문 전 과정이 완전히 수행된 상태이다. 완료된 Order 객체는 주문 내역으로 이동된다.
- **Done**
 - 주문 상태 관리가 끝난 상태로, 취소된 주문은 state 0 의 상태로, 배달 완료된 주문은 state 5 의 상태로 유지된다.