

<무어>

<MrGreat>

설계 산출물

문서버전	4.1
문서 ID	무어-2022-001
최종변경일	2022.11.27.
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		<무어> <MrGreat> 설계 산출물	
문서 ID			
버전		변경일	설명
1	0	2022-10-07	아키텍처도를 추가했다. “Employee Client” 클래스도를 추가했다. “Customer Client” 클래스도를 추가했다.
	1	2022-10-09	“Server Client” 클래스도를 추가했다. “Customer Client” 서브시스템 세부설계 설명을 추가했다. “Staff Client” 서브시스템 세부설계 설명을 추가했다.
	2	2022-10-13	“Server” 서브시스템 세부설계 설명을 추가했다.
2	0	2022-10-15	릴리즈
	1	2022-10-16	오타 수정
3	0	2022-10-18	[피드백 반영] 서브시스템 설계 및 클래스도 수정
4	0	2022-11-24	서브시스템 설계 및 클래스도 수정
	1	2022-11-26	교류도 및 세부 설명 수정

1 개 요

1.1 목 적

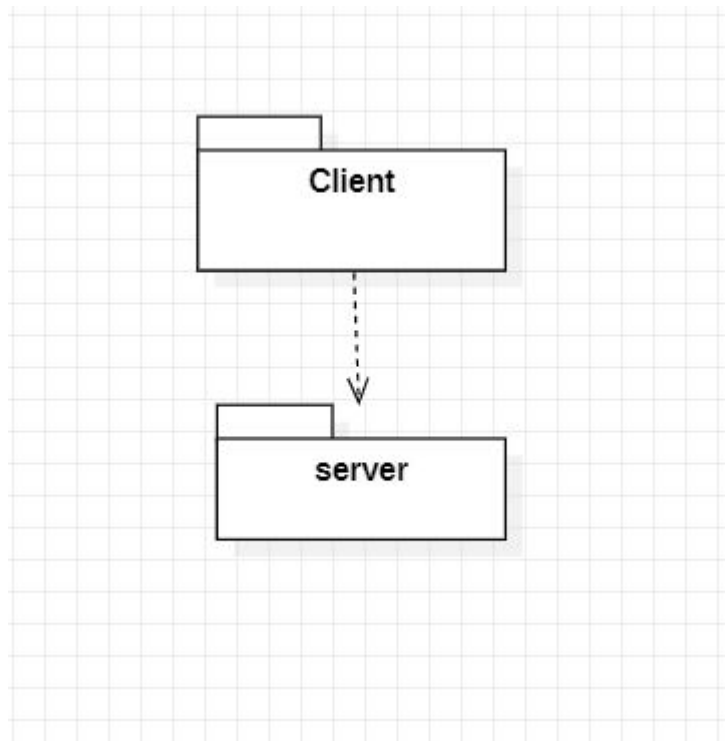
본 문서는 미스터 대박 디너 서비스(Mr.Great)의 산출물을 기술한다.

- ✓ 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.

- Client
- Sever

Client

- 서버와 통신하며 사용자에게 정보를 보여주는 역할을 한다.
- 리액트와 node.js를 이용하여 작성된다.

Server

- 메뉴, 고객 정보, 주문 정보를 담고 있으며 주문 저장 및 이력 관리의 역할을 한다.
- Node.js로 작성된다.

3 서브시스템 세부 설계

3.1. "Client" 서브시스템 세부 설계

App은 MainPage를 호출하고 MainPage에서 기능에 해당하는 modal을 띄운다. 각 modal은 사용자에게 메뉴 화면과 주문 화면을 보여준다. 또한 user_action 클래스를 통해 서버와 통신하며 사용자의 주문을 수행한다.

Client 의 주요 함수는 다음과 같다.

✓ checkhandler (event)

- 로그인한 ID의 토큰을 판별하여 사용자가 고객인지 직원인지 판별한다.

✓ onEmailHandler()

- 입력받은 email정보를 저장한다.

✓ onPasswordHandler()

- 입력받은 pw 정보를 저장한다.

✓ onAddressHandler()

- 입력받은 주소 정보를 저장한다.

✓ onAmountHandler()

- 입력받은 수량 정보를 저장한다.

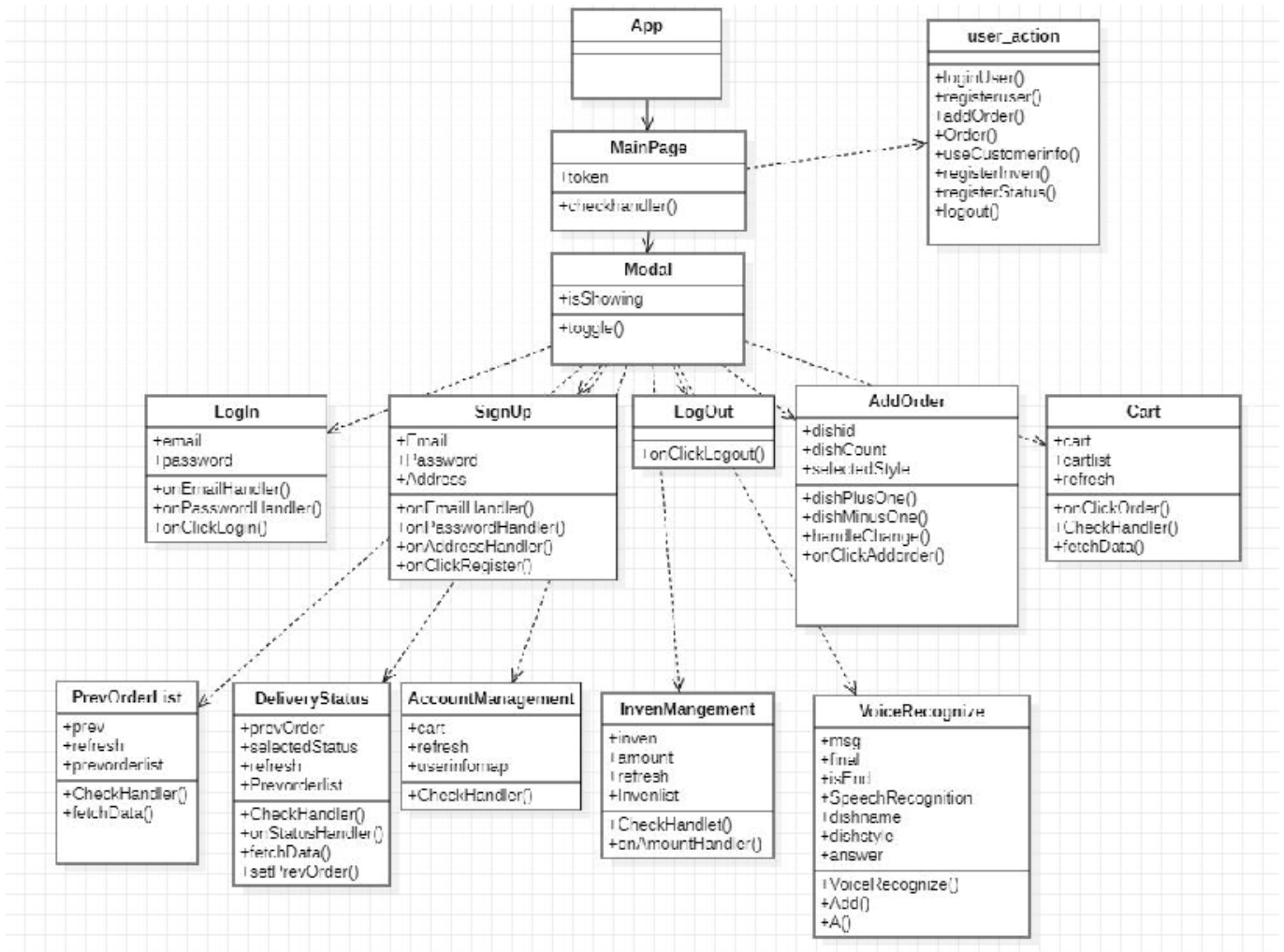
✓ onStatusHandler()

■ 입력받은 주문 상태 정보를 저장한다.

✓ onClick()

■ 버튼을 눌렀을때 해당하는 함수를 호출한다.

- “Client” 서브시스템 메인 클래스도



App

✓ 최초로 호출되는 클래스이다. MainPage를 호출한다.

MainPage

✓ 사용자의 UI를 보여주는 클래스이다.

✓ token의 role을 분석하여 사용자가 고객인지 직원인지 구분하여 기능을 표시한다.

✓ 각 기능에 해당하는 modal을 띄운다

Modal

✓ modal의 형식을 정해준다.

✓ 열고 닫는 창을 보여준다.

Login

- ✓ 로그인을 수행하는 클래스이다.

Sign_up

- ✓ 회원가입을 수행하는 클래스이다.

AddOrder

- ✓ 해당메뉴를 주문하는 기능을 수행한다.
- ✓ 주문하기를 수행해서 고객이 주문하려는 주문을 장바구니에 담을 수 있다.

LogOut

- ✓ 로그아웃을 수행하는 클래스이다.
- ✓

Cart

- ✓ 장바구니 기능을 수행하는 클래스이다.
- ✓ 장바구니에서 결제하는 기능을 포함한다.

InvenManagement

- ✓ 재고를 관리하는 클래스이다.
- ✓ 재고의 수량을 변경할 수 있다.

AccountManagement

- ✓ 고객정보를 관리하는 클래스이다.
- ✓ 고객정보를 변경할 수 있다.

DeliveryStatus

- ✓ 결제된 주문의 상태를 관리하는 클래스이다.
- ✓ 주문의 상태를 변경할 수 있다.

PrevOrderList

- ✓ 과거에 주문한 내역을 관리하는 클래스이다.

VoiceRecognize

- ✓ 음성인식을 구현하는 클래스이다.

user_action

- ✓ app의 모달에서 서버에 요청하는 api를 구현한 클래스이다.
- ✓ 각 모달에서 사용되는 api를 저장한다.

3.2 “Server” 서브시스템 세부 설계

“Server” 는 사용자에게 메뉴 관련 정보를 제공하고 주문 정보를 저장한다. 또한, App 에서 받은 요청을 처리한다.

“Server”의 주요 함수는 다음과 같다.

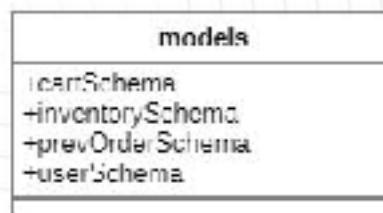
✓ post()

■ 데이터베이스에 정보를 전달하는 함수다.

✓ get()

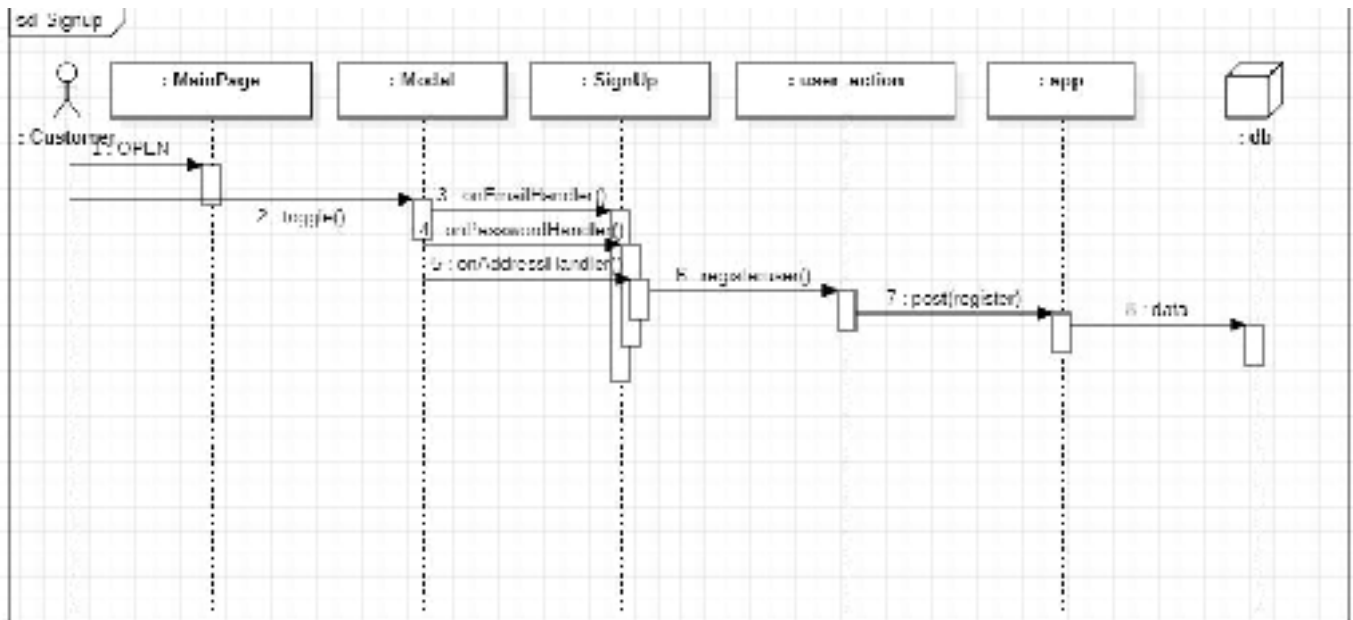
■ 데이터베이스에서 정보를 전달받는 함수다.

- “Server” 서브시스템 메인 클래스도



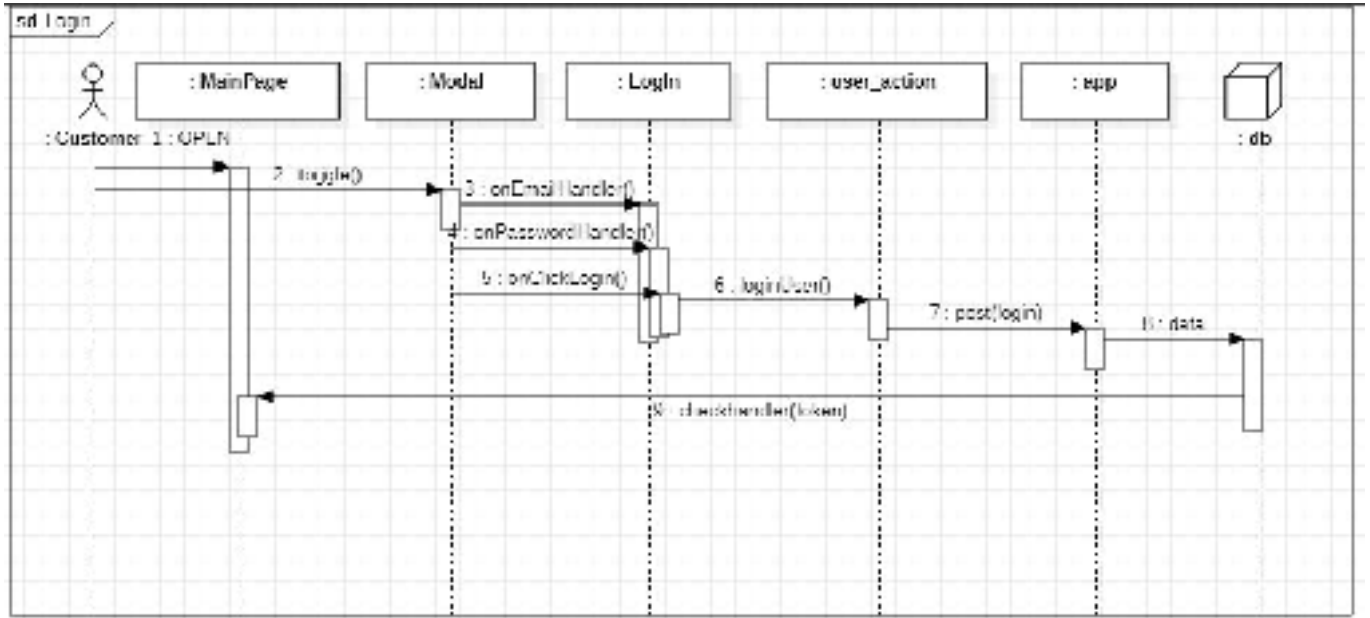
4 교류도

4.1 회원가입 (Sign Up)



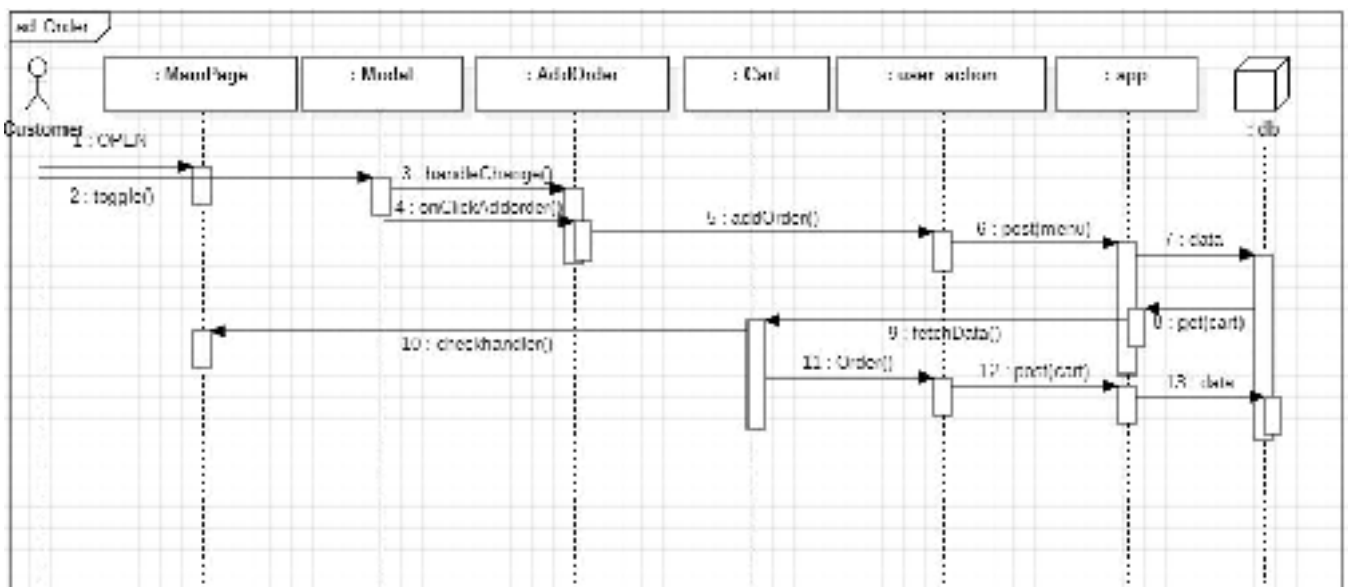
1. 고객이 MainPage를 연다
2. 고객이 MainPage에서 SignUp modal을 클릭한다.
3. SignUp에서는 onEmailHandler()의 onPasswordHandler(), onAddressHandler()를 통해서 정보를 입력받고 onClickRegister로 전송한다.
4. user_action에서는 registerUser를 통해서 app에 회원가입 요청을 한다.
5. app에서 post(register)을 통해 서버의 데이터베이스에 정보를 저장한다.

4.2 로그인 (Login)



1. 사용자가 이메일, 비밀번호를 입력하고 “로그인” 버튼을 누르면 onEmailHandler, onPaswordHandler()로 입력된 정보를 저장하고 onClickLogin()으로 정보를 전송한다.
2. user_action에서 loginUser()로 app에 로그인을 요청한다.
3. app에서 post(login)으로 데이터베이스에 로그인 요청한다.
4. 만약 일치하는 id, pw가 있다면 로그인을 허용하고 checkhandler로 정보를 갱신한다.

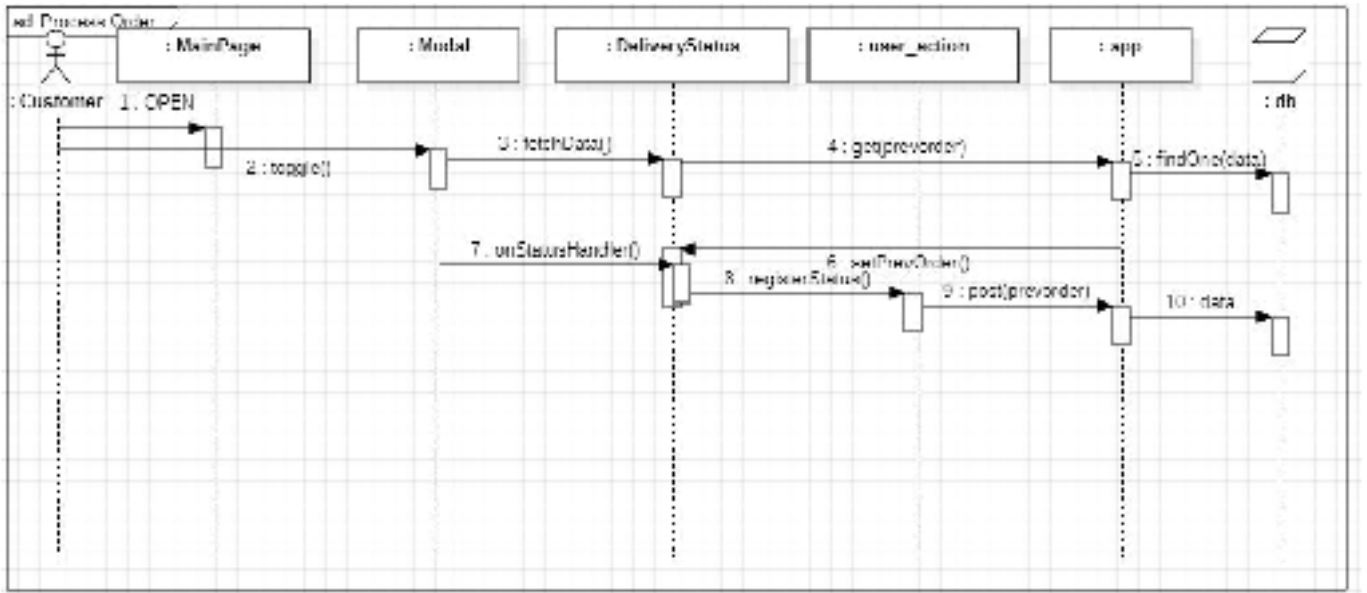
4.3 주문 (Order menu)



1. “메뉴”를 선택하면 handleChange()로 정보를 저장하고 onClickAddorder()로 저장된 정보를 전달한다.
2. user_action에서 addOrder()로 app에 장바구니 저장하기를 요청한다.

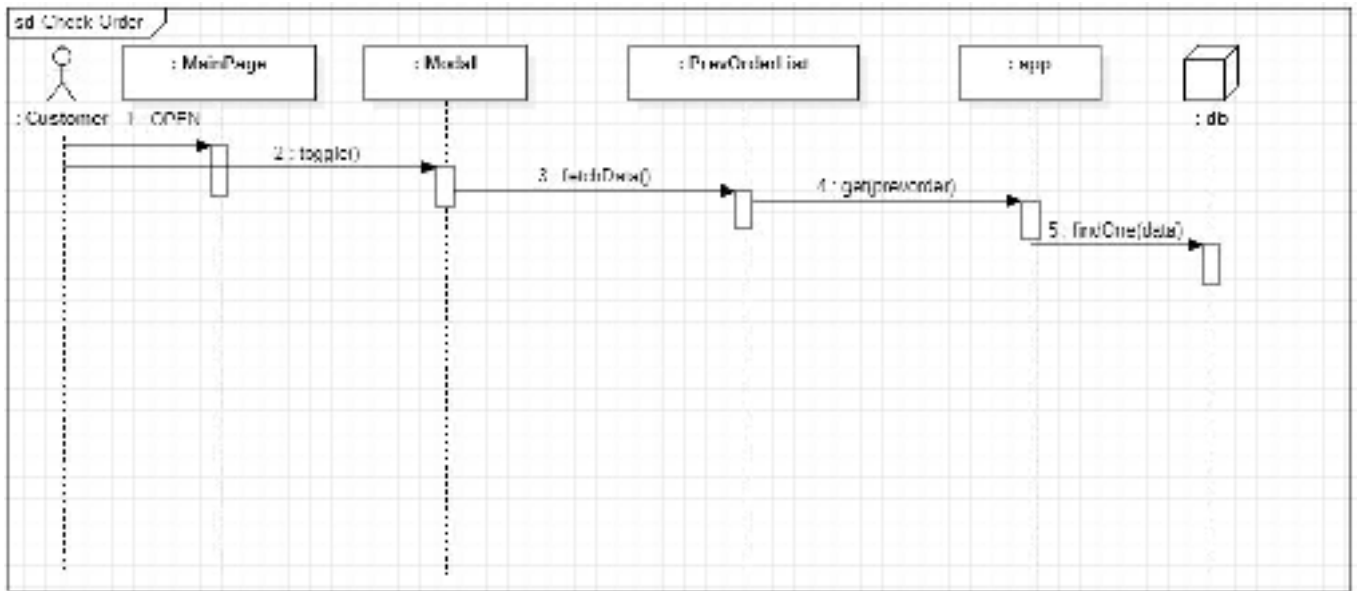
3. app 에서 post(menu)로 장바구니 내역을 데이터베이스에 저장한다.
4. Cart 에서 fetchdata()로 주문한 내용을 장바구니에 띄운다.
5. MainPage에서 checkhandler로 정보를 갱신한다.
6. user_action에서 Order()로 app에 주문하기를 요청한다.
7. app에서 post(cart)로 주문내역을 데이터베이스에 저장한다.

4.4 주문 처리 (Process Order)



1. 직원이 DeliveryStatus 를 선택하면 fetchdata()가 호출되고 app 에서 get(prevorder)로 데이터베이스에서 주문내역을 가져온다.
2. 가져온 내역을 setPrevOrder()로 저장한다.
3. DeliveryStatus 에서 변경한 주문상태의 값을 onStatusHandler()를 통해서 저장한다.
4. user_action 에서 변경된 상태를 registerStatus()를 통해서 app 에 전달한다.
5. app 에서 post(prevorder)로 데이터베이스에 저장한다.

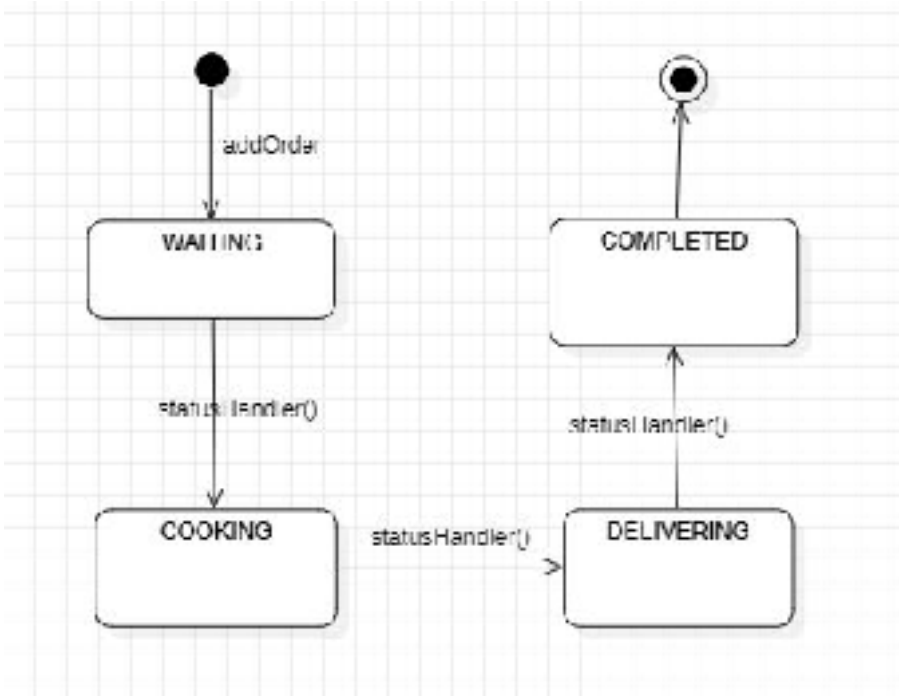
4.5 주문 내역 확인 (Check order)



1. PrevOrderList를 선택하면 fetchData()가 실행되고 app에서 app.get(prevorder)가 실행된다.
2. get(prevorder)로 데이터를 받아온다.

5 Order 객체 상태도

Cart에서 스타일, 수량 변경같은 정보 변경 상태도



Order 클래스 주문을 나타내는 type 클래스이다. Order 클래스는 다음과 같은 상태 변화를 갖는다.

✓ WAITING.

- add_order 메소드를 통해 Order 객체가 생성되고, 초기 상태는 WAITING이 된다.

✓ COOKING

- 조리중인 상태를 나타낸다..
- statusHandler 메소드를 통해 WAITING에서 COOKING 상태로 전환된다.

✓ DELIVERING

- 배달 중인상태를 나타낸다. .
- statusHandler 메소드를 통 COOKING에서해 DELIVERING 상태로 전환된다.

✓ COMPLETED

- 배달이 완료된상태를 나타낸다. .
- statusHandler 메소드를 통 DELIVERING에서 COMPLETED 상태로 전환된다.