

2MM

미스터 대박 디너 서비스

설계 산출물

문서버전	1.0
최종변경일	2020-11-05
문서상태	1차 제출

요 약

미스터 대박 디너 서비스 시스템의 설계 산출물을 기술.
서브 시스템의 구성과 각 서브 시스템의 구조를 기술

- 주요 산출물
- 아키텍처도
 - 패키지 클래스도
 - 서브시스템 클래스도
 - 순차도
 - 상태도

표 1 문서 변경 기록

문서 이름		2MM 미스터 대박 디너 서비스 설계 산출물	
버전		변경일	설명
0	0	2020-10-26	개요와 목적, 아키텍처도와 설명을 추가하였다.
	1	2020-10-27	3가지 서브시스템 (Member, Order, Stock)에 대한 패키지 클래스도를 추가하였다.
	2	2020-11-01	클래스도에 따른 클래스의 설명을 추가하였다.
	3	2020-11-03	8개의 유즈케이스에 대하여 순차도를 추가하였다. <ul style="list-style-type: none"> ✓ SignUp ✓ LogIn ✓ Update my Info ✓ Check My OrderList ✓ Order ✓ Check OrderList ✓ Manage Member ✓ Manage Stock
	4	2020-11-04	8개의 유즈케이스에 대하여 대안 흐름의 순차도를 추가하였다. 각각의 흐름에 대한 설명도 추가하였다. Order의 기본 흐름에 대한 순차도를 변경하였다. 각각 순차도에 대한 메소드 및 변수 표현을 수정하였다.
1	0	2020-11-05	아키텍처의 패키지 분류를 수정하였다. 아래 네 개의 클래스도를 추가하였다. <ul style="list-style-type: none"> ✓ Controller ✓ Service ✓ DAO ✓ DTO Order 객체의 상태도를 추가하였다.

1 개 요

1.1 목 적

본 문서는 “미스터 대박 디너 서비스”의 설계 산출물을 기술한다.

전체 시스템이 어떤 패키지로 구성되어 있는지 아키텍처도를 통해 보여준다.

패키지 클래스도를 통해 각각의 패키지를 구성하는 클래스를 보여준다.

시스템을 기능적 측면으로 나눈 세 가지의 서브시스템(Member, Order, Stock)에 대한 클래스도를 통해 기능적인 측면에서의 클래스의 관계를 보여준다.

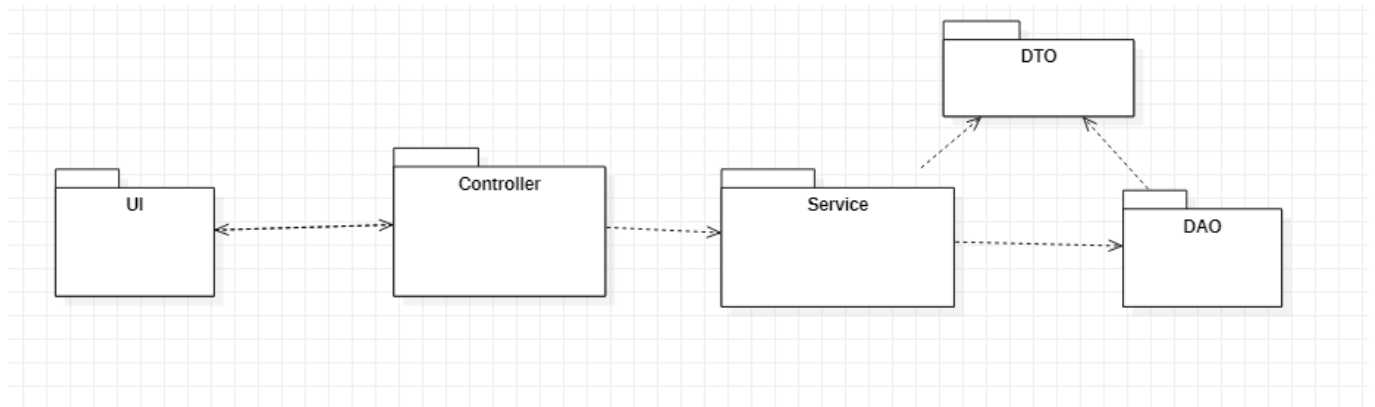
각각의 유스케이스에 따른 순차도를 통해 구체적인 서비스의 실행 과정과 대안 상황에 따른 흐름을 보여준다.

상태도를 통해 서비스의 흐름에 따른 객체의 상태 변화를 보여준다.

1.2 참고 문헌

없음.

2. 아키텍처도



전체 시스템을 패키지 단위로 표현한 아키텍처도이다.

전체 시스템은 아래의 패키지로 구성된다.

- UI
- Controller
- Service
- DTO
- DAO

UI

시스템을 웹 화면으로 출력해 유저와 상호작용하는 시스템이다.

HTML, CSS, JavaScript를 사용하여 작성된다.

Controller

UI와 서비스를 연결하는 패키지로, 해당하는 웹 페이지를 디스플레이하거나, Service에게 관련 로직을 실행하도록 하는 패키지이다.

기능에 따라 MemberController, OrderController, ServiceController의 세 패키지로 나뉜다.

Java 언어를 사용하여 작성된다.

Service

Controller와 DAO를 연결하는 패키지로 Controller의 메시지에 따라 해당하는 기능을 직접적으로 수행하는 패키지이다. DTO에서의 클래스를 활용하여 DAO로부터 데이터를 얻어와 서비스를 수행한다.

기능에 따라 MemberService, OrderService, ControllerService의 세 패키지로 나뉜다.

Java 언어를 사용하여 작성된다.

DAO

전체 데이터베이스와 상호작용하는 시스템이다.

Service패키지 의해 메시지를 받아 DB에서의 추가, 조회, 수정, 삭제의 역할을 수행한다. DTO에서의 클래스를 활용한다.

Java 언어와 sql 언어를 사용하여 작성된다.

DTO

DAO와 Service에서 사용할 데이터의 형태를 정의한 패키지이다.

DB의 컬럼에 대응하는 속성값을 가지고 있다.

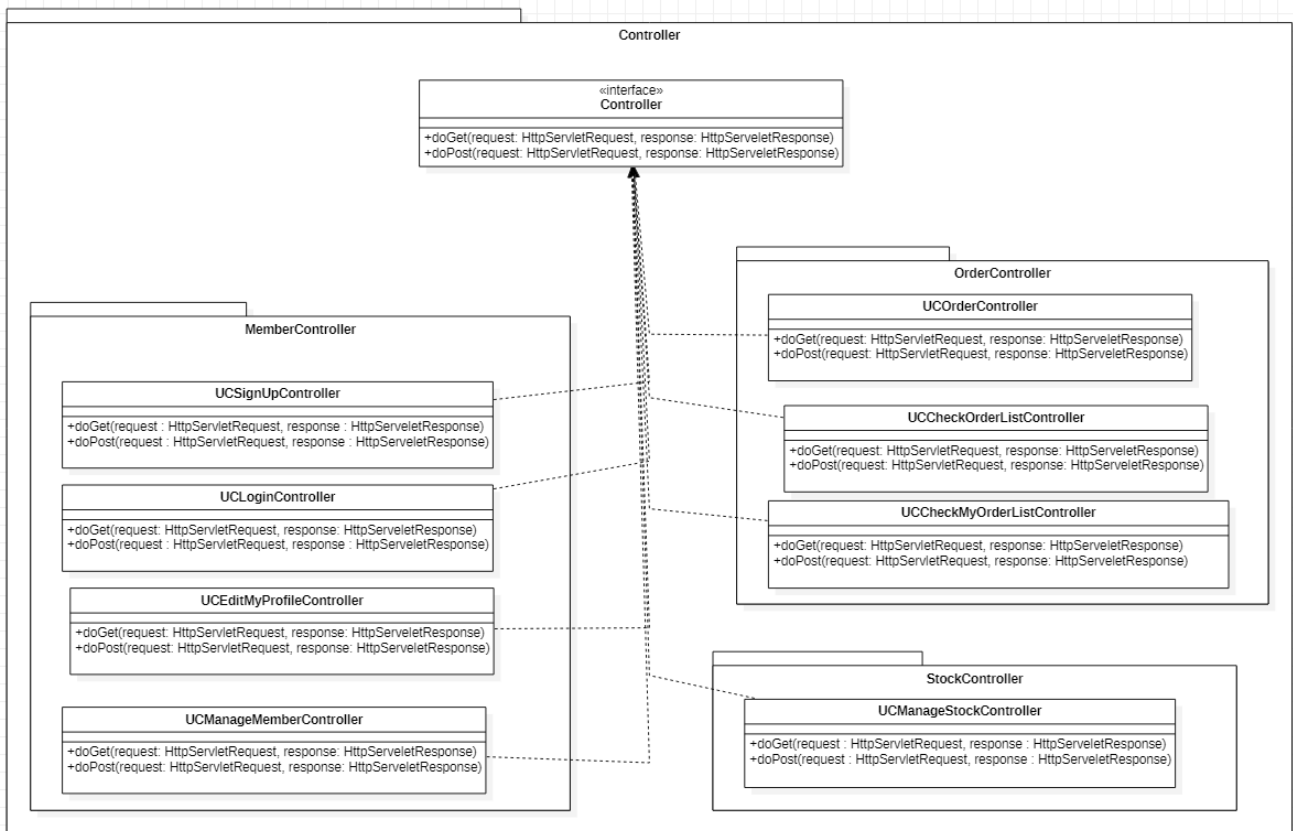
JAVA언어를 사용하여 작성된다.

3. 패키지 클래스도

Controller, Service, DAO, DTO의 패키지를 구성하는 클래스를 보여주는 클래스도이다.

Controller, Service 관련한 클래스들은 4장에서 추가적인 설명이 되어있다.

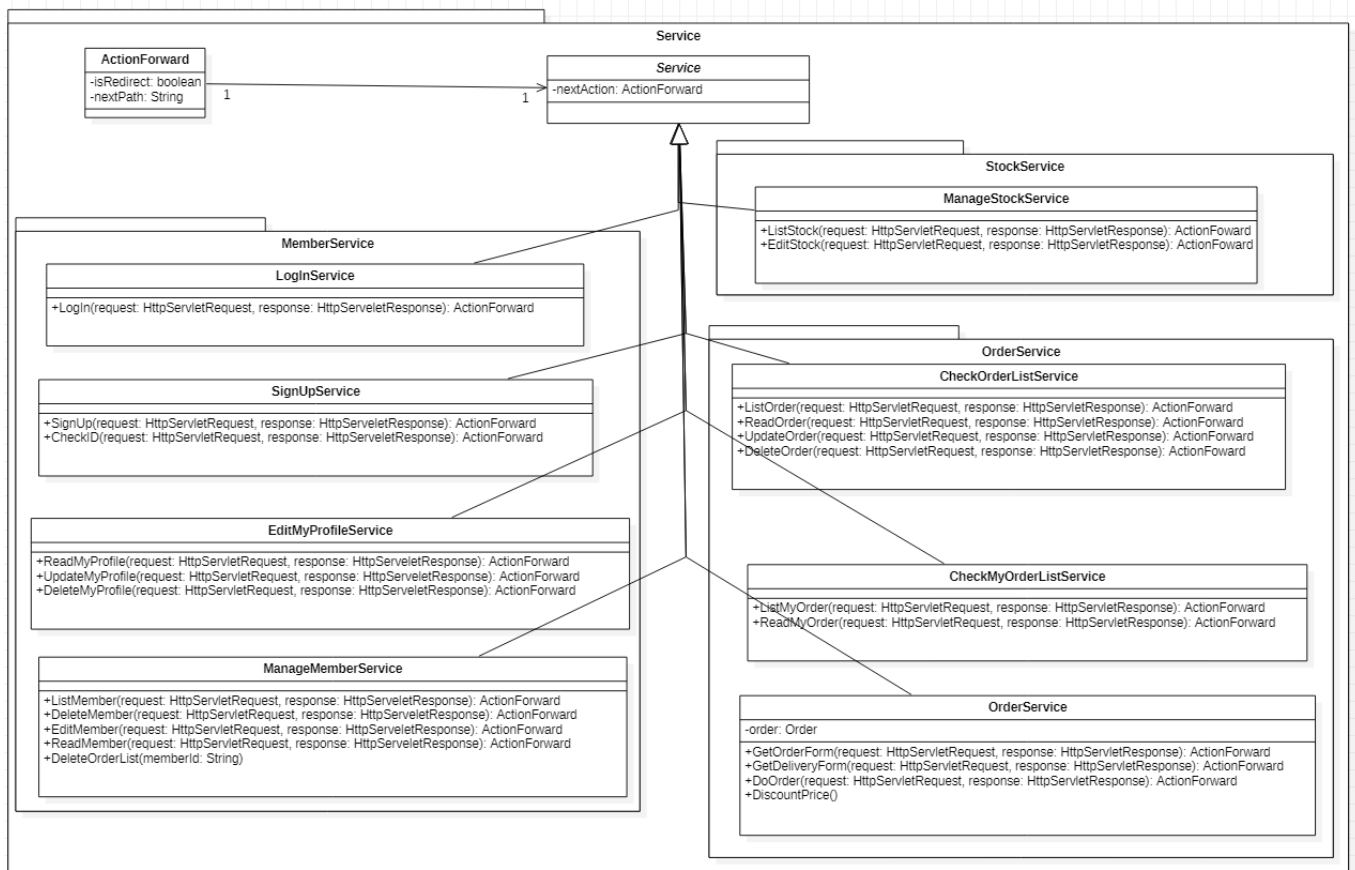
3.1 Controller 클래스도



Controller 내부의 기능에 따라 MemberController, OrderController, StockController 세 패키지로 나뉘어진다.

모든 클래스는 Controller 인터페이스를 구현하여 `doGet()`, `doPost()` 메소드를 가지고 있다.

3.2 Service 클래스도

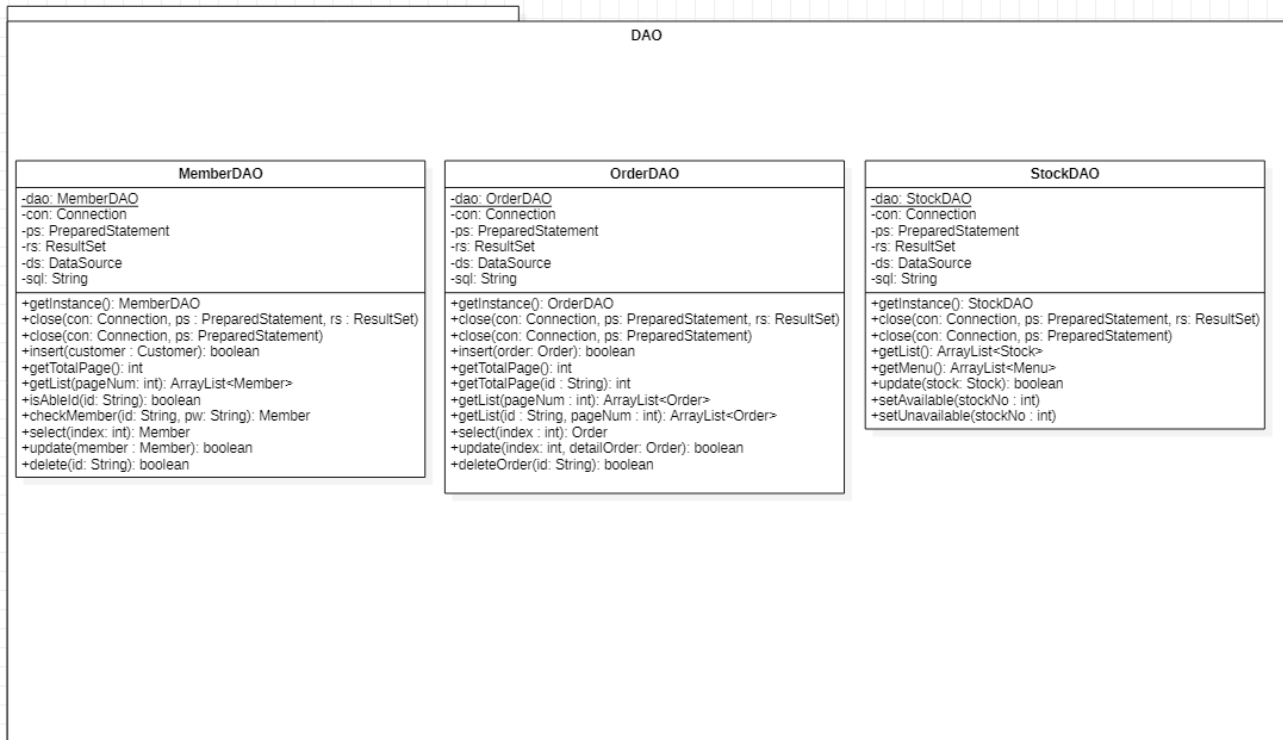


Service 내부의 기능에 따라 MemberService, OrderService, StockService 세 패키지로 나뉘어진다.

Service 클래스는 추상클래스로 속성으로 ActionForward 클래스를 가지고 있다. ActionForward 클래스는 속성으로 재연결여부와 다음 경로에 대한 정보를 가지고 있다.

나머지 클래스들은 모두 Service 클래스를 상속받아 각각의 기능들을 구현한다.

3.3 DAO 클래스도



MemberDAO

- ✓ 멤버와 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- ✓ 데이터베이스 연결을 위한 속성값들이 존재하고, DTO 패키지의 Member 클래스를 활용한다.
- ✓ 속성값으로 스스로를 전역변수형태로 선언하여 정적 바인딩을 통해 인스턴스가 여러 개 생기는 것을 방지한다.
- ✓ insert, select, update, delete의 메소드를 통해 삽입, 조회, 수정, 삭제의 기능을 수행한다.

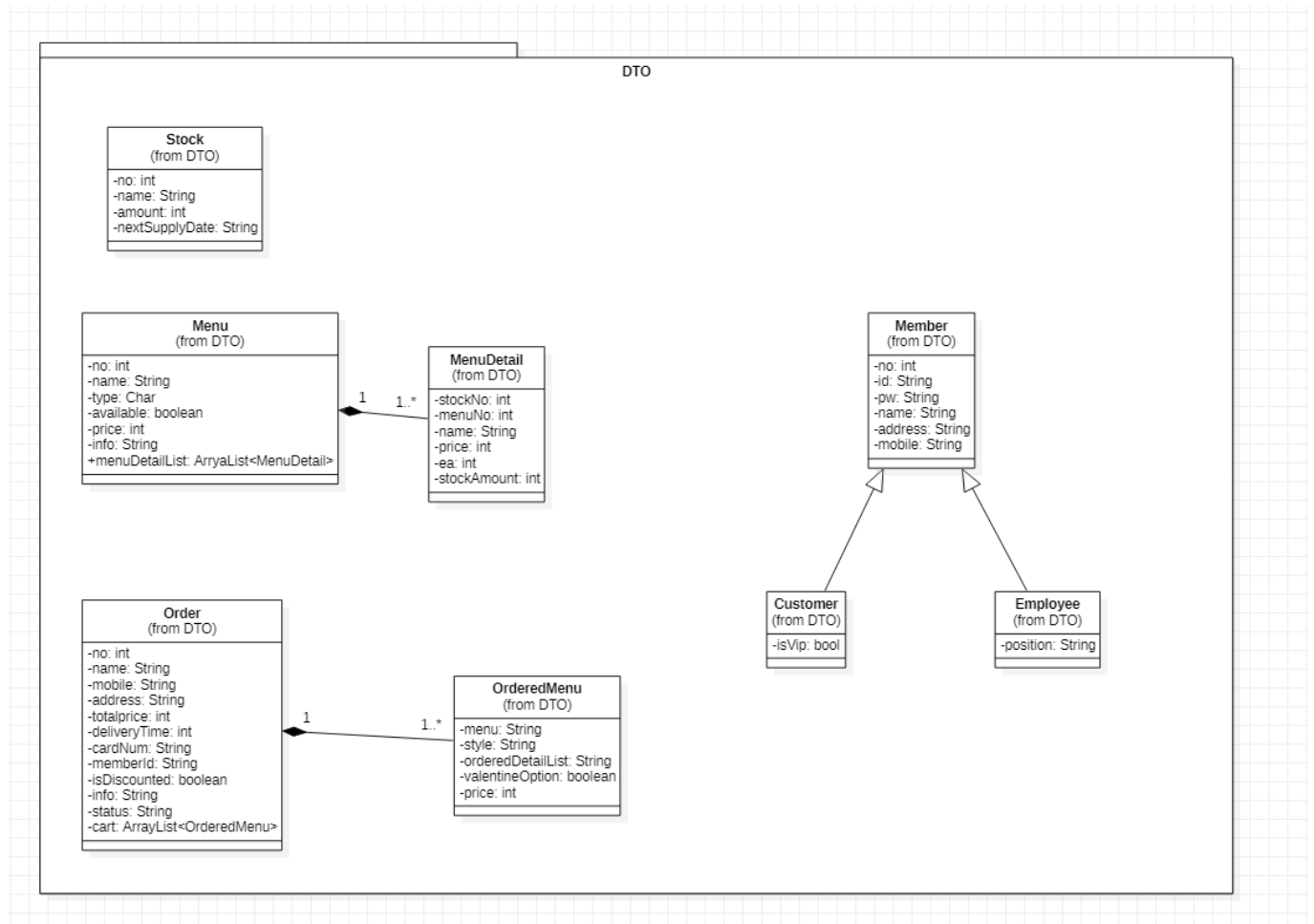
OrderDAO

- ✓ 주문, 메뉴와 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- ✓ 데이터베이스 연결을 위한 속성값들이 존재하고, DTO 패키지의 Order, Menu 클래스를 활용한다.
- ✓ 속성값으로 스스로를 전역변수형태로 선언하여 정적 바인딩을 통해 인스턴스가 여러 개 생기는 것을 방지한다.
- ✓ insert, select, update, delete의 메소드를 통해 삽입, 조회, 수정, 삭제의 기능을 수행한다.
- ✓ 주문 리스트를 가져옴에 있어 메소드 오버라이딩을 통해 다양한 상황에 사용될 수 있게 하였다.

StockDAO

- ✓ 재고와 관련된 데이터베이스 테이블에 접근하는 클래스이다.
- ✓ 데이터베이스 연결을 위한 속성값들이 존재하고, DTO 패키지의 Stock 클래스를 활용한다.
- ✓ 속성값으로 스스로를 전역변수형태로 선언하여 정적 바인딩을 통해 인스턴스가 여러 개 생기는 것을 방지한다.
- ✓ insert, select, update, delete의 메소드를 통해 삽입, 조회, 수정, 삭제의 기능을 수행한다.
- ✓ 주문 리스트를 가져옴에 있어 메소드 오버라이딩을 통해 다양한 상황에 사용될 수 있게 하였다.
- ✓ setAvailable()과 setUnavailable()은 예외적으로 파생테이블을 통해 메뉴테이블에 접근하여 해당 재고가 속한 메뉴의 속성을 수정하게 한다.

3.4 DTO 클래스도



Member

- ✓ 사용자의 정보를 캡슐화하는 단위이다. 데이터베이스에서 멤버 테이블에 대응한다.
- ✓ 고객을 의미하는 Customer와 직원을 의미하는 Employee 가 각각 상속을 받는다.
- ✓ Customer는 추가적으로 단골손님(Vip)여부를, Employee는 직책(position)을 속성으로 가진다.

Order

- ✓ 주문 내용에 대한 정보를 담은 클래스이다. 데이터베이스에서 주문 테이블에 대응한다.
- ✓ 이름, 전화번호, 주소, 주문시간, 카드번호, 가격, 할인여부, 아이디, 추가정보, 상태 속성을 가진다.
- ✓ 또한 한 번의 주문에 여러 개의 디너 선택이 있으므로, 주문 메뉴(각각의 디너 세트)에 대한 정보를 담은 OrderedMenu의 리스트를 속성으로 가진다.

OrderedMenu

- ✓ 주문한 메뉴 각각에 대한 정보를 담은 클래스이다. Order객체와 1대다 집약관계에 있다.
- ✓ 디너, 스타일과 함께 추가음식에 세부사항에 대해 열거한 디테일리스트, 발렌타인 옵션 여부, 가격을 속성으로 가진다.

Menu

- ✓ 기본적인 메뉴(디너 혹은 스타일)에 대한 정보를 담은 클래스이다. 데이터베이스에서 멤버 테이블에 대응한다.
- ✓ 디너 서비스 시스템에서 미리 제공하는 기본적인 디너와 스타일에 대해서 이름, 가능여부, 가격, 추가 정보와 함께 제공되는 음식들의 세부사항을 담은 MenuDetail 객체의 리스트를 속성으로 가진다.

MenuDetail

- ✓ 메뉴에 대한 세부 구성요소에 대한 정보를 담은 클래스이다. Menu객체와 1대다 집약관계에 있다.
- ✓ 메뉴와 재고의 코드번호, 구성요소의 이름, 가격, 메뉴에서의 개수, 구성요소의 재고수를 속성으로 가진다.

Stock

- ✓ Stock 클래스는 재고에 관련된 정보를 담은 클래스이다.
- ✓ DB에서 DAO를 통해 정보를 얻고, UI로 정보를 전송할 때 해당 객체를 사용한다.

4. 서브시스템 클래스도

아키텍처도, 패키지 클래스도는 모듈 측면에서의 표현 관점이었다면, 4장 에서의 클래스도는 사용자에게 제공하는 서비스에 따라 실행시의 사용 관계로 분류한 클래스들을 표현하였다. 시스템을 기능적인 측면에서 크게 3가지의 서브시스템으로 분류하였고, Member, Order, Stock 세가지 서브시스템에 대하여 설명하도록 하겠다.

4.1 Member 서브시스템 세부 설계

Member 시스템은 전체 시스템에서 재고와 관련된 동작에 연관된 서브시스템이다. UI 패키지, DAO패키지와 상호작용하여 유즈케이스 중 회원 가입, 로그인, 개인정보 관리, 회원 관리 부분을 수행한다.

Member 시스템은 다음과 같은 중요한 함수들로 구성되어 있다. 각 함수는 Member 시스템에서 중요한 기능적인 역할을 수행한다.

SignUp

- ✓ 회원 가입을 수행한다.

LogIn

- ✓ 로그인을 수행한다.

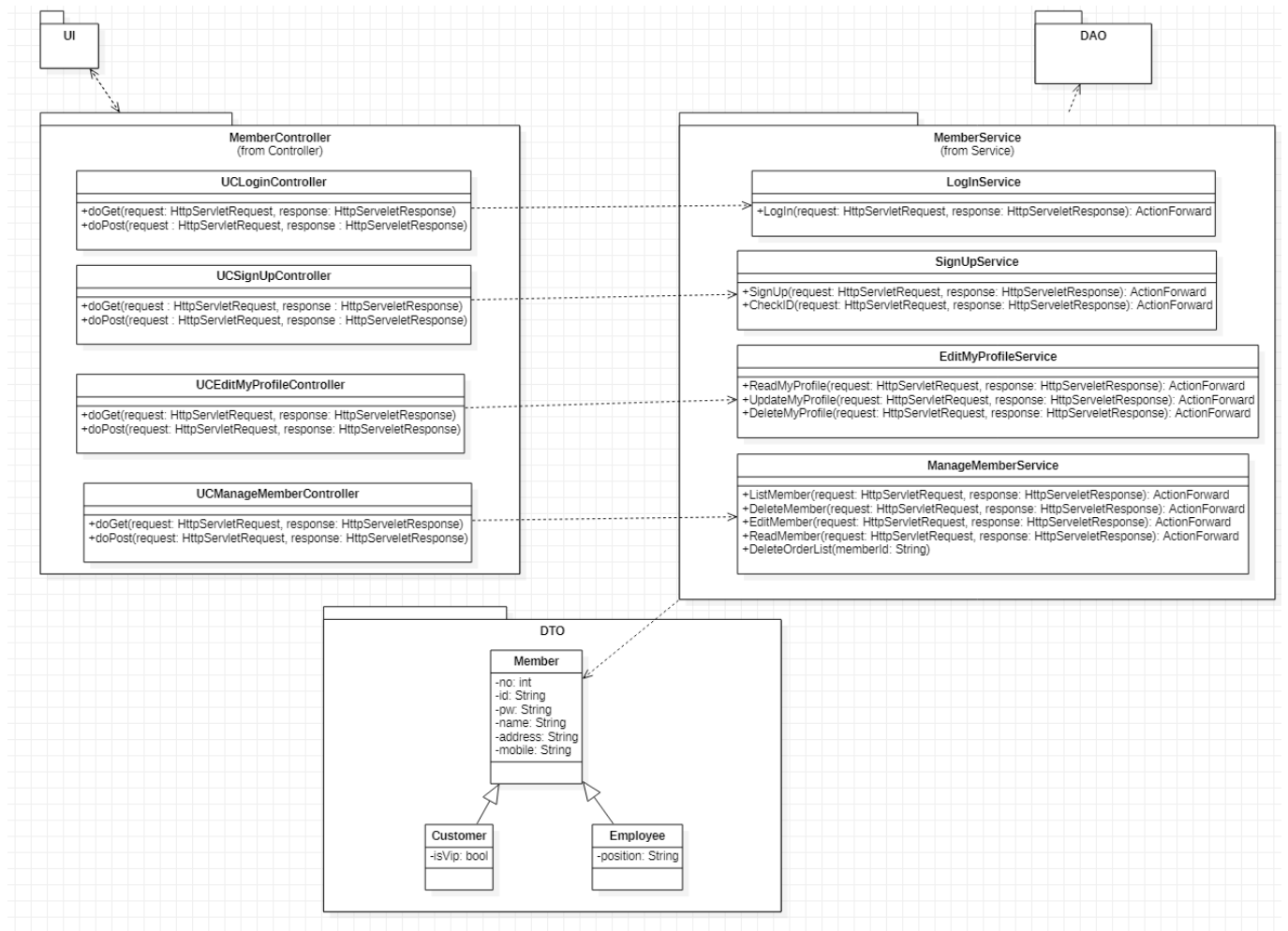
UpdateMyProfile

- ✓ 개인정보 수정을 수행한다

EditMember

- ✓ 회원정보 수정을 수행한다.

4.1.1 메인 클래스도



UCSignUpController

- ✓ 회원 가입과 관련된 UI와의 상호작용을 처리한다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.
- ✓ 사용자가 UI를 통해 서비스를 요청할 때, 컨트롤러가 이를 받아서 서비스로 연결해주고 서비스에서 요청을 받아올 때 대신 받아온다.

UCLoginController

- ✓ 로그인과 관련된 UI와의 상호작용을 처리한다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.

UCEditMyProfileController

- ✓ 개인 수정과 관련된 UI와의 상호작용을 처리한다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.

UCManageMemberController

- ✓ 회원 관리와 관련된 UI와의 상호작용을 처리한다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.

SignUpService

- ✓ 회원 가입을 담당하는 객체이다.
- ✓ UCSignUpController로부터 메시지를 받아 DAO를 호출하는 방식으로 기능을 수행하며, CheckID 함수로 아이디 중복 확인, SignUp함수로 회원 가입을 수행한다.

LoginService

- ✓ 로그인을 담당하는 객체이다.
- ✓ UCLoginController로부터 메시지를 받아 DAO를 호출하는 방식으로 기능을 수행하며, Login 함수로 기능을 수행한다.

EditMyProfileService

- ✓ 개인 정보 수정을 담당하는 객체이다.
- ✓ UCEditMyProfileController로부터 메시지를 받아 DAO를 호출하는 방식으로 기능을 수행하며, ReadMyProfile, UpdateMyProfile, DeleteMyProfile 함수로 개인정보 조회, 수정, 삭제 기능을 수행한다.

ManageMemberService

- ✓ 회원 관리를 담당하는 객체이다.
- ✓ UCMangeMemberController로부터 메시지를 받아 DAO를 호출하는 방식으로 기능을 수행하며, ListMemeber, DeleteMember, EditMember, DeleteMember함수로 회원 리스트 조회, 정보 조회, 수정, 삭제 기능을 수행한다.

4.2 Order 서브시스템 세부 설계

Order 시스템은 전체 시스템에서 주문과 관련된 동작에 연관된 서브시스템이다. UI 패키지, DAO패키지와 상호 작용하여 유즈케이스 중 주문하기, 내 주문목록 확인, 주문 확인 부분을 수행한다.

Member 시스템은 다음과 같은 중요한 함수들로 구성되어 있다. 각 함수는 Member 시스템에서 중요한 기능적인 역할을 수행한다.

DoOrder

- ✓ 고객이 입력한 주문정보를 바탕으로 주문을 수행한다.

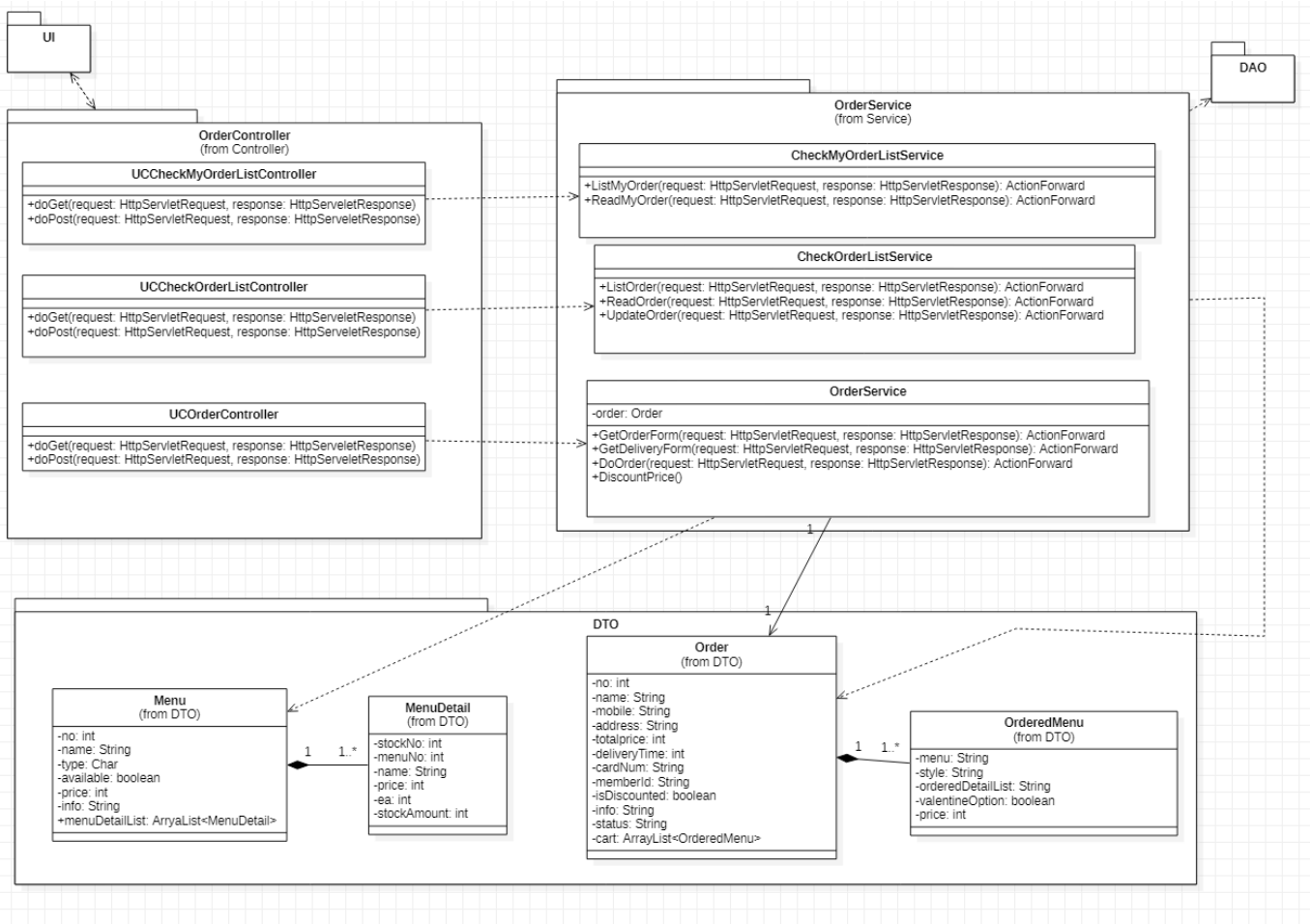
UpdateOrder

- ✓ 주문에 대한 상태를 업데이트하는 작업을 수행한다.

ReadMyOrder

- ✓ 로그인된 아이디의 주문에 대한 정보를 제공하는 작업을 수행한다.

4.2.1 메인 클래스도



UCOrderController

- ✓ 주문과 관련된 UI와의 상호작용을 처리한다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.

UCCheckOrderListController

- ✓ 주문 확인과 관련된 UI와의 상호작용을 처리한다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.

UCCheckMyOrderListController

- ✓ 내 주문목록 확인과 관련된 UI와의 상호작용을 처리한다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.

OrderService

- ✓ 주문에 대한 동작을 담당하는 객체이다.
- ✓ UCOOrderController로부터 메시지를 받아 DAO를 호출하는 방식으로 기능을 수행하며, GetOrderForm, GetDeliveryForm, DoOrder함수로 주문을 위한 정보 제공 및 수집, 배달을 위한 정보 제공 및 수집, 주문기능을 수행한다.

CheckOrderListService

- ✓ 주문 확인에 대한 동작을 담당하는 객체이다.
- ✓ UCCheckOrderListController로부터 메시지를 받아 DAO를 호출하는 방식으로 기능을 수행하며, ListOrder, ReadOrder, UpdateOrder 함수로 전체 주문목록 조회, 정보 조회, 상태 수정 기능을 수행한다.

CheckMyOrderListService

- ✓ 내 주문 확인에 대한 동작을 담당하는 객체이다.
- ✓ UCCheckMyOrderListController로부터 메시지를 받아 DAO를 호출하는 방식으로 기능을 수행하며, ListMyOrder, ReadOrder 함수로 내 주문목록 조회, 정보 조회 기능을 수행한다.

4.3 Stock 서브시스템 세부 설계

Stock 시스템은 전체 시스템에서 재고와 관련된 동작에 연관된 서브시스템이다. UI 패키지, DAO패키지와 상호 작용하여 유즈케이스 중 재고관리 부분을 수행한다.

Stock 시스템은 다음과 같은 중요한 함수들로 구성되어 있다. 각 함수는 Stock 시스템에서 중요한 기능적인 역할을 수행한다.

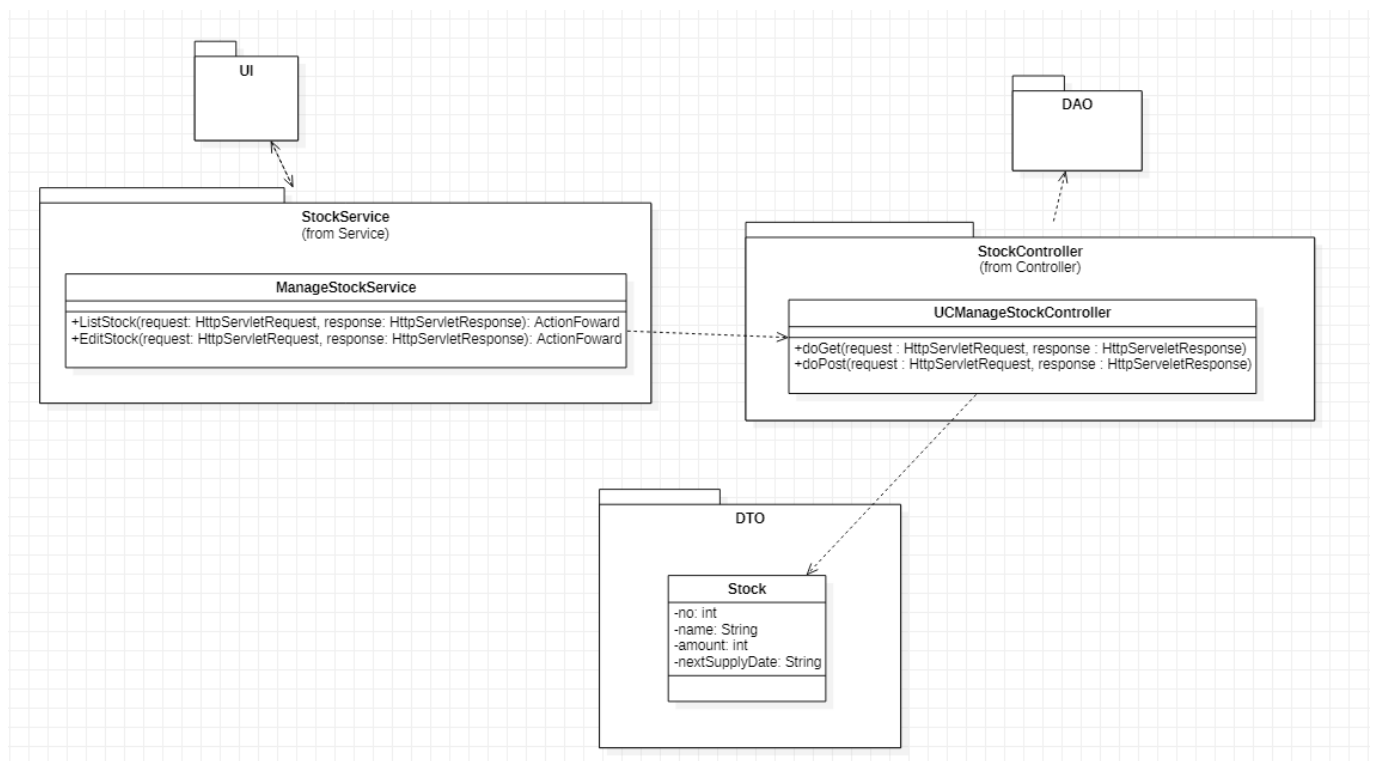
ListStock

- ✓ 전체 재고 리스트를 요청한다.

EditStock

- ✓ 재고 정보 수정을 요청한다.

4.3.1 메인 클래스도



UCManageStockController

- ✓ UI와 Service를 연결하는 객체이다. requestURI에 따라 해당하는 서비스객체로 메시지를 전송한다.

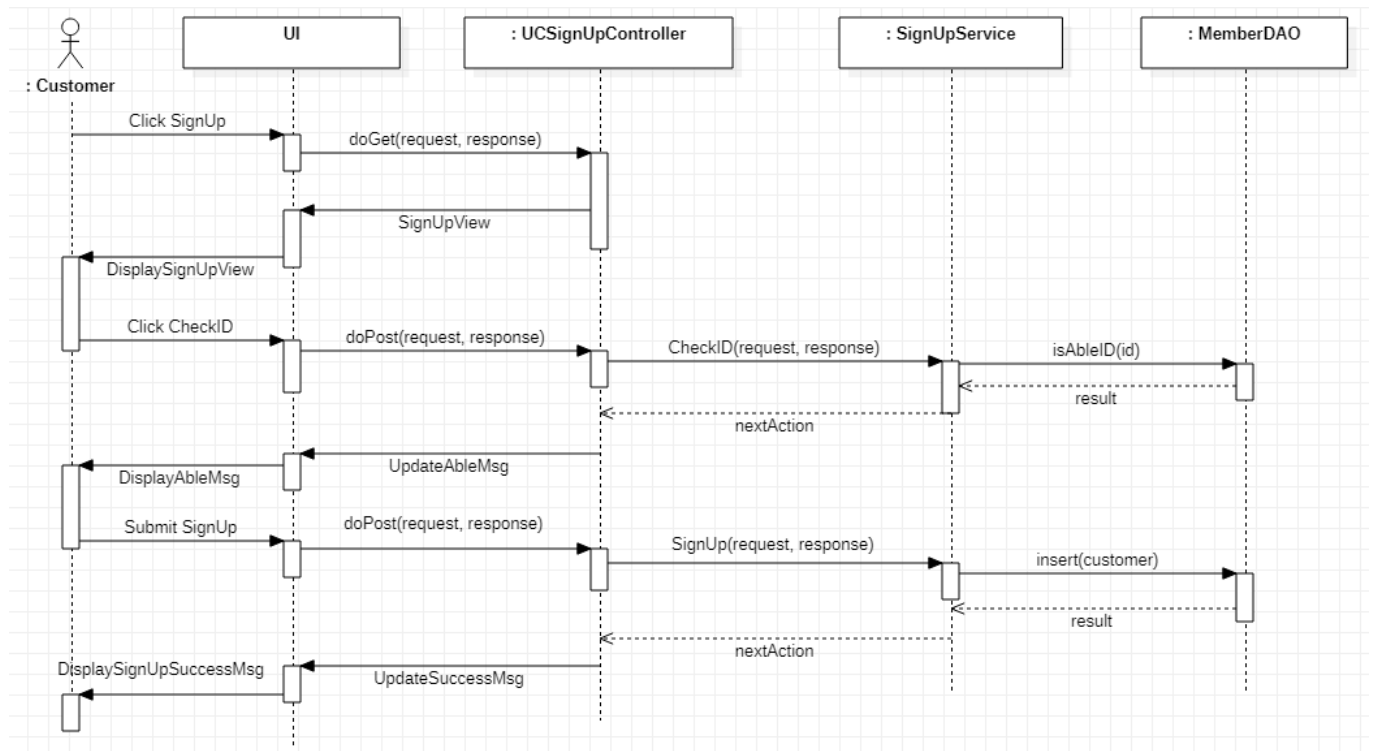
StockActionForward

- ✓ 서비스에서 다음 경로를 저장하는 객체이다. 컨트롤러에서 판단하여 웹 페이지 혹은 메소드를 실행한다.

ManageStockService

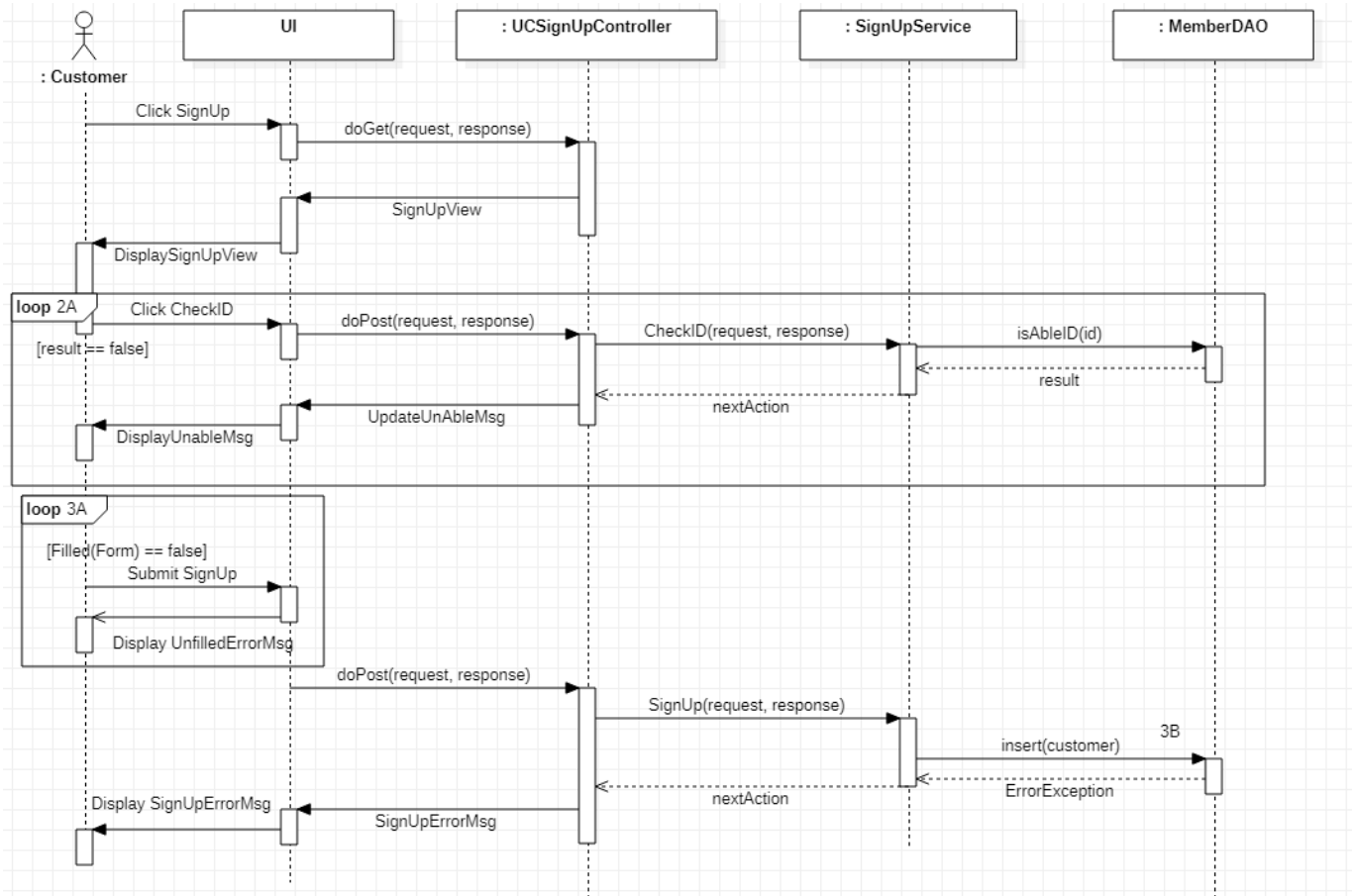
- ✓ 재고 관리를 담당하는 객체이다. 컨트롤러로부터의 메시지를 받아 DAO로 메시지를 전송하여 재고 리스트 얻기, 재고 수정을 수행한다.

5.1 교류도 (usecase: 회원가입 (SignUp))



정상흐름

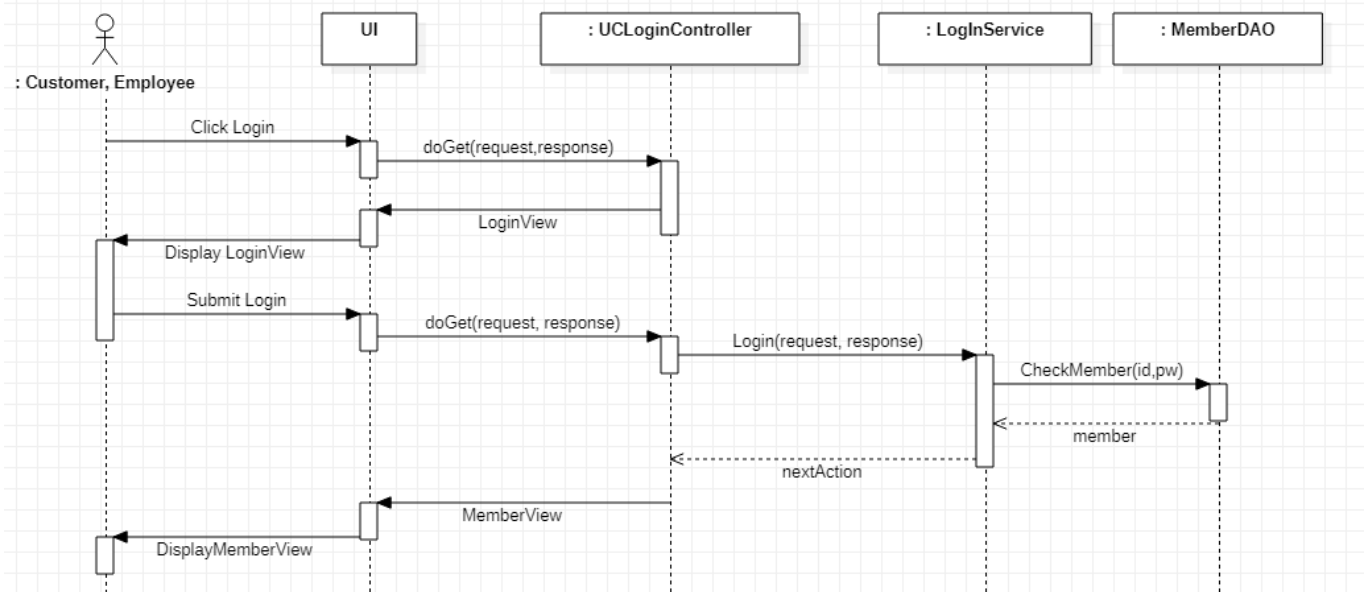
1. Sign Up을 클릭하면 UI에서 View를 요청하여 SignUp View를 띄운다.
2. ID가 적절한지 검사를 위해 CheckID가 실행되고 MemberDAO 객체에서 isAbleID를 통해 사용가능한 ID인지를 체크하여 가능여부를 결정한다.
3. 가입을 클릭하면 SignUpService에서 정보를 받아 MemberDAO에 insert메소드를 호출한다. 성공적으로 DB에 저장 후 true가 리턴되면 UpdateSuccessMsg를 띄운다.



대안흐름

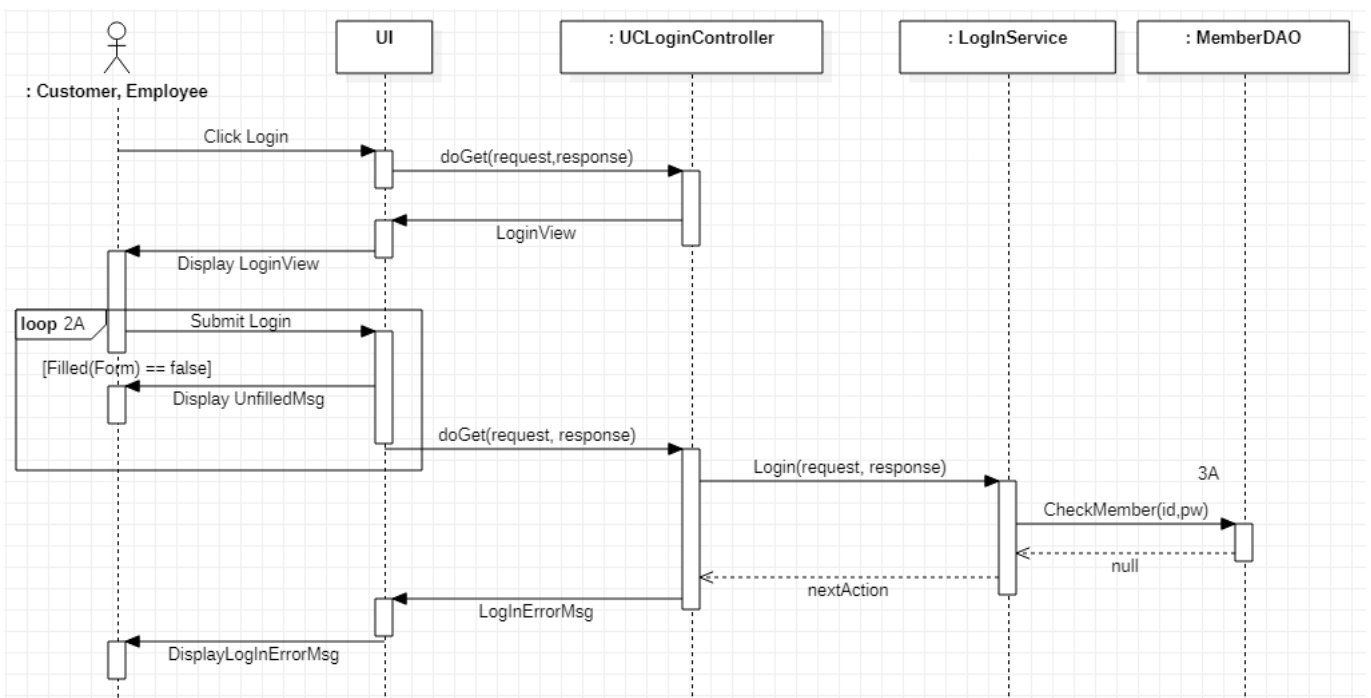
- 2A.
1. 중복되는 아이디가 DB에 존재하여 isAbleID의 result가 false일 경우 UnableMsg를 띄우고 다시 Id 입력을 받는다.
- 3A.
1. SignUp을 실행후에 UI에서 빈 양식이 있는 경우 UnfilledErrorMsg를 띄워 사용자가 마저 입력하게 한다.
- 3B.
1. Insert에 실패해 발생한 경우 Controller를 통해 SignUpErrorMsg를 UI로 전달해 디스플레이 한다.

5.2 교류도 (usecase: 로그인 하기(Login))



기본 흐름

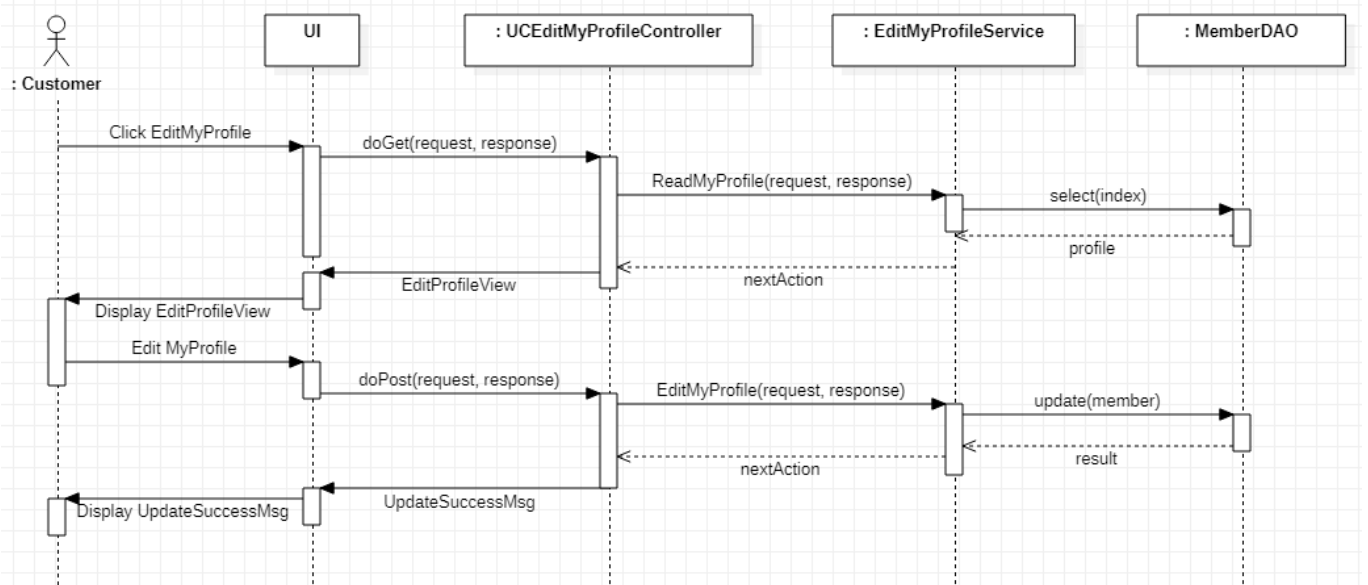
1. 고객이 로그인을 누르면 doGe()메소드를 실행해 Controller가 Login View를 UI에 보낸다.
2. 고객이 양식을 제출하면 LoginService의 Login()으로 MemberDAO 객체의 CheckMember를 호출한다.
3. MemberDAO에서 성공적으로 로그인한 고객의 정보인 member를 받아오면 Controller를 거쳐 화면에 알맞은 MemberView를 띄운다.



대안 흐름

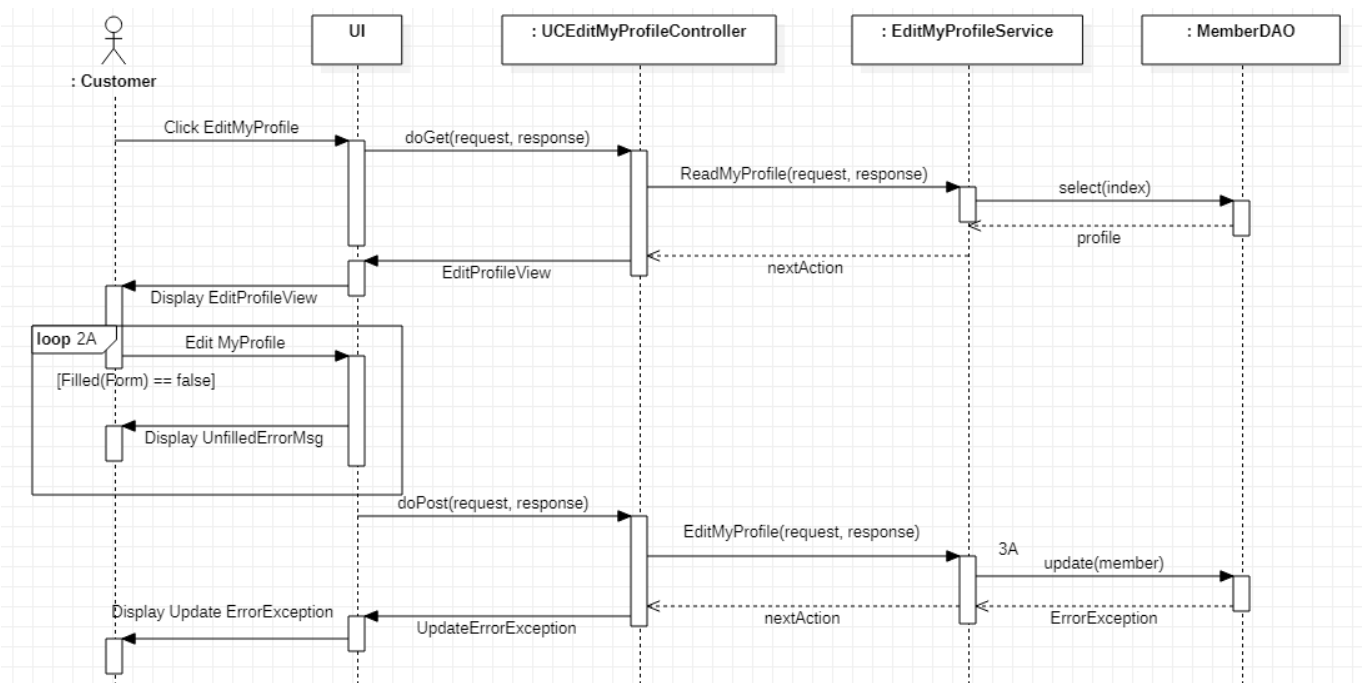
- 2A. 양식이 제대로 입력돼있지 않으면 Display UnfilledMsg를 실행한다.
- 3A. checkMember를 실행 후에 멤버를 찾지 못하면 null값이 반환되는데, 이후 Controller를 거쳐 사용자에게 LoginErrorMsg를 띄운다.

5.3 교류도 (usecase: 개인정보 수정 (Edit My Profile))



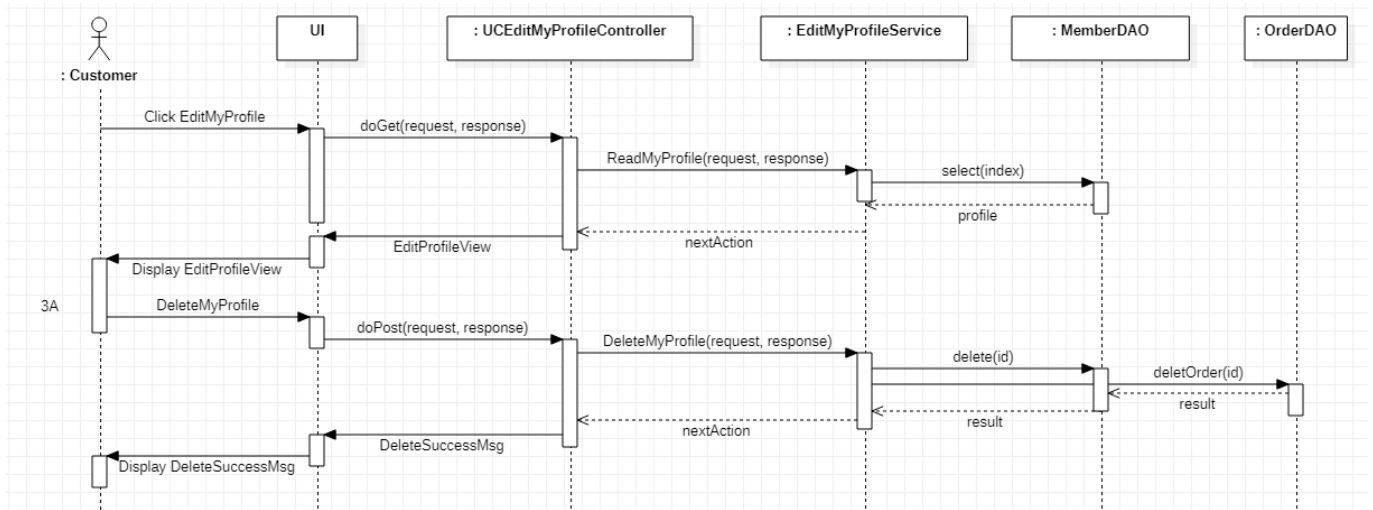
기본 흐름

1. 고객이 내정보수정을 누르면 doGet메소드를 실행하여 EditMyProfileService의 ReadMyProfile함수를 실행해 내정보를 가져와 Controller가 정보수정 화면을 UI에 보낸다.
2. 고객이 수정 후 정보를 제출하면 컨트롤러가 doPost메소드를 실행한다.
3. EditMyProfile 함수를 실행해 MemberDAO객체의 update()함수를 실행하여 편집된 정보를 저장한 뒤 성공 메시지를 띄운다.



대안 흐름 (2A, 3A)

- 2A.
 1. 올바르지 않은 양식일 경우 UI에서 UnfilledErrorMsg를 띄운 뒤에 다시 Profile View에서 대기한다.
- 3A.
 1. MemberDAO의 update함수를 실행한 뒤에 Error가 발생하면 UpdateErrorException 메세지를 띄운다.

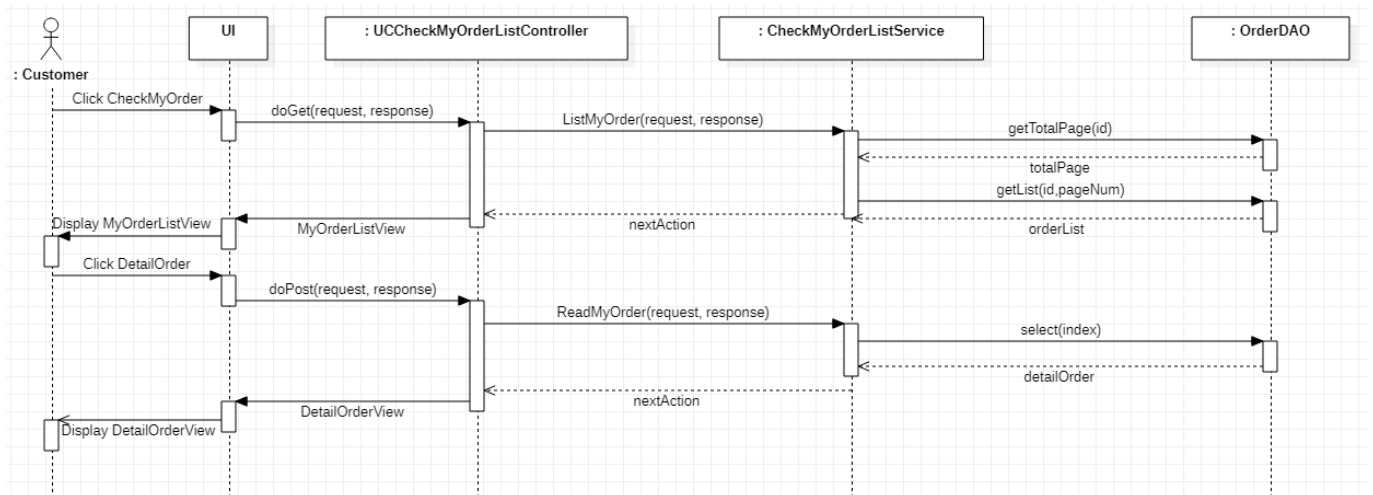


대안 흐름 (2B)

2B

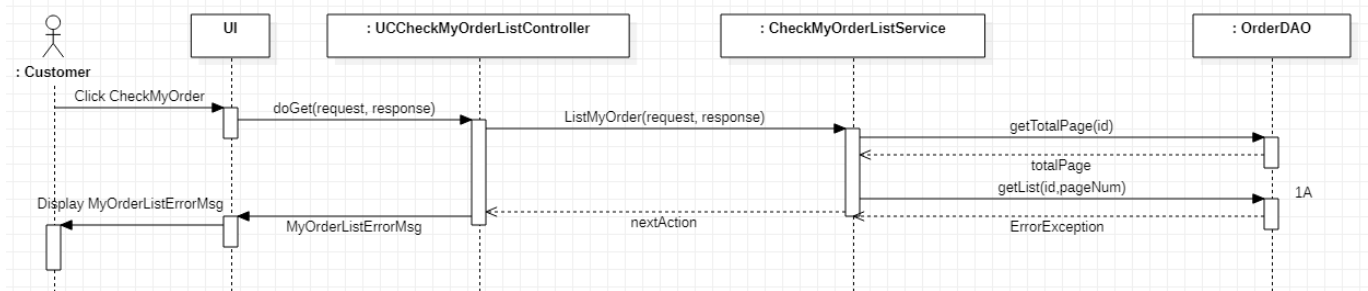
1. 고객이 정보 삭제를 할 경우 EditMyProfileService 객체의 DeleteMyProfile 메소드를 실행하여 MemberDAO의 delete 함수를 통해 고객정보를 삭제한다.
2. OrderDAO의 delteOrder 함수를 통해 해당 고객의 주문정보 역시 삭제하고, 성공적으로 완료 후 DeleteSuccessMsg를 디스플레이하며 종료한다.

5.4 교류도 (usecase: 내 주문목록 확인 (Check My OrderList))



기본 흐름

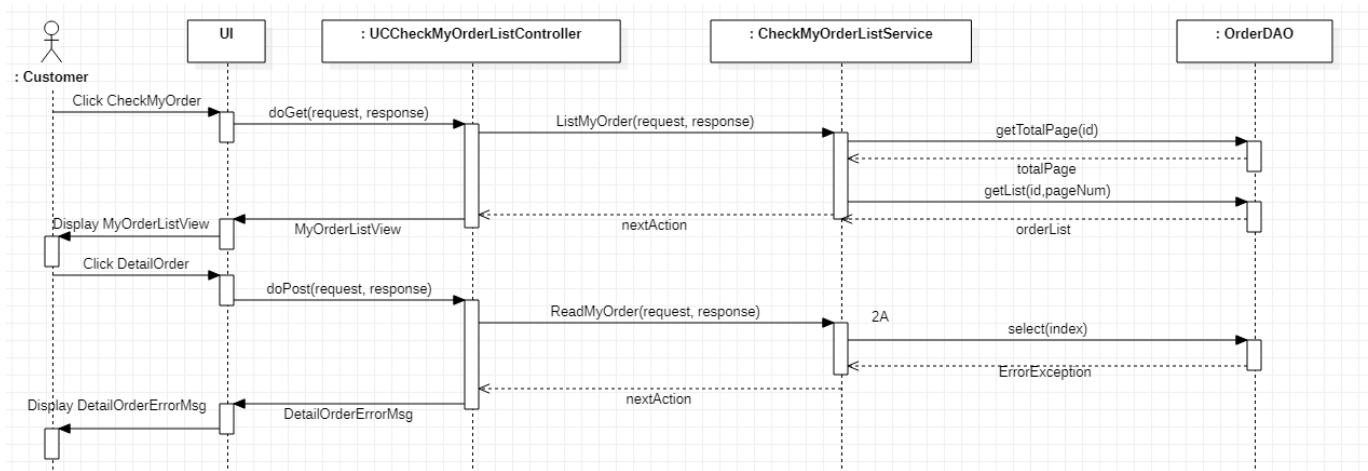
1. 내 주문목록 확인을 클릭하면 CheckMyOrderListService의 ListMyOrder메소드를 실행하여 OrderDAO 객체의 getTotalPage를 실행하여 전체 페이지 수를 얻고, 맨 앞 페이지부터 getList 메소드를 통해 orderList를 받아와 MyOrderListView를 띄운다.
2. 특정 주문목록을 클릭하면 ReadMyOrder 함수를 실행하여 OrderDAO의 select(index) 메소드를 통해 detailOrder를 받아와서 DetailOrderView에 띄운다.



대안 흐름 (1A)

1A.

1. getList메소드의 실행에 Error가 발생하면 MyOrderListErrorMsg를 띄우고 유스케이스를 종료한다.

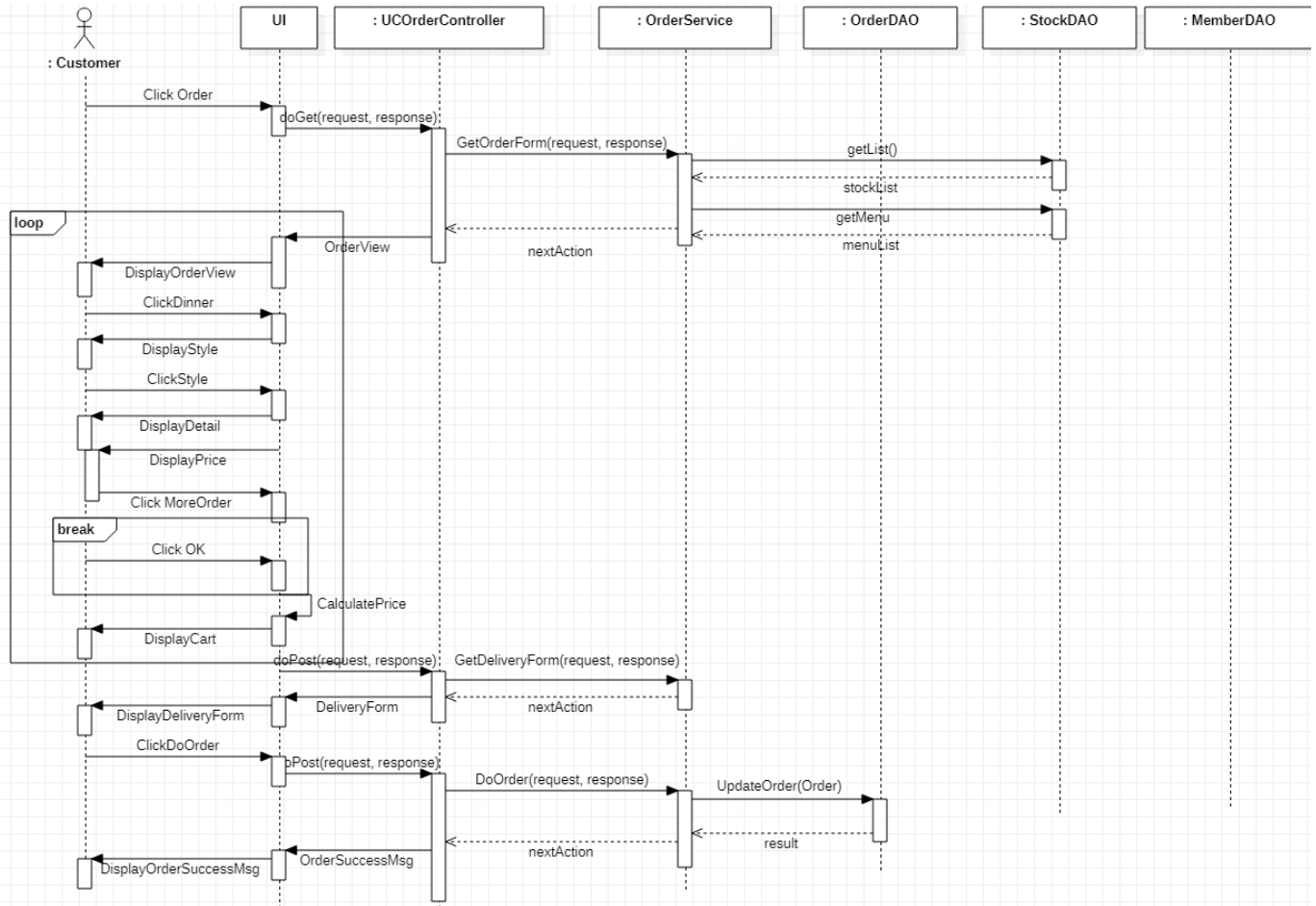


대안 흐름 (2A)

2A.

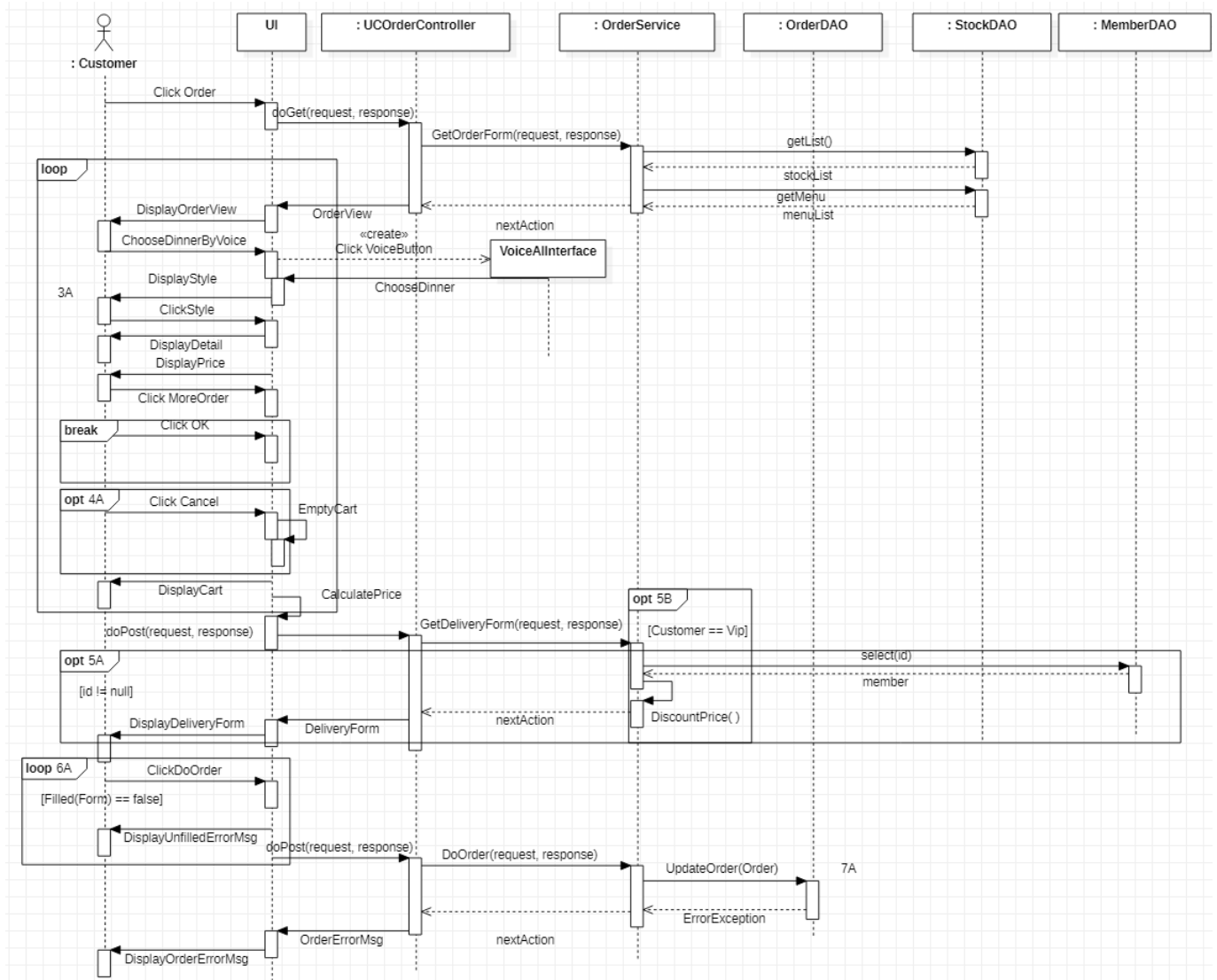
1. select(index) 메소드의 실행 후에 Error가 발생하면 DetailOrderErrorMsg를 띄우고 유스케이스를 종료한다.

5.5 교류도 (usecase: 주문하기 (Order))



기본 흐름

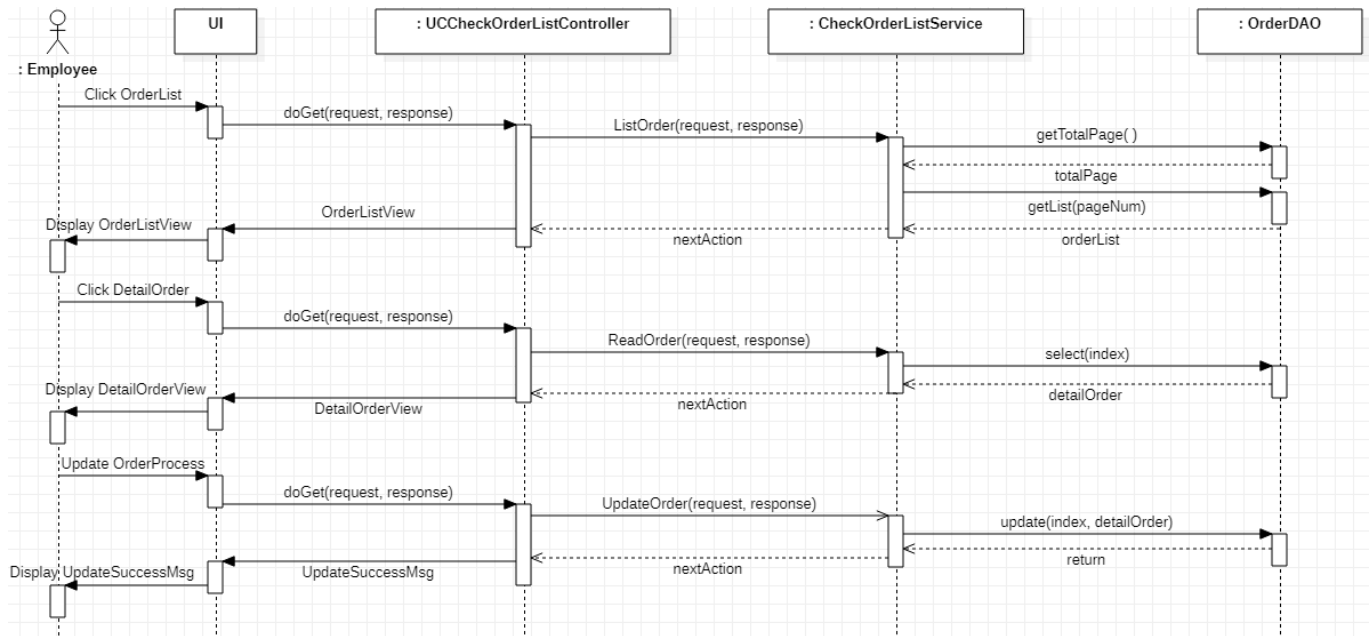
1. 고객이 주문하기를 클릭하면, UCOrderController에서 OrderService의 GetOrderForm()을 실행한다.
2. GetOrderForm()은 StockDAO의 getList(), getMenu()를 호출하여 전체 디너 메뉴와 재고의 리스트를 받아온다.
3. 고객에게 주문화면에서 전체 디너 메뉴와 재고를 바탕으로 디너, 스타일, 추가음식을 고르게 하고 가격까지 제시한다.
4. 고객이 더주문하기 버튼을 누르면, 해당 세트의 가격을 계산한 다음 정보를 묶어 장바구니 항목으로 디스플레이한다.
5. 고객이 모든 선택을 완료하면 UCOrderController에서 배달 양식을 디스플레이한다.
6. 고객이 주문하기를 누르면 UCOrderController에서 OrderService의 DoOrder()를 실행시켜 OrderDAO의 UpdateOrder를 통해 주문을 DB에 저장하게 한다.
7. 이후 정상적으로 완료되어 result로 true가 리턴되면 OrderSuccessMsg를 디스플레이하고 유스케이스를 종료한다.



대안 흐름

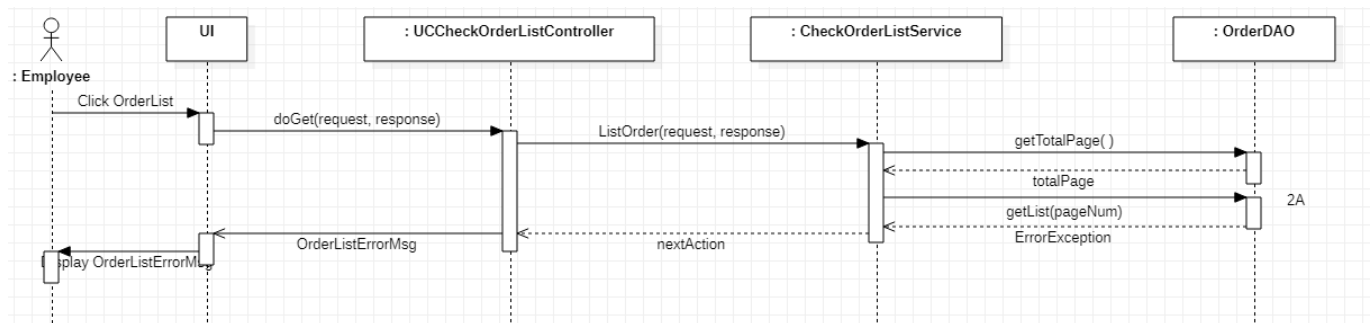
- 3A
- 음성 인식 버튼을 누르고 마이크를 통해 디너의 이름을 음성으로 선택하려 하면, 외부 음성인식 인터페이스를 통해 해당하는 디너를 자동으로 선택한다.
- 4A
- 고객이 취소 버튼을 선택하면 장바구니의 모든 항목을 비운다.
- 5A
- 로그인 되어 있어서 id가 null이 아니라면, MemberDAO의 select 함수를 통해 시스템에서 고객의 정보를 받아와 이후 DeliveryForm에 함께 전송시켜 양식에 미리 적힌 채로 디스플레이 한다.
- 5B
- 고객의 정보가 Vip라면 DiscountPrice를 통해 금액을 할인한다.
- 6A
- DoOrder를 진행할 때 배달의 양식이 모두 입력되지 않았다면, DisplayUnfilledErrorMsg를 디스플레이 한다.
- 7A
- OrderDAO에서 UpdateOrder에 에러가 발생하였다면, OrderErrorMsg를 디스플레이하고 유스케이스를 종료한다.

5.6 교류도 (usecase: 주문 확인 (Check OrderList))



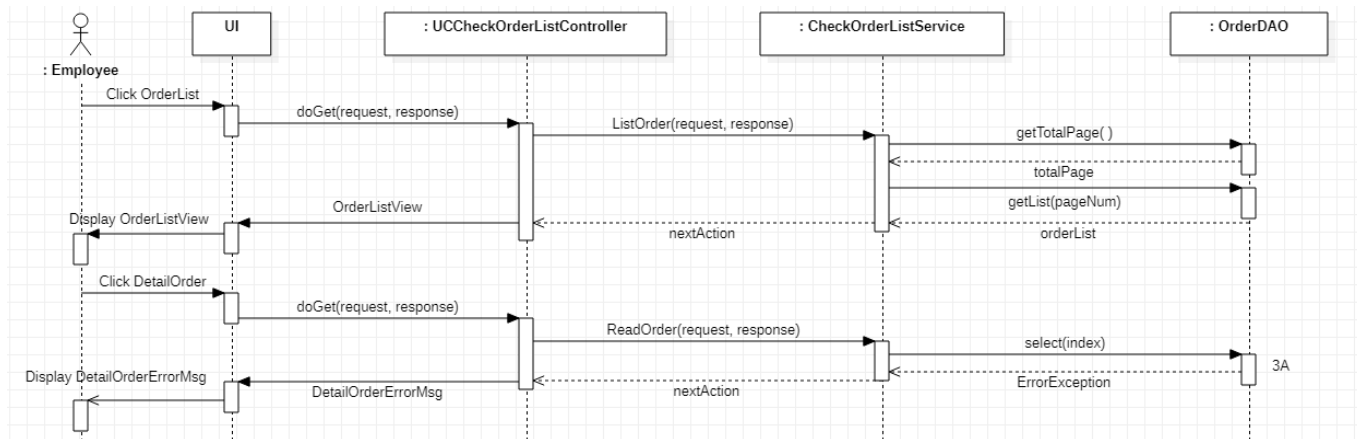
기본 흐름

1. 주문 확인을 클릭하면 CheckOrderListService의 ListOrder메소드를 실행한다.
2. OrderDAO의 getTotalPage()로 전체 페이지수를, getList로 맨 첫 페이지부터의 orderList를 받아와 OrderListView를 디스플레이한다.
3. 사용자가 특정 주문목록을 클릭하면 CheckOrderListService의 ReadOrder함수를 통해 OrderDAO의 select()를 실행하여 detailOrder를 가져와 DetailOrderView를 띄운다.
4. 주문상태를 업데이트 하면 CheckOrderListService의 UpdateOrder를 실행하여 OrderDAO의 update함수를 실행한 뒤, UpdateSuccessMsg를 띄운다.



대안 흐름(2A)

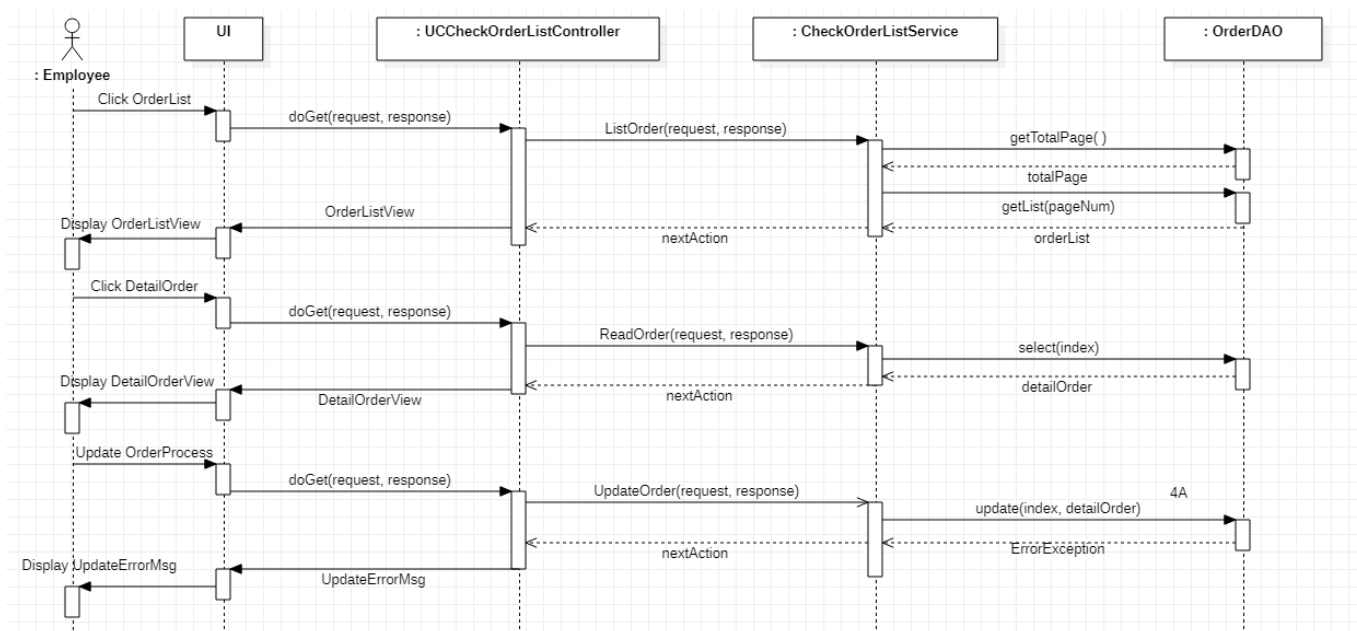
- 2A.
1. OrderDAO객체의 getList메소드를 실행할때 Error가 발생하면 OrderListErrorMsg를 디스플레이하고 유스케이스를 종료한다.



대안 흐름(3A)

3A.

1. OrderDAO객체의 select메소드를 실행할 때 Error가 발생하면 OrderListErrorMsg를 디스플레이하고 유스케이스를 종료한다.

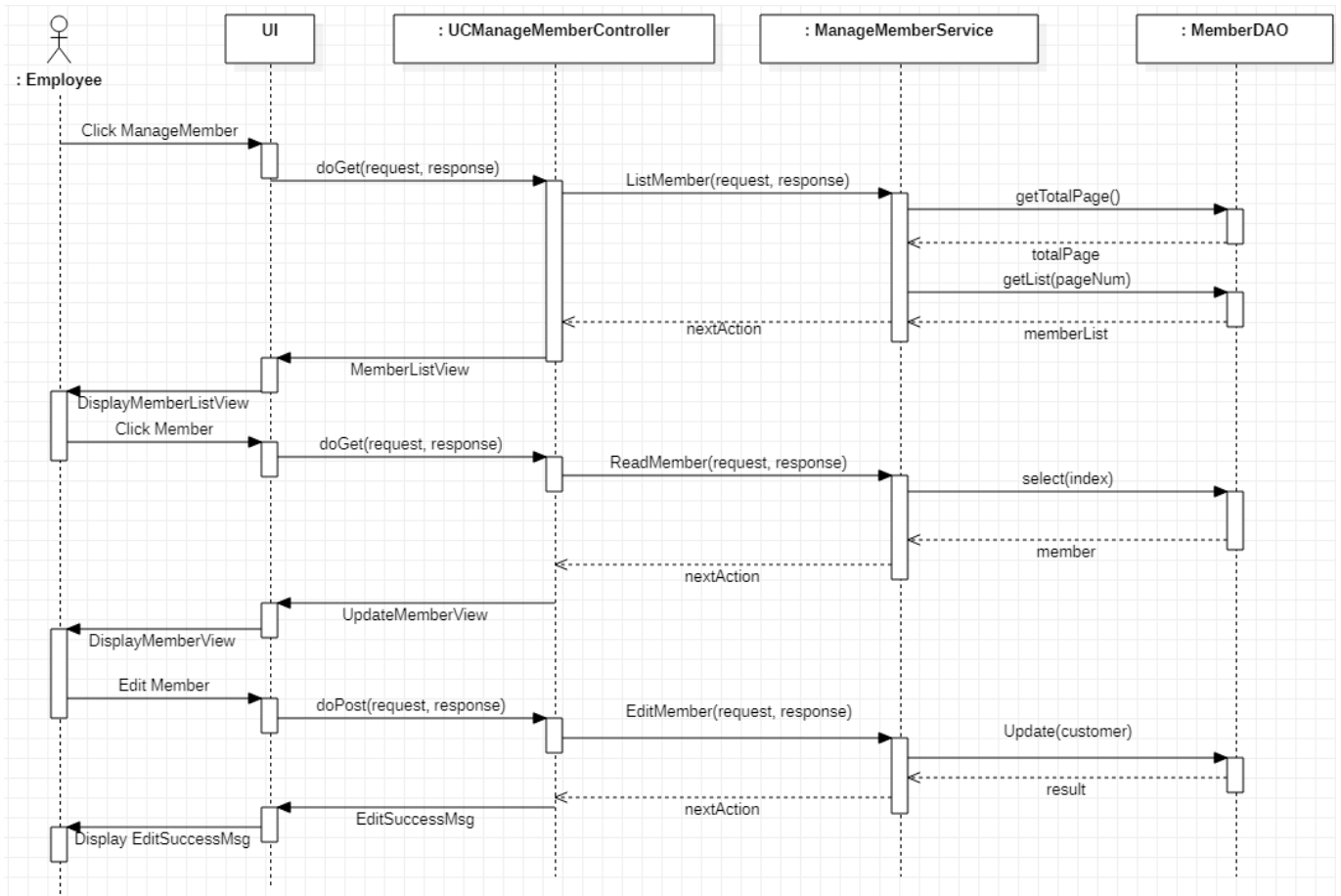


대안 흐름(4A)

4A.

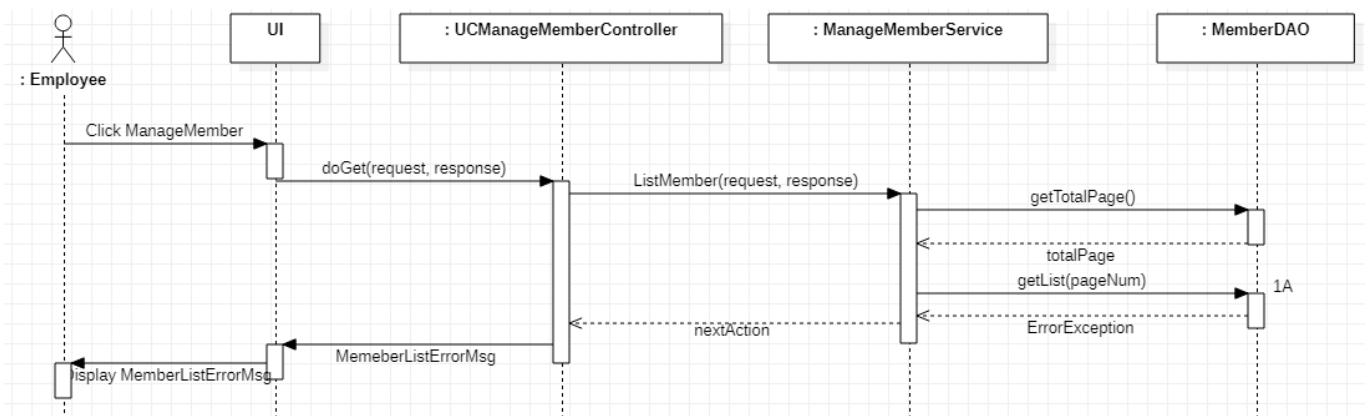
1. 주문상태의 업데이트를 위한 update 메소드를 실행할때 Error가 발생하면 CheckOrderListController를 거쳐 UI에 UpdateErrorMsg를 디스플레이하고 유스케이스를 종료한다.

5.7 교류도 (usecase: 회원 관리 (Manage Member))



기본 흐름

1. 회원관리를 선택하면 MemberDAO 객체에서 현재 pageNum인덱스를 주어서 해당 페이지의 memberList를 받아와 MemberListView를 나타낸다.
2. 특정 Member를 클릭하면 ReadMember메소드를 실행해 Member DB에서 select(index)를 통한 멤버정보를 얻어와 MemberView를 나타낸다.
3. 해당 MemberView를 수정하고 ManageMmberService객체의 EditMember메소드를 실행하여 MemberDAO Update 함수를 통해 result를 받아와 SuccessMsg를 띄운다.

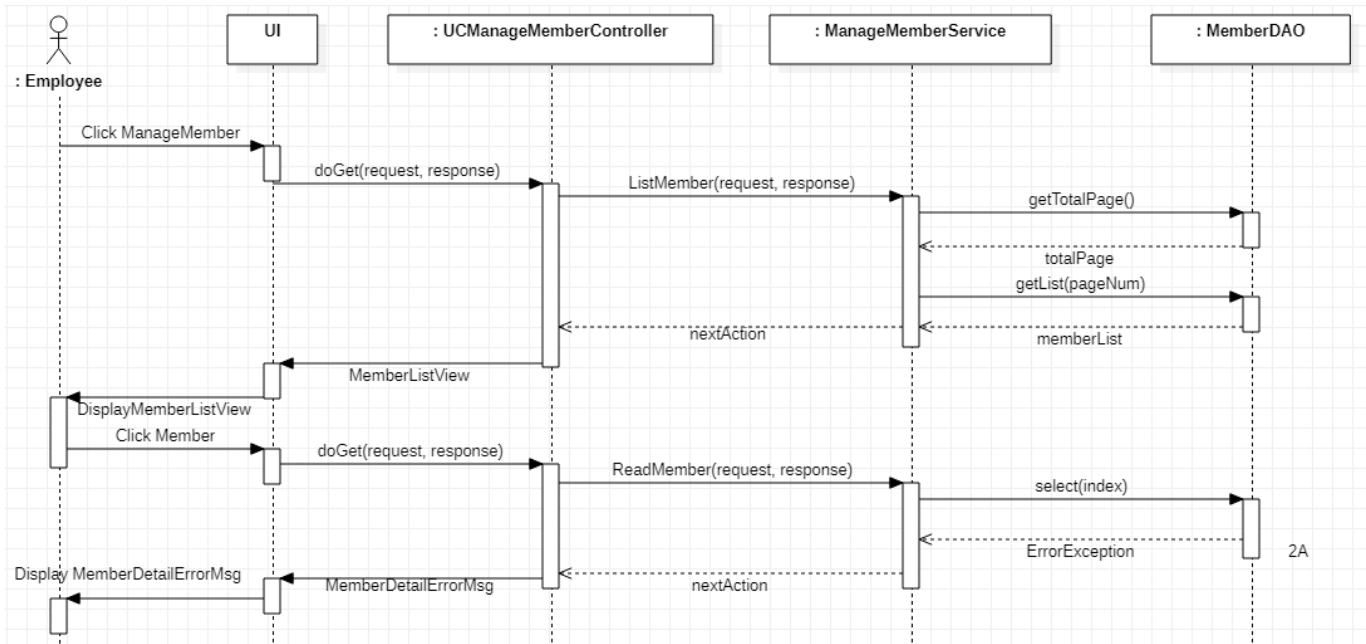


대안 흐름(1A)

1A

1. MemberDAO객체에서 전체 memberList를 받아오는 것이 Error 가 발생할 경우 MemberListErrorMsg

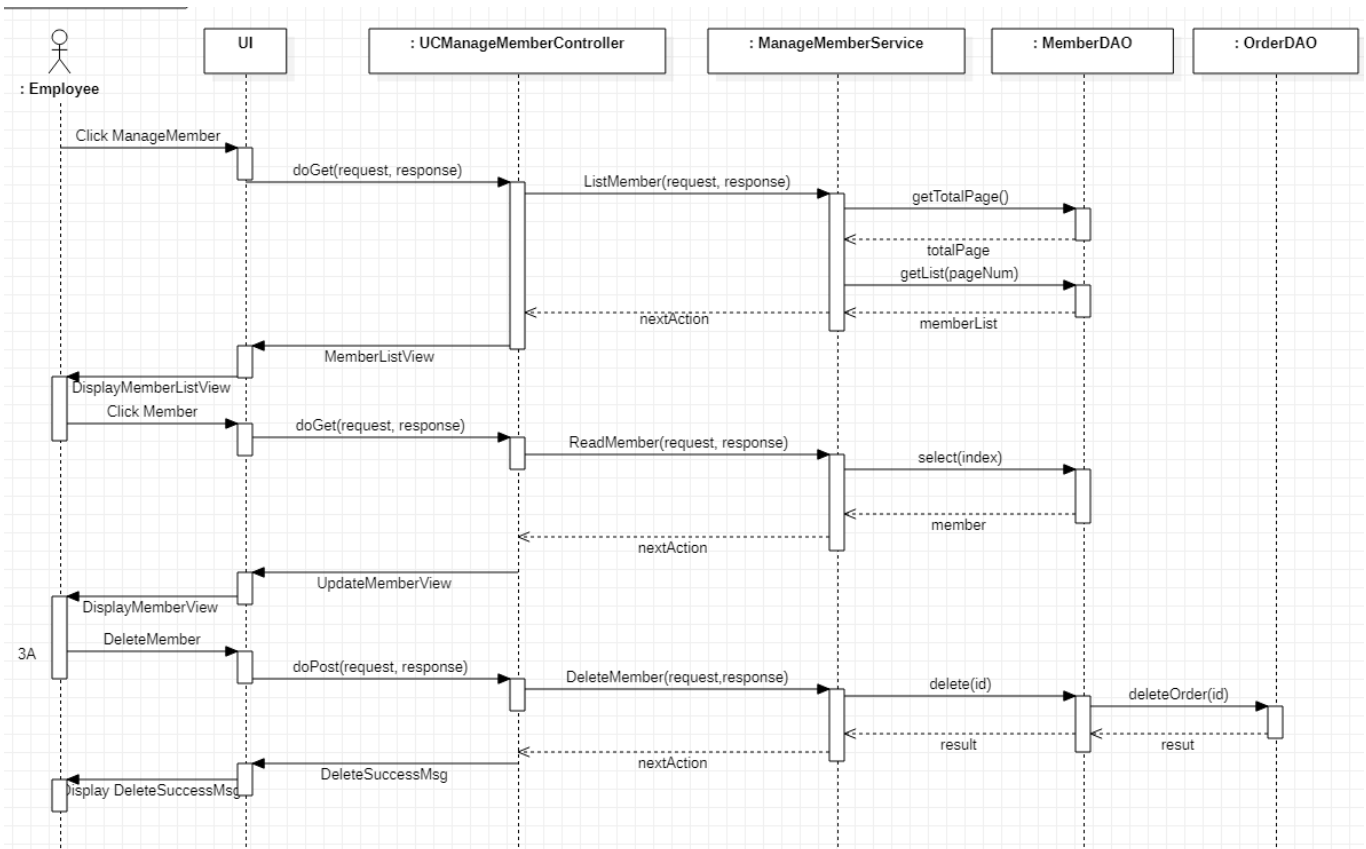
를 띄운다.



대안 흐름(2A)

2A

1. MemberDAO에서 select 함수로 특정 멤버의 정보를 얻어오는데 Error가 발생하면 DetailOrderErrorMsg를 디스플레이 한다.



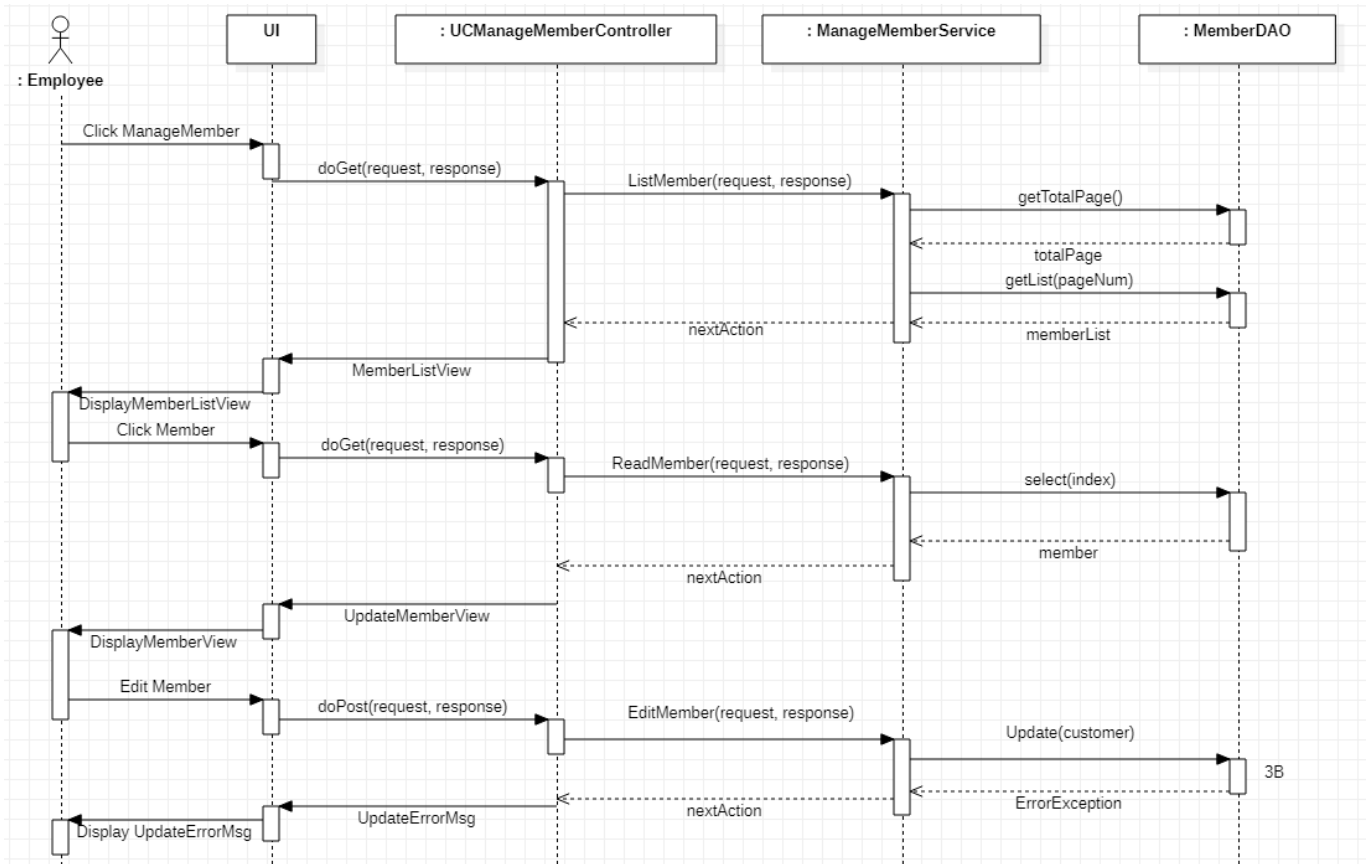
대안 흐름(3A)

3A

1. 해당 멤버 삭제시에 ManageMemberService의 DeleteMember를 실행하여 MemberDAO의 delete 메소

드를 통하여 멤버의 정보를 삭제한다.

- 또한 deleteOrder 메소드를 통하여 해당 멤버의 주문기록을 삭제하고, 성공적으로 완료되어 result에 true가 return되면 DeleteSuccess메시지를 디스플레이 한다.

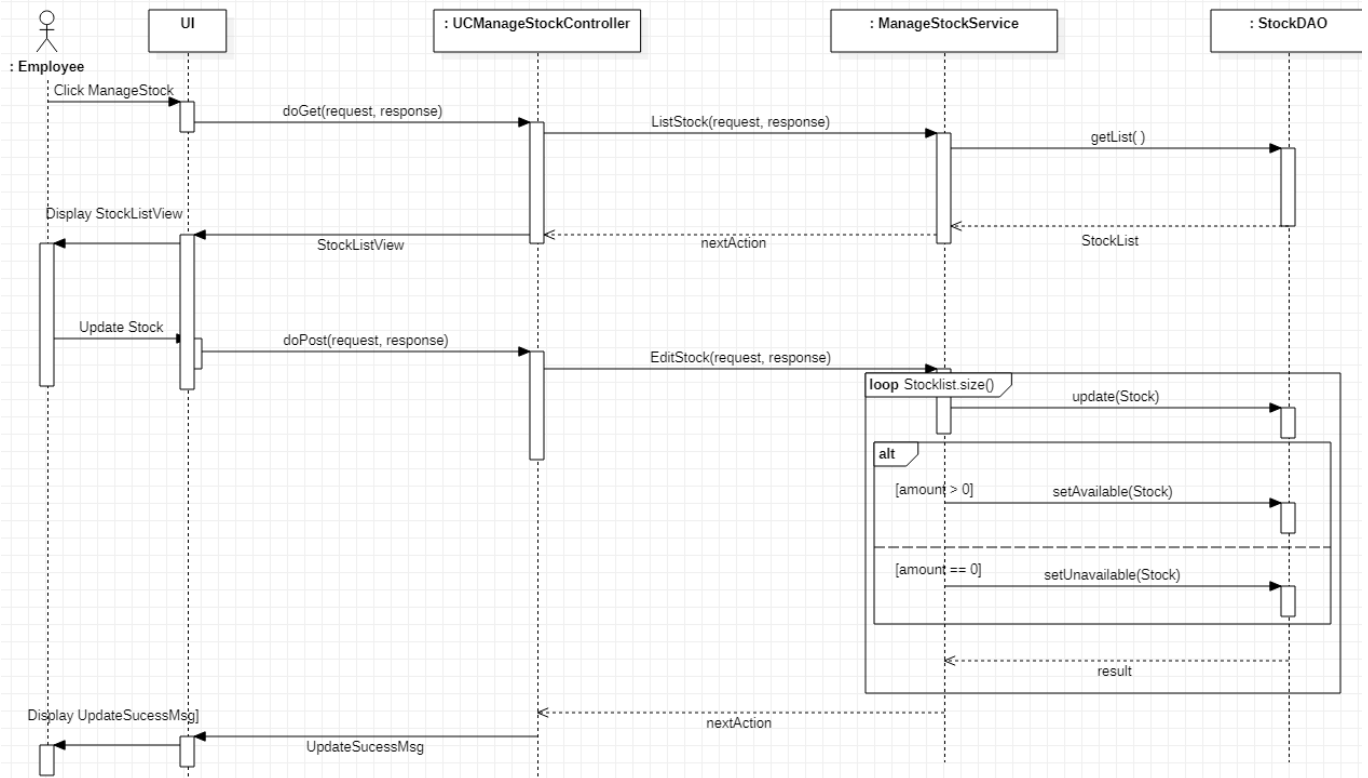


대안 흐름(3B)

3B

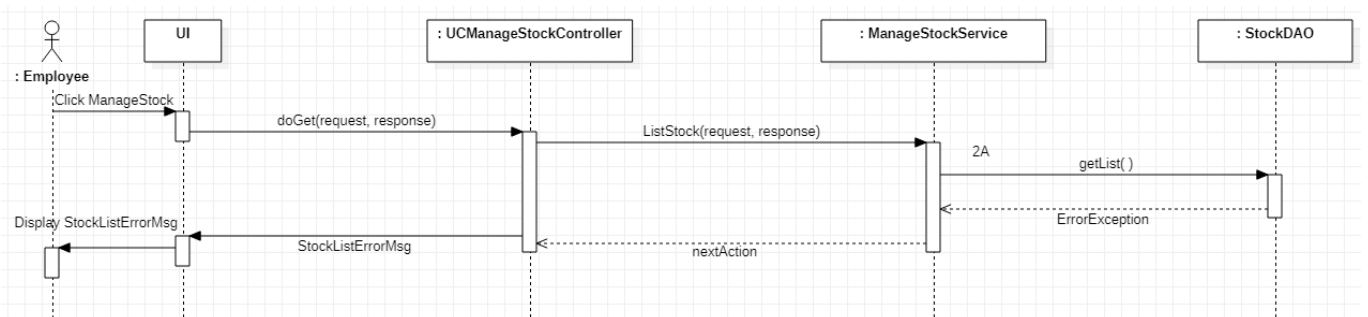
- 멤버정보를 MemberDAO에서 update메소드로 업데이트하는데 Error가 발생한 경우 UpdateError메시지를 디스플레이 한다.

5.8 교류도(usecase: Manage Stock)



기본 흐름

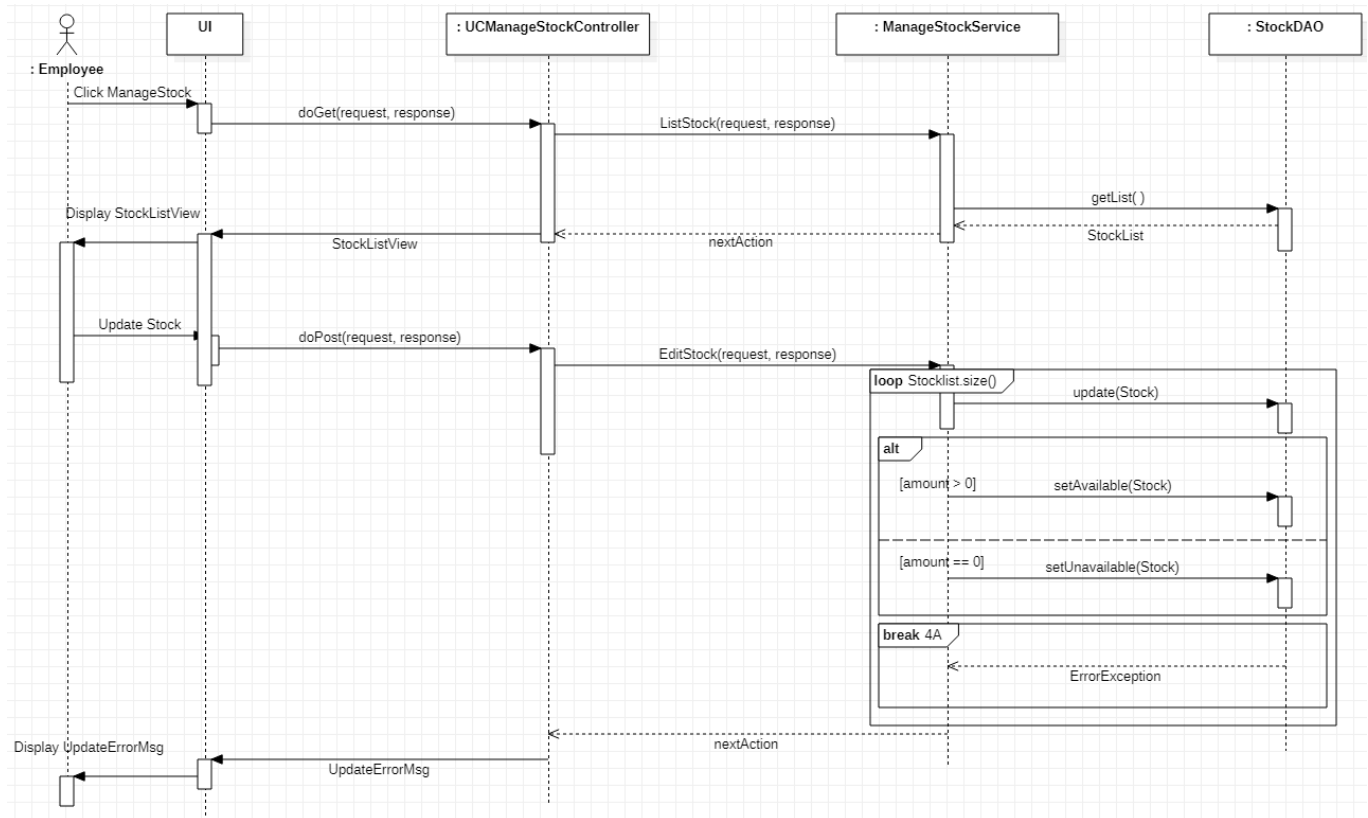
1. 재고관리를 클릭하면 ManageStockService객체의 ListStock을 실행하여 StockDAO객체의 getList()함수를 호출한다.
2. StockList를 받아와 UManageStockController를 통해 StockListView를 띄운다.
3. 재고목록의 수정사항을 저장하면 ManageStockService의 EditStock이 실행되고 StockDAO의 update(stock)이 실행되어 재고가 저장된다.
4. ManageStockService에서 Stock 개수만큼 loop를 돌며 개수가0 인 재고는 StockDAO의 setUnAvailable 함수를 실행한다.
5. StockDAO객체에서 수정이 완료되면 UpdateSuccessMsg를 출력한다.



대안 흐름(2A)

2A.

1. GetList 함수의 호출 과정 중 Error가 발생하면 UManageStockController를 통해 StockListErrorMsg를 띄운다.

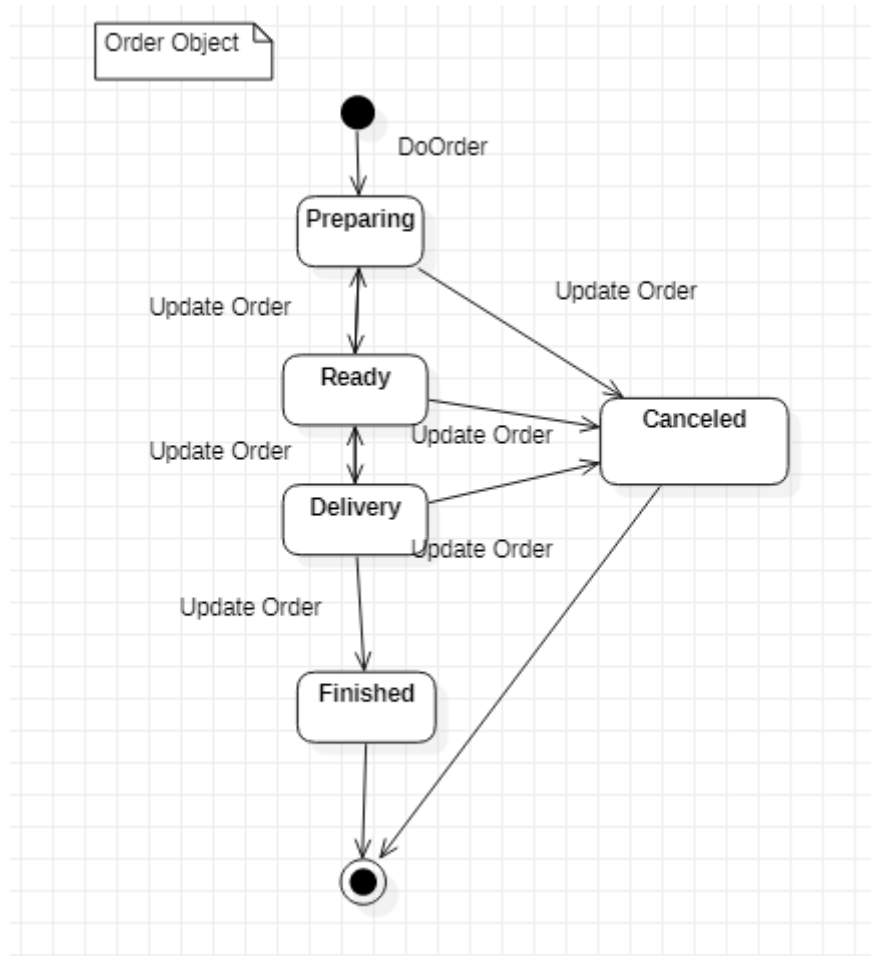


대안 흐름(4A)

- 4A.
1. loop를 도는 동안 StockDAO에서 Error가 발생할 경우 UpdateErrorException을 ManageStockService에 호출하고 Controller가 UpdateErrorMsg를 보내 UI에 띄운다.

6. Order 객체 상태도

Order 객체는 주문정보와 상태를 나타내는 클래스로, Order는 Preparing, Ready, Delivery, Finished, Canceled 5개의 상태 중 하나를 가진다.



1. Preparing

- ✓ 일반적으로 Order에서 처음 저장되는 상태는 Preparing이다.
- ✓ 주문하기 유스케이스에서 DoOrder에 의해 생성되는 Order 객체 상태이다.

2. Ready, Delivery

- ✓ 이후 직원이 준비 완료, 배달중에 있을 때 상태이다.
- ✓ Preparing 단계와 자유롭게 상태를 변경할 수 있다.
- ✓ 직원이 주문 확인 유스케이스에서 Update Order 메소드를 통해 변경할 수 있다.

3. Finished

- ✓ 직원이 배달을 완료하면 Completed 상태로 변경할 수 있다.
- ✓ 직원이 주문 확인 유스케이스에서 Update Order 메소드를 통해 변경할 수 있다.
- ✓ Delivery 상태에서 변경될 수 있으며 더 이상 상태가 변경되지 않고 종료된다.

4. Canceled

- ✓ 고객이 주문을 취소하면 직원에 의해 Canceled 상태로 변경된다.
- ✓ 주문 확인 유스케이스에서 Update Order 메소드를 통해 변경할 수 있다.
- ✓ Preparing, Ready, Delivery 상태에서 변경될 수 있으며 더 이상 상태가 변경되지 않고 종료된다.