

<똑배기> <MrGreat> 설계 산출물

문서버전	2.0
문서 ID	SE-2020-002
최종변경일	2020-11-05
문서상태	릴리즈

요 약

미스터 대박 디너 서비스(Mr.Great)의 설계 산출물을 기술.

서브 시스템의 구성과 각 서브 시스템의 구조를 기술.

주요 산출물

- ✓ 아키텍처도
- ✓ 클래스도
- ✓ 교류도
- ✓ 상태도

표 1 문서 변경 기록

문서이름		<똑배기> <MrGreat> 설계 산출물	
문서 ID		SE-2020-002	
버전		변경일	설명
1	0	2020-11-4	아키텍처도를 추가했다. “Staff Client” 클래스도를 추가했다. “Server” 클래스도를 추가했다.
	1	2020-11-4	“Customer Client” 클래스도를 추가했다. “Customer Client” 서브시스템 세부설계 설명을 추가했다. “Staff Client” 서브시스템 세부설계 설명을 추가했다. “Server” 서브시스템 세부설계 설명을 추가했다.
	2	2020-11-5	교류도를 추가했다. 상태도를 추가했다.
2	0	2020-11-5	릴리즈

1 개 요

1.1 목 적

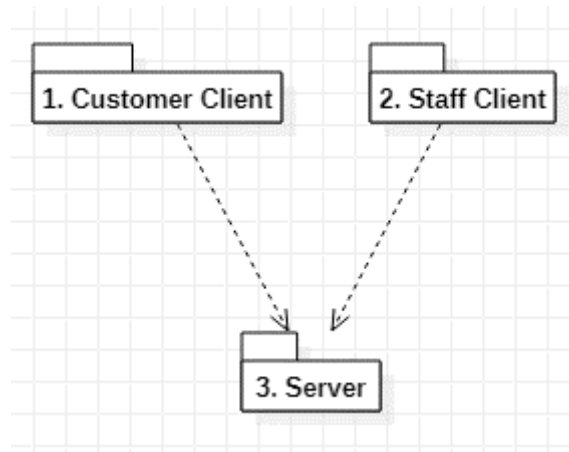
본 문서는 미스터 대박 디너 서비스(Mr.Great)의 산출물을 기술한다.

- ✓ 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.

- Customer Client 시스템
- Staff Client 시스템
- Sever 시스템

고객 Client 시스템

- 서버와 통신하며 고객에게 정보를 보여주는 역할을 한다.
- Javascript를 이용하여 작성된다.

직원 Client 시스템

- 서버와 통신하며 직원에게 정보를 보여주는 역할을 한다.
- Javascript를 이용하여 작성된다.

Server 시스템

- 메뉴, 고객 정보, 주문 정보를 담고 있으며 주문 저장 및 알림의 역할을 한다.
- Node.js를 이용하여 작성된다.

3 서브시스템 세부 설계

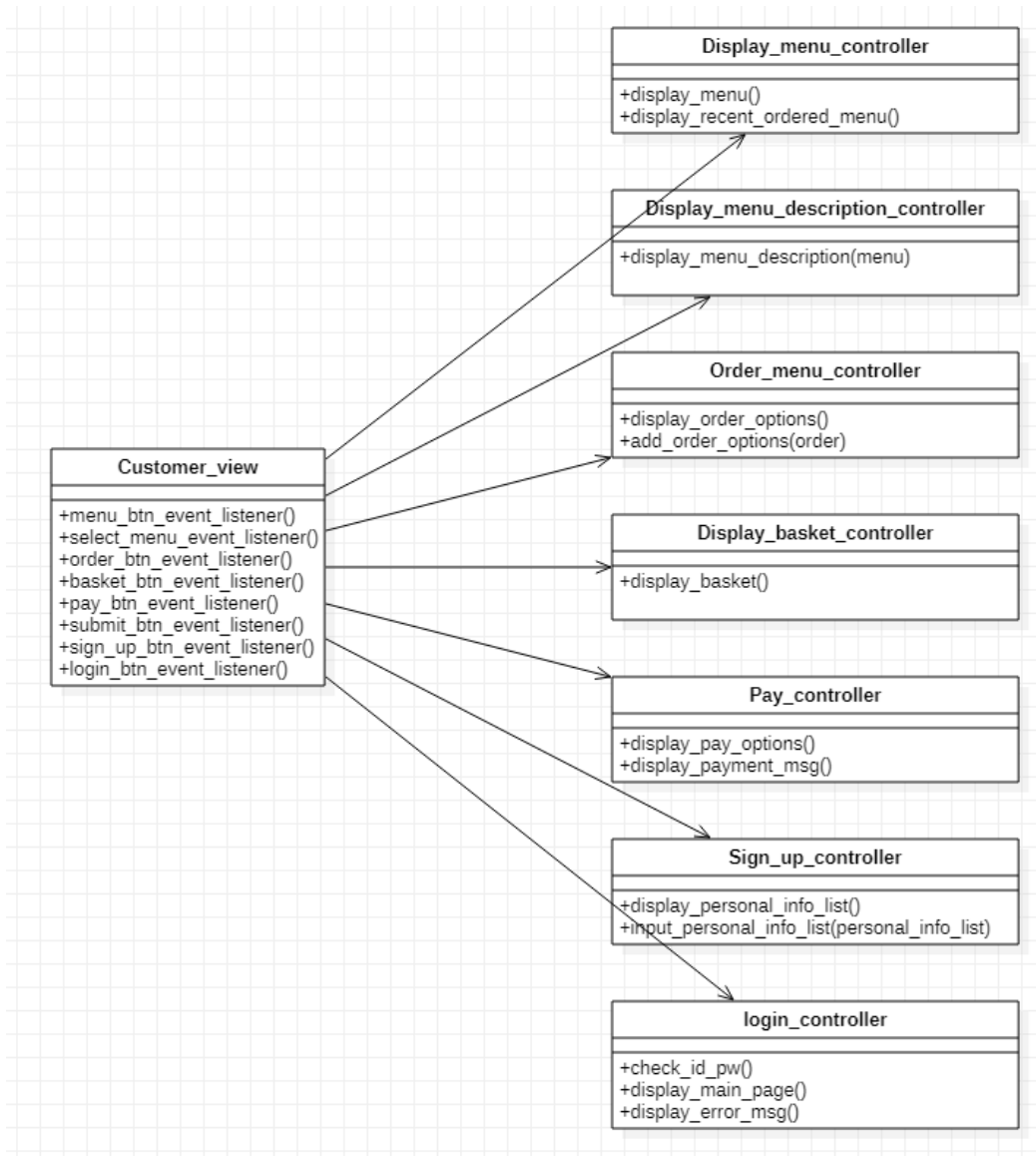
3.1 “Customer Client” 서브시스템 세부 설계

“Customer Client” 시스템은 고객에게 필요한 정보를 “Server” 시스템으로부터 받아와서 보여준다. 또한, 고객으로부터 입력을 받고 “Server” 시스템으로 전송한다.

“Customer Client” 시스템의 주요 함수는 다음과 같다.

- ✓ 각종 display
 - “Customer Client” 시스템에서 event_listener 가 호출되었을 때 필요한 정보를 보여준다.
- ✓ 각종 event_listener
 - 각종 버튼이 눌렸을 때 호출되는 함수로 “Server” 시스템에 특정 요청을 한다.

- “Customer Client” 서브시스템 메인 클래스도



Customer_view

- ✓ “Customer Client” 시스템은 Customer_view 클래스를 통해 사용자의 입력을 처리한다.
- ✓ Customer_view 클래스는 입력에 따라 Controller 클래스의 함수를 호출한다.

각종 Controller

- ✓ “Customer Client” 시스템은 각종 Controller 클래스를 통하여 “Server”에 요청을 보낸다.

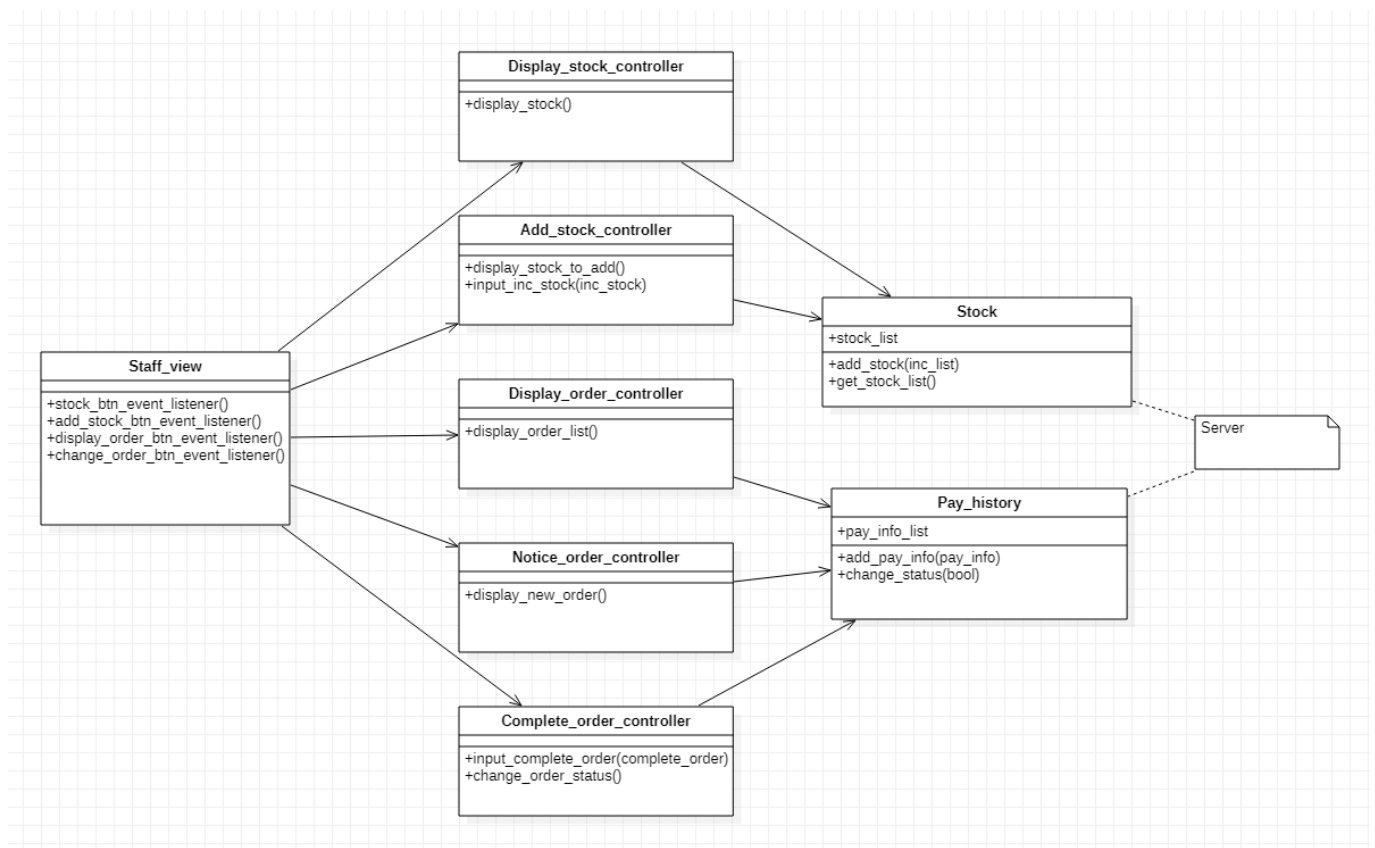
3.2 “Staff Client” 서브시스템 세부 설계

“Staff Client” 시스템은 주문 내역과 재고 내역을 “Server” 시스템으로부터 받아와서 보여준다. 그리고 납품으로 늘어난 재고의 양과 완료된 주문 목록을 입력으로 받고 “Server” 시스템으로 전송한다.

“Staff Client” 시스템의 주요 함수는 다음과 같다.

- ✓ 각종 display
 - “Staff Client” 시스템에서 event_listener 가 호출되었을 때 필요한 정보를 보여준다.
- ✓ 각종 event_listener
 - 각종 버튼이 눌렸을 때 호출되는 함수로 “Server” 시스템에 특정 요청을 한다.

- “Staff Client” 서브시스템 메인 클래스도



Staff_view

- ✓ “Staff Client” 시스템은 Staff_view 클래스를 통해 사용자의 입력을 처리한다.
- ✓ Staff_view 클래스는 입력에 따라 Controller 클래스의 함수를 호출한다.

각종 Controller

- ✓ “Staff Client” 시스템은 각종 Controller 클래스를 통하여 “Server”에 요청을 보낸다.

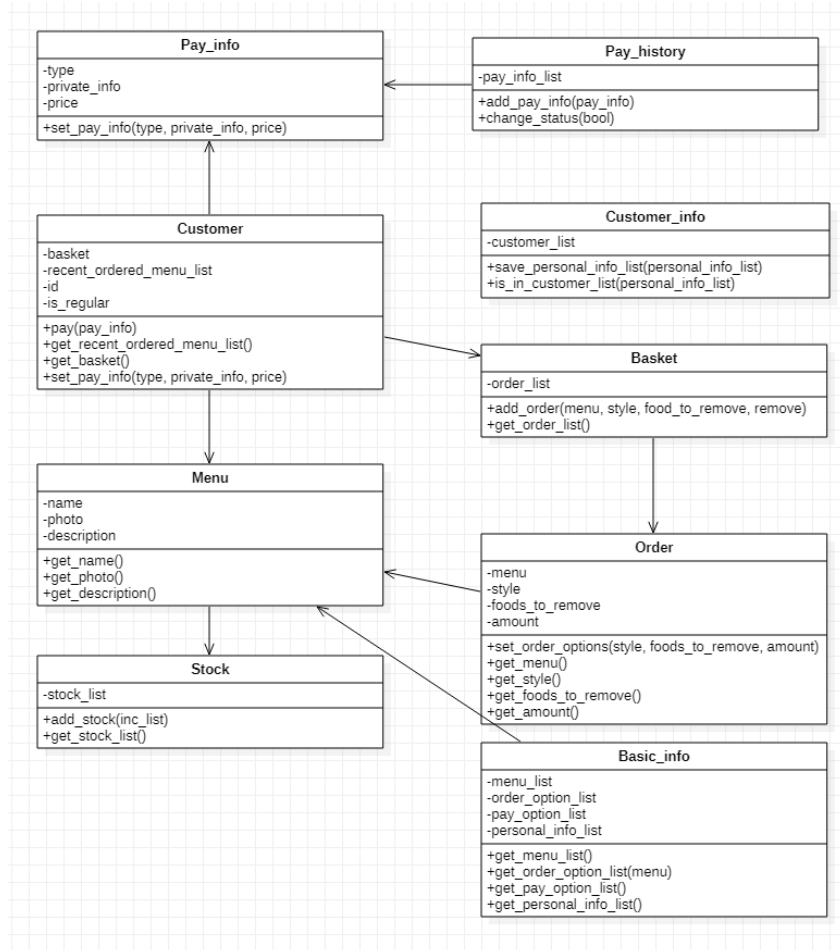
3.3 “Server” 서브시스템 세부 설계

“Server” 시스템은 고객에게 메뉴 관련 정보를 제공하고 주문 정보를 저장한다. 또한, “Customer Client”, “Staff Client”에게 받은 요청을 처리하는 역할을 한다.

“Server” 시스템의 주요 함수는 다음과 같다.

- ✓ **set_order_options**
 - “Customer Client”에서 주문 option 을 전송받고 주문 정보를 생성한다.
- ✓ **add_order**
 - “Customer Client” 시스템으로부터 전송받은 정보를 토대로 주문 정보를 생성하고 장바구니에 추가한다.
- ✓ **set_pay_info**
 - “Customer Client” 시스템에서 결제 정보를 받는다.
- ✓ **add_pay_info**
 - “Customer Client” 시스템에서 결제가 성사된 pay_info 를 저장한다.
- ✓ **change_status**
 - “Staff Client”에서 전송받은 주문을 완료 상태로 전환한다.
- ✓ **add_stock**
 - “Staff Client”에서 전송받은 납품 정보를 반영한다.
- ✓ **pay**
 - 결제 시스템에게 결제 정보를 넘기며 결제를 요청한다.

- “Server” 서비스시스템 메인 클래스도



Basic_info

- ✓ 미스터 대박 서비스의 기본적인 정보를 담고 있는 클래스이다.
- ✓ “Server” 시스템은 Basic_info 를 통해 “Customer Client” 시스템에게 메뉴 정보, 메뉴에 따른 주문 옵션 정보, 결제 정보를 넘겨준다.

Menu

- ✓ 메뉴 정보를 저장하는 type 을 나타내는 클래스이다.
- ✓ 예시 사진과 메뉴 설명에 대한 정보를 담고 있다.

Customer

- ✓ 고객을 나타내는 클래스이다.
- ✓ 고객을 식별하는 id, 단골 고객 여부, 장바구니, 최근 주문 내역을 저장하고 있다.
- ✓ Pay 함수를 통해 장바구니에 담긴 주문에 대해 최종 결제를 할 수 있다.

Basket

- ✓ 장바구니 type 클래스이다.
- ✓ 장바구니는 주문 정보의 집합이며 add_order 를 통해 장바구니에 주문 정보를 추가할 수 있다.

Order

- ✓ 주문 정보를 저장하는 type 을 나타내는 클래스이다.
- ✓ 메뉴, 스타일, 뽕 음식, 수량 정보를 저장하고 있으며 set_order_options 로 주문 정보를 설정할 수 있다.

Pay_info

- ✓ 결제 정보를 저장하는 type 을 나타내는 클래스이다.
- ✓ 결제 수단, 결제 관련 개인 정보, 최종 결제 가격 정보를 저장하고 있으며 set_pay_info 로 결제 정보를 설정할 수 있다.

Stock

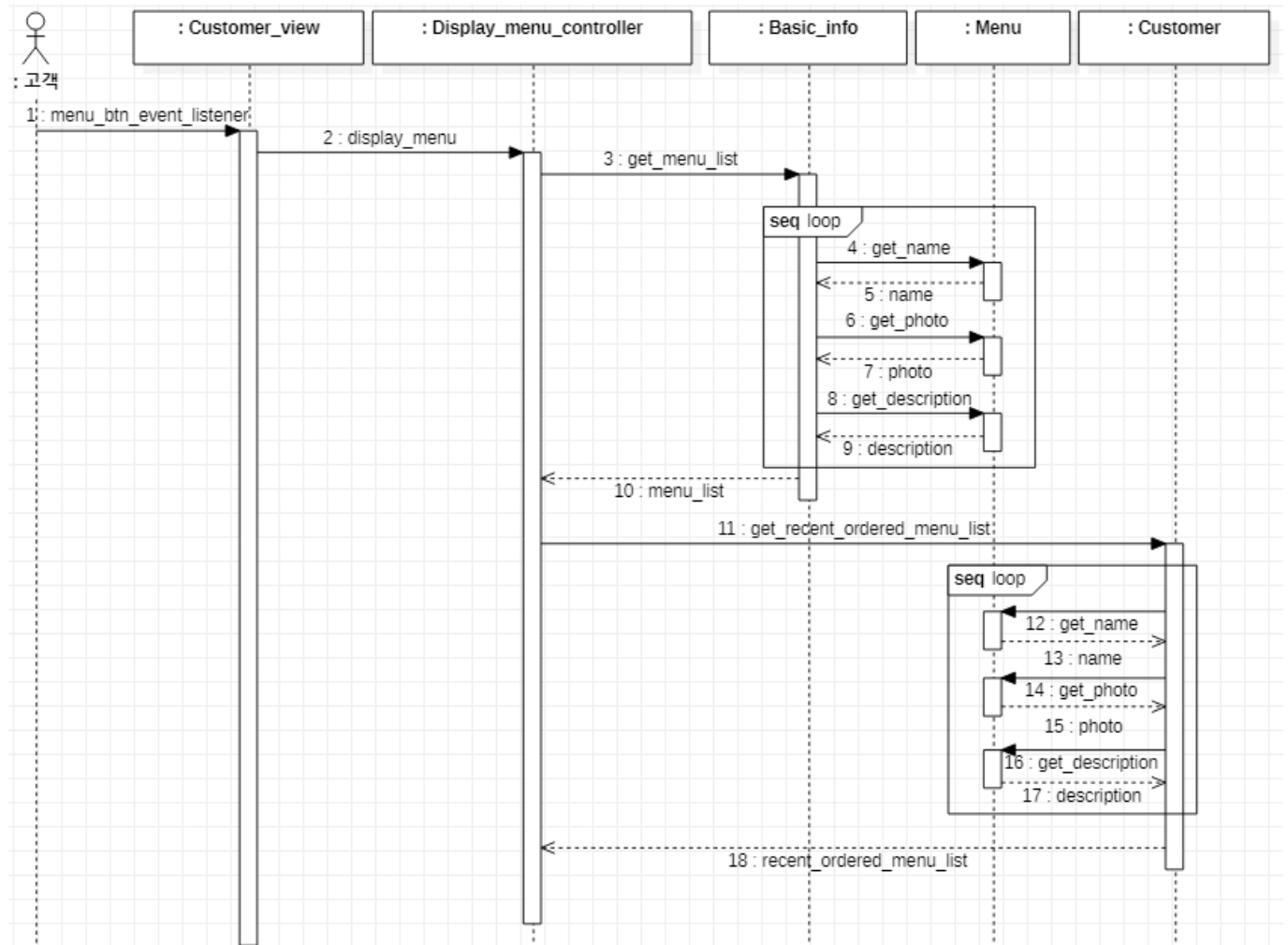
- ✓ 재고의 정보를 저장하는 type을 나타내는 클래스이다.
- ✓ 재고의 종류, 재고의 수량을 저장하고 있으며 add_stock을 통해서 납품된 재고를 추가할 수 있다.

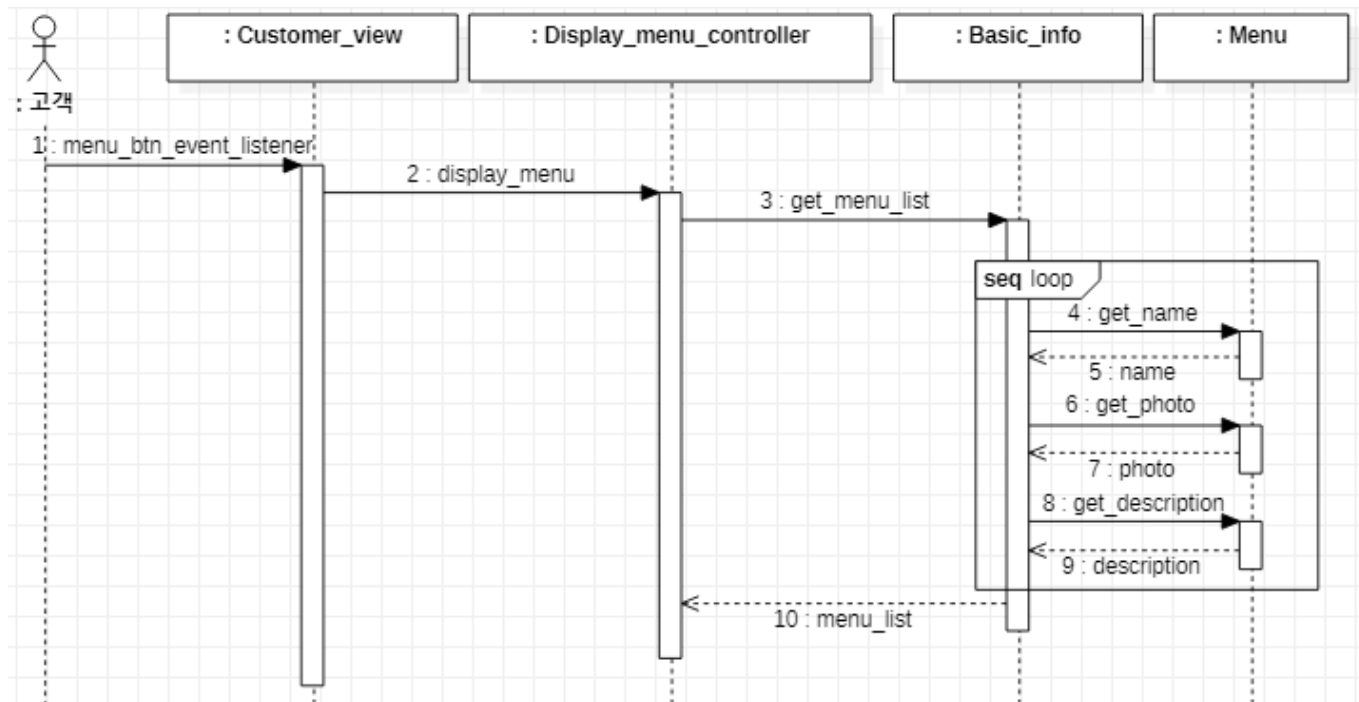
Pay_history

- ✓ 성사된 주문의 정보를 저장하는 클래스이다.
- ✓ 새로 성사된 주문을 add_pay_info를 통해서 추가할 수 있다.
- ✓ change_status를 통해서 배달이 완료된 주문을 처리 완료 상태로 전환할 수 있다.

4 교류도

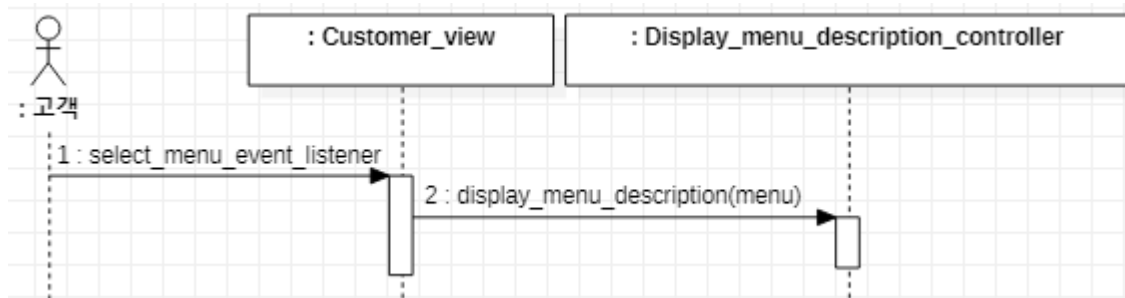
4.1 메뉴 열람 (Display menu)





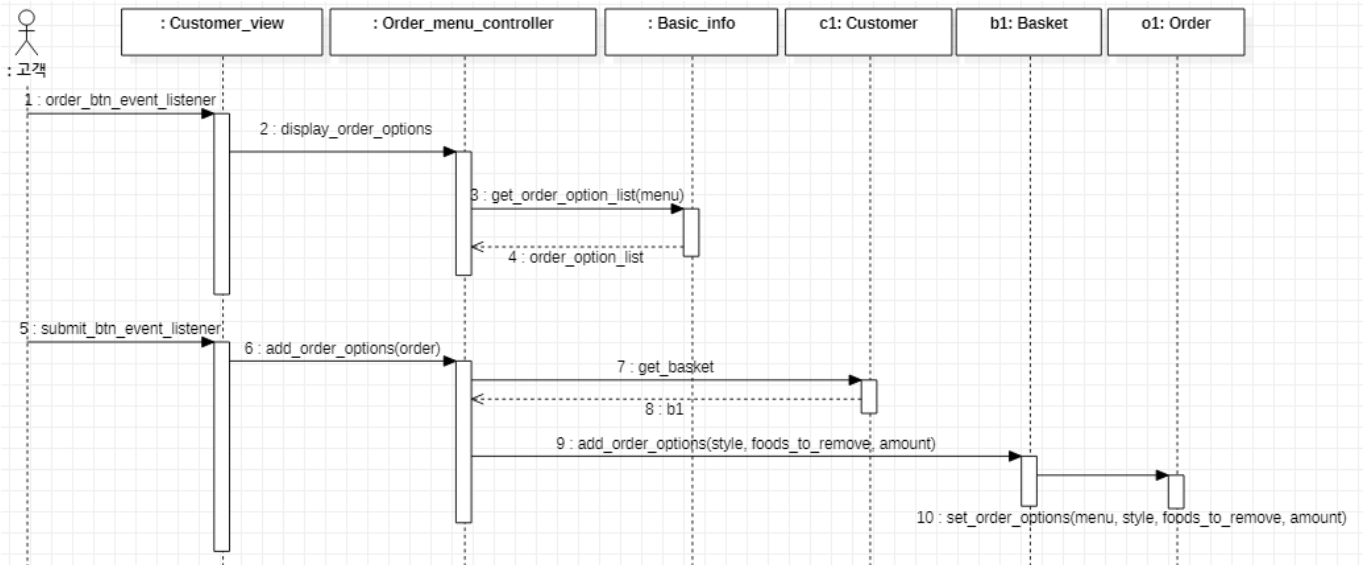
1. 고객이 “메뉴” 버튼을 누르면 menu_btn_event_handler()이 호출되고 Display_menu_controller의 display_menu()를 호출하게 된다.
2. get_menu_list()으로 menu_list를 불러오고 불러온 메뉴 정보들을 화면에 출력한다.

4.2 메뉴 설명 확인 (Display menu description)



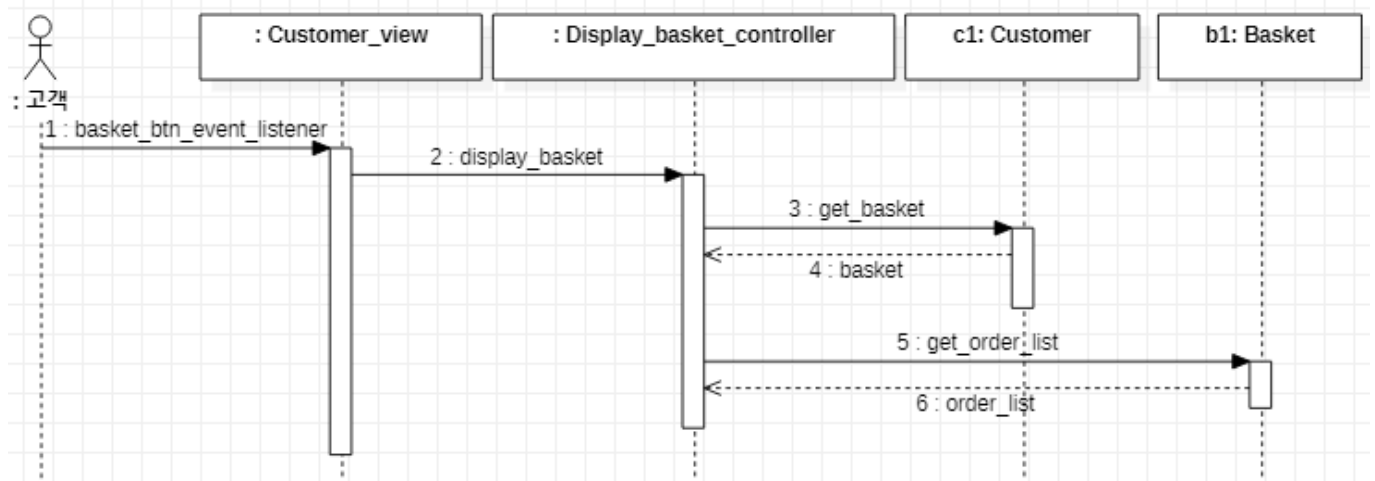
1. 고객이 메뉴 중 하나를 선택하면 select_menu_event_listener()가 호출된다.
2. 선택한 메뉴에 따라 display_menu_description(menu)를 호출하여 메뉴의 사진과 설명을 화면에 출력한다.

4.3 주문하기 (Order menu)



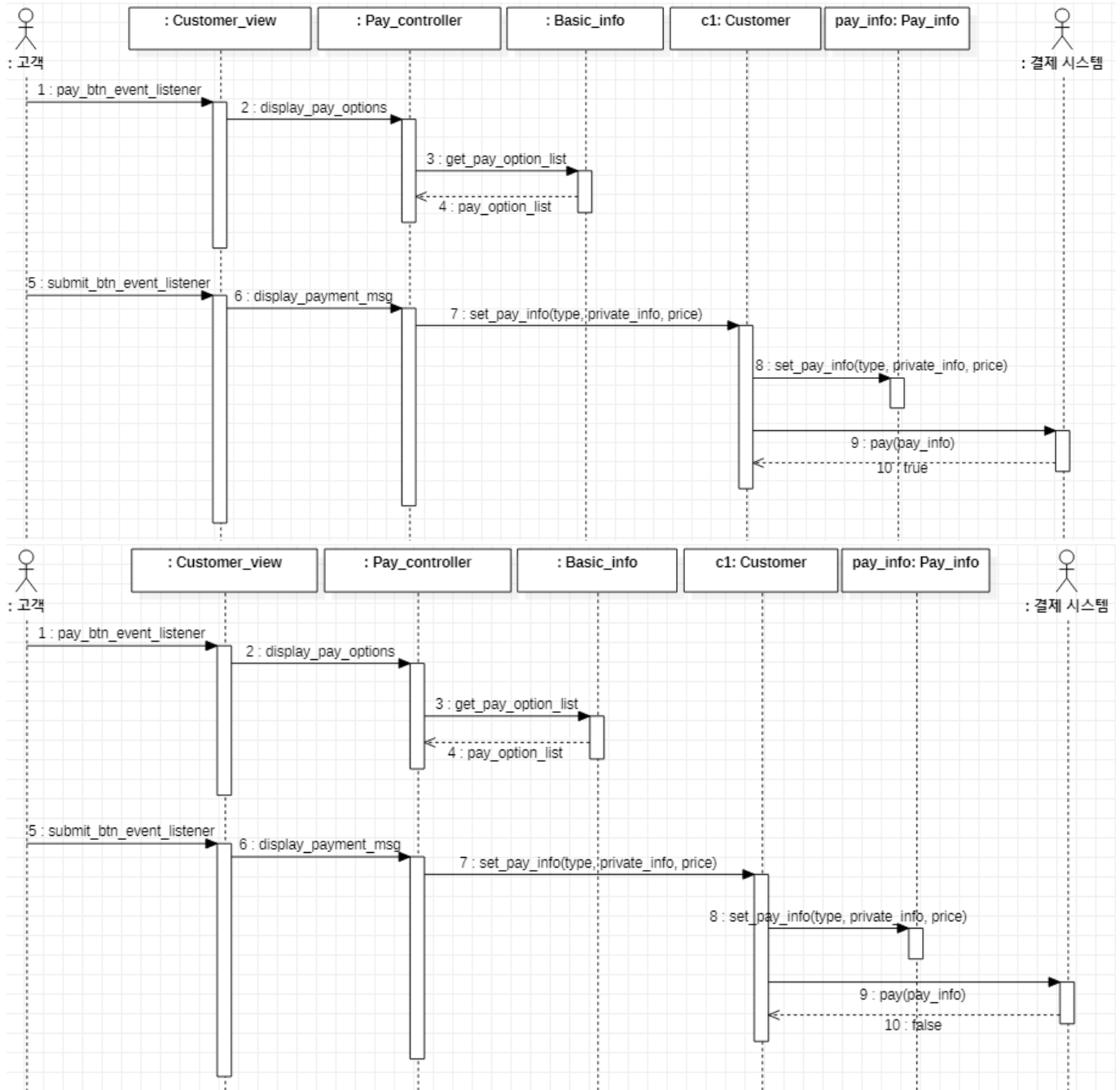
1. “주문하기” 버튼을 누르면 order_btn_event_listener()가 호출되고 Order_menu_controller의 display_order_options()를 호출하게 된다.
2. Basic_info로부터 order_option_list를 받아와서 화면에 출력한다.
3. 고객이 주문 정보를 입력하고 submit 버튼을 누르면 submit_btn_event_listener()이 호출되고 Order_menu_controller의 add_order_options(order)를 호출하게 된다.
4. get_basket() 함수를 통해 Customer 객체로부터 Basket 객체(b1)를 받아온다.
5. 해당 객체에 add_order_options 함수로 order를 추가한다.

4.4 장바구니 확인 (Display basket)



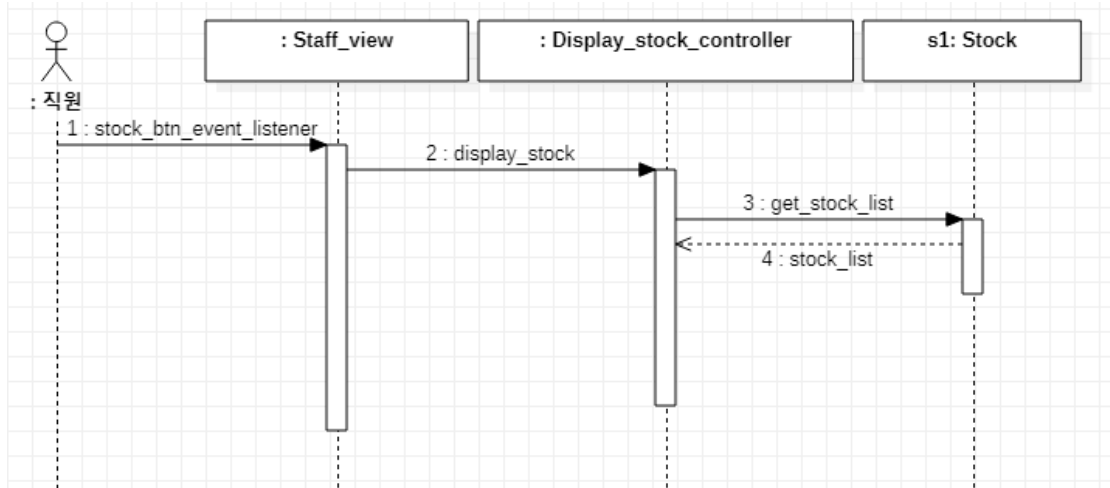
1. 고객이 “장바구니” 버튼을 누르면 basket_btn_event_listener가 호출되고 Display_basket_controller의 display_basket()이 호출된다.
2. get_basket() 함수를 통해 Customer 객체(c1)의 Basket 객체(b1)를 넘겨받는다.
3. 넘겨받은 Basket 객체에 대해 get_order_list()를 호출하여 order_list를 넘겨받는다.
4. order_list를 화면에 출력한다.

4.5 결제 (Pay)



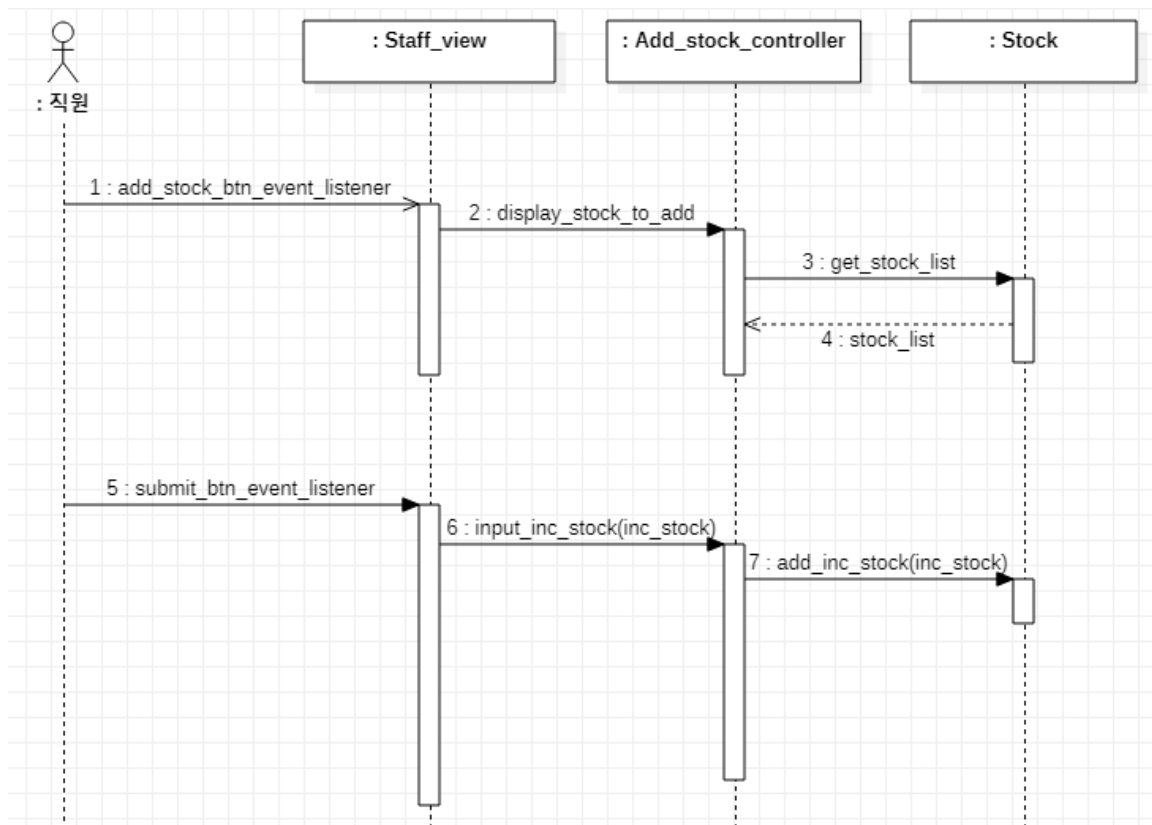
1. 고객이 “결제” 버튼을 누르면 `pay_btn_event_listener()`가 호출되고 `Pay_controller`의 `display_pay_options()`가 호출된다.
2. `get_pay_option_list()`를 통해 `Basic_info`로부터 `pay_option_list`를 받아온다.
3. `pay_option_list`를 출력한다.
4. 고객이 결제 방법을 선택한 뒤 “제출” 버튼을 누르면 `submit_btn_event_listener()`가 호출되고 `Pay_controller`의 `display_payment_msg()`가 호출된다.
5. `set_pay_info(type, private_info, price)`를 통해 `Customer`의 `pay` 정보를 설정한다.
6. 설정된 `pay` 정보를 토대로 `Pay_info` 객체를 만들고 결제 시스템에 승인을 요청한다.
7. 반환값을 토대로 사용자에게 확인 메시지를 보여준다.

4.6 재고 조회 (Display Stock)



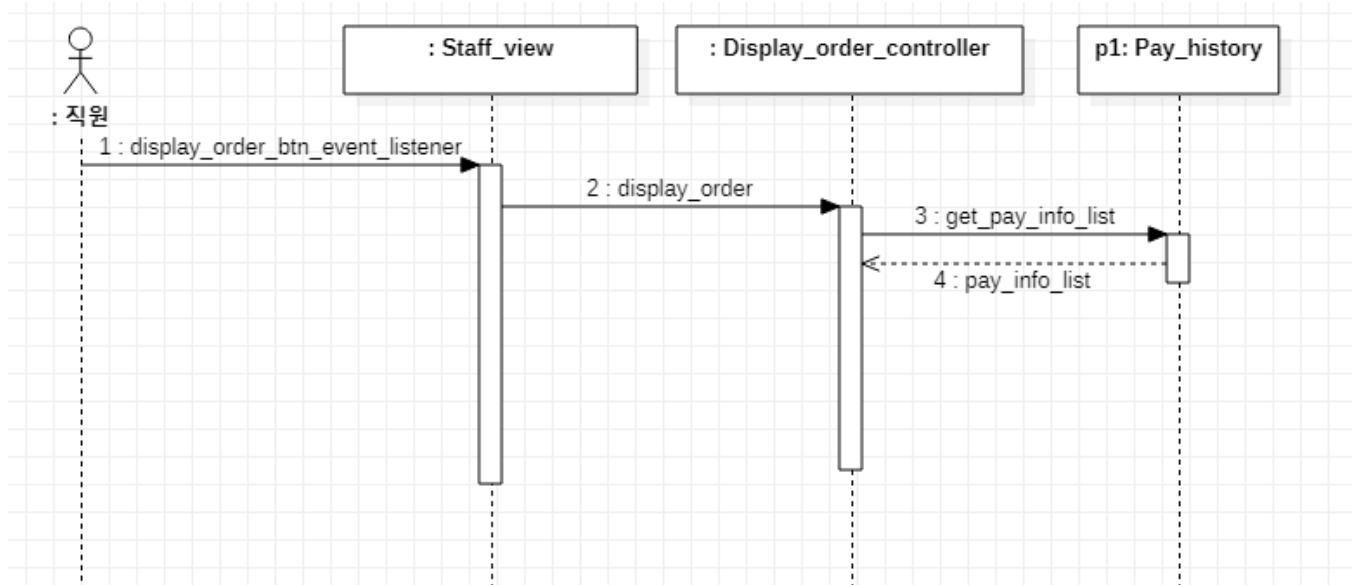
1. 직원이 '재고 확인'을 선택하면 stock_btn_event_listener()가 호출된다.
2. display_stock()를 호출하여 재고 항목과 재고의 수를 보여준다.

4.7 재고 납품 (Add Stock)



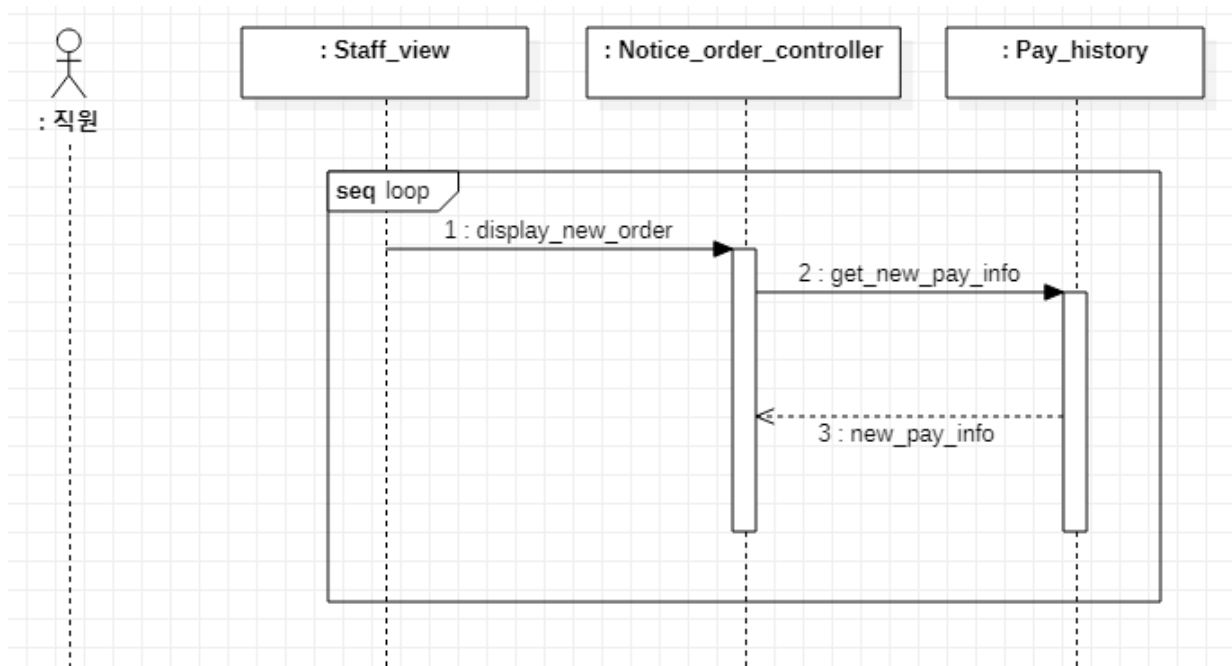
1. 직원이 “재고 납품” 버튼을 누르면 add_stock_btn_event_listener()가 호출되고 Add_stock_controller의 display_stock_to_add()가 호출된다.
2. get_stock_list()를 통해 Stock 으로부터 stock_list 를 받아온다.
3. stock_list 를 출력한다.
4. 직원이 납품된 물품의 수량을 입력한 뒤 “제출” 버튼을 누르면 submit_btn_event_listener()가 호출되고 Add_stock_controller의 input_inc_stock(inc_stock)가 호출된다.
5. 입력된 납품 물품 수량 정보를 add_inc_stock(inc_stock)을 통해서 Stock 객체에 반영한다.

4.8 주문 내역 확인 (Display Order)



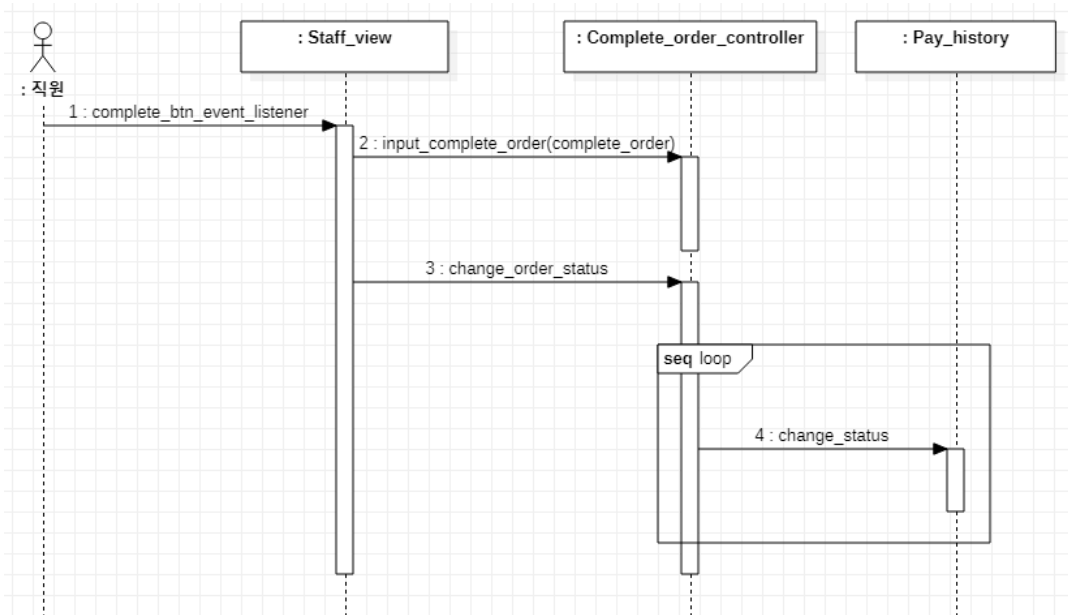
1. 직원이 '주문 내역 확인'을 선택하면 display_order_btn_event_listener()가 호출된다.
2. display_order()를 호출하여 모든 주문 내역을 보여준다.

4.9 주문 알림 (Notice Order)



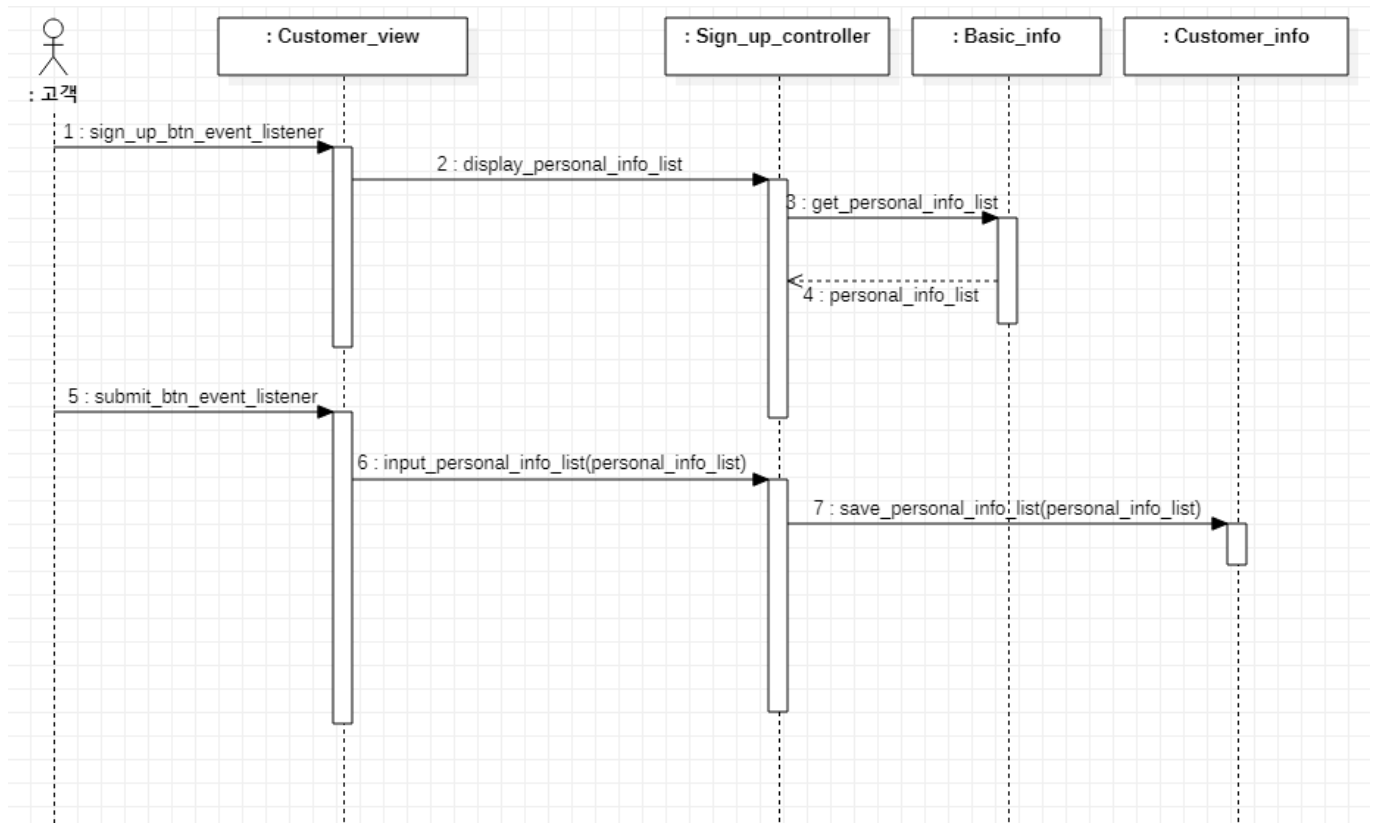
1. 직원이 Staff_view 를 사용하는 동안 display_new_order 가 호출된다.
2. 만약 새로운 결제가 발생하면 Pay_history 객체에서 Notice_order_controller 에 새로운 주문 정보인 new_pay_info 를 넘겨준다.
3. Staff_view 에 새로운 주문 정보가 팝업으로 직원에게 알려진다.

4.10 주문 완료 (Complete Order)



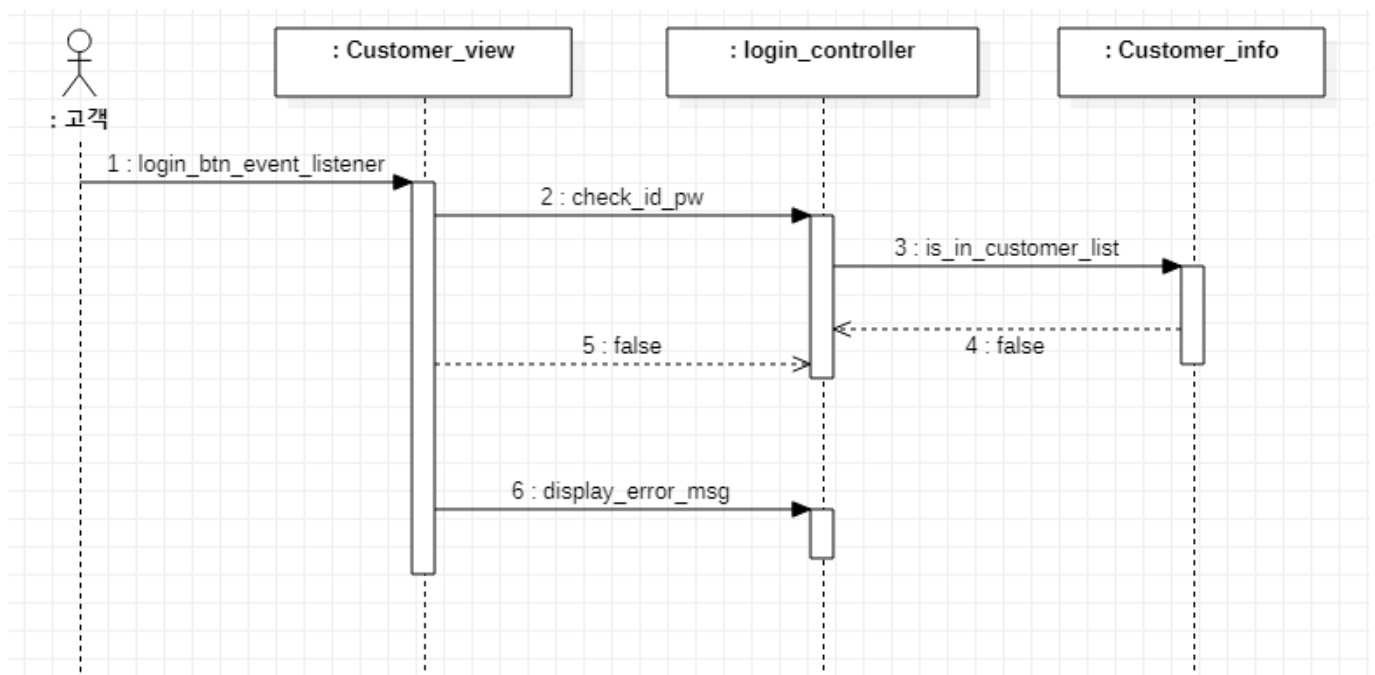
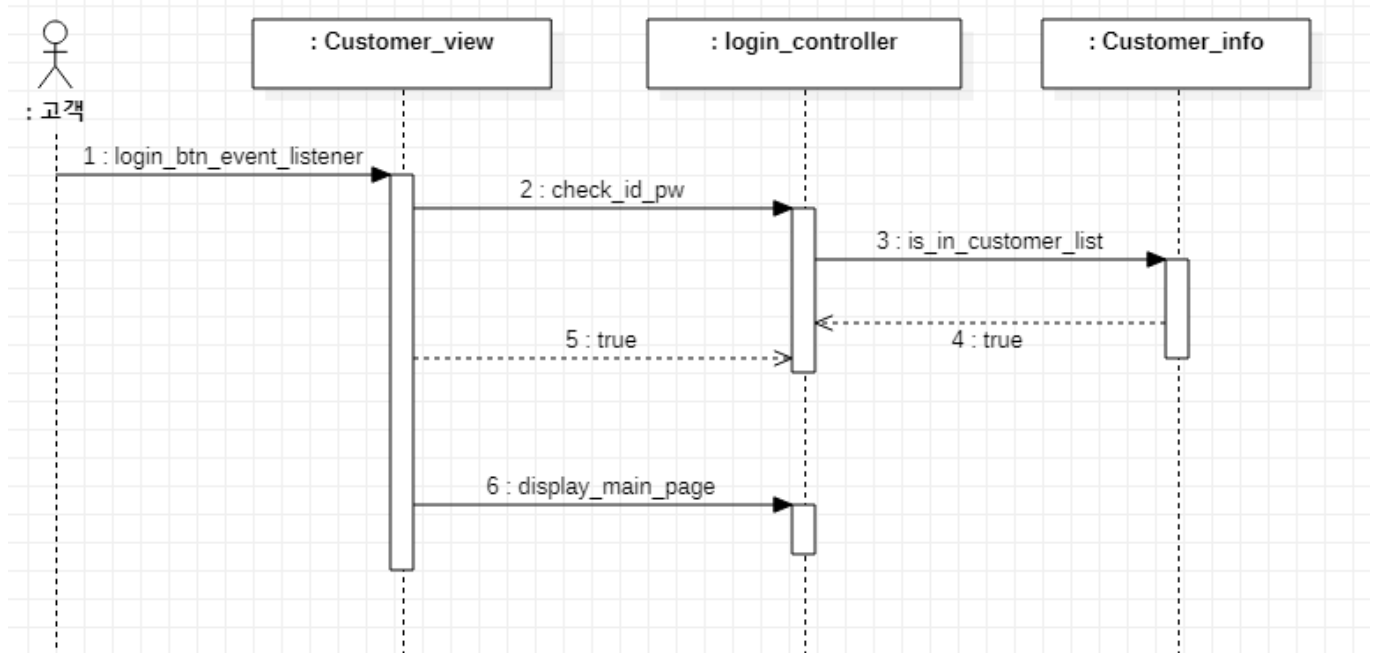
1. 직원이 '주문 완료'를 선택하면 complete_btn_event_listener()가 호출된다.
2. Staff_view에서는 input_complete_order를 통해서 주문 완료하려는 주문 항목을 Complete_order_controller에게 전달해준다.
3. change_order_status를 호출하면 complete_order에 있는 항목을 각각 change_status를 호출하여 주문 완료 상태(complete)로 전환한다.

4.11 회원가입 (Sign Up)



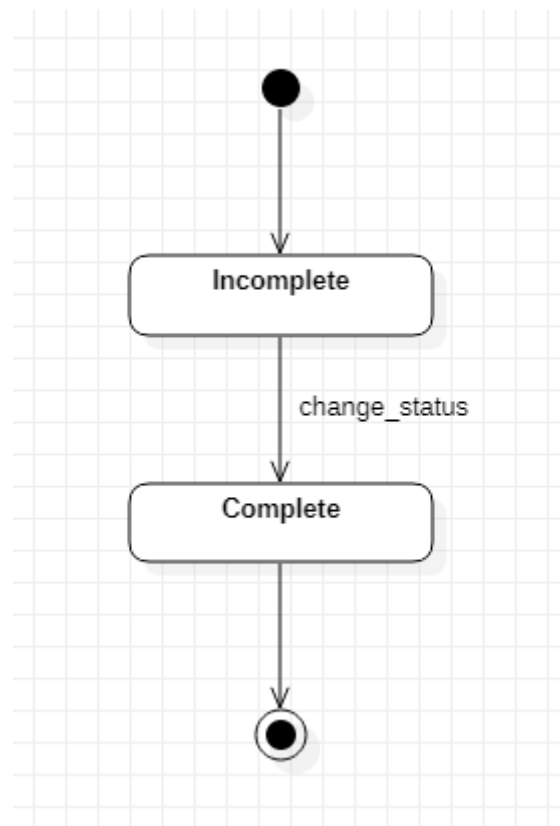
1. 고객이 '회원가입'을 선택하면 sign_up_btn_event_listener()가 호출된다.
2. Customer_view에서는 Sign_up_controller의 display_personal_info_list를 통해서 입력해야하는 개인정보를 보여준다.
3. Sign_up_controller에서는 get_personal_info_list를 통해서 Basic_info에 있는 personal_info_list를 받는다.
4. 고객이 personal_info_list의 각 항목을 입력하고 입력 화면에서 '회원가입'을 누르면 Sign_up_controller에 input_personal_info_list로 고객의 개인정보를 넘겨준다.
5. save_personal_info_list를 통해서 Customer_info에 고객 정보를 저장한다.

4.12 로그인 (Login)



1. 고객이 아이디, 비밀번호를 입력하고 “로그인” 버튼을 누르면 `login_btn_event_listener()`가 호출되고 `Login_controller`의 `check_id_pw()`가 호출된다.
2. `is_in_customer_list()`를 통해 `Customer_info`에 고객의 아이디, 비밀번호가 있는지 확인한다.
3. 만약 반환값이 `true`이면 `display_main_page`를 통해서 메인 화면을 보여준다.
4. 반환값이 `false` 라면 `display_error_msg`를 통해서 에러 메시지를 보여준다.

5 Pay_info 객체 상태도



`Pay_info` 객체는 결제 정보는 나타내는 클래스이다. `Pay_info`는 다음 두 개의 상태 중 하나를 가진다.

- ✓ **Incomplete**
 - “Customer Client”에서 결제가 완료되었지만 아직 배달이 완료되지 않은 상태이다.
 - 처음 결제가 완료되면 `Incomplete` 상태로 초기화되며 직원이 배달이 완료된 주문을 `change_status()`를 호출하면 `Complete` 상태가 된다.
- ✓ **Complete**
 - “Customer Client”에서 결제가 완료되었고 배달까지 완료된 상태이다.