

2. SDS (Software Design Specification)

Dressing



학번	이름	이메일
22112092	조하은	cocomind2015@gmail.com
22110611	문성윤	castle9612@gmail.com
22112072	박예원	keroro010112@gmail.com
22112132	양유진	yujin1875@gmail.com
22110612	이효주	nicehyoju@gmail.com

[Revision history]

Revision date	Version #	Description	Author
10/27/2023	0.1	전체적인 보고서 틀 작성	조하은, 문성윤, 박예원, 양유진, 이효주
10/28/2023	0.2	class diagram 작성	조하은, 문성윤, 박예원, 양유진, 이효주
10/30/2023	0.3	sequence diagram 작성	조하은, 문성윤, 박예원, 양유진, 이효주
11/28/2023	0.4	Use case diagram, Class diagram, User interface prototype 수정	조하은, 문성윤, 박예원, 양유진, 이효주
11/29/2023	0.5	sequence machine diagram, state 수정	조하은, 문성윤, 박예원, 양유진, 이효주
12/4/2023	0.6	User interface prototype 수정	조하은, 문성윤, 박예원, 양유진, 이효주

= Contents =

1. Introduction	5
2. Use case analysis	6
3. Class diagram	39
4. Sequence diagram	82
5. State machine diagram	103
6. User interface prototype	105
7. Implementation requirements	117
8. Glossary	119
9. References	120

= Authors for each section =

Introduction - 문성윤

Use case analysis - 조하은, 문성윤, 박예원, 양유진, 이효주

Class diagram - 조하은, 문성윤, 박예원, 양유진, 이효주

Sequence diagram - 조하은, 문성윤, 박예원, 양유진, 이효주

State machine diagram - 조하은, 문성윤, 박예원, 양유진, 이효주

User interface prototype - 조하은, 문성윤, 박예원, 양유진, 이효주

Implementation requirements - 조하은, 문성윤, 박예원, 양유진, 이효주

Glossary - 조하은, 문성윤, 박예원, 양유진, 이효주

References - 조하은, 문성윤, 박예원, 양유진, 이효주

1. Introduction

본 문서는 우리 조가 개발하고자 하는 시스템인 Dressing의 design specification(SDS)이다. 문서에서는 Dressing을 위해 기존에 식별한 기능적 요구사항을 구현하기 위해 시스템을 여러 가지 관점으로 바라보고 설계한다. Use case analysis는 사용자 관점에서 소프트웨어가 제공하는 기능을 서술했고 class diagram은 시스템의 구조적 관점, sequence diagram과 state machine diagram은 시스템의 동적 관점을 묘사했다. User interface는 사용자 인터페이스의 관점에서 시스템을 설계한다.

본 문서를 작성할 때 가장 중요하게 생각한 점은 diagram 간의 일관성이다. 즉 다른 diagram 간에 모순이 없는지 가장 주의를 들여 작성했다. 예를 들어, 특정 기능의 sequence diagram에서 쓰인 object, message는 모두 같은 기능의 class diagram에 있는 association, attribute, operation과 일치한다. 또한 sequence diagram의 흐름은 use case description의 main success scenario와 일치한다. 이와 같은 일관성은 장마다 다시 설명한다. 또한 같은 종류의 diagram 내에서는 형식의 통일을 추구했다.

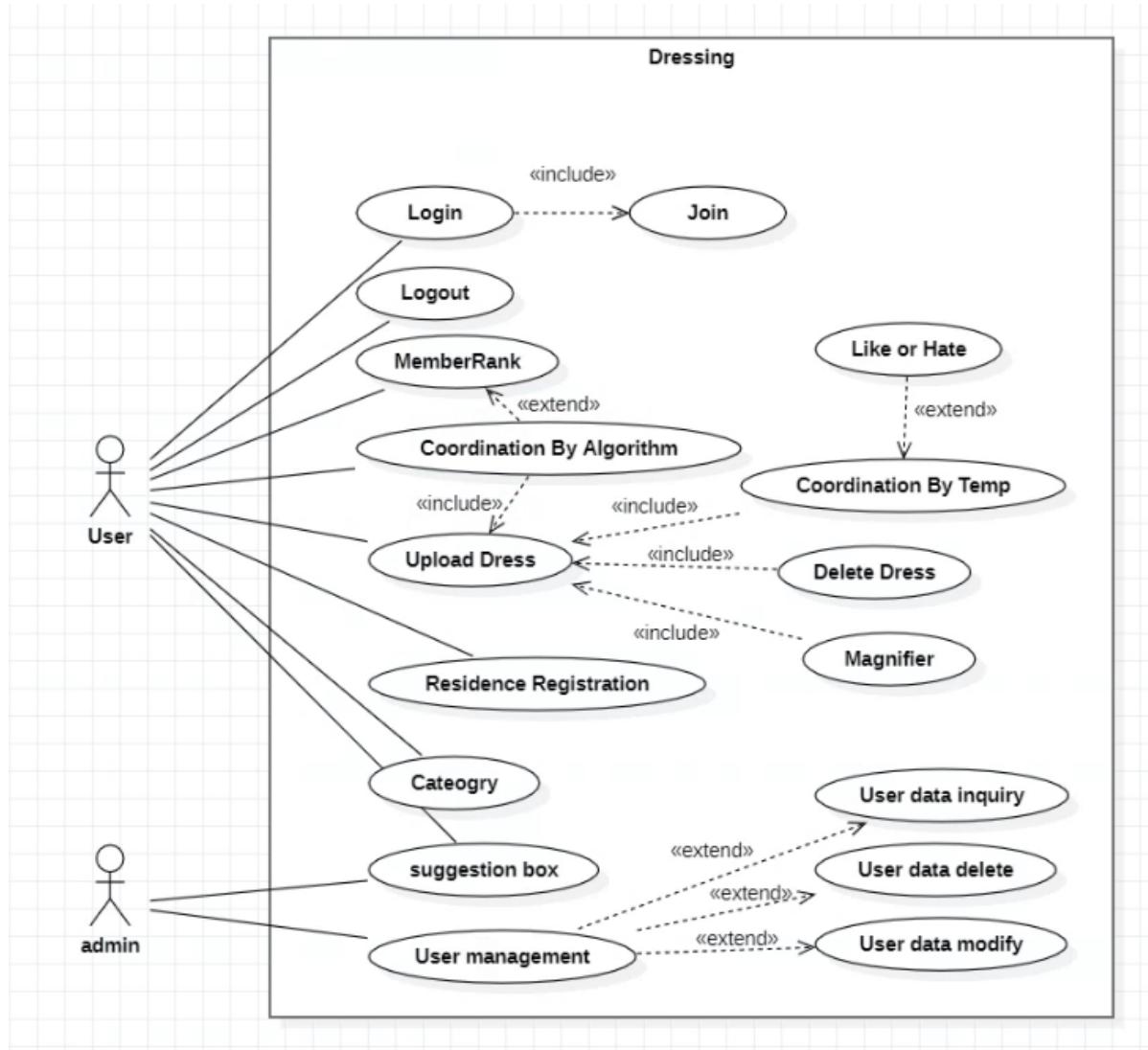
Dressing은 서버와 웹으로 이뤄져 있다. 웹의 개발 환경은 백엔드는 spring boot와 mySQL, 프론트 엔드는 react, AI는 pytorch를 채택했다. 본 문서는 웹 설계를 중점적으로 다루며 서버에 대한 설계는 DB의 관점에서 보는 class diagram과 server state machine diagram에 간략히 설명되어 있다. 이 문서의 가장 중요한 부분은 use case description과 기능 class diagram이다. use case description은 후에 그려지는 거의 모든 UML diagram에 영향을 미칠 뿐 아니라 요구사항이 잘 반영되어 있는지 알 수 있으며 기능 class diagram은 구현 시에 가장 직접적인 영향을 미칠 뿐만 아니라 시스템 구조를 파악할 수 있게 도와준다.

2. Use case analysis

이번 장은 use case diagram과 use case description을 제공한다. Diagram에 관한 고려 사항을 다음과 같다.

- Use case 다이어그램은 주로 부가적인 정보를 제공하기 위한 도구로 활용되며, 본 문서의 독자는 Use case 다이어그램보다는 Use case 설명서에서 충분한 정보를 얻길 바란다.
- Use case diagram에 나타난 use case는 모두 user-goal level use cases이다.
- User는 로그인한 사용자이다. Dressing은 기본적으로 로그인을 하지 않으면 기능을 이용할 수 없다. (회원가입 기능 제외)

[그림 2-1]은 Dressing의 use case diagram이다.



[그림 2 - 1] Use case diagram

이번 장은 use case description에 할당한다. Use case description 을 읽기 전 고려해야할 사항은 아래와 같다.

- 각 use case의 level은 summary, user-goal, subfunction로 구분한다.
 - Summary는 전체 시스템의 문맥을 나타낸다.
 - 시나리오의 각 단계는 Dressing의 핵심 user-goal use case이다.
 - extension scenario에서 오류(실패)와 관련된 부분은 다른 모든 use case에서도 동일하게 일어날 수 있다.

- User-goal은 사용자가 지금 가장 원하는 기능이다.
 - 다른 level에 비해 가장 높은 중요도를 가진다.
 - description에서 user level로 표시한다.
 - 모든 user-goal use cases는 summary use case의 각 단계를 unfolding한 것이거나 각 단계에서 추가적으로 제공하는 기능이다.
 - Use case #1, Use case #4~9, Use case #11~13
- Subfunction은 user-goal use case의 시나리오 한 부분을 떼어내 만든 것이다.
 - 문서에서 모든 subfunction use cases를 고려하지 않지만 Use case #2, Use case #3, Use case #10은 중요도와 필요성을 고려해 따로 use case로 만들어 관리한다.
 - Use case #2 Login, Use case #3 Logout, Use case #10 Suggestion Box: 시스템 사용을 위해 사용자에게 필수적으로 요구되며 기능에 대해 상대적으로 많은 요구사항(api 사용 등)이 있어 중요도가 높다고 판단함.



Use case #1 : Join

GENERAL CHARACTERISTICS

Summary	웹 사이트를 이용하기 위한 필수 절차이며 모든 사용자는 웹사이트를 사용하기 전 회원가입을 해야한다.
Scope	Dressing
Level	User level
Author	조하은
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	웹사이트 첫화면에서 회원가입 버튼을 눌러야 한다.
Trigger	Dressing에 회원가입을 하려고 할 때
Success Post Condition	중복되지 않은 아이디를 입력하고 비밀번호와 전화번호를 형식에 맞춰 작성하였을 때 회원가입에 성공한다.
Failed Post Condition	중복된 아이디를 입력하거나 비밀번호와 전화번호를 형식에 맞지 않게 작성하였을 때 회원가입에 실패한다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 Dressing에 회원가입을 한다.
1	이 Use case 는 사용자가 웹사이트 첫화면에서 회원가입 버튼을 눌렀을 때 시작된다.
2	사용자는 회원가입 창에서 회원정보(이름, id, password, 전화번호)를 입력하여 회원가입을 한다.
3	중복된 아이디인지 체크한다.
4	비밀번호 입력 형식에 적합한지 체크한다.
5	전화번호가 입력 형식에 적합한지 체크한다.
6	이 Use case 는 회원가입에 성공하면 끝난다 .

EXTENSION SCENARIOS

Step	Branching Action
3	3a. 아이디가 중복되어 회원가입에 실패한다. …3a1. 이미 사용 중인 아이디라는 오류 메시지를 보여준다. …3a2. 아이디를 다시 입력받도록 한다.
4	4a. 비밀번호 확인이 맞지 않으면 회원가입에 실패한다. …4a1. 비밀번호 불일치는 오류 메시지를 보여준다. …4a2. 비밀번호를 다시 입력받도록 한다.
5	5a. 전화번호 규칙에 어긋나 회원가입에 실패한다. …5a1. ‘010-xxxx-xxxx’ 형식에 맞게 입력하라는 오류 메시지를 보여준다. …5a2. 전화번호를 다시 입력받도록 한다.

RELATED INFORMATION

Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #2 : Login

GENERAL CHARACTERISTICS

Summary	웹 사이트를 이용하고자 하는 사용자가 회원가입한 정보로 로그인을 한다.
Scope	Dressing
Level	User level
Author	박예원
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	사용자가 Dressing 에 회원가입한 상태여야 한다 .
Trigger	로그인창에서 아이디와 비밀번호를 입력한 후 회원 인증을 받으려고 할 때
Success Post Condition	사용자는 로그인에 성공하여 Dressing의 모든 기능을 사용할 수 있다.
Failed Post Condition	사용자는 로그인에 실패하여 시스템의 회원가입을 제외한 모든 기능을 사용할 수 없다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 Dressing 에 로그인 한다
1	이 Use case 는 사용자가 Dressing 에 로그인할 때 시작된다 .
2	사용자는 로그인창에서 ID 와 Password 를 입력하고 로그인 버튼을 클릭한다 .
3	Dressing DB에서 등록된 회원인지 체크하고 등록된 회원이라면 로그인에 성공한다 .
4	이 Use case 는 로그인이 성공하면 끝난다 .

EXTENSION SCENARIOS	
Step	Branching Action
3	<p>3a. 아이디와 비밀번호를 입력을 실패하여 로그인을 실패한다.</p> <ul style="list-style-type: none"> …3a1. 아이디나 비밀번호가 잘못되었다(failed login!)는 메시지를 보여준다. …3a2. 아이디와 비밀번호를 입력하는 단계로 돌아간다. <p>3b. DB 연결 실패로 인해 로그인을 실패한다.</p> <ul style="list-style-type: none"> …3b1. DB와 접속이 원활하지 않다면 로그인에 실패한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #3 : Logout

GENERAL CHARACTERISTICS

Summary	로그인 상태에서 더이상 웹 사이트를 이용하고 싶지 않을 때 사용자는 로그아웃을 한다.
Scope	Dressing
Level	User level
Author	양유진
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	관리자, 회원
Preconditions	로그아웃을 하기 위해서는 사이트를 관리하는 관리자와 사이트를 사용하는 회원은 Dressing 사이트에 로그인된 상태여야 한다.
Trigger	관리자와 회원이 Dressing 사이트에 로그인된 상태에서 로그아웃 버튼을 눌렀을 때 시작된다.
Success Post Condition	관리자와 회원은 로그아웃에 성공한다.
Failed Post Condition	관리자와 회원은 로그아웃에 실패하고 로그인 상태가 유지된다.

MAIN SUCCESS SCENARIO

Step	Action
S	관리자, 회원이 Dressing 에 로그아웃한다.
1	이 Use case 는 관리자, 회원이 Dressing에 로그인된 상태에서 로그아웃을 눌렀을 때 시작된다.
2	관리자, 회원은 Dressing 메인 사이트에서 로그아웃 버튼을 누른다.
3	시스템은 로그아웃이 성공했는지 판단한다.
4	이 Use case 는 로그아웃이 성공하면 끝난다 .

EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 통신에 문제가 있으면 로그아웃에 실패한다.. …3a1. 로그아웃 실패 메시지를 띠운다. …3a2. 로그인 상태인 메인 페이지로 돌아간다. (Use case #3-2)

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	관리자, 회원당 로그인당 1 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case 4 : MemberRank

GENERAL CHARACTERISTICS

Summary	사용자가 회원가입을 한 날짜를 기준으로 회원가입을 한지 오래되었을수록 더 좋은 기능을 사용할 수 있다.
Scope	Dressing
Level	User level
Author	이효주
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	회원등급을 이용하기 위해서는 Dressing 사이트에 회원가입이 되어있어야 한다.
Trigger	사용자가 Dressing 사이트에 로그인을 한다.
Success Post Condition	사용자가 회원가입을 한 날짜 기준으로 회원등급이 표시되고 등급에 따라 더 좋은 기능 혹은 기능제한이 해제된다.
Failed Post Condition	사용자의 회원등급이 표시되지 않고 회원의 등급에 맞는 기능을 사용할 수 없다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 자신의 회원등급을 확인하고 그에 맞는 기능을 사용하고자한다.
1	사용자가 로그인을 성공한다 .
2	시스템은 사용자가 회원가입을 한 날짜를 확인하고 회원가입을 한 날짜를 기준으로 회원등급이 표시된다.
3	시스템은 사용자에게 사용자의 회원등급에 맞는 기능을 제공한다.
4	이 Use case 는 로그아웃이 성공하면 끝난다 .

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a. 회원가입을 한 날짜가 확인되지 않아 회원등급을 표시하지 못한다.</p> <ul style="list-style-type: none"> …2a1. 회원등급을 확인할 수 없다는 메시지를 보여준다 . …2a2. 회원등급이 표시되지 않는다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #5 : UploadDress	
GENERAL CHARACTERISTICS	
Summary	웹 사이트에 사용자가 옷 사진을 업로드함으로써 사용자가 소유한 옷들을 확인할 수 있다.
Scope	Dressing
Level	User level
Author	양유진
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	회원
Preconditions	옷 사진을 업로드하려는 회원은 Dressing 사이트에 정상적으로 로그인한 상태여야 한다.
Trigger	Dressing 사이트 메인화면에서 옷 사진을 추가하는 버튼을 클릭하면 옷 사진을 추가할 수 있다.
Success Post Condition	사용자는 옷 사진을 업로드한다.
Failed Post Condition	사용자는 옷 사진 업로드에 실패한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	회원이 옷 사진을 업로드하여 회원이 소유한 옷을 확인하고자 한다.
1	이 Use case 는 회원이 옷 사진을 추가하는 버튼을 클릭한다.
2	첨부파일에 사진 파일을 선택하고 업로드 버튼을 클릭한다.
3	AI를 통해 옷인지 판별 및 종류별로 분류후 추가한다.
4	이 Use case는 옷이 웹사이트에 추가되면 끝난다.

EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 올바른 파일 형식이 아니면 옷추가에 실패한다. …3a1. 올바른 파일 형식이 아니라는 메시지를 보여준다. …3a2. 파일을 선택하는 단계로 돌아간다.(Use case #5-2)

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #6 : DeleteDress

GENERAL CHARACTERISTICS

Summary	웹사이트에 사용자가 업로드한 옷을 삭제할 수 있다.
Scope	Dressing
Level	User level
Author	문성윤
Last Update	2023. 10. 27
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	사용자는 Dressing에 로그인된 상태여야 한다.
Trigger	자신이 찍은 옷을 웹사이트에서 삭제하기 위해 옷사진을 삭제 할때
Success Post Condition	Dressing 웹사이트에 자신의 옷장에 찍은 옷이 삭제된다.
Failed Post Condition	오류로 인하여 옷 삭제에 실패한다.

MAIN SUCCESS SCENARIO

Step	Action
S	유저가 옷사진을 웹사이트에 삭제를 성공한다.
1	이 Use case 는 유저가 옷장에 옷을 삭제버튼을 누를때 시작된다.
2	유저는 x버튼을 눌러 자신이 가지고 있는 옷사진을 삭제한다.
3	DB에 저장된 옷정보를 지우고 웹사이트에서 지워진다.
4	이 Use case는 옷이 웹사이트에 삭제되면 끝난다.

EXTENSION SCENARIOS

Step	Branching Action

EXTENSION SCENARIOS

3	3a. 이미지 파일이 아니어 옷삭제에 실패한다. …3a1. 오류가 발생하여 삭제에 실패했다는 메시지를 보여준다. …3a2. 옷삭제를 선택하는 부분으로 돌아간다..(Use case #6-2)
---	---

RELATED INFORMATION

Performance	$\leq 1 \text{ sec.}$
Frequency	회원당 하루에 평균 5 번
Concurrency	제한 없음
Due Date	2023.12.21

Use case #7 : Coordination By Temp

GENERAL CHARACTERISTICS

Summary	사용자의 점수를 기반으로 사용자의 취향에 맞는 옷을 추천해준다.
Scope	Dressing
Level	User level
Author	문성윤
Last Update	2023. 10. 27
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	사용자는 Dressing에 로그인된 상태여야 하며, 옷 추천을 받으려는 사용자가 옷을 추가한 상태이여야 한다.
Trigger	옷 추천을 받기 위해 기온별 옷 추천 버튼을 눌렀을 때
Success Post Condition	기온에 따른 옷을 추천받으며 좋아요와 싫어요 버튼을 띄운다.
Failed Post Condition	기온에 따른 옷을 추천받지 못한다.

MAIN SUCCESS SCENARIO

Step	Action
S	기온에 따른 옷 추천 버튼을 누른다.
1	이 Use case 는 회원이 기온에 따른 옷 추천버튼을 누를 때 시작된다.
2	유저는 기온에 따른 옷 추천버튼을 클릭한다.
3	옷이 추가되어 있는 상태인 사용자이면 기온에 따른 옷추천에 성공해야 한다.
4	이 Use case 는 옷이 추천되면 끝난다.

EXTENSION SCENARIOS	
Step	Branching Action
3	<p>3a. 사용자가 옷을 추가 되어있지 않아 옷추천에 실패한다.</p> <p>…3a1. 옷이 추가되지 않은 상태라는 메시지를 보여준다 .</p> <p>…3a2. main page로 돌아간다.</p>

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원의 랭크 별 지정 횟수만큼
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #8 : Like or Hate
GENERAL CHARACTERISTICS

Summary	추천받은 옷을 보고 사용자가 좋은지, 싫은지 판단한다. 이는 사용자의 취향에 맞는 옷을 추천하는 데 기반이 된다.
Scope	Dressing
Level	User level
Author	조하은
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	사용자는 Dressing에 로그인된 상태여야 하며, 사용자의 옷 사진이 1개 이상 등록된 상태여야 한다.
Trigger	사용자가 옷 코디를 추천받고자 할 때
Success Post Condition	사용자가 최소 등록 옷의 수를 만족하고, 코디 추천 횟수가 남아있을 때 성공한다.
Failed Post Condition	사용자가 등록한 옷이 없을 때 조합할 수 있는 코디가 없으므로 실패한다.

MAIN SUCCESS SCENARIO

Step	Action
S	회원이 옷 코디를 추천받아 취향에 맞게 평가한다.
1	이 Use case 는 회원이 옷추천 버튼을 눌렀을 때 시작된다.
2	카테고리에 코디를 할 수 있는 옷이 있는지 확인한다.
3	추천해주는 코디를 보고 취향에 맞게 좋아요나 싫어요 버튼을 누른다.
4	사용자의 랭크에 맞는 추천 횟수가 끝나거나 더이상 코디할 수 없을 때 이 Use case는 끝난다.

EXTENSION SCENARIOS	
Step	Branching Action
2	2a. 카테고리에 옷이 없을 때 코디 추천을 받을 수 없다. …2a1. 등록된 옷이 없으니 옷을 추가하라는 메시지를 보여준다. …2a2. 메인 페이지로 돌아간다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원의 랭크 별 지정된 횟수만큼
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #9 : Coordination By algorithm	
GENERAL CHARACTERISTICS	
Summary	알고리즘을 토대로 기온별로 코디를 추천하게 된다.
Scope	Dressing
Level	User level
Author	문성윤
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	사용자는 Dressing에 로그인된 상태여야 하며, 알고리즘 기반으로 옷 추천을 받으려는 사용자는 랭크가 “diamond”이여야 하며, 옷이 추가된 상태이여야 한다.
Trigger	알고리즘을 기반으로 한 코디를 추천받으려고 할때
Success Post Condition	사용자는 알고리즘에 따라 추천한 코디를 확인한다.
Failed Post Condition	등급에 맞지 않거나 오류가 있어 사용자가 AI가 추천한 코디를 받지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	회원이 코디추천을 받는다.
1	이 Use case 는 회원이 algorithm기반 추천을 누를때 시작한다.
2	회원은 AI기반 추천 버튼을 클릭한다.
3	등급을 확인하고 등급이 “diamond”이면 algorithm기반 옷 추천을 받는다.
4	이 Use case 는 회원이 algorithm기반 옷 추천을 받으면 끝난다 .

EXTENSION SCENARIOS	
Step	Branching Action
3	<p>3a. 등급이 Diamond가 아니라서 실패한다 .</p> <p>…3a1. 등급이 총족되지 않았다는 메시지를 보여준다 .</p> <p>…3a2. main page로 돌아간다.</p> <p>3b. 저장된 옷이 없어 실패한다 .</p> <p>…3b1. 저장된 옷이 없다는 메시지를 보여준다 .</p> <p>…3b2. main page로 돌아간다.</p> <p>3c. 오류가 발생하여 실패한다 .</p> <p>…3c1. 오류가 발생했다는 메시지를 보여준다 .</p> <p>…3c2. main page로 돌아간다.</p>

RELATED INFORMATION	
Performance	≤ 30 seconds
Frequency	회원당 하루에 평균 5 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #10 : Suggestion Box

GENERAL CHARACTERISTICS

Summary	사용자가 Dressing을 사용하면서 가진 건의를 작성하는 건의함이다.
Scope	Dressing
Level	Subfunction level
Author	박예원
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	사용자는 Dressing에 로그인 한 상태여야 한다.
Trigger	사용자가 건의함 화면에서 건의하기 버튼을 누를 때
Success Post Condition	사용자는 본인이 원하는 건의글을 작성한다.
Failed Post Condition	사용자가 건의글을 작성하지 못한다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 건의함에 글을 작성한다.
1	건의함 화면에서 건의하기 버튼을 클릭하면 use case가 실행된다.
2	사용자가 제목과 내용을 입력하고 등록하기 버튼을 누르면 건의글이 등록된다.

EXTENSION SCENARIOS

Step	Branching Action

EXTENSION SCENARIOS

S.2	2a. DB와의 연동이 실패하여 건의글이 제대로 등록되지 않는다. …3a1. 작성한 건의글이 DB에 등록되지 않는다
-----	---

RELATED INFORMATION

Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #11 : Magnifier
GENERAL CHARACTERISTICS

Summary	회원이 옷장에서 자신이 업로드한 옷 사진을 클릭하여 확대하여 화면에 표시하고 닫는 둘보기 기능이다.
Scope	Dressing
Level	User level
Author	양유진
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	회원
Preconditions	회원이 Dressing 사이트에 로그인되어 있으면서 옷장에 업로드한 옷이 존재해야 한다.
Trigger	회원이 옷장에서 자신이 업로드한 옷 사진에서 확대 버튼을 클릭하고 확대 후 X 버튼을 누를 때
Success Post Condition	회원이 클릭한 옷 사진이 확대되어 화면에 나타난다. 회원이 X 표시를 누르면 확대한 창이 닫힌다.
Failed Post Condition	회원이 클릭한 옷 사진이 확대되어 화면에 나타나지 않는다. 회원이 X 표시를 누르면 확대한 창이 닫히지 않는다.

MAIN SUCCESS SCENARIO

Step	Action
S	회원은 옷 사진을 확대한 창을 볼 수 있다.
1	회원이 옷장에서 옷을 확대하는 버튼을 누를 때 시작한다.
2	회원은 옷장에서 옷을 확대하는 버튼을 클릭한다.
3	시스템은 선택한 옷을 확대하여 나타내는 창을 띄우면 이 기능은 종료된다.
M	회원은 확대한 창을 닫을 수 있다.
1	회원이 확대한 창을 닫는 X 버튼을 누를 때 시작한다.

MAIN SUCCESS SCENARIO	
2	확대한 창을 닫는 X 버튼을 클릭한다.
3	화면에 나타났던 창이 사라지면 이 기능은 종료된다.

EXTENSION SCENARIOS	
Step	Branching Action
S.3	3a. 해당 옷 사진을 확대한 이미지를 찾을 수 없다. …3a.1 확대 창을 띄울 수 없다는 메시지를 나타낸다.
M.2	3a. 해당 창을 닫을 수 없다. …3a.1 해당 창을 닫을 수 없다는 메시지를 나타낸다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 5 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #12 : Category	
GENERAL CHARACTERISTICS	
Summary	회원가입을 한 모든 사용자들이 사용할 수 있으며 사용자의 옷들과 이전까지 사용자가 점수를 매긴 코디들을 볼수있다.
Scope	Dressing
Level	User level
Author	이효주
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	사용자
Preconditions	사용자가 로그인에 성공한 상태여야 한다.
Trigger	드레싱에 로그인하기 위해 아이디와 비밀번호를 입력한 후 회원 인증을 받으려고 할 때
Success Post Condition	사용자가 누른 버튼에 해당되는 사용자의 옷들 또는 이전까지 사용자가 점수를 매긴 코디들을 확인할 수 있다.
Failed Post Condition	사용자가 누른 버튼에 해당되는 사용자의 옷들과 이전까지 사용자가 점수를 매긴 코디들을 확인하지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 확인하고 싶어하는 옷 카테고리의 버튼을 누른다.
1	사용자는 사용자가 확인하고 싶어하는 옷 카테고리의 버튼을 누른다 .
2	시스템은 사용자가 누른 버튼에 해당되는 옷들의 정보를 가져온다.
3	사용자는 사용자가 누른 버튼에 해당되는 옷들의 정보를 확인한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a. 저장된 옷들의 정보를 확인하지 못해 사용자가 옷들의 정보를 확인하지 못한다.</p> <ul style="list-style-type: none"> …2a1. 저장된 옷들의 정보를 확인할 수 없다는 메시지를 보여준다 . …2a2. 사용자가 누른 버튼에 해당되는 사용자의 옷들 또는 사용자가 점수를 매긴 코디들을 확인할 수 없다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #13 : User Data Inquiry

GENERAL CHARACTERISTICS

Summary	사용자들이 작성한 건의함을 조회할 수 있다.
Scope	Dressing
Level	User level
Author	양유진
Last Update	2023. 11. 28.
Status	Analysis (Finalize)
Primary Actor	관리자
Preconditions	관리자가 로그인에 성공한 상태여야 한다.
Trigger	관리자 페이지에서 회원 관리 버튼을 누르고 조회 버튼을 누를 때
Success Post Condition	해당 건의함을 작성한 id, 이름, 아이디, 비밀번호, 전화번호, Rank, 지역, 회원가입일을 조회할 수 있다.
Failed Post Condition	해당 건의함을 작성한 id, 이름, 아이디, 비밀번호, 전화번호, Rank, 지역, 회원가입일을 조회할 수 없다.

MAIN SUCCESS SCENARIO

Step	Action
S	관리자는 관리자 페이지에서 회원관리를 할 수 있다.
1	관리자가 확인하고 싶어하는 건의의 조회 버튼을 누른다.
2	시스템은 관리자가 누른 버튼에 해당되는 건의의 정보를 가져온다.
3	관리자는 누른 버튼에 해당되는 건의 정보를 확인한다.

EXTENSION SCENARIOS

Step	Branching Action

EXTENSION SCENARIOS

2	2a. 서버와의 문제가 발생하여 건의함을 조회하지 못한다. …2a1. 에러창이 뜬다.
---	--

RELATED INFORMATION

Performance	\leq 2 seconds
Frequency	관리자당 하루에 평균 5 번
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #14 : User Data Delete
GENERAL CHARACTERISTICS

Summary	회원가입을 한 모든 사용자들의 데이터중 하나를 관리자가 삭제할 수 있다.
Scope	Dressing
Level	Admin level
Author	문성윤
Last Update	2023. 11. 28.
Status	Analysis (Finalize)
Primary Actor	Admin
Preconditions	관리자 로그인에 성공한 상태여야 한다.
Trigger	관리자가 사용자의 데이터를 삭제하려고 할때
Success Post Condition	관리자가 해당되는 사용자의 데이터를 서버에서 삭제한다.
Failed Post Condition	관리자가 해당되는 사용자의 데이터를 서버에서 삭제하지 못한다.

MAIN SUCCESS SCENARIO

Step	Action
S	관리자가 유저의 데이터를 삭제한다.
1	관리자는 유저의 정보의 삭제버튼을 누른다 .
2	시스템은 관리자가 삭제를 원하는 유저의 데이터를 서버에서 삭제한다.
3	시스템은 웹페이지에 삭제된 유저의 데이터에 대한 정보를 지운다.

EXTENSION SCENARIOS

Step	Branching Action

EXTENSION SCENARIOS	
2	2a. 서버에서 유저의 데이터를 삭제하지 못한다. …2a1. 삭제 실패라는 message box를 띄운다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
Concurrency	제한 없음
Due Date	2023.12.21.

Use case #15 : User Data Modify

GENERAL CHARACTERISTICS

Summary	관리자는 유저의 데이터를 수정한다.
Scope	Dressing
Level	Admin level
Author	문성윤
Last Update	2023. 10. 27.
Status	Analysis (Finalize)
Primary Actor	Admin
Preconditions	관리자 계정으로 로그인이 되어있어야 한다.
Trigger	관리자가 회원의 정보를 수정하고자 할때
Success Post Condition	관리자가 회원의 정보를 성공적으로 수정한다.
Failed Post Condition	관리자가 회원의 정보를 수정하지 못한다.

MAIN SUCCESS SCENARIO

Step	Action
S	관리자가 회원정보 수정버튼을 누른다.
1	관리자가 수정하고 하는 정보를 수정한다.
2	시스템은 관리자가 수정하고자 하는 정보를 서버에 전송한다.
3	시스템은 관리자가 수정하고자 하는 정보를 web에 반영한다.

EXTENSION SCENARIOS

Step	Branching Action
2	2a. 회원정보 수정에 실패한다. … 2a1. 예기치 못한 오류로 인하여 수정에 실패했다는 메세지를 띄운다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	
Concurrency	제한 없음
Due Date	2023.12.21.

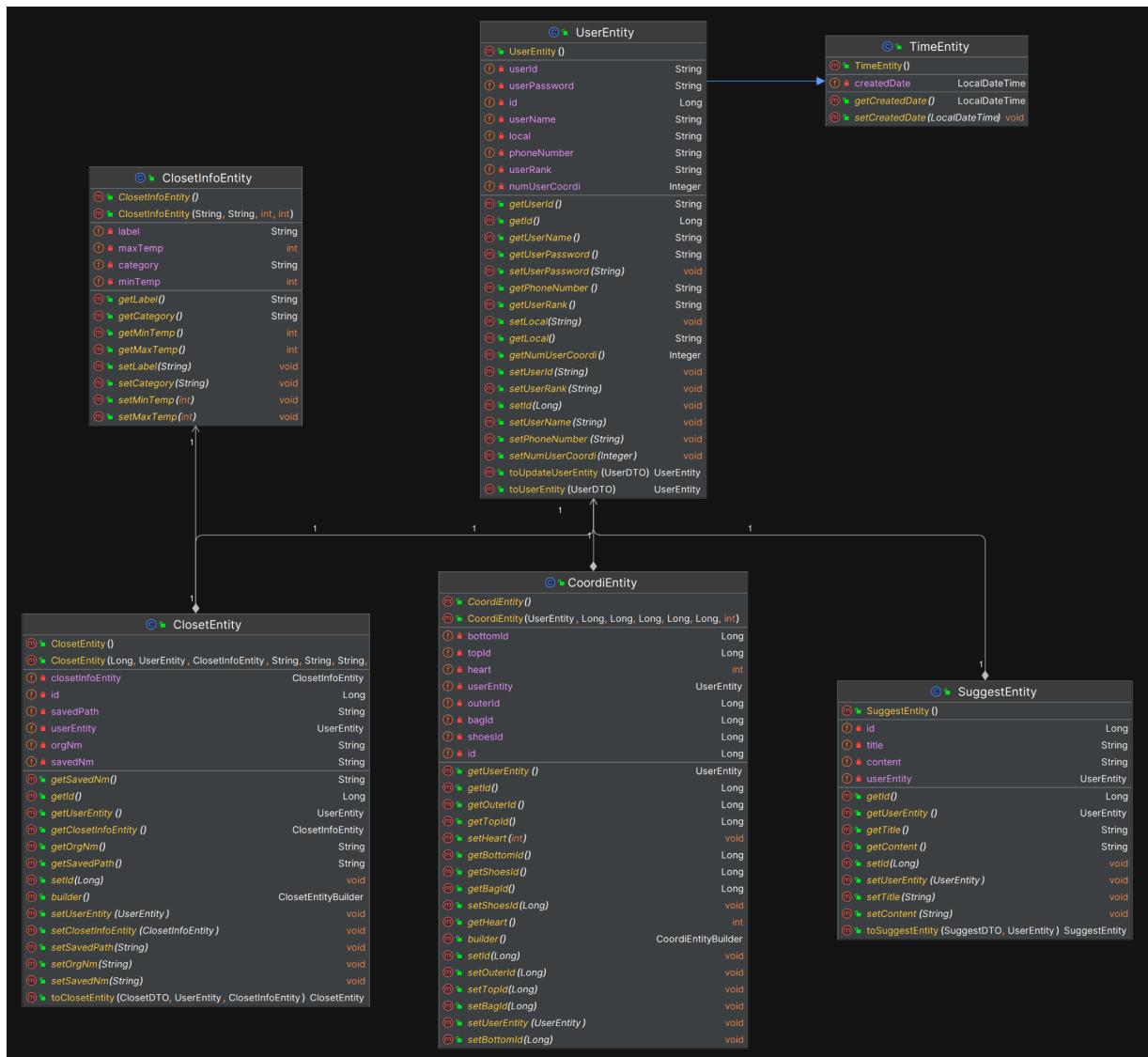
3. Class diagram

이번 장은 다양한 관점에서 바라본 Class diagram과 각각에 대한 설명을 첨부한다. Class diagram를 볼 때 고려해야 할 사항은 다음과 같다.

- 본 시스템의 Class diagram은 웹 전용이다.
- 웹으로 작성한 Class diagram은 기능, 상속, DB 접근으로 나눌 수 있다.
 - 각각의 기능 Class diagram은 하나 혹은 여러 개의 use case와 연결된다.
 - 기능을 제외한 Class diagram은 구현 시 고려해야 할 사항들에 대한 것이며 각각의 description파트에서 역할을 자세히 설명한다.
- 데이터 타입과 Util, static은 Class diagram으로 그리지는 않았지만 추출한 클래스들을 한곳으로 모아 별도의 장으로 설명한다.

3.1 DB Class diagram

- Dressing 프로그램의 데이터베이스 구조를 Class diagram으로 작성했다.
 - ER diagram을 먼저 작성하여 DB 구조 골격을 세운 다음 이를 CD으로 변환하여 얻은 결과이다.
 - DB에서 직접 값을 주고 받는 Entity 클래스들의 구조를 나타낸 것이다.



[그림 3 - 1] DB Class Diagram

ClosetEntity			
Class	데이터베이스의 'closet_table' 테이블과 매핑되는 엔터티이다.		
Description			
구분	Name	Type	Visibility
Description			
Attributes	id	Long	private
	entity의 기본키로 사용되는 필드로, 자동으로 증가하는 값으로 설정되어 있다.		
	userEntity	UserEntity	private
	n:1 관계로 설정된 필드로, 'UserEntity' 엔터티와의 관계를 나타낸다. 'uid' 컬럼이 외래키로 사용된다.		
	closetInfoEntity	ClosetInfoEntity	private
	n:1 관계로 설정된 필드로, 'ClosetInfoEntity' 엔터티와의 관계를 나타낸다. 'label' 컬럼이 외래키로 사용된다.		
	orgNm	String	private
	оригинальный файл имя 나타내는 문자열 필드이다.		
	savedNm	String	private
	저장된 파일 이름을 나타내는 문자열 필드이다.		
	savedPath	String	private

	파일이 저장된 경로를 나타내는 문자열 필드이다.		
구분	Name	Arguments	Returns
	Description		
Operations	ClosetEntity		
	기본생성자와 '@Builder' 어노테이션을 사용한 빌더 패턴을 지원하는 생성자가 정의되어 있다.		
	ClosetEntity	UserDTO, String, HttpServletRespo nse	String
빌더 패턴을 사용하여 객체를 생성하고 반환한다.			

ClosetInfoEntity			
Class	데이터베이스의 'closetinfo_table' 테이블과 매핑되는 엔터티이다. 이 엔터티는 의상 아이템의 정보를 나타내는데 사용되며, 각 의상 아이템은 특정 라벨에 속하고, 해당 라벨의 카테고리와 온도 범위를 가지고 있다.		
구분	Name	Type	Visibility
	Description		
Attributes	label	String	private



	의상 아이템이 속하는 라벨을 나타내는 문자열 필드이다. ‘ClosetEntity’에서 외래키로 사용된다.				
	category	String			
	private				
	의상 아이템의 카테고리를 나타내는 문자열 필드이다.				
	maxTemp	int			
	private				
	의상의 적합한 최대 온도를 나타내는 정수형 필드이다.				
	minTemp	int			
	private				
	의상의 적합한 최소 온도를 나타내는 정수형 필드이다.				
구분	Name	Arguments	Returns		
	Description				
Operations	ClosetInfoEntity	String label, String category, int minTemp, int maxTemp			
	이 클래스의 생성자는 의상에 대한 정보를 초기화하는 데 사용된다. 이 생성자를 통해 라벨, 카테고리, 최소 온도, 최대 온도를 설정할 수 있다.				

CoordiEntity

Class	데이터베이스의 'coordi_table' 테이블과 매핑되는 엔터티이다. 이 엔터티는 사용자의 코디 정보를 나타내며, 코디에 필요한 의상들과 좋아요/싫어요 정보를 저장한다.		
구분	Name	Type	Visibility
Description			
Attributes	id	Long	private
	코디의 고유 식별자로 사용되는 필드로, 자동으로 생성되는 값이다.		
	userEntity	UserEntity	private
	n:1 관계로 설정되는 필드로, 'UserEntity' 엔터티와의 관계를 나타낸다. 'uid' 컬럼이 외래키로 사용된다		
	outerId	Long	private
	코디에 사용된 아우터의 고유 식별자를 나타내는 필드이다		
	topId	Long	private
	코디에 사용된 상의의 고유 식별자를 나타내는 필드이다		
	bottomId	Long	private
	코디에 사용된 하의의 고유 식별자를 나타내는 필드이다		
	shoesId	Long	private
	코디에 사용된 신발의 고유 식별자를 나타내는 필드이다		
	bagId	Long	private

구분	코디에 사용된 가방의 고유 식별자를 나타내는 필드이다		
	heart	int	private
	코디에 대한 좋아요/싫어요 정보를 나타내는 정수형 필드로, 좋아요가 1이고, 싫어요가 -1이다.		
Operations	Name	Arguments	Returns
	Description		
	CoordiEntity	UserEntity userEntity, Long outerId, Long topId, Long bottomId, Long shoesId, Long bagId, int heart	
	사용자의 코디 정보를 데이터베이스에 저장하고 조회하는 데 사용된다. 주로 속성만 정의하고 있고, 실제로 구체적인 동작 메서드는 포함되어 있지 않다.		

SuggestEntity			
Class	데이터베이스의 ‘suggest_table’ 테이블과 매핑되는 엔터티이다. 사용자의 건의사항을 나타내는 엔터티로, 건의사항에 대한 정보를 저장한다.		
Description	Name	Type	Visibility

구분	Description							
Attributes	id	Long	private					
	건의사항의 고유 식별자로 사용되는 필드로, 자동으로 생성되는 값이다.							
	userEntity	UserEntity	private					
	n:1 관계로 설정된 필드로, 'UserEntity' 엔터티와의 관계를 나타낸다. 'uid' 컬럼이 외래키로 사용된다.							
	title	String	private					
	건의사항의 제목을 나타내는 문자열 필드이다.							
Operations	content	String	private					
	건의사항의 내용을 나타내는 문자열 필드이다.							
	<table border="1"> <thead> <tr> <th>Name</th> <th>Arguments</th> <th>Returns</th> </tr> </thead> <tbody> <tr> <td colspan="3">Description</td></tr> </tbody> </table>			Name	Arguments	Returns	Description	
Name	Arguments	Returns						
Description								
Operations	toSuggestEntity	SuggestDTO suggestDTO, UserEntity userEntity	String					
	회원가입 페이지 품 작성 데이터를 받아온다.							

TimeEntity			
Class	여러 엔터티에서 공통적으로 사용되는 시간정보를 가지고 있는 추상 클래스이다. 실제 테이블과 매핑되지 않고, 다른 엔터티 클래스에서 공통적으로 사용될 속성을 정의한다.		
구분	Name	Type	Visibility
	Description		
Attributes	createdDate	LocalDateTime	private
	엔터티가 생성된 날짜와 시간을 나타내는 필드이다. 엔터티가 생성될 때 자동으로 현재 날짜와 시간이 할당된다.		
구분	Name	Arguments	Returns
	Description		
Operations	없음		

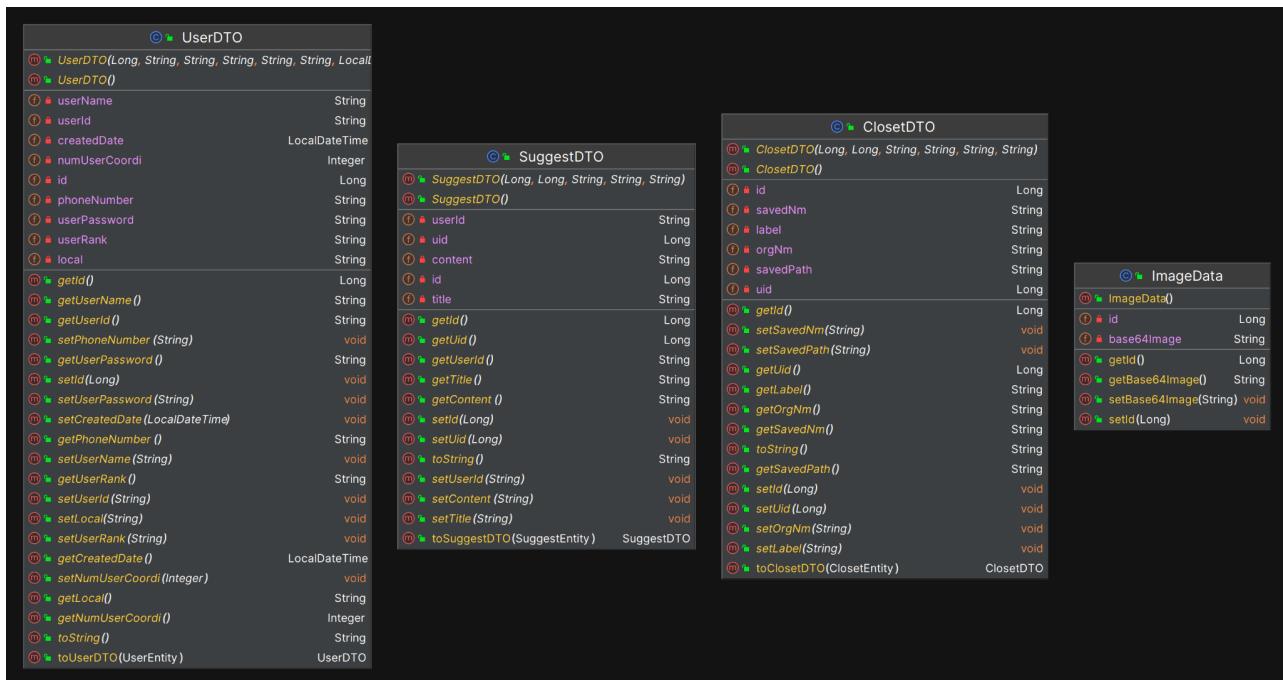
UserEntity			
Class	데이터베이스의 ‘user_table’ 테이블과 매핑되는 엔터티이다. ‘TimeEntity’를 상속함으로써, 회원가입 날짜와 시간을 자동으로 기록하는 기능을 가지고 있다.		
구분	Name	Type	Visibility
	Description		

	Description		
Attributes	id	UserService	private
	엔터티의 기본키로 사용되는 필드로, 자동으로 증가하는 값으로 설정되어 있다.		
	userNamed	OtherComponen t	private
	사용자의 이름을 나타내는 문자열 필드이다.		
	userId	String	private
	사용자의 아이디를 나타내는 문자열 필드로, 중복이 허용되지 않도록 '@Column(unique = true)' 어노테이션이 적용되어 있다.		
	userPassword	String	private
	사용자의 비밀번호를 나타내는 문자열 필드이다.		
	phoneNumber	String	private
	사용자의 전화번호를 나타내는 문자열 필드이다.		
	userRank	String	private
	사용자의 등급을 나타내는 문자열 필드로, 기본값이 'Bronze' 이다.		
	local	String	private
	사용자의 거주지를 나타내는 문자열 필드로, 기본값이 'Daegu' 이다.		
	numUserCoordi	Integer	private

	사용자가 추천받기 버튼을 누른 횟수를 나타내는 정수형 필드로, 기본값은 0이다.		
구분	Name	Arguments	Returns
	Description		
Operations	toUserEntity	UserDTO userDTO	String
	'UserDTO' 객체를 입력으로 받아서 'UserEntity'로 변환하는 정적 메서드이다. 이를 통해 DTO에서 엔터티로의 변환을 수행한다.		
	toUpdateUserEntity	UserDTO userDTO	String
	회원 업데이트에 사용되는 유저 DTO를 입력으로 받아서 'UserEntity'로 변환하는 정적 메서드이다. 회원 업데이트에서는 이미 존재하는 사용자의 ID가 존재하기 때문에, ID 값을 설정해주어야 한다.		

3.2 Data Type Class diagram

- Dressing 프로그램의 Data type을 Class diagram으로 작성했다.
 - 주로 Entity 클래스에서 받아온 값을 Controller와 Service 등에서 사용하기 위해 저장하는 클래스들의 구조이다.
 - 또한, 자주 사용되는 data를 저장하는 클래스 구조이다.



[그림 3 - 2] Data Type Class diagram

ClosetDTO			
Class	사용자의 옷장 아이템에 대한 데이터 전송 객체로, 옷장 아이템의 정보를 표현하고 전달하는 역할을 한다.		
Description	Name	Type	Visibility
구분			

	Description		
Attributes	id	Long	private
	온 아이템의 고유 식별자 (Primary key)		
	uid	Long	private
	사용자의 고유 식별자(Foregin Key)		
	label	String	private
	온 아이템의 라벨 또는 카테고리		
	orgNm	String	private
	온 아이템의 원본 파일명		
	savedNm	String	private
Operations	저장된 온 아이템의 파일명		
	savedPath	String	private
저장된 온 아이템의 파일 경로			
구분	Name	Arguments	Returns
Description			
Operations	toCloset	ClosetEntity closetEntity	ClosetDTO
	ClosetEntity 인스턴스를 받아 해당 정보로 ClosetDTO 객체를 생성하고		

	반환한다.
--	-------

ImageData			
Class Description	데이터 전송 객체이다.		
구분	Name	Type	Visibility
	Description		
Attributes	base64Image	String	private
	이미지의 Base64 인코딩된 문자열을 저장한다. 이 문자열은 웹 페이지에 이미지를 표시하는 데 사용된다.		
	id	Long	private
이미지의 고유 식별자를 저장한다. 이 식별자는 이미지를 관리하거나 참조하는 데 사용된다.			
구분	Name	Arguments	Returns
	Description		
Operations	getBase64Image	-	String
	base64Image 필드의 값을 반환한다.		
	setBase64Image	String	Void

	base64Image	
base64Image 필드의 값을 설정한다.		
getId	-	Long
id 필드의 값을 반환한다.		
setId	-	Long
id 필드의 값을 설정한다.		

SuggestDTO			
Class	사용자 건의사항에 관한 데이터를 전달하는 객체로, 건의사항의 ID, 사용자 ID, 사용자명, 제목, 내용을 포함한다.		
구분	Name	Type	Visibility
	Description		
Attributes	id	Long	private
	건의사항의 고유 식별자이다.		
	uid	Long	private
건의를 제출한 사용자의 고유 식별자이다.			
userId	String	private	

	건의를 제출한 사용자의 식별 이름이다.		
	title	String	private
	건의사항의 제목이다.		
	content	String	private
	건의사항의 내용이다.		
구분	Name	Arguments	Returns
	Description		
Operations	toSuggestDTO	SuggestEntity suggestEntity	SuggestDTO
	SuggestEntity 인스턴스를 받아 해당 정보로 SuggestDTO 객체를 생성하고 반환한다. 이 과정에서 건의사항의 ID, 사용자 ID와 이름, 제목, 내용을 SuggestDTO에 설정한다.		

UserDTO			
Class	사용자 정보를 저장하고 전달하는 데이터 전송 객체로, 사용자의 기본 정보와 상태를 포함한다.		
구분	Name	Type	Visibility
	Description		

Attributes	id	Long	private
	사용자의 고유 식별자이다.		
	userName	String	private
	사용자의 이름이다.		
	userId	String	private
	사용자의 식별 아이디이다.		
	userPassword	String	private
	사용자의 비밀번호이다.		
	phoneNumber	String	private
	사용자의 전화번호이다.		
userRank	String	private	
사용자의 등급이다.			
createDate	LocalDateTime	private	
사용자 계정의 생성 날짜 및 시간이다.			
local	String	private	
사용자의 지역, 대구로 고정되어 있다.			
numUserCoordi	Integer	private	
사용자가 생성한 코디(조합)의 수이다.			

구분	Name	Arguments	Returns
	Description		
Operations	toUserDTO	UserEntity userEntity	UserDTO UserEntity 객체를 받아 해당 정보를 UserDTO 객체로 변환하고 반환한다. 사용자의 ID, 이름, 아이디, 전화번호, 등급, 생성날짜, 지역, 코디 수를 설정한다.

3.3 Component Classes

ApplicationRunnerComponent			
Class	Spring Boot 애플리케이션 구동 후 사용자 등급 업데이트를 실행하는 class		
Description			
구분	Name	Type	Visibility
	Description		
Attributes	userService	userService	private
	사용자 관련 서비스를 제공하며, 서버 시작 시 사용자 등급 업데이트에 사용		
구분	Name	Argument	Returns
	Description		
Operations	run	ApplicationArguments args	void
	서버 시작 시 자동으로 호출되어 사용자 등급을 업데이트하는 기능을 수행		

OtherComponent	
Class	웹 어플리케이션에서 클라이언트에게 JavaScript 경고 메시지를 보내는
Description	기능을 제공하는 클래스이다.

구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
구분	Name	Arguments	Returns
	Description		
Operations	AlertMessage	HttpServletResponse response, String message	void
	클라이언트에게 JavaScript 경고창을 표시하고, 이전 페이지로 되돌아가는 스크립트를 전송		

PythonModelComponent			
Class Description	주어진 이미지 경로에 대한 레이블을 파이썬 스크립트를 실행하여 얻는 기능을 제공하는 클래스		
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
	Name	Arguments	Returns

구분	Description		
	getLabel	String imagePath	String
Operations	주어진 이미지 경로를 파이썬 스크립트에 전달하고, 스크립트를 실행하여 이미지의 레이블을 얻는다. 파이썬 스크립트의 출력 결과는 표준 출력으로 읽어와 마지막으로 읽은 줄을 레이블로 반환한다.		

3.4 Controller Classes

AdminController			
Class Description	웹 어플리케이션에서 관리자 관련 기능, 사용자 관리 기능을 제공하는 컨트롤러이다.		
구분	Name	Type	Visibility
	Description		
Attributes	userService	UserService	private
	사용자 관련 서비스를 제공하는 객체, 사용자 데이터의 조회, 수정, 삭제 등의 로직을 처리한다.		
구분	Name	Arguments	Returns
	Description		
Operations	findAll	Model model	String
	모든 사용자의 목록을 조회하여 “admin/list” 뷰에 전달한다.		
	findById	Long id, Model model	String
	특정 ID를 가진 사용자의 상세 정보를 조회하여 “admin/detail” 뷰로 전달한다.		
	updateForm	Long id, Model model	String
	특정 ID를 가진 사용자의 정보를 수정하기 위한 폼을 “admin/update” 뷰에 전달한다.		
	update	UserDTO	String
	제공된 UserDTO 객체의 정보로 사용자 정보를 업데이트하고, 사용자 목록 페이지로 리다이렉트한다.		
	deleteById	Long id	String
	특정 ID를 가진 사용자를 삭제하고, 사용자 목록 페이지로 리다이렉트한다.		

CategoryController

Class Description	웹 어플리케이션에서 다양한 카테고리에 따른 상품 표시를 관리하고, 사용자 세션에 선택된 카테고리를 설정한 후 메인 페이지로 리다이렉트하는 기능을 제공하는 컨트롤러이다.		
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
구분	Name	Arguments	Returns
	Description		
Operations	showMainCategory[CategoryName] (CategoryName은 Total, Outer, Top, Bottom, Shoes, Bag, Like, Dislike)	HttpSession session	String
	사용자 세션에 특정 카테고리를 설정하고, 사용자를 메인 페이지로 리다이렉트한다.		

ClosetController			
Class Description	사용자의 옷장 관련 기능, 파일 업로드 및 이미지 삭제 기능을 관리하는 컨트롤러이다.		
구분	Name	Type	Visibility
	Description		
Attributes	closetService	ClosetService	private
	사용자 옷장 관련 서비스를 제공하는 객체, 파일 저장 및 이미지 삭제와 같은 로직을 처리한다.		
구분	Name	Arguments	Returns
	Description		
Operations	testUploadForm	-	String
	파일 업로드를 테스트하기 위한 뷰("uploadTest")로 이동한다.		
	uploadFile	List<MultipartFile> files, HttpSession httpSession	String
	여러 파일을 업로드하고, 업로드된 파일을 처리한 후 메인 페이지로 리다이렉트한다.		

CoordiController			
Class Description	사용자에게 날씨에 맞는 옷 조합을 추천하고, 사용자 반응(좋아요, 싫어요)를 처리하는 컨트롤러이다.		
구분	Name	Type	Visibility
	Description		
Attributes	closetService	ClosetService	private
	사용자 옷장 관련 서비스를 제공하는 객체, 파일 저장 및 이미지 삭제와 같은 로직을 처리한다.		
	userService	UserService	private
	사용자 데이터 및 관련 로직을 처리하는 서비스이다.		
	otherComponent	OtherComponen t	private
	사용자에게 경고 메시지를 보내는데 사용된다.		
	closetService	ClosetService	public
	옷장 관련 데이터와 상호작용을 담당하는 서비스이다.		
Operations	coordiService	CoordiService	public
	옷 조합 추천 및 관련 데이터 처리를 담당하는 서비스이다.		
	구분	Name	Arguments
	Description		
Operations	coordiForm	HttpSession session, HttpServletRequest response	String
	날씨에 맞는 옷 조합을 추천하고, 조건이 맞지 않을 경우 경고 메시지를 보내며, 추천 결과를 coordi 뷰에 전달한다.		
	likeCoordi	HttpSession session	String
	사용자가 좋아요를 누른 코디에 대한 반응을 처리하고 JSON 형태로 결과를 반환한다.		
	dislikeCoordi	HttpSession	String

		session	
사용자가 싫어요를 누른 코디에 대한 반응을 처리하고 JSON 형태로 결과를 반환한다.			
	handleUserReaction	HttpSession session, boolean isLike	String
사용자의 좋아요/싫어요 반응을 공통적으로 처리하고 사용자의 코디 추천 횟수를 JSON 형태로 반환한다.			

HomeController			
Class Description	기본 페이지 요청을 처리하는 컨트롤러 클래스이다.		
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
구분	Name	Arguments	Returns
	Description		
Operations	index		String
	기본 페이지 요청을 처리하는 메서드로 index 문자열을 반환한다. 기본 페이지 요청이 들어왔을 때 index 템플릿 페이지로 이동하도록 사용한다.		

ShowMainController			
Class Description	웹 어플리케이션의 메인 페이지를 표시하며, 사용자의 사진과 좋아요/싫어요 리스트, 날씨 정보를 포함하여 뷰에 전달하는 컨트롤러이다.		
구분	Name	Type	Visibility
	Description		
Attributes	weatherService	WeatherService	private

	날씨 정보를 제공하는 서비스이다.		
	closetService	ClosetService	private
	옷장과 관련된 데이터와 상호작용을 처리하는 서비스이다.		
	coordiService	CoordiService	private
	옷 조합 추천 및 관련 데이터 처리를 담당하는 서비스이다.		
	userService	UserService	private
	사용자 관련 데이터 처리를 담당하는 서비스이다.		
구분	Name	Arguments	Returns
	Description		
Operations	show MainPage	Model model, HttpSession session	String
	대구의 날씨 정보와 사용자의 카테고리별 사진 또는 좋아요/싫어요 리스트를 뷰에 전달하고, “main” 페이지를 반환한다.		

SuggestController			
Class Description	사용자의 건의사항을 처리하고, 관리자가 건의함을 확인할 수 있는 서비스이다.		
구분	Name	Type	Visibility
	Description		
Attributes	suggestService	SuggestService	private
	사용자 건의사항 관련 서비스를 제공하는 객체, 건의사항 등록, 건의 목록 확인과 같은 로직을 처리한다.		
구분	Name	Arguments	Returns
	Description		
Operations	saveForm		String
	사용자가 건의하기 페이지에 접근할 때 호출되며, “suggest”라는 뷰를 반환한다.		
	save	SuggestDTO suggestDTO,	String

		HttpSession httpSession	
사용자가 건의하기 페이지에서 등록 버튼을 눌렀을 때 호출되며, 사용자가 입력한 데이터를 받아 'SuggestDTO' 객체로 변환한다. 그리고 해당 건의를 데이터베이스에 저장하고, 마지막에는 '/main'으로 redirect한다.			
boardFindAll	Model model	String	
관리자가 건의함 페이지에 접근할 때 호출되며, 건의된 목록을 조회하여 'admin/board' 부에 전달한다.			

UserController			
Class Description	웹 어플리케이션의 회원가입과 로그인을 표시하며 사용자의 회원가입, 로그인,email 체크 정보를 포함하여 뷰에 전달하는 컨트롤러이다.		
구분	Name	Type	Visibility
	Description		
Attributes	userService	UserService	public
	UserService 객체를 주입 받는다		
	otherComponent	OtherComponent	public
	otherComponent 객체를 주입 받는다.		
구분	Name	Arguments	Returns
	Description		
Operations	joinForm		String
	회원가입 페이지 폼 작성 데이터를 받아온다.		
	join	UserDTO, String, HttpServletRequest	String
	회원가입을 하여 회원가입 실패시 "Failed Join!!"이라는 메세지를 반환하며 성공시 login창을 실행시킨다.		
	loginForm		String
	로그인 페이지 폼 작성 데이터를 받아온다.		
	login	UserDTO, HttpSession, HttpServletRequest	String

		Response	
로그인을 하여 관리자는 관리자 일반 사용자는 일반 사용자로 로그인이 된다.			
logout	HttpSession	String	
logout를 하여 세션을 무효화 한다.			
emailCheck	String	String	
emaill을 check하여 결과를 반환한다.			

3.5 Service Classes

ClosetInfoService			
Class Description	옷 정보에 관한 서비스 처리를 담당하는 클래스이다.		
구분	Name	Type	Visibility
	Description		
Attributes	closetInfoRepository	ClosetInfoRepository	private
	ClosetInfoRepository 객체를 주입받는다.		
구분	Name	Arguments	Returns
	Description		
Operations	init		void
	클래스 인스턴스화 후 초기화를 위한 메소드		
	insertClosetInfoData		void
	옷장 정보 데이터를 저장소에 삽입하는 메소드		

ClosetService			
Class Description	사용자가 등록할 옷 관련 로직 서비스 처리를 담당하는 클래스이다.		
구분	Name	Type	Visibility
	Description		
Attributes	pythonModelComponent	PythonModelComponent	private
	PythonModelComponent 객체를 주입받는다.		
Attributes	closetRepository	ClosetRepository	private
	ClosetRepository 객체를 주입받는다.		
Attributes	userRepository	UserRepository	private
	UserRepository 객체를 주입받는다.		

구분	closetInfoRepository	ClosetInfoRepository	private
	ClosetInfoRepository 객체를 주입받는다.		
	fileStoragePath	String	private
	이미지가 저장될 하드디스크 폴더의 경로를 저장할 변수이다.		
	Name	Arguments	Returns
	Description		
	saveFile	MultipartFile , Long	Long
	사용자가 디스크에서 선택한 옷 사진을 DB에 저장하도록 하는 메소드이다		
	createDirectoryIfNotExists	String	void
	하드디스크에 경로명과 일치하는 폴더가 없다면 생성하는 메소드이다.		
Operations	getUserPhotos	Long	List<ClosetEntity
	사용자 id와 일치하는 옷 리스트들을 반환하는 메소드이다.		
	getBase64Image	String	String
	사용자의 좋아요/싫어요 반응을 공통적으로 처리하고 사용자의 코디 추천 횟수를 JSON 형태로 반환한다.		
	deleteImage	Long , Long	void
	이미지의 경로를 받아서 Base64Image를 얻는 메소드이다.		
	deleteFileFromSystem	String	void
	이미지 파일을 디스크에서 삭제하는 메소드이다.		
	findUserPhotosByCategory	Long ,String	List<ClosetEntity>
	카테고리와 일치하는 사용자의 옷 리스트를 반환하는 메소드이다.		
	toImageDataList	List<ClosetEntity>	List<ImageData>
	옷 리스트를 ImageData 리스트로 변환하여 반환하는 메소드이다.		

SchedulerService			
Class Description	Dressing 어플리케이션의 서버가 주기적으로 스케줄링해서 실행해야 할 일들을 관리하는 서비스 클래스이다		
구분	Name	Type	Visibility
	Description		
Attributes	userRepository	UserRepository	private
	UserRepository 객체를 주입 받는다.		
구분	Name	Arguments	Returns
	Description		
Operations	updateRankMidnight		void
	매일 오전 12시에 모든 사용자들의 rank가 update 된다		

SuggestService			
Class Description	건의 서비스 처리를 담당하는 클래스이다		
구분	Name	Type	Visibility
	Description		
Attributes	suggestRepository	SuggestRepository	private
	사용자 옷장 관련 서비스를 제공하는 객체, 파일 저장 및 이미지 삭제와 같은 로직을 처리한다.		
Operations	userRepository	UserRepository	private
	사용자 데이터 및 관련 로직을 처리하는 서비스이다.		
구분	Name	Arguments	Returns
	Description		
Operations	coordiForm	HttpSession session, HttpServletResponse response	String
	날씨에 맞는 옷 조합을 추천하고, 조건이 맞지 않을 경우 경고 메시지를 보내며, 추천 결과를 coordi 뷰에 전달한다.		

	likeCoordi	HttpSession session	String
사용자가 좋아요를 누른 코디에 대한 반응을 처리하고 JSON 형태로 결과를 반환한다.			
	dislikeCoordi	HttpSession session	String
사용자가 싫어요를 누른 코디에 대한 반응을 처리하고 JSON 형태로 결과를 반환한다.			
	handleUserReaction	HttpSession session, boolean isLike	String
사용자의 좋아요/싫어요 반응을 공통적으로 처리하고 사용자의 코디 추천 횟수를 JSON 형태로 반환한다.			

UserService			
Class Description	User의 정보와 login/logout 정보를 database에서 조회 혹은 업데이트를 하는 class		
구분	Name	Type	Visibility
Description			
Attributes	userRepository	UserRepository	private
userRepository를 가져온다.			
구분	Name	Arguments	Returns
Description			
Operations	join	UserDTO, String	int
	아이디, 비밀번호 중복을 확인하고 비밀번호를 추가하여 Database에 저장한다.		
	login	UserDTO	UserDTO
	회원이 입력한 아이디로 database에서 조회후 비밀번호가 일치하는지 일치안하는지 확인한다.		
	findAll		List
database에서 모든 유저를 리스트에 담아 DTO리스트로 반환한다.			

	findById	Long	UserDTO
	database에서 id를 이용해 찾은 유저의 DTO를 리턴한다.		
	updateForm	Long	UserDTO
	관리자 페이지로 업데이트 옮기고 수정한다.		
	update	UserDTO	void
	database의 회원정보를 업데이트한다.		
	deleteById	Long	void
	database에 저장된 회원을 id를 이용하여 삭제한다.		
	idCeck	String	String
	id를 database에서 검색후 조회결과를 반환한다.		
	updateRank		void
	모든 유저의 랭크를 조건에 의거하여 업데이트한다.		
	getNumLimit	String	int
	Rank에 따라 하루최대 옷추천개수를 반환한다.		
	saveUser	UserDTO	void
	UserDTO를 database에 저장하기 위해 UserEntity로 변환하여 업데이트 한다.		

WeatherService			
Class Description	대구의 날씨와 온도정보를 가져와 뷰에 전달하는 역할을 한다.		
구분	Name	Type	Visibility
	Description		
Attributes			
구분	Name	Arguments	Returns
	Description		
Operations	getDaeguWather		CompletableFuture<String>

			ure
대구의 현재 온도와 날씨정보를 파싱하여 뷰에 전달한다.			
getDaeguTempCelsius	daeguWeather	double	
대구의 현재 온도를 가져온다.			

3.6 Repository Class

ClosetinfoRepository			
Class	label(category명)으로 database에서 옷 데이터를 찾기위한 것이다.		
Description			
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
구분	Name	Arguments	Returns
	Description		
Operations	-	-	-

ClosetRepository			
Class	database에서 uid와 category로 closetEntity를 찾는것이다.		
Description			
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-

구분	Name	Arguments	Returns
	Description		
Operations	-	-	-

CoordiRepository			
Class Description	database에서 uid와 heart를 이용하여 CoordiEntity를 찾는것이다.		
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
구분	Name	Arguments	Returns
	Description		
Operations	-	-	-

SuggestRepository	
Class	SuggestRepository로 database에 접근하기 위한것이다.

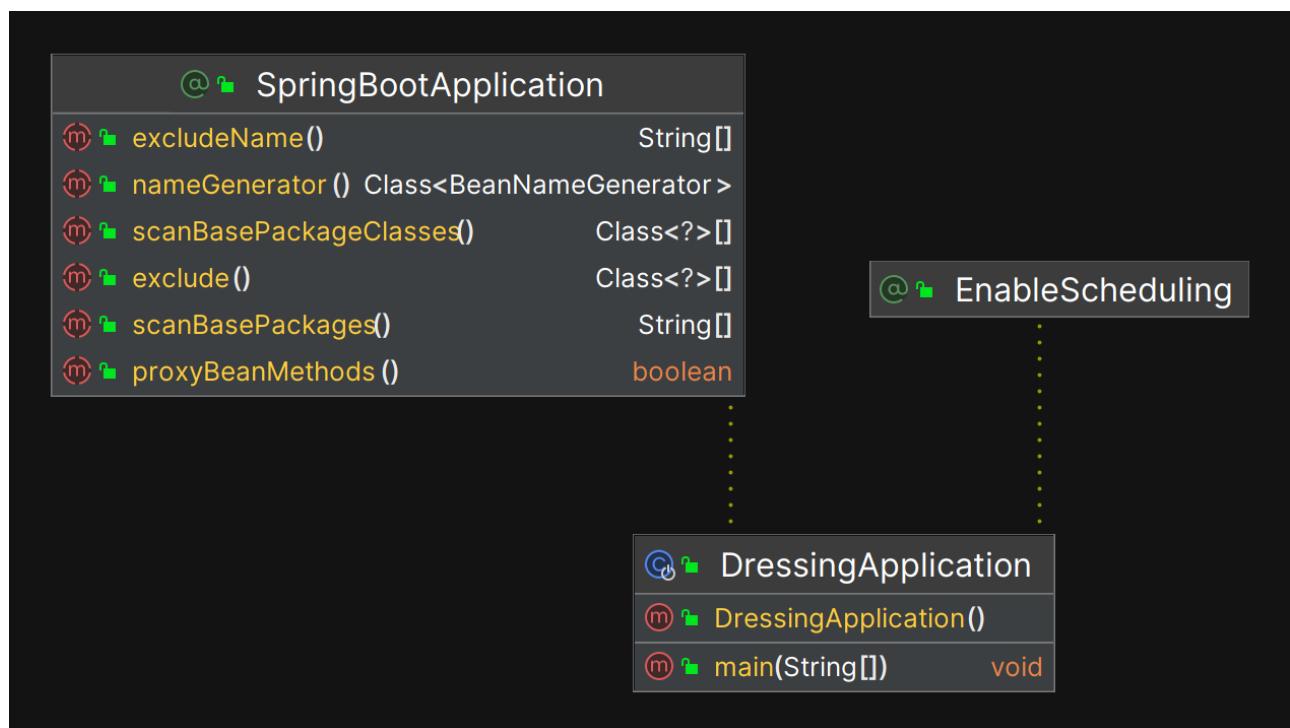
Description			
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
구분	Name	Arguments	Returns
	Description		
Operations	-	-	-

ClosetRepository			
Class	UserRepository로 database에 접근하여 id로 회원정보를 조회하고 사용자		
Description	id를 이용하여 user의 rank를 업데이트하는것이다.		
구분	Name	Type	Visibility
	Description		
Attributes	-	-	-
구분	Name	Arguments	Returns
	Description		
Operations	-	-	-

3.7 기능 Class diagram

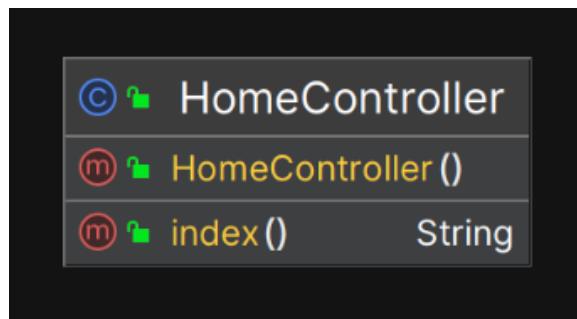
- Use case 기능을 기반으로 Class 들의 상관관계를 나타내는 Class Diagram이다.
- MVC 패턴을 적용해 설계했다.
- Class diagram을 집중적으로 설계했다.

3.7.1 DressingApplication Class diagram



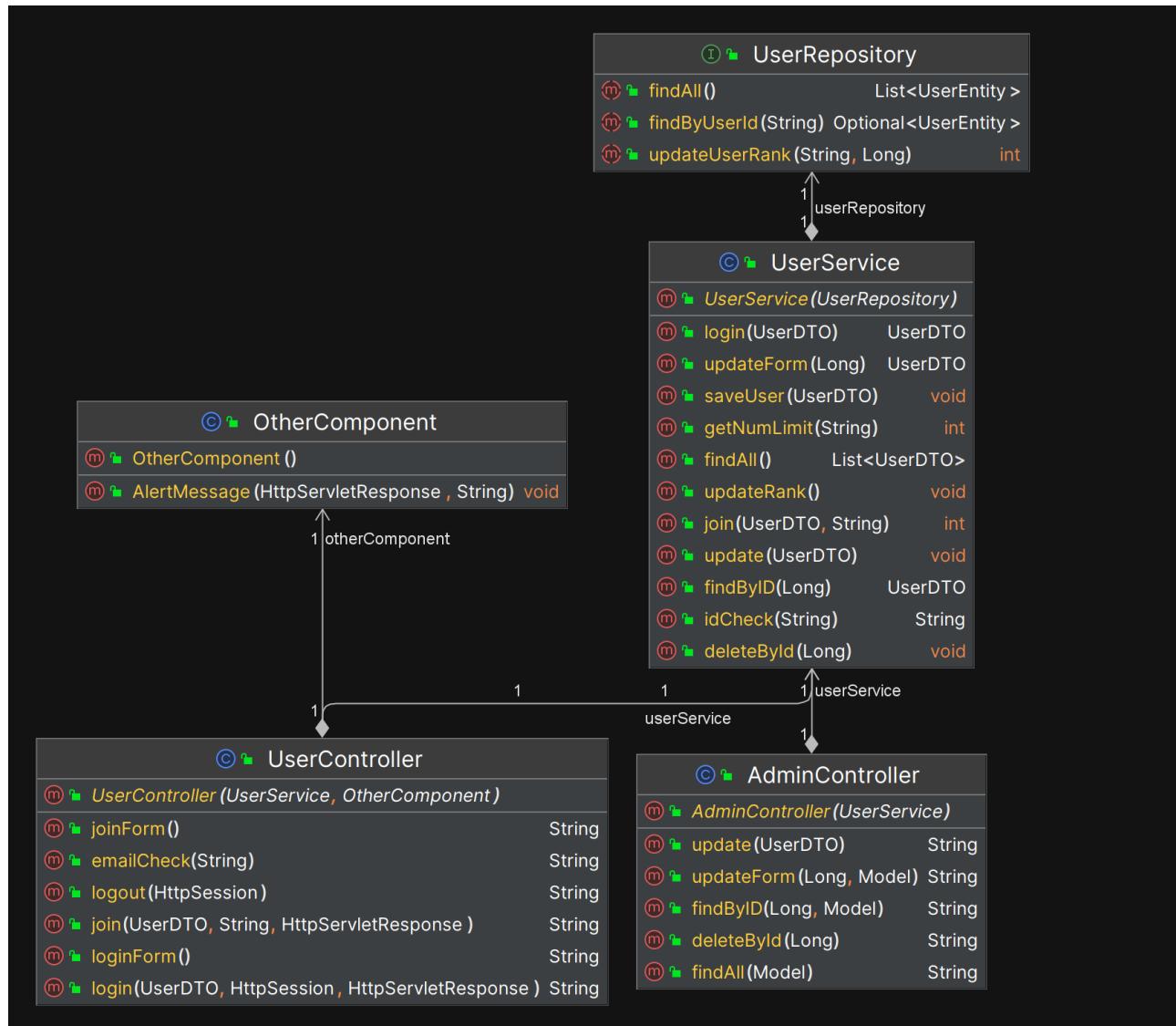
[그림 3 - 3] DressingApplication Class diagram

3.7.2 Home Class diagram



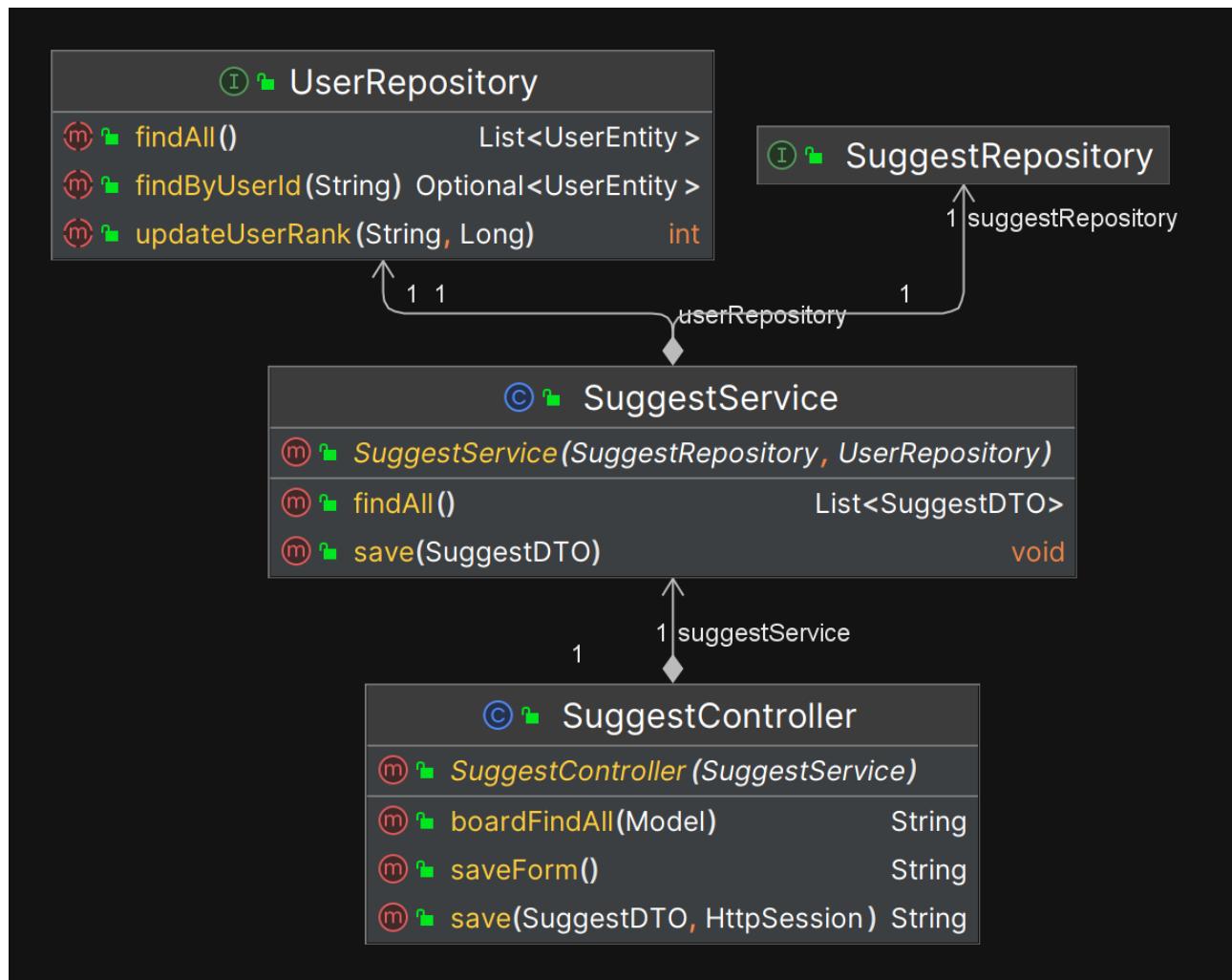
[그림 3 - 4] Home Class diagram

3.7.3 User, Admin Class diagram



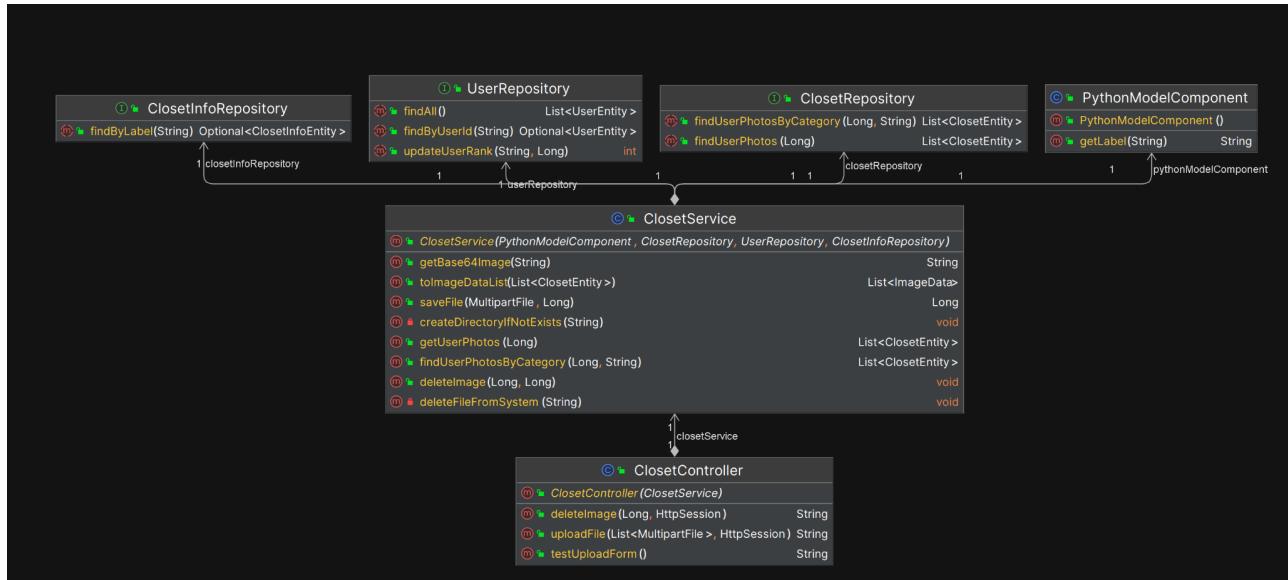
[그림 3 - 5] User, Admin Class diagram

3.7.4 Suggest Class diagram



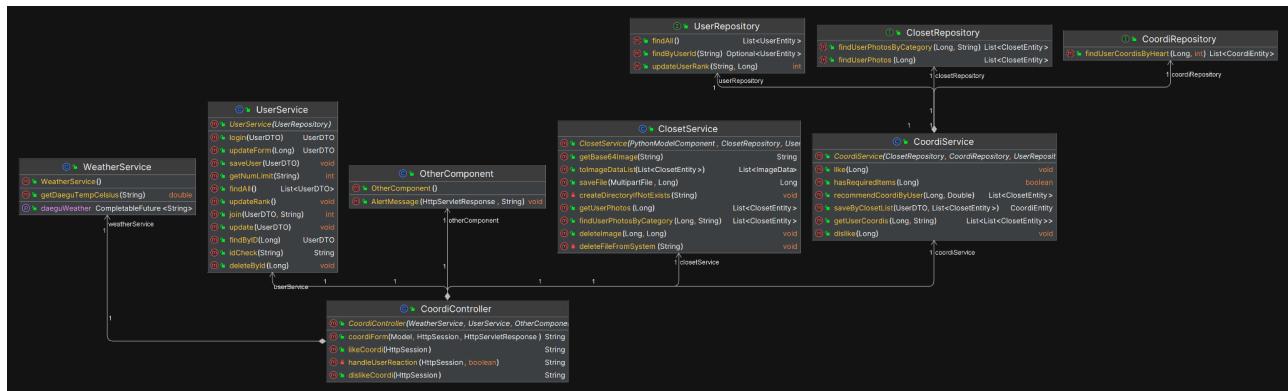
[그림 3 - 6] Suggest Class diagram

3.7.5 Closet Class diagram



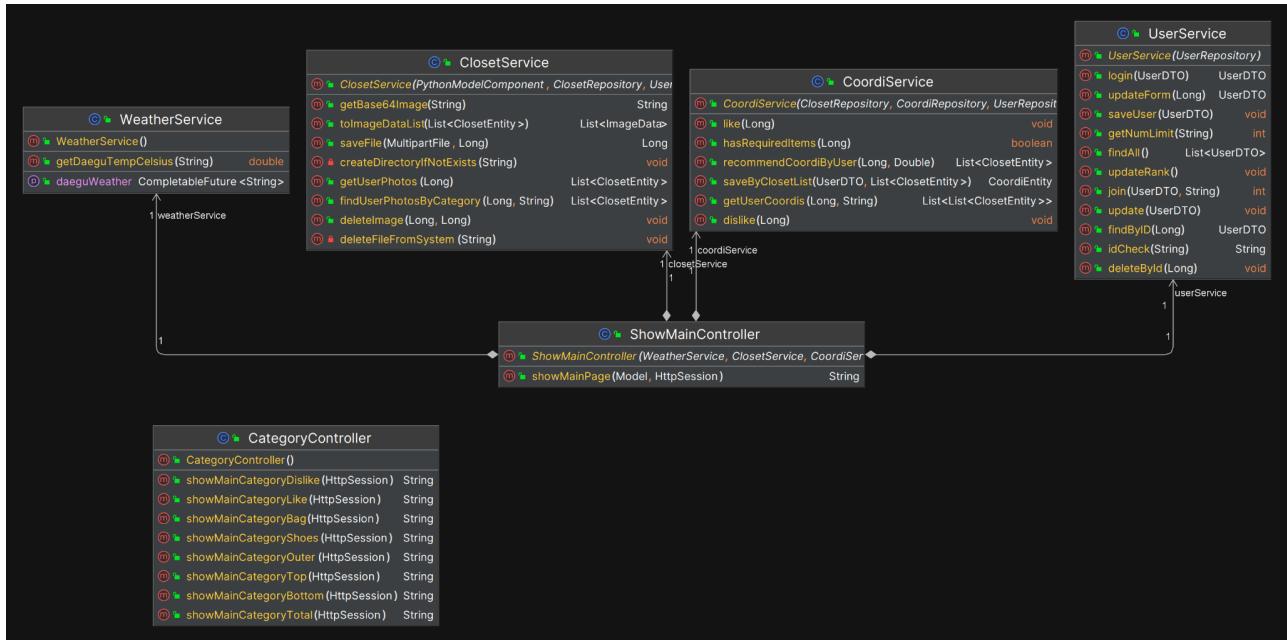
[그림 3 - 7] Closet Class diagram

3.7.6 Coordi Class diagram



[그림 3 - 8] Coordi Class diagram

3.7.7 ShowMain, Category Class diagram



[그림 3 - 9] ShowMain, Category Class diagram

4. Sequence diagram

이 장은 sequence diagrams와 그 설명을 제공한다. 고려 사항은 다음과 같다.

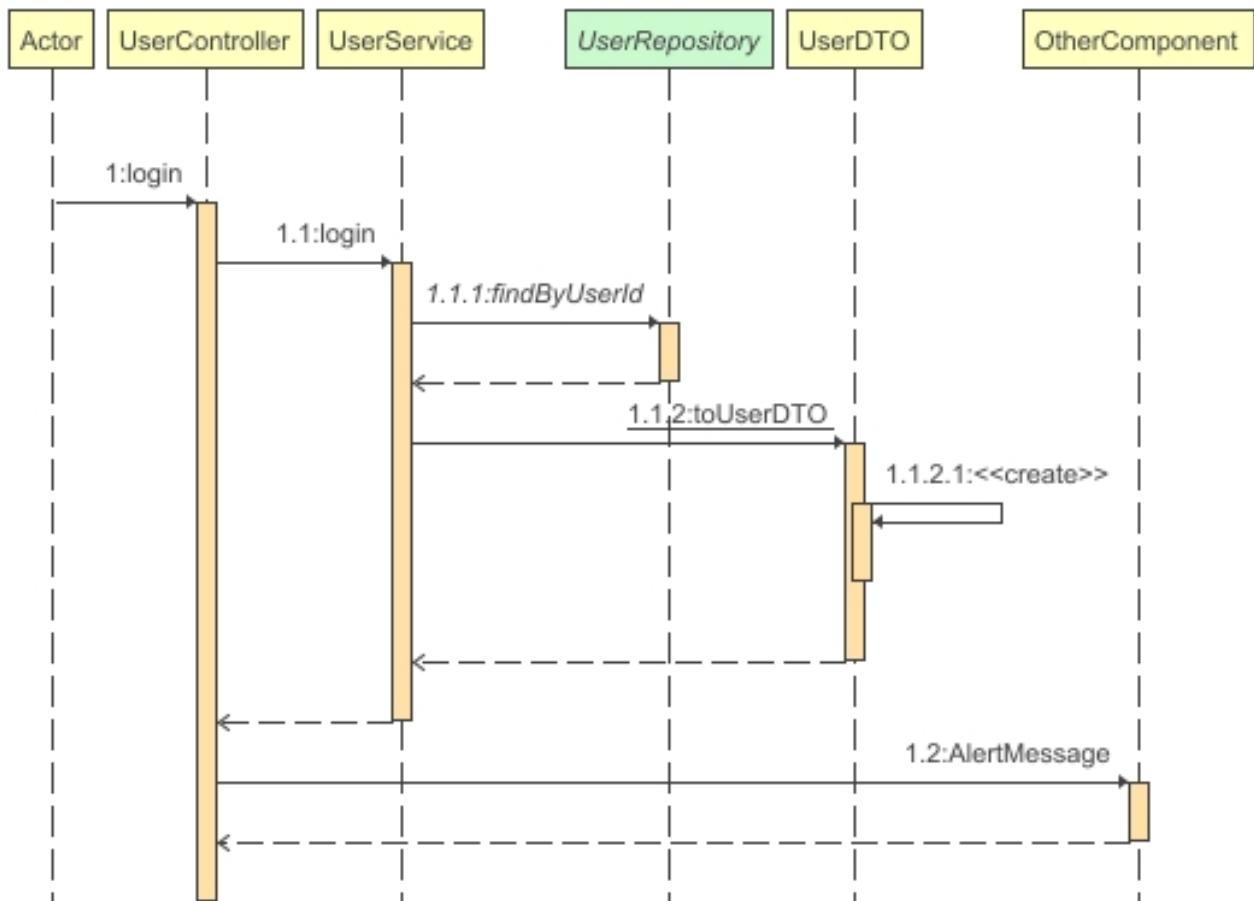
- 모든 sequence diagrams는 특정 use case와 대응된다.
- 모든 sequence diagrams는 특정 use case의 main/extension success scenario의 흐름을 따른다.
- 본 시스템에서는 객체 이름을 중요하게 생각하지 않으므로 생명선의 class 명칭은 객체이름과 함께 표현하지 않고 클래스 이름만 적었다.
- 메시지를 주고받는 클래스들은 모두 class diagram에서 연결 관계가 있다.
- sequence diagrams의 메시지는 CD의 operation 혹은 attributes이거나 CD에 없는 개념적인 (혹은 시스템 내부의) 과정(e.g. observing)/결과(e.g. creating plan complete)를 나타낸다.
- User는 실제 코드 상에 있는 클래스가 아니라 사용자를 뜻하는 개념적 클래스다.
- Repository에서 서버에 요청을 보낼 때는 Retrofit을 사용하는 것을 전제로 설계했으며 비동기 메시지 전달 방법(Call.enqueue()[8]로 구현)을 사용한다. -> 함수 호출 시 주로 비동기 메시지로 호출한다.
- Http request를 여러 개 보낼 때 편의를 위해 결과가 오는 순서를 지정했지만 실제 환경에서 그 순서는 다를 수 있다.
- Repository에서 요청을 보낼 때 (request~ 연산) 모두 ~Service의 연산을 사용한다. 자세히 하자면 repository에서 Service~의 request~를 호출하면 Call 객체를 반환한다. Repository는 call을 이용해 요청을 보낸다. 하지만 모두 공통적인 흐름일 뿐만 아니라 가독성을 해치기 때문에 request~를 repository 내부의 연산으로 단순화했다.
- 모든 request~는 code를 반환한다. 모든 sequence diagrams는 code = 0(성공)일 때를 가정한다. (success scenario)
- sequence diagrams의 description은 관련된 use case, class diagram과 흐름 묘사로

이뤄져 있다.

- 흐름 묘사는 sequence diagrams를 그대로 글에 옮겨 적은 것이다. 일치성과 모순성을 검사하기 위해 사용한다. 혹은 diagram으로 이해되지 않는 부분을 해소하는 데 사용한다.

4.1 User sequence diagram

4.1.1 Login

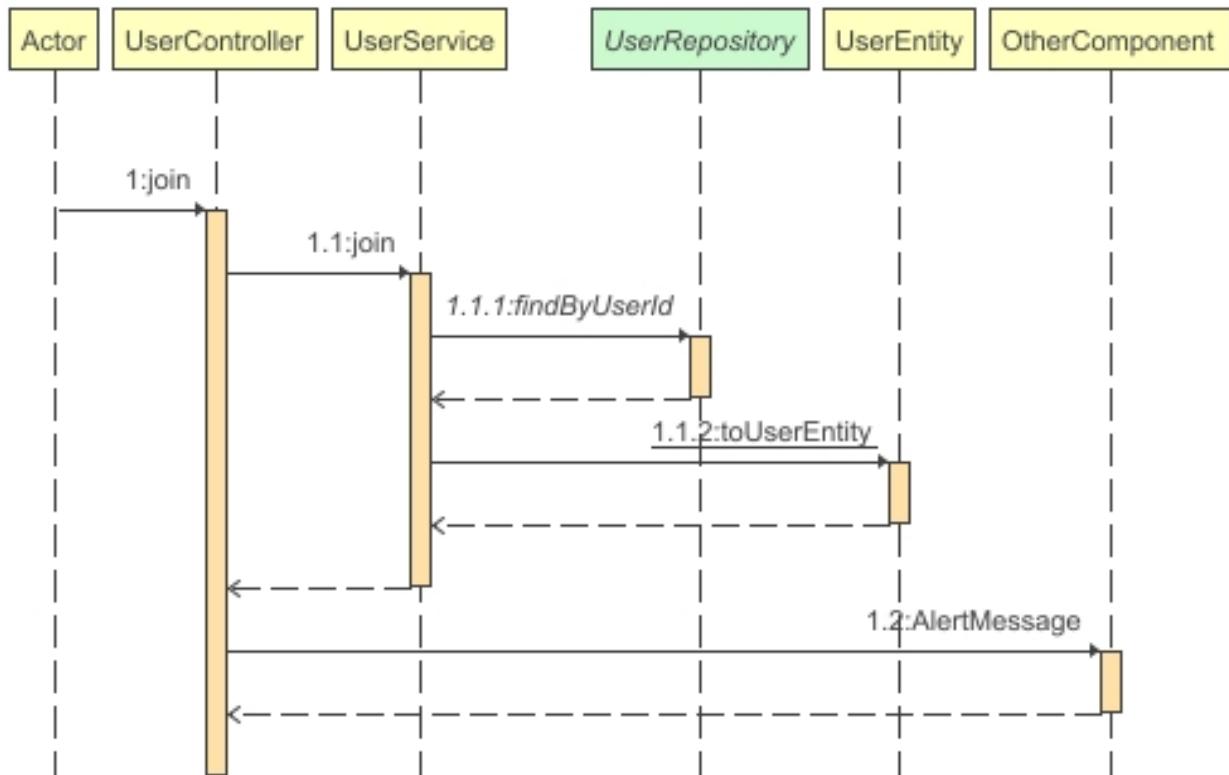


[그림 4 - 1] Login sequence diagram

사용자가 시스템에 로그인을 하는 use case를 나타내는 sequence diagram이다. User class description에서 <Use case #2>의 경우이다.

사용자가 웹페이지에 접속하면 기능을 시작한다. 사용자가 웹페이지에서 아이디 비밀번호를 입력한다. 이후 DB에서 해당 아이디 비밀번호의 존재유무를 파악한다. DB에 계정이 존재시 서버에 접속하였다는 요청을 보내고 약속 목록 화면으로 전환한다. DB에 계정이 존재하지 않는경우 계정명이 틀렸거나 비밀번호가 틀렸다는 메시지를 반환한다.

4.1.2 Join



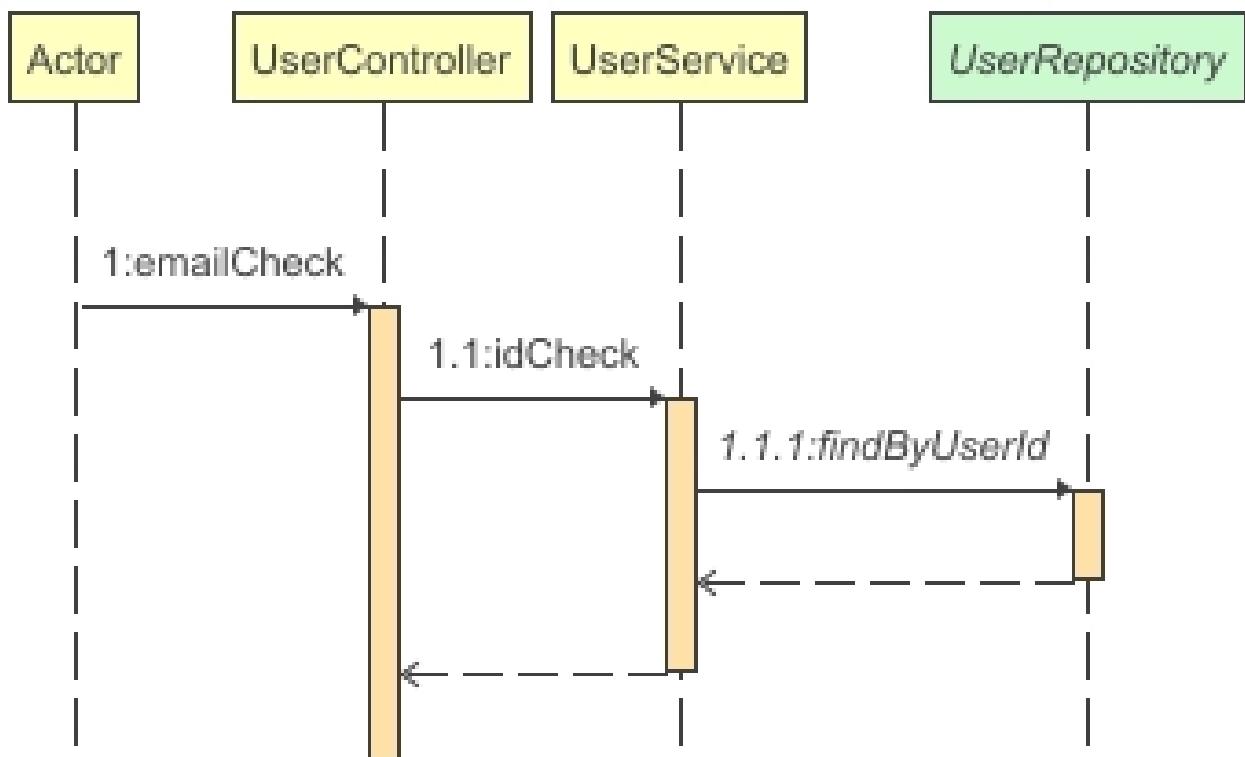
[그림 4 - 2] Join sequence diagram

사용자가 시스템에 로그인을 하는 use case를 나타내는 sequence diagram이다. User class description에서 <Use case #1>의 경우이다.

사용자가 회원가입버튼을 누르면 회원가입을 하는 화면이 나온다. 회원가입 화면에 이름, 아이디, 비밀번호, 비밀번호확인, 휴대폰 번호를 입력한후 회원가입 버튼을 누르면 조건에 만족시 서버에 계정을 저장후 로그인 화면으로 전환한다. 만약 아래의 조건을 충족하지 않는다면 회원가입화면으로 다시 반환된다. 조건은 아래와 같다.

- 1.비밀번호와 비밀번호 확인이 일치해야한다.
- 2.아이디가 DB에서 중복되지 않아야한다.
- 3.휴대폰번호가 DB에서 중복되지 않아야한다.

4.1.3 이메일 확인



[그림 4 - 3] email check sequence diagram

이 시퀀스 다이어그램은 사용자 또는 액터와 시스템의 여러 컴포넌트 간의 상호작용을 나타낸다. 이 다이어그램은 사용자 또는 액터와 시스템의 여러 컴포넌트 간의 상호작용을

나타낸다. 컴포넌트들은 'UserController', 'UserService', 그리고 'UserRepository'로 구성되어 있다. 다이어그램에서 보이는 상호작용은 다음과 같다.

액터가 'UserController'에 'emailCheck'라는 요청을 한다.

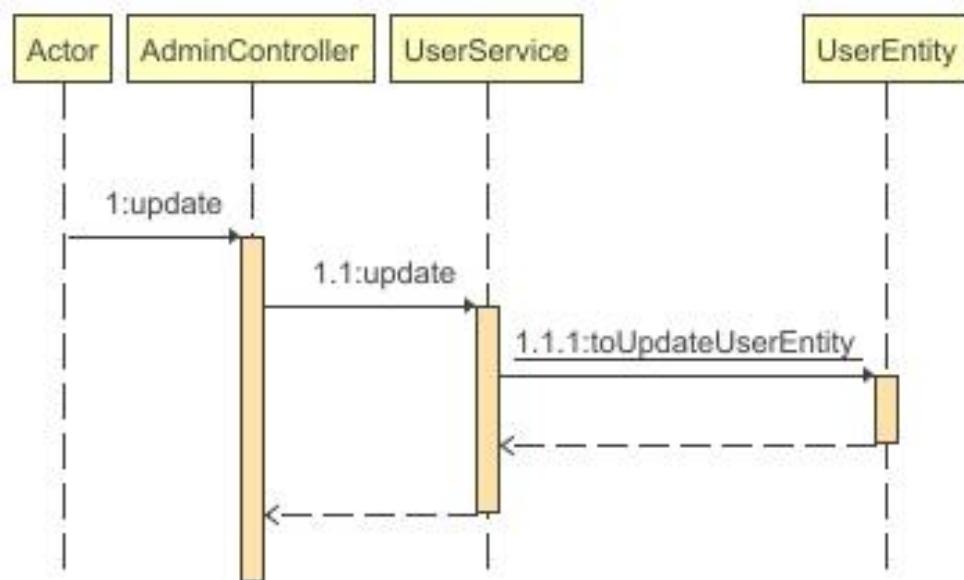
'UserController'는 'UserService'에 'idCheck'를 요청한다.

'UserService'는 'UserRepository'에 'findById'를 요청한다.

'UserRepository'는 요청된 정보를 찾아 응답으로 돌려준다.

4.2 Admin sequence diagram

4.2.1 회원정보 수정



[그림 4 - 4] modify user data sequence diagram

이 시퀀스 다이어그램은 사용자 엔티티의 업데이트 프로세스를 나타낸다. 다이어그램의 구성 요소는 'Actor', 'AdminController', 'UserService', 그리고 'UserEntity'로 구성되어 있다.

다이어그램에서 보이는 상호작용은 다음과 같다.

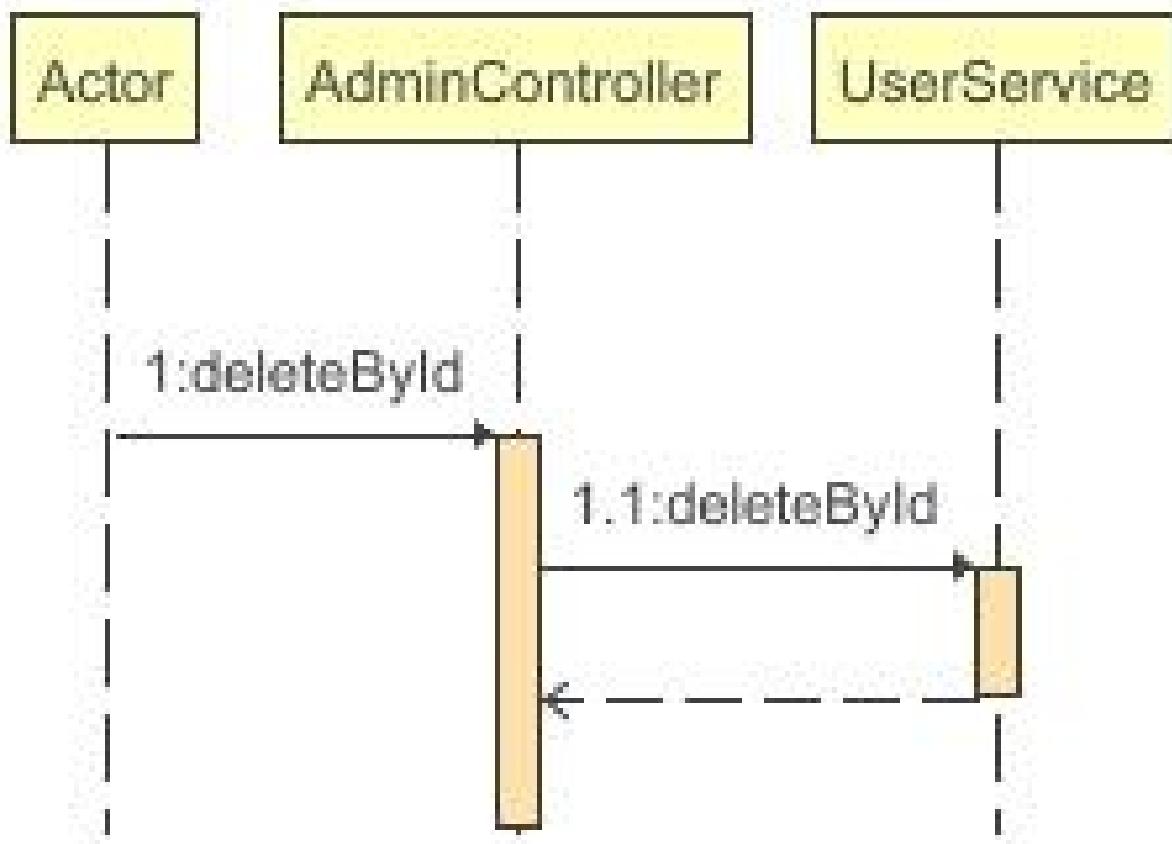
액터는 'AdminController'에 'update'라는 요청을 한다.

'AdminController'는 이 요청을 처리하기 위해 'UserService'에 'update'를 요청한다.

'UserService'는 'UserEntity'를 업데이트하기 위해 'toUpdateUserEntity' 메소드를 호출한다.

'UserEntity'는 업데이트가 완료되면 그 결과를 'UserService'로 비동기적으로 반환한다.

4.2.2 회원 삭제



[그림 4 - 5] delete user sequence diagram

이 시퀀스 다이어그램은 ID를 기준으로 사용자 정보를 삭제하는 프로세스를 나타낸다.

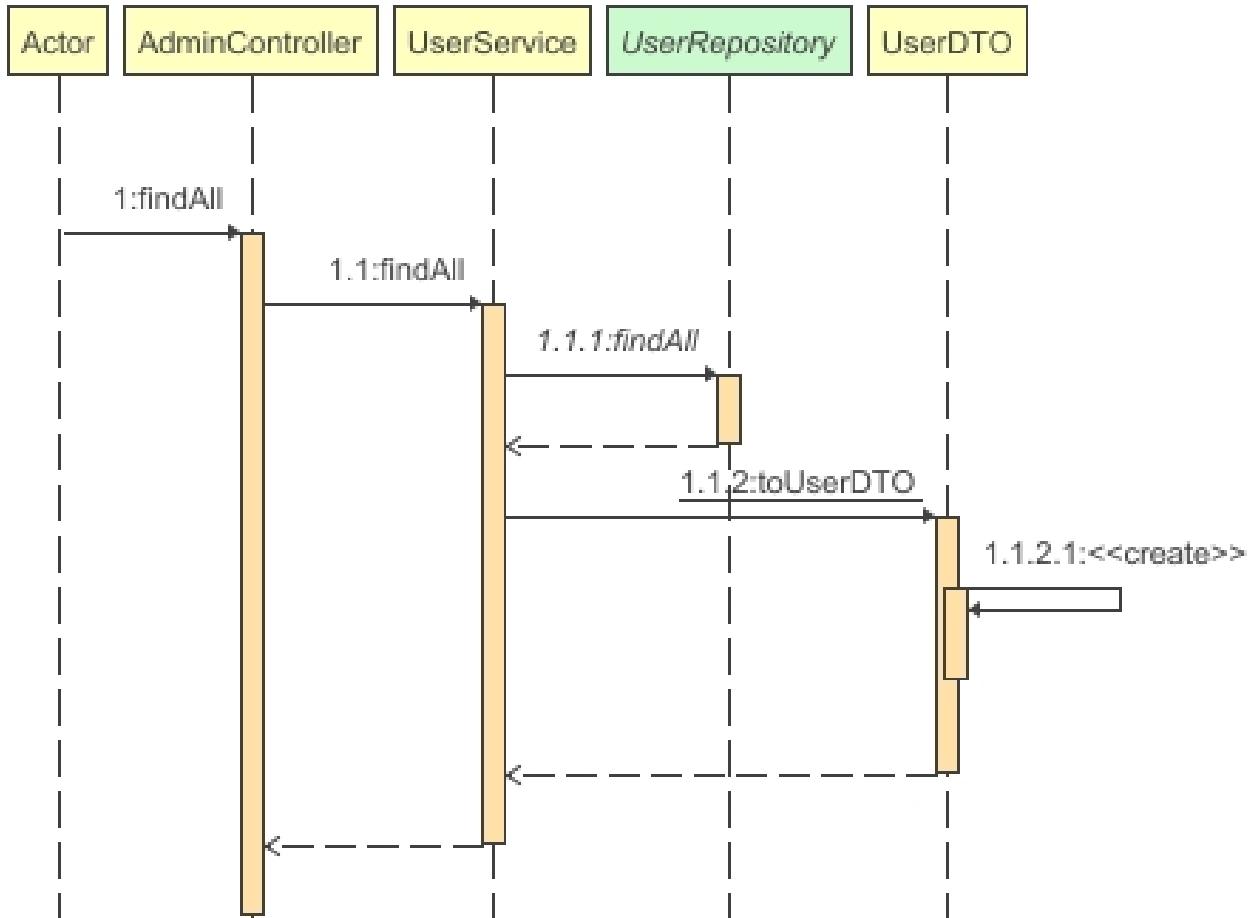
다이어그램에는 'Actor', 'AdminController', 그리고 'UserService' 세 가지 주요 컴포넌트가 표시되어 있다. 다이어그램에서 보이는 상호작용은 다음과 같다.

액터는 'AdminController'에 'deleteById'라는 명령을 전달한다.

'AdminController'는 이 요청을 처리하기 위해 'UserService'에 같은 이름의 'deleteById' 메소드를 호출한다.

'UserService'는 요청된 ID로 사용자 정보를 삭제하고, 그 결과를 'AdminController'로 반환합니다. 반환되는 정보는 점선의 화살표로 나타나 있는데, 이는 통상적으로 비동기 응답을 의미할 수 있다.

4.2.3 회원목록출력



[그림 4 - 6] print userlist sequence diagram

이 시퀀스 다이어그램은 ID를 통해 특정 사용자 정보를 조회하는 과정을 나타낸다.

다이어그램에는 'Actor', 'AdminController', 'UserService', 그리고 'UserDTO'로 구성되어 있다. 다이어그램에서 보이는 상호작용은 다음과 같다.

액터는 'AdminController'에게 특정 ID로 사용자를 조회하는 'findById' 요청을 한다.

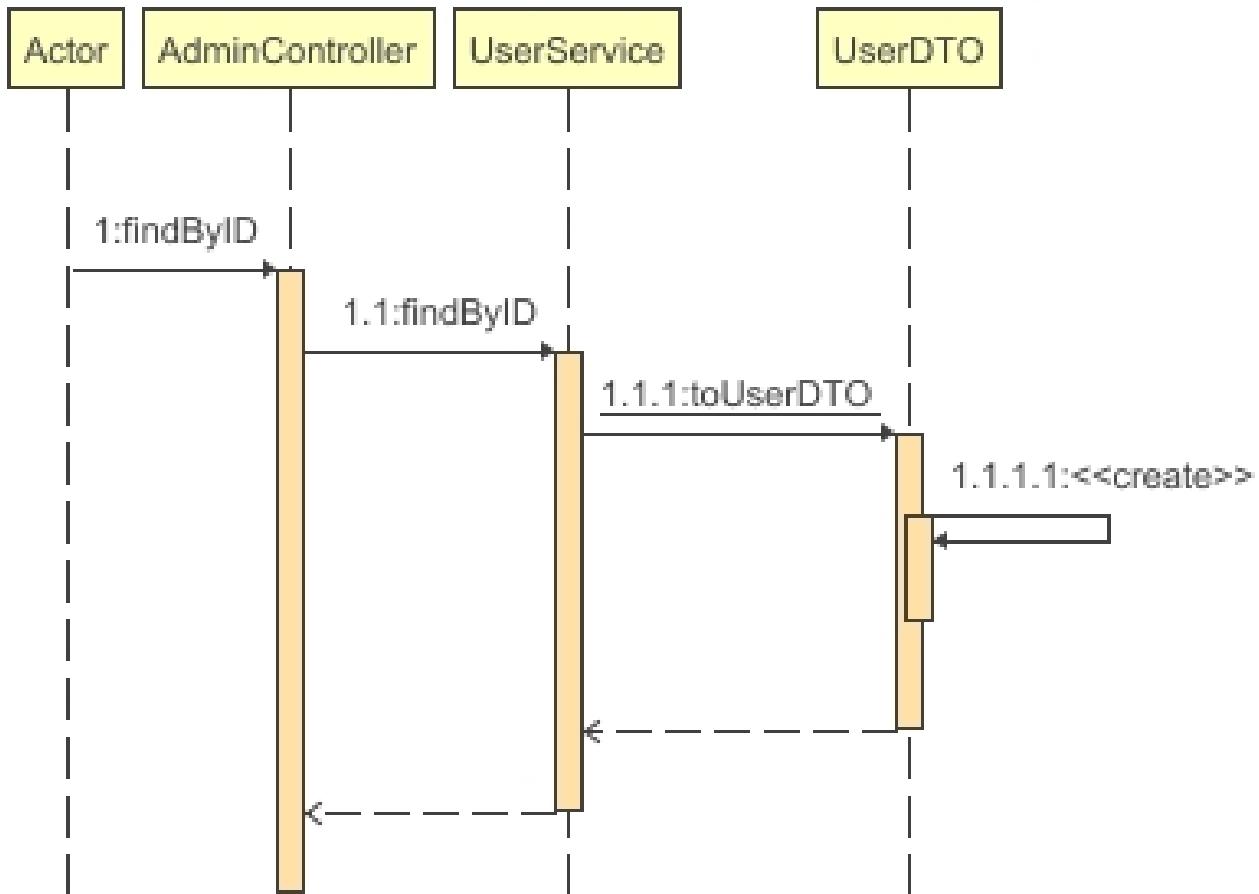
'AdminController'는 이 요청을 'UserService'에 전달한다.

'UserService'는 요청받은 ID에 해당하는 사용자 데이터를 처리하고 'UserDTO' 객체로 변환하는 'toUserDTO' 메소드를 호출한다.

'UserDTO' 객체가 생성된다('create' 작업이 표시됨).

생성된 'UserDTO' 객체가 'UserService'를 통해 'AdminController'로 반환되고, 최종적으로 액터에게 전달된다.

4.2.4 회원 상세정보 출력



[그림 4 - 7] print User Detail data sequence diagram

이 시퀀스 다이어그램은 ID를 통해 특정 사용자 정보를 조회하는 과정을 나타낸다. 다이어그램에는 'Actor', 'AdminController', 'UserService', 그리고 'UserDTO' 네 가지 주요 컴포넌트가 표시되어 있다. 다이어그램에서 보이는 상호작용은 다음과 같다.

액터는 'AdminController'에게 특정 ID로 사용자를 조회하는 'findById' 요청을 한다. 'AdminController'는 이 요청을 'UserService'에 전달한다.

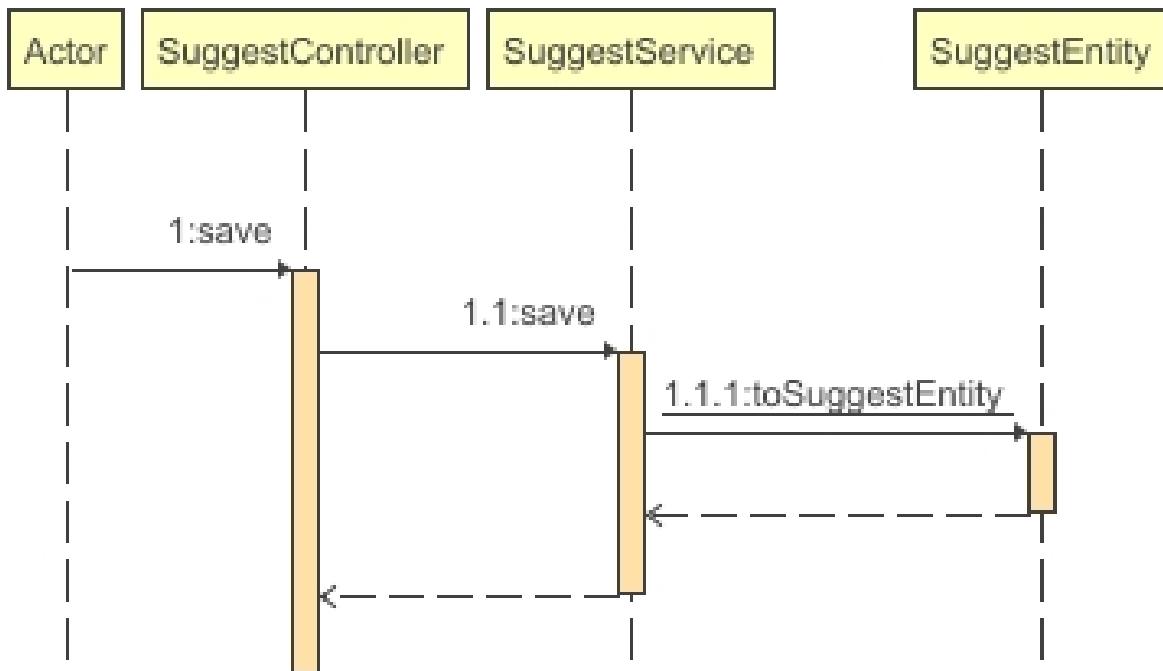
'UserService'는 요청받은 ID에 해당하는 사용자 데이터를 처리하고 'UserDTO' 객체로 변환하는 'toUserDTO' 메소드를 호출한다.

'UserDTO' 객체가 생성된다('create' 작업이 표시됨).

생성된 'UserDTO' 객체가 'UserService'를 통해 'AdminController'로 반환되고, 최종적으로 액터에게 전달된다.

4.3 Suggest sequence diagram

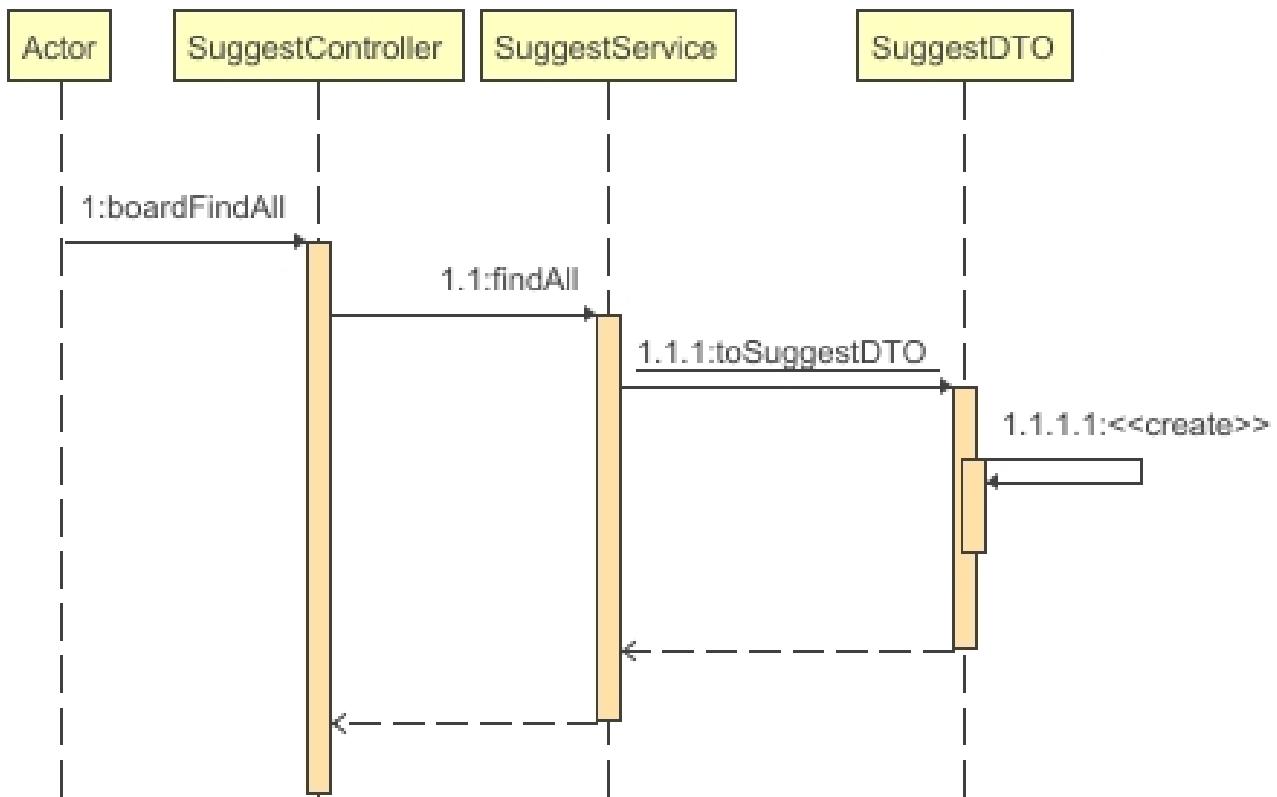
4.3.1 건의 작성하기



[그림 4 - 8] wire suggest sequence diagram

사용자가 시스템에 건의 작성하는 과정을 나타내는 sequence diagram으로 ‘SuggestController’, ‘SuggestService’, ‘SuggestEntity’ 컴포넌트간의 이루어지는 상호작용에 대한 diagram이다. 먼저 ‘Actor’가 ‘SuggestController’에 ‘save’라는 요청을 하면 ‘SuggestController’는 ‘SuggestService’에게 ‘save’ 요청을 하게 된다. ‘SuggestService’가 ‘SuggestEntity’에 ‘toSuggestEntity’ 메소드를 호출하고 응답을 함으로써 건의 작성이 이루어지게 된다.

4.3.2 건의목록 출력

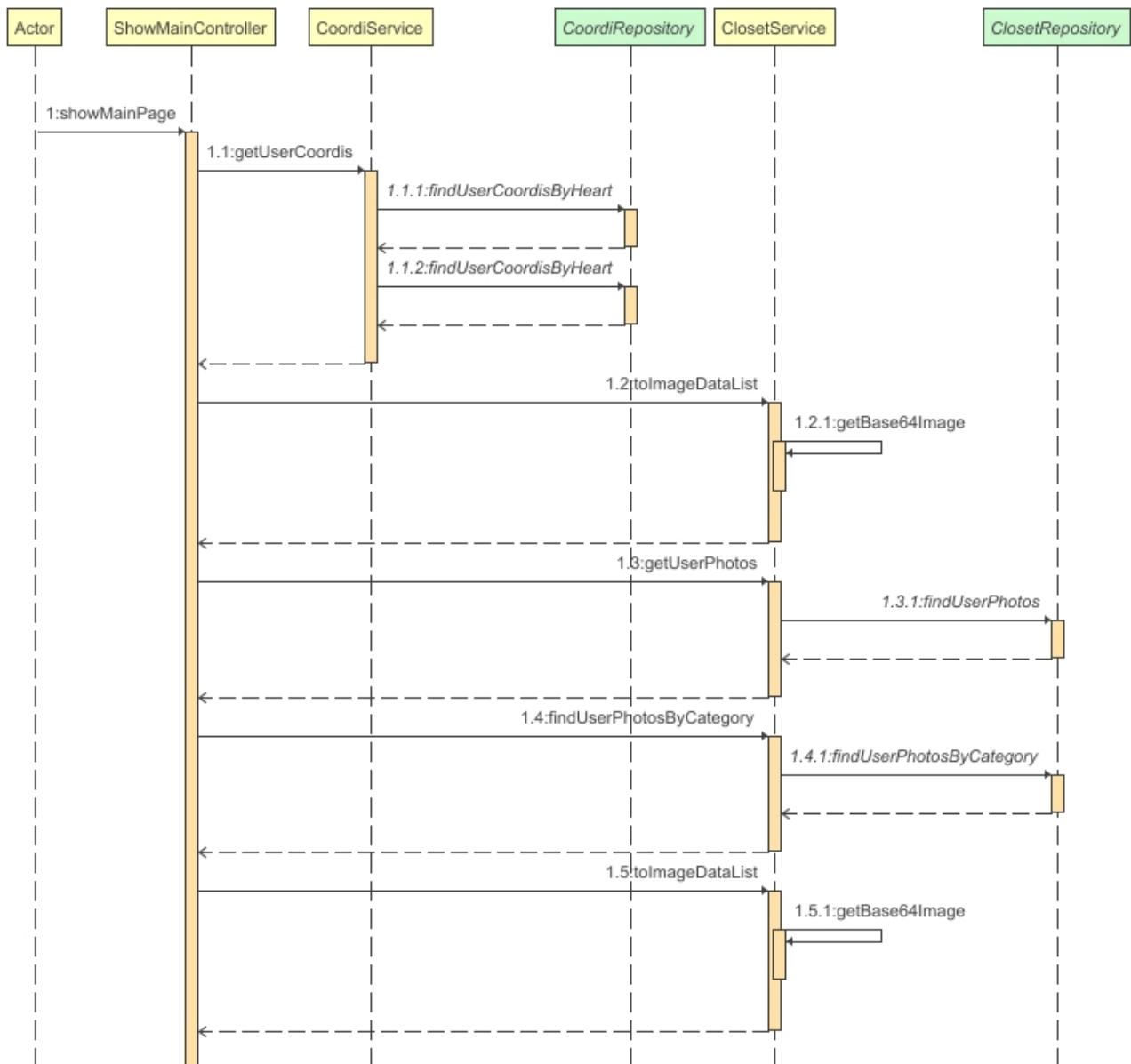


[그림 4 - 9] print suggest list sequence diagram

사용자가 시스템의 건의 내용 데이터를 조회하기 위해 데이터를 출력하는 과정을 나타내는 sequence diagram으로 ‘SuggestController’, ‘SuggestService’, ‘SuggestDTO’ 컴포넌트간의 이루어지는 상호작용에 대한 diagram이다. 사용자가 ‘SuggestController’에 ‘boardFindAll’ 요청을 하면 ‘SuggestController’는 ‘SuggestService’에 ‘findAll’ 요청을 하게 된다. ‘SuggestService’는 ‘SuggestDTO’ 객체로 변환하는 메소드를 호출하게 되면 ‘create’ 작업이 되면서 ‘SuggestDTO’ 객체가 생성되게 된다. 그렇게 생성된 ‘SuggestDTO’ 객체가 ‘SuggestService’, ‘SuggestController’를 통해 반환됨으로써 건의목록이 출력된다.

4.4 Mainpage sequence diagram

4.4.1 mainpage 출력



[그림 4 - 10] print mainpage sequence diagram

이 시퀀스 다이어그램은 사용자의 메인 페이지를 보여주는 과정을 설명한다.

다이어그램에는 'Actor', 'ShowMainController', 'CoordiService', 'CoordiRepository', 'ClosetService', 그리고 'ClosetRepository'가 표시되어 있다. 다이어그램에서 보이는

상호작용은 다음과 같다.

'Actor'는 메인 페이지를 요청한다 (show MainPage).

'ShowMainController'는 다음 두 가지 서비스를 호출한다.

1. 'CoordiService'에 'getUserCoordis'를 요청하여 사용자의 좌표 데이터를 가져온다.

2. 'ClosetService'에 'getUserPhotos'를 요청하여 사용자의 사진을 가져온다.

'CoordiService'는 'CoordiRepository'에 'findUserCoordisByHeart'를 요청하여 "좋아요"를 기준으로 좌표 데이터를 검색한다.

'ClosetService'는 'ClosetRepository'에 'findUserPhotos'와 'findUserPhotosByCategory'를 요청하여 사용자의 사진 및 카테고리별 사진을 검색한다.

'CoordiRepository'와 'ClosetRepository'는 각각 요청받은 데이터를 찾아 서비스 레이어로 비동기적으로 반환한다.

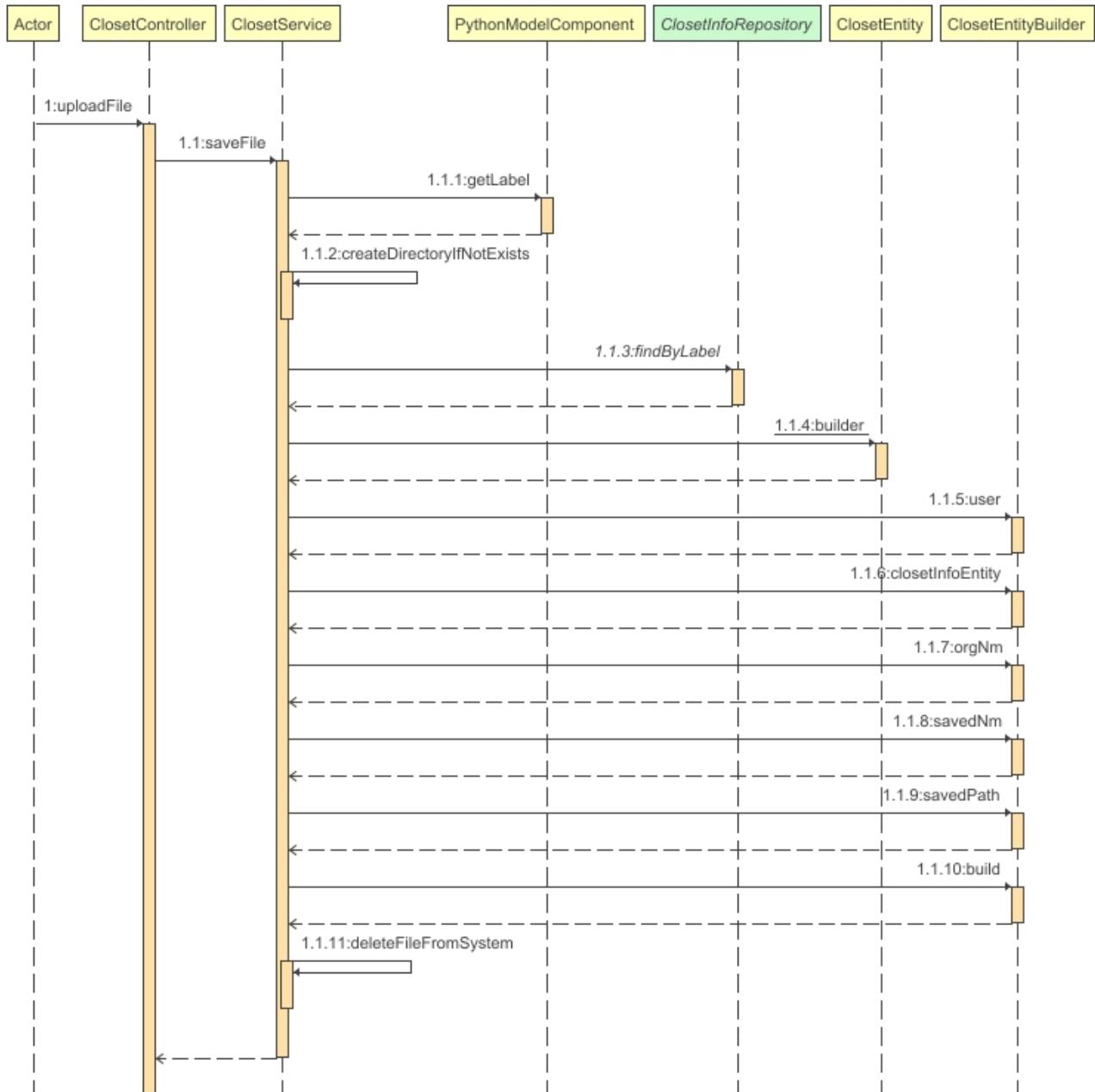
'CoordiService'는 반환받은 좌표 데이터를 이미지 데이터 리스트로 변환하는 과정(toImageDataList)을 수행한다.

'ClosetService'는 반환받은 사진 데이터를 Base64 이미지 포맷으로 변환하는 과정(getBase64Image)을 수행한다.

이 변환된 이미지 데이터는 'ShowMainController'로 반환되어 최종적으로 'Actor'에게 전달되어 메인 페이지에 표시된다.

4.5 Closet sequence diagram

4.5.1 옷 사진 등록하기



[그림 4 - 11] upload cloth picture sequence diagram

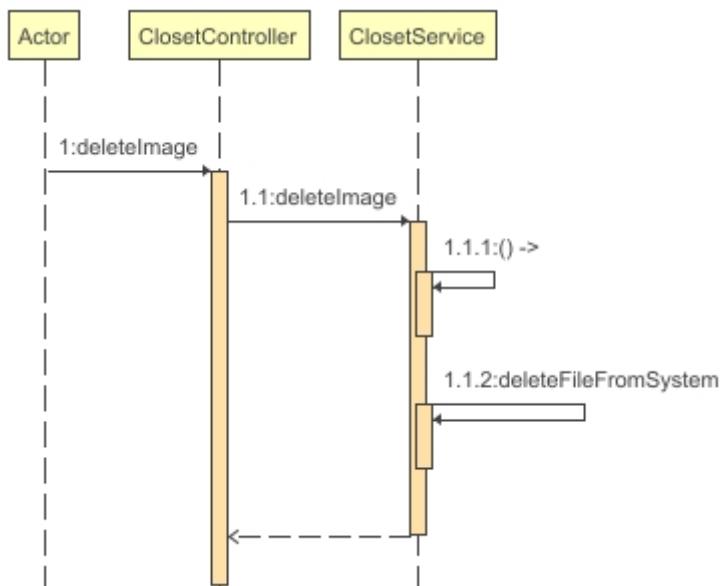
이 시퀀스 다이어그램은 사용자가 옷 사진을 등록하는 과정을 설명한다.

다이어그램에는 'Actor', 'ClosetController', 'ClosetService', 'PythonModelComponent', 'ClosetInfoRepository', 'CloseEntity', 그리고 'ClosetEntityBuilder'가 표시되어 있다.

다이어그램에서 보이는 상호작용은 다음과 같다.

'Actor'가 사진 추가 버튼을 누르면 uploadFile 요청이 입력된다. 그러면 'ClosetService'의 saveFile 서비스를 실행하여 파일 등록 로직을 실행한다. 이 때 사용자가 등록한 옷을 AI 모델이 분류하여 옷에 해당하는 라벨을 얻는다. 라벨을 얻으면 하드디스크에 사진을 저장하는데, 이 때 해당하는 디렉토리 경로가 없다면 생성한다. 그리고 옷 사진 경로를 DB의 Closet 테이블에 저장한다.

4.5.2 옷 사진 삭제



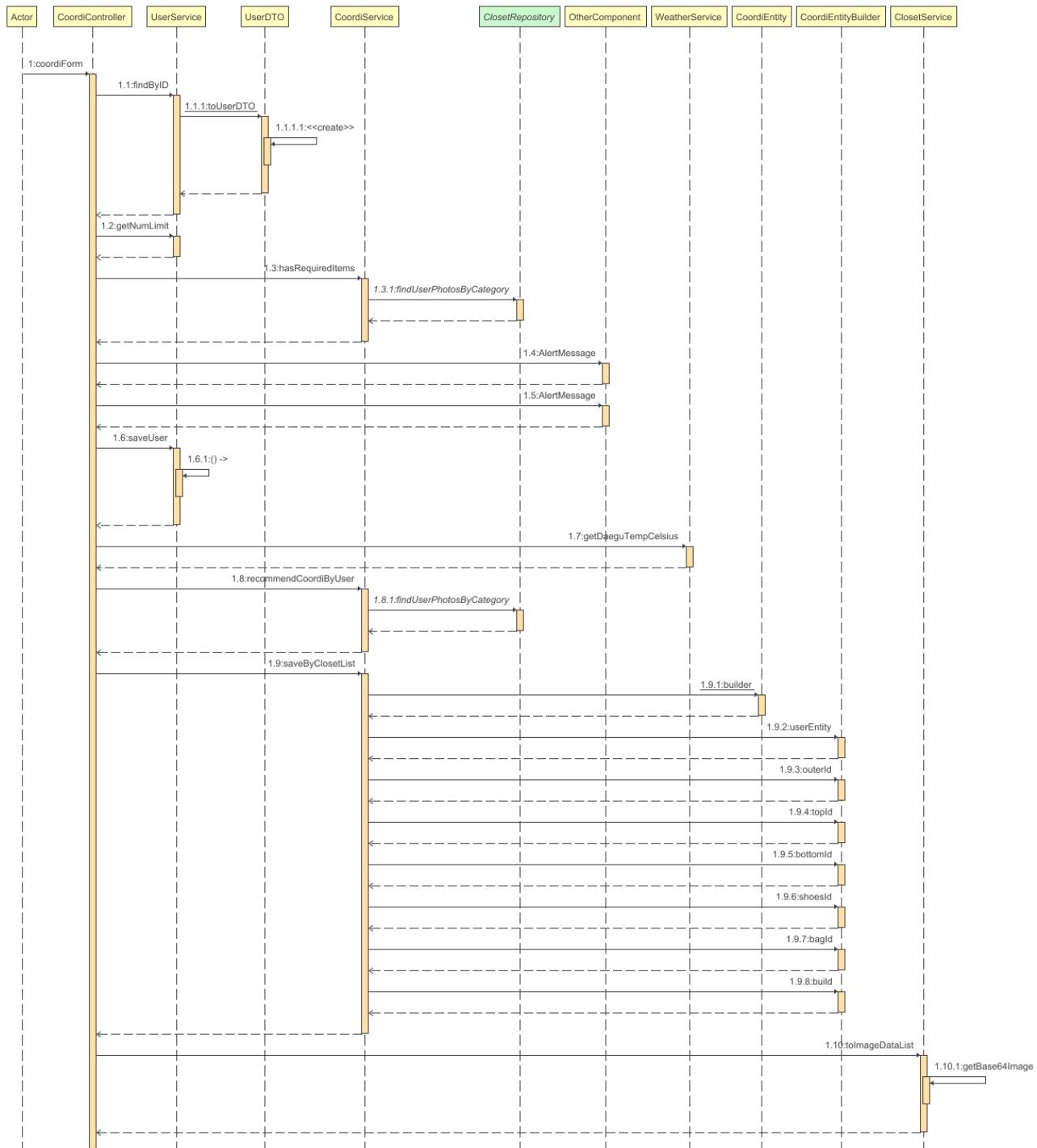
[그림 4 - 12] delete cloth picture sequence diagram

이 diagram은 사용자가 옷 사진을 삭제하는 과정을 나타내는 sequence diagram으로 ‘ClosetController’, ‘ClosetService’ 컴포넌트간의 이루어지는 상호작용에 대한 diagram이다. 사용자가 ‘ClosetController’에 ‘deleteImage’ 요청을 하면 ‘ClosetController’는 ‘CloserService’에 ‘deleteImage’ 요청을 하게 된다. ‘deleteImage’ 요청을 받은 ‘ClosetService’는 삭제할 옷사진의 데이터를 찾아내어 ‘deleteFileFromSystem’, 시스템에서 옷 사진의 데이터를 삭제를 하는 과정을 수행한다. 이 과정을 모두 수행한 후의 데이터는 ‘ClosetController’에 보내져 사용자가 선택한 옷 사진이 삭제되는 과정이 마무리된다.



4.6 Coordi sequence diagram

4.6.1 옷 추천 화면 출력

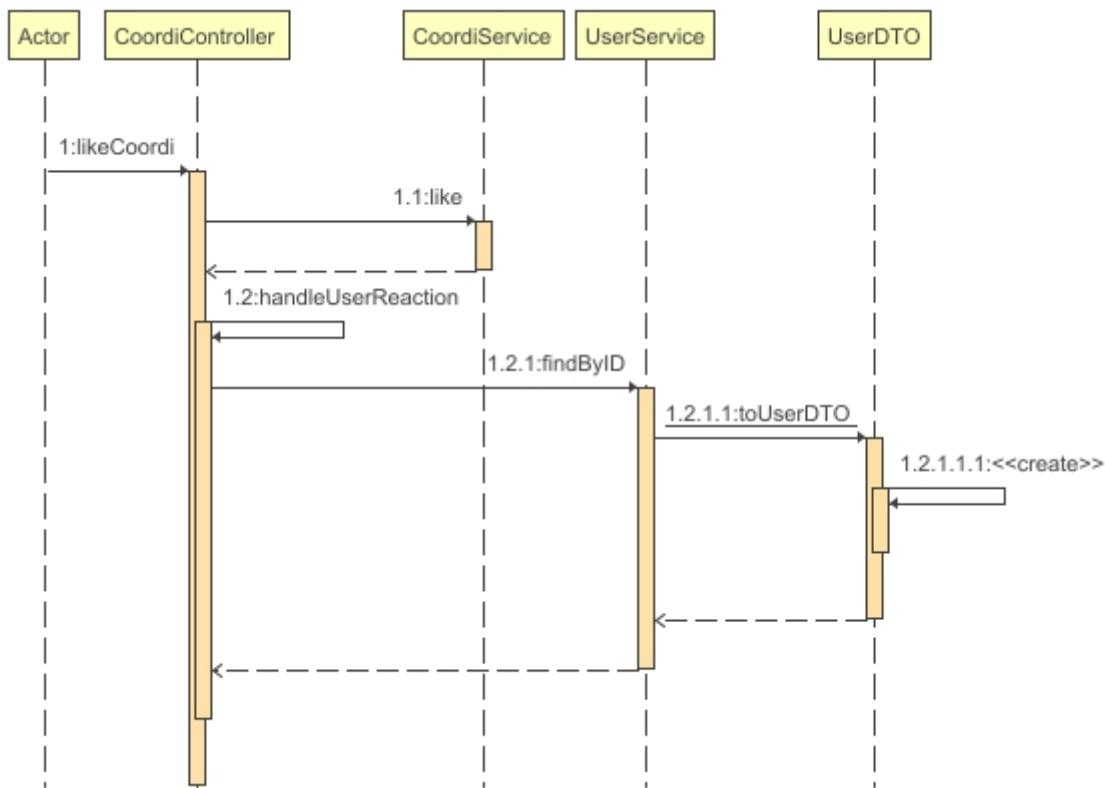


[그림 4 - 13] print recommand cloth sequence diagram

이 diagram은 사용자가 코디 추천 버튼을 눌렀을 때 옷 추천 과정을 나타내는 sequence diagram으로 ‘Actor’, ‘CoordiController’, ‘UserService’, ‘UserDTO’, ‘CoordiService’, ‘ClosetRepository’, ‘OtherComponent’, ‘WeatherService’, ‘CoordiEntity’, ‘CoordiEntityBuilder’, ‘ClosetService’ 객체간에 이루어지는 상호작용에 대한 diagram이다.

사용자가 코디 추천 버튼을 누르면 coordiForm 요청이 이뤄진다. 이 때 사용자의 아이디를 이용하여 findById를 통해 UserDTO에 사용자의 정보를 담는다. 그리고 사용자마다 하루 옷 추천 한도가 있기 때문에 getNumLimit 를 이용해 구한다. 그리고 옷 추천을 받기 위해선 최소 옷 등록 제한사항으로 상의, 하의, 신발이 하나 이상 있어야 하기 때문에 hasRequiredItems으로 이를 검사한다. 이 두가지 제한사항을 만족하지 않는다면 AlertMessage를 통해 경고차ing 표시된다. 현재 날씨에 적정한 옷을 선택하기 위해 getDaeguTempCelsius를 통해 대구시의 온도를 가져온다. 이러한 정보와 사용자에게 등록된 옷을 조합하여 recommendCoordiByUser를 통해 사용자에게 맞는 옷을 추천받는다. 그리고 이런 추천받은 옷을 saveByBloselList를 통해 DB에 저장한다. 그리고 저장한 옷 리스트를 tolImageDataList를 이용해 이미지 리스트로 반환한다.

4.6.2 옷 코디 좋아요



[그림 4 - 14] like cloth coordi sequence diagram

이 시퀀스 다이어그램은 사용자가 특정 항목('Coordi'라는 객체)에 "좋아요(dislike)" 반응을 보낼 때 시스템 내에서 일어나는 상호작용을 설명한다. 다이어그램은 'Actor', 'CoordiController', 'CoordiService', 'UserService', 그리고 'UserDTO'로 구성되어 있다. 다이어그램에서 보이는 상호작용은 다음과 같다.

'Actor'는 'CoordiController'에게 'likeCoordi'라는 작업을 요청한다. 이는 사용자가 특정 항목을 좋아한다는 것을 나타낸다.

'CoordiController'는 이 요청을 처리하기 위해 'CoordiService'에 'like' 작업을 요청한다.

'CoordiService'는 사용자의 반응을 처리하는 'handleUserReaction' 메소드를 실행한다.

이 과정에서 'UserService'에게 특정 ID로 사용자를 찾는 'findById' 작업을 요청한다.

'UserService'는 해당 ID를 가진 사용자를 찾아 'UserDTO' 객체로 변환합니다. 이 변환 과정은

'toUserDTO' 메소드로 표현된다.

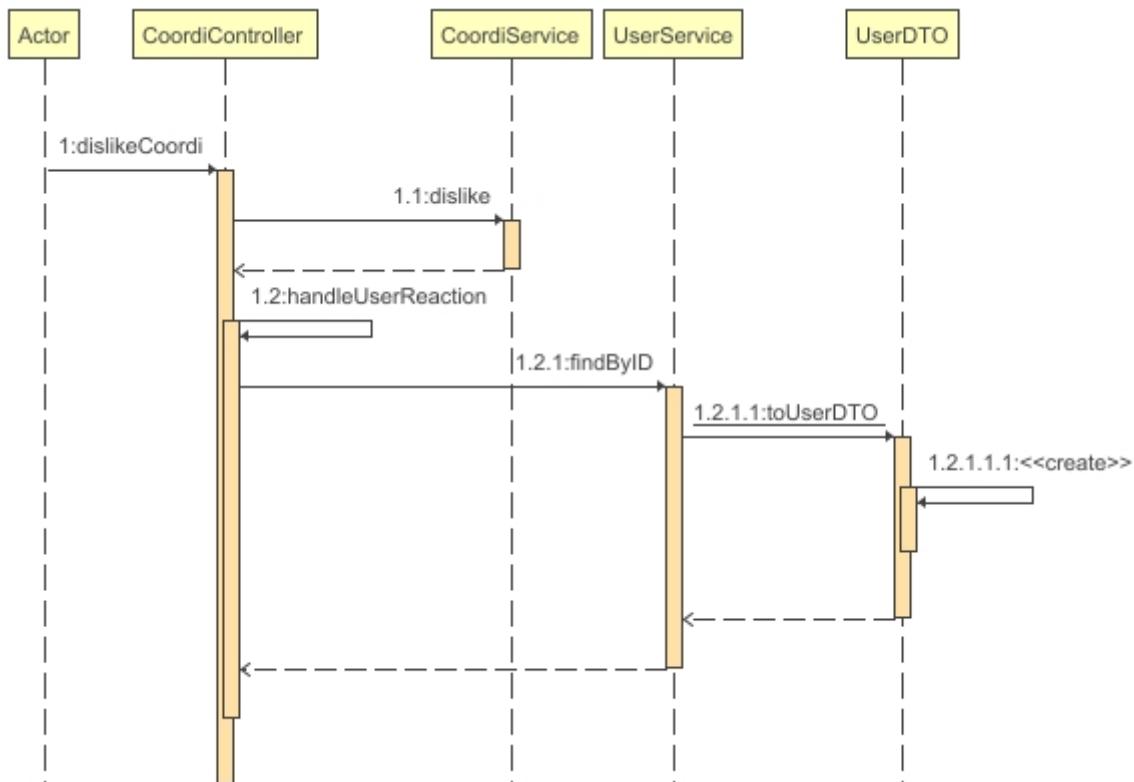
변환된 'UserDTO' 객체는 'UserService'를 통해 'CoordiService'로 반환되며, 이 과정은 객체

생성('create')을 포함할 수 있다.

'CoordiService'는 최종적으로 'CoordiController'에게 처리 결과를 반환하고,

'CoordiController'는 이를 'Actor'에게 응답으로 전달한다.

4.6.3 옷 코디 싫어요



[그림 4 - 15] dislike cloth coordi sequence diagram

이 시퀀스 다이어그램은 사용자가 특정 항목('Coordi'라는 객체)에 "싫어요(dislike)" 반응을 보낼 때 시스템 내에서 일어나는 상호작용을 설명한다. 다이어그램은 'Actor', 'CoordiController', 'CoordiService', 'UserService', 그리고 'UserDTO'로 구성되어 있다.

다이어그램에서 보이는 상호작용은 다음과 같다.

'Actor'는 'CoordiController'에게 특정 항목에 대한 "싫어요" 반응을 전달한다 (dislikeCoordi).

'CoordiController'는 이 반응을 처리하기 위해 'CoordiService'에 'dislike' 작업을 요청한다.

'CoordiService'는 사용자의 반응을 처리하는 'handleUserReaction' 메소드를 실행한다.

이 과정에서 'CoordiService'는 사용자의 정보를 처리하기 위해 'UserService'에 'findById'를 요청한다.

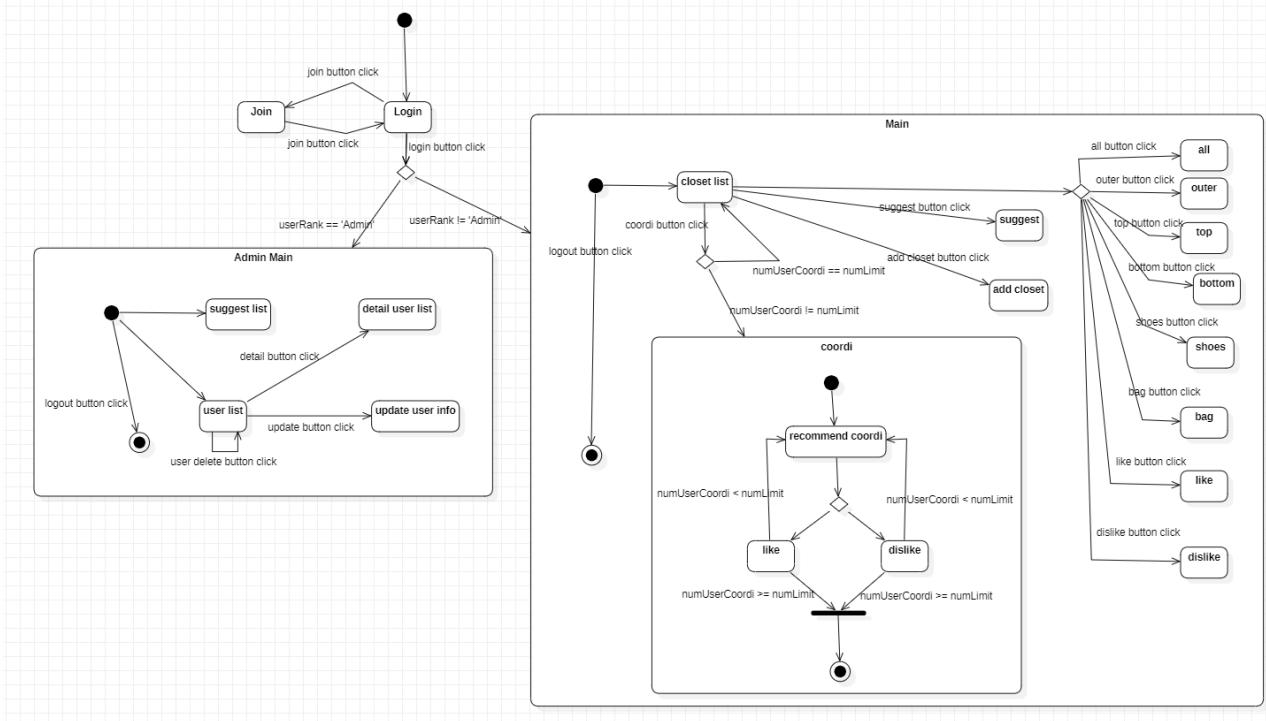
'UserService'는 해당 ID로 사용자를 찾아내고 'UserDTO' 객체로 변환하는 작업을 수행한다 (toUserDTO).

생성된 'UserDTO' 객체는 'UserService'를 통해 'CoordiService'로 반환된다.

'CoordiService'는 최종적으로 처리 결과를 'CoordiController'로 반환하며, 이는 'Actor'에게 응답으로 전달될 수 있다.

5. State machine diagram

이 장은 시스템의 state machine diagram(이하 SMD)을 그리고 설명한다. [그림 5 - 1]은 클라이언트 측의 시스템(웹)에 관한 diagram이다.



[그림 5 - 1] State machine diagram

State machine diagram는 두 개의 주요 섹션으로 구성된 상태 머신 다이어그램을 보여준다. "Admin Main"과 "Main". 상태 머신 다이어그램은 소프트웨어나 시스템의 상태 변화를 시각적으로 표현한 것으로, 다양한 상태와 그 상태들 간의 전이를 나타낸다. 사용자의 행동이나 시스템 이벤트에 따라 상태가 변경된다.

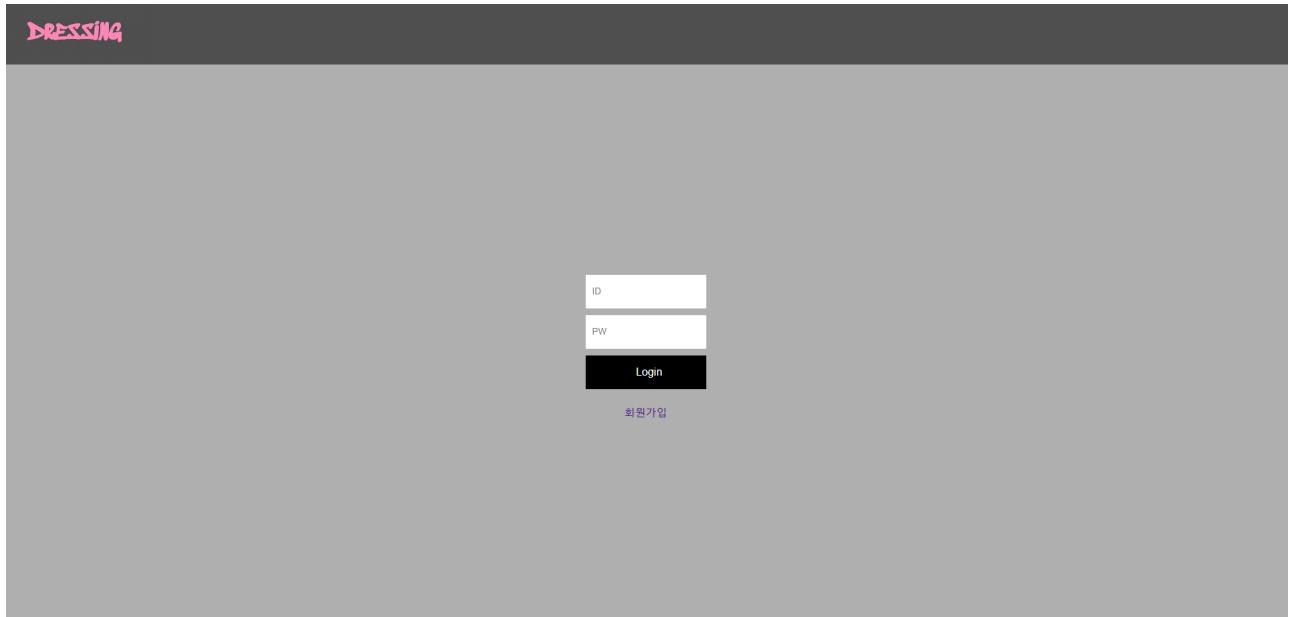
"Admin Main" 섹션은 관리자의 주요 기능들을 보여준다. "Join" 버튼을 클릭하면 "Login" 상태로 이동한다. "Login" 상태에서 사용자 등급이 "Admin"이면 "Admin Main" 상태로 진입한다. "Admin Main" 상태에서는 "suggest list", "user list", "detail user list", "update user info" 등의 기능이 있다. 관리자는 사용자 목록을 볼 수 있고, 상세 정보를 볼 수 있는

버튼을 클릭하거나 사용자 정보를 업데이트할 수 있다. "logout button"을 클릭하면 관리자 세션이 종료된다.

"Main" 섹션은 일반 사용자를 위한 섹션으로 보인다. "closet list"에서 시작하여, 사용자는 여러 옷장 항목을 추천받을 수 있다. "coordi" 버튼을 클릭하여 옷 조합을 볼 수 있고, 추천받을 수도 있다. 사용자는 옷 조합을 "like" 또는 "dislike" 할 수 있으며, 이는 "numUserCoordi"의 값에 영향을 줄 수 있다. 이는 사용자가 좋아요 또는 싫어요를 할 수 있는 한계를 나타낼 수 있다. 옷장에 아이템을 추가하거나, 상의, 하의, 신발, 가방 등의 카테고리를 볼 수 있는 버튼도 있다. 전반적으로 이 다이어그램은 어떤 웹 애플리케이션 또는 시스템의 사용자 인터페이스 흐름을 나타낸다. 사용자와 관리자의 상호작용을 통해 시스템이 어떻게 반응하는지를 설명하며, 버튼 클릭과 같은 사용자의 행동에 따라 어떤 프로세스가 실행되는지를 보여준다.

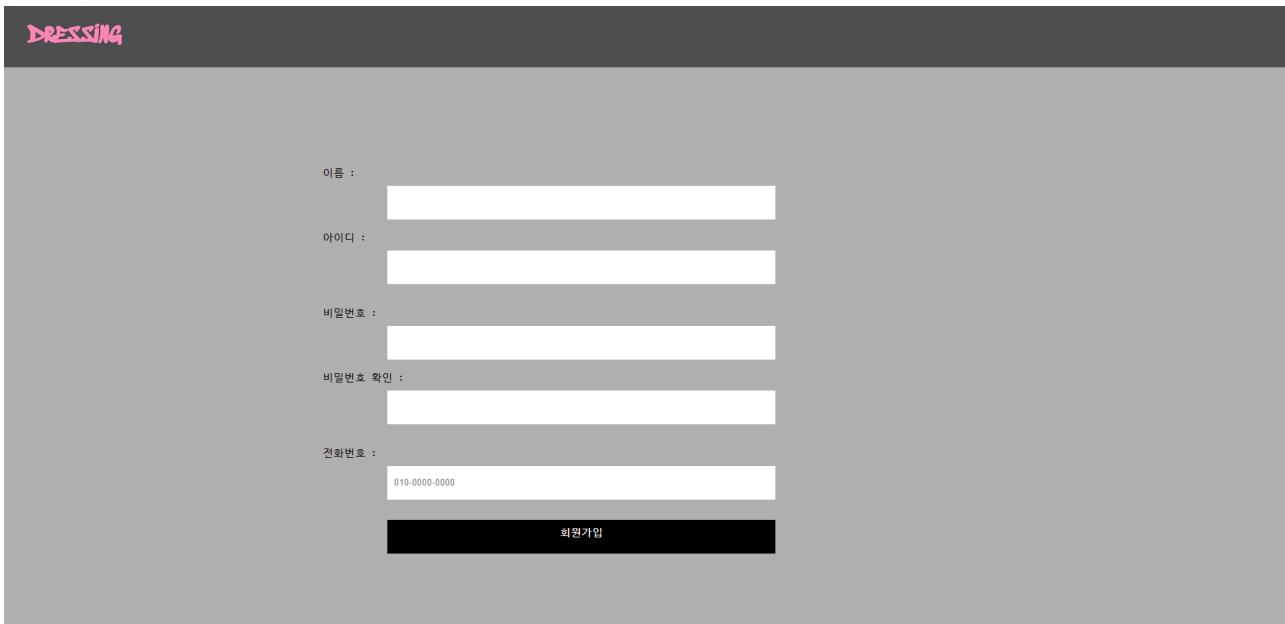
6. User interface prototype

이 장은 예상 웹사이트 UI와 UI안의 각 구성요소를 설명한다. 실제 개발된 어플의 UI와 디자인이고, 내용은 아래와 같다.



[그림 6 - 1] 로그인 화면

위 사진은 로그인 화면이다. 아이디와 비밀번호를 입력하여 로그인을 할 수 있으며 회원가입이 되어있지 않다면 회원가입을 할수있는 창으로 넘어가는 회원가입 버튼이 존재한다.



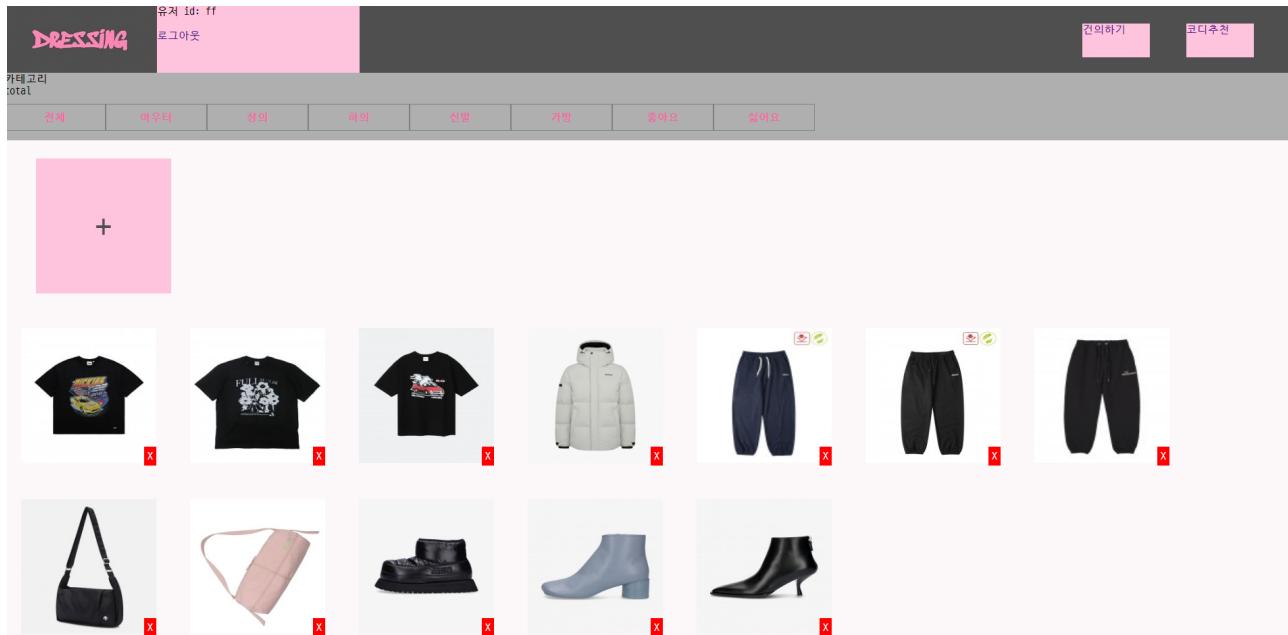
The screenshot shows a sign-in form with the following fields:

- 이름 : [Redacted input field]
- 아이디 : [Redacted input field]
- 비밀번호 : [Redacted input field]
- 비밀번호 확인 : [Redacted input field]
- 전화번호 : [Redacted input field]
010-0000-0000

A blacked-out button labeled "회원가입" (Member Registration) is at the bottom.

[그림 6 - 2] 회원가입 화면

위 사진은 회원가입 화면으로 이름, ID, PW, 휴대폰번호를 넣어 회원가입을 할 수 있다. 전화번호는 ‘010-0000-0000’이라는 양식이 있기 때문에, 텍스트 박스에 양식을 입력하여 회원이 정보 입력을 하기 편하도록 하였다.



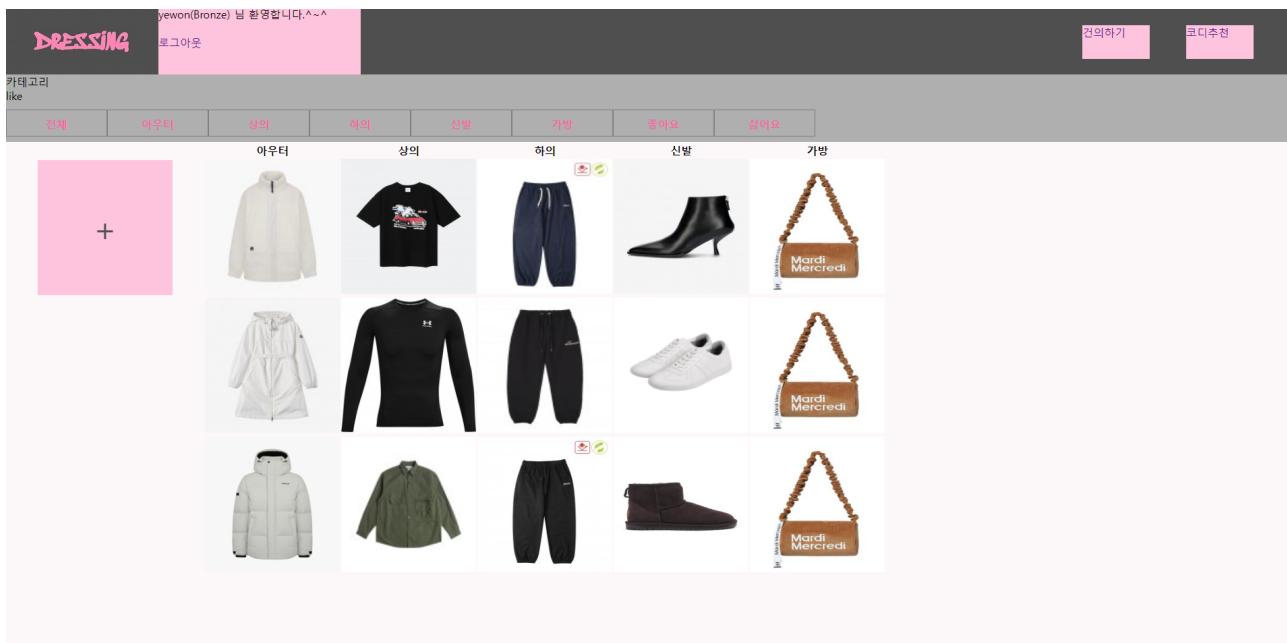
[그림 6 - 3] 메인화면

위 사진은 로그인페이지에서 유저가 로그인시 볼수있는 메인화면으로, 프로그램 로고 옆에는 로그인한 유저 id를 확인할 수 있고, 로그아웃 버튼이 있다. 우측 상단에 있는 코디 추천 버튼 클릭시 AI를 기반으로 한 코디를 추천받을 수 있다. 카테고리 내에 있는 버튼을 클릭시 버튼명에 해당하는 자신이 추가한 옷들을 카테고리별로 볼 수 있다. + 버튼을 클릭시 옷을 추가할 수 있고, 추가한 옷은 화면에 나타난다. 추가한 옷을 삭제하고 싶을 때는 옷에 있는 x버튼을 눌러 삭제할 수 있다.



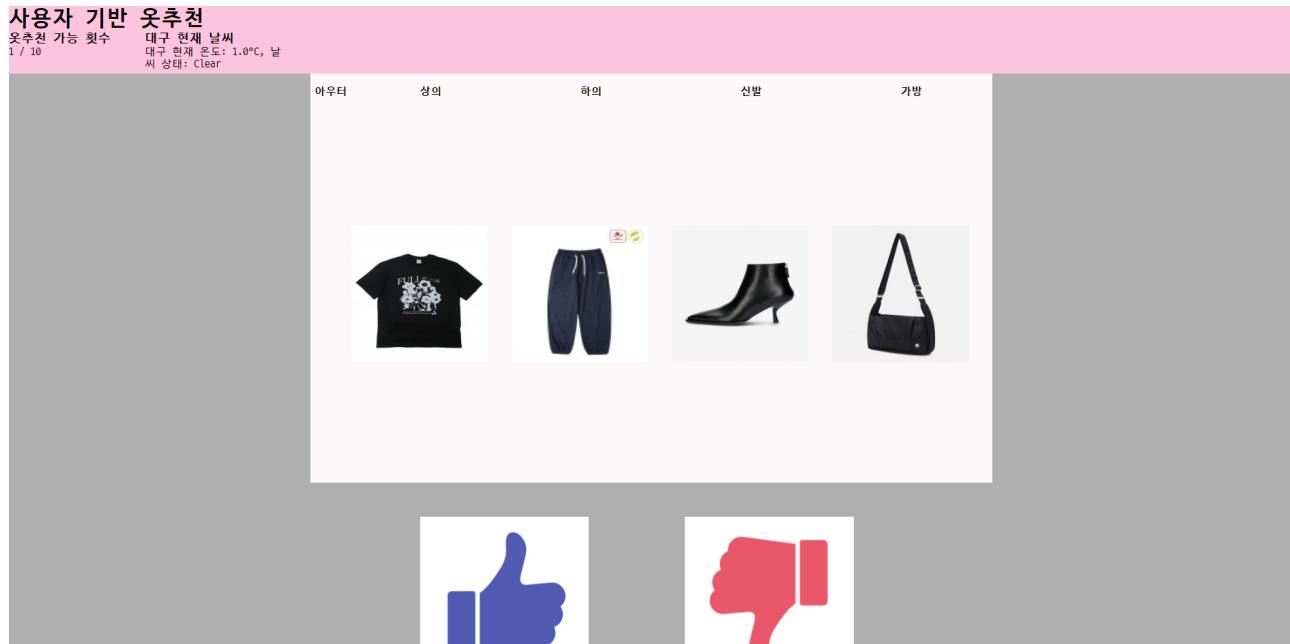
[그림 6 - 4] 상의 카테고리 화면

메인화면에서 카테고리 칸의 상의 버튼을 눌렀을 때 나오는 화면이다. 사용자가 등록한 옷 사진 중에 카테고리가 상의인 것만 보여준다. 상의 말고도 아우터, 하의, 신발, 가방 등이 있다.



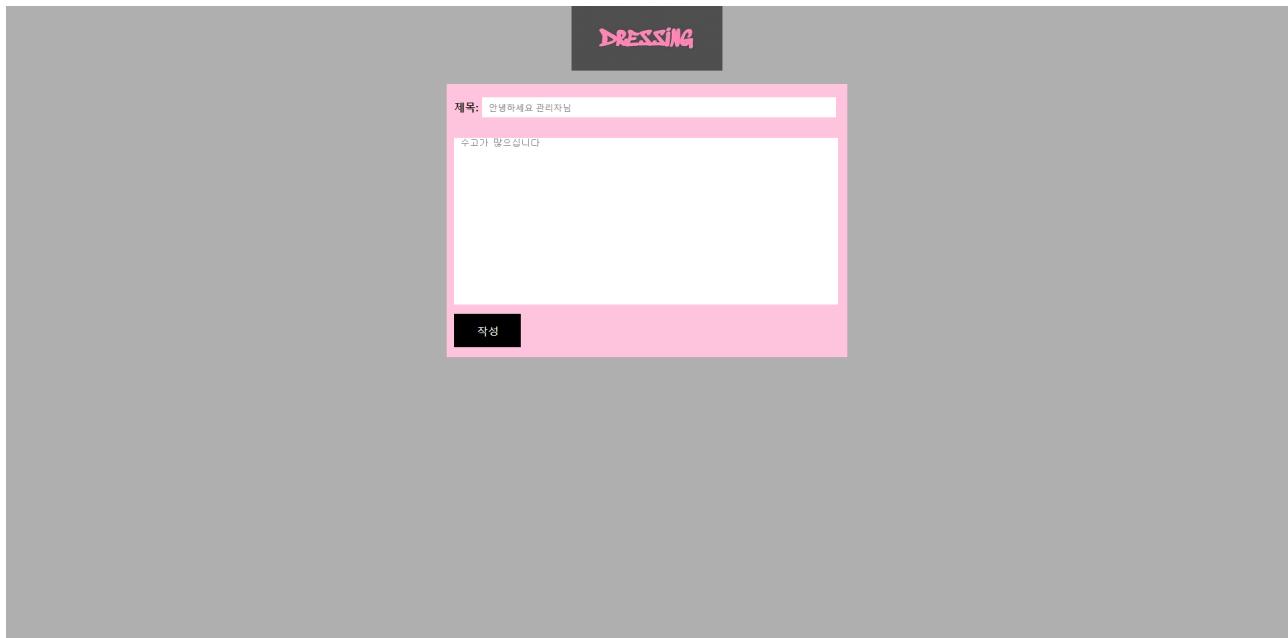
[그림 6 - 5] 좋아요 카테고리 화면

이 화면은 사용자가 메인화면에서 좋아요 카테고리를 눌렀을 때 나오는 화면이다. 만약 사용자가 본인이 등록한 옷으로 코디 추천을 받아 좋아요 또는 싫어요로 평가를 했다면 이 화면에 해당 코디가 뜬다. 아우터, 상의, 하의, 신발, 가방 순으로 사진이 뜨며, 만약 코디 추천에 포함되지 않은 옷 종류가 있다면 빈칸으로 보여진다. 좋아요 말고도 싫어요가 있다.



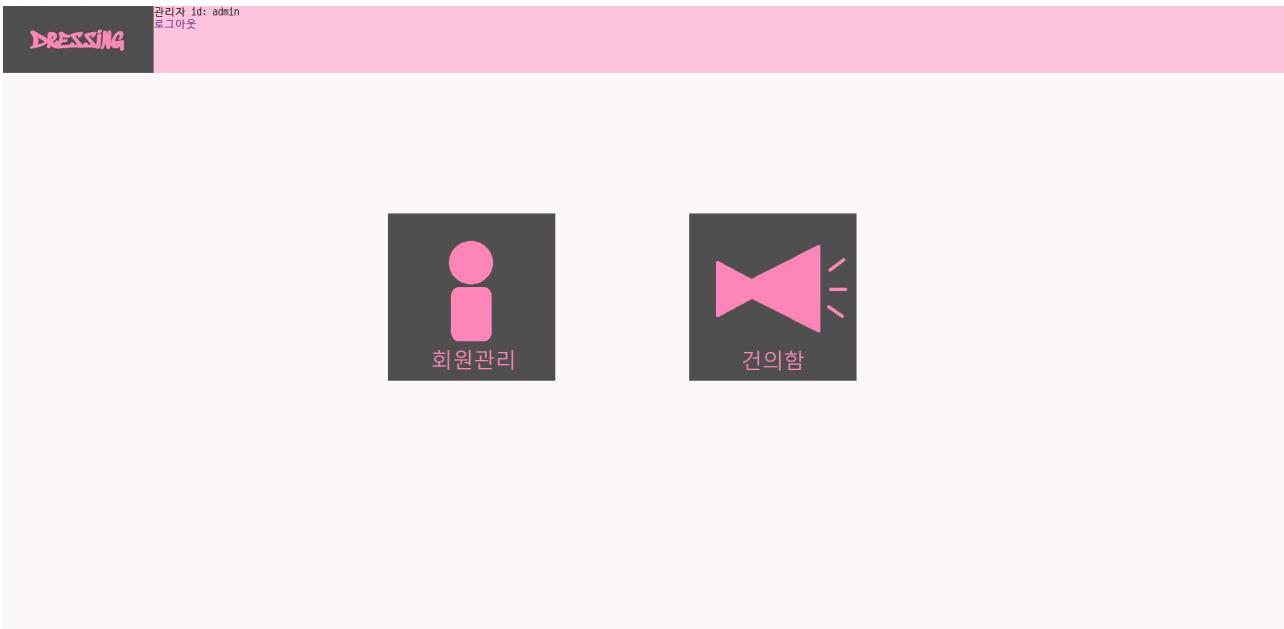
[그림 6 - 6] 코디 추천 화면

위 사진은 AI기반 혹은 날씨 기반 코디추천 버튼을 눌렀을때 볼수있는 화면으로 좋아요와 싫어요를 통하여 사용자의 점수를 추가하여 코디추천에 반영할 수 있다. 하얀 박스 안에 추천된 코디 사진이 뜬다. 좋아요나 싫어요 버튼을 누를 때마다 결과가 반영되기 때문에 웹의 뒤로가기 버튼을 눌러 나가도 코디와 그 코디에 대한 평가가 저장된다.



[그림 6 - 7] 건의하기 화면

위 사진은 사용자가 메인화면에서 건의하기 버튼을 눌렀을 때 나오는 화면이다. 사용자는 건의할 사항을 제목과 내용을 포함하여 작성할 수 있다. 사용자는 텍스트 입력 상자를 통해 제목과 내용을 작성하고 제출하는 버튼을 통해 전송된다. 제목 또는 내용에 빈칸이 있으면 제출하기 버튼을 눌렀을 때 다시 작성하라는 오류 메시지가 뜬다. 사용자가 웹사이트에 대한 평가를 작성할 수 있다. 이는 Dressing 웹사이트를 유지보수 하는 데에 도움을 준다.



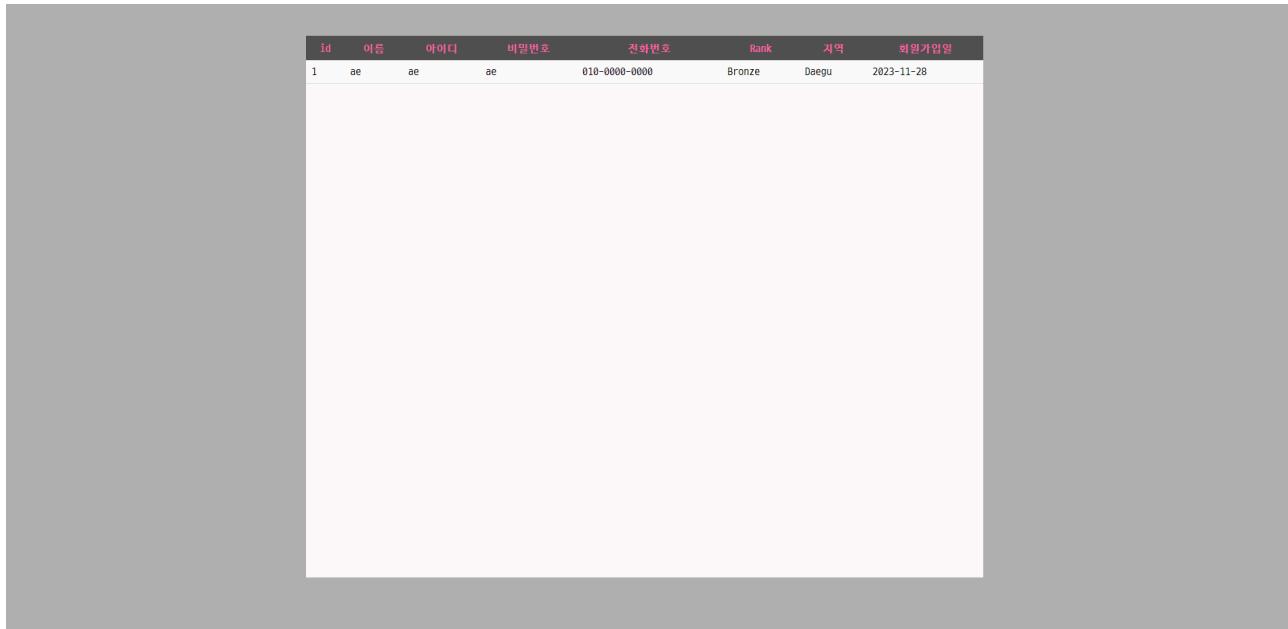
[그림 6 - 8] 관리자 화면

위 사진은 관리자 화면으로 로그인페이지에서 관리자 아이디로 접속시 볼 수 있는 화면이다. 프로그램 로고 옆 로그인한 관리자의 id를 확인할 수 있고, 로그아웃 버튼도 있다. 건의함을 클릭하면 유저들이 적어둔 건의사항 리스트들을 확인할 수 있고, 회원관리를 클릭시 전체 유저 리스트를 확인할 수 있는 페이지로 넘어간다.

회원 리스트						
id	이름	아이디	상세조회	삭제	수정	
1	ae	ae	[조회]	삭제	수정	
2	admin	admin	[조회]	삭제	수정	
3	aa	aa	[조회]	삭제	수정	
4	ff	ff	[조회]	삭제	수정	

[그림 6 - 9] 회원관리 화면

위 사진은 관리자 화면에서 회원관리 버튼을 눌렀을 때 이동되는 페이지이다. 가입된 전체 회원 목록을 볼 수 있다. id, 이름, 아이디를 확인할 수 있고, 회원의 정보를 상세 조회할 수 있는 버튼과, 회원 삭제 버튼, 회원 수정 버튼이 있다.



[그림 6 - 10] 회원 상세조회 화면

위 사진은 회원 리스트 화면에서 특정 유저의 상세 조회 버튼을 눌렀을 때 볼 수 있는 화면이다. 회원 리스트에서 확인할 수 있는 정보와 추가로 비밀번호, 전화번호, 등급, 거주 지역, 회원가입일을 확인할 수 있다.



[그림 6 - 11] 회원 건의사항 화면

위 사진은 관리자 화면에서 건의함 버튼을 눌렀을 때 나오는 화면이다. 사용자들이 건의한 내용을 사용자의 id, 작성자와 함께 확인할 수 있다.



[그림 6 - 12] 회원 정보 수정 화면

위 사진은 관리자 화면에서 수정 버튼을 눌렀을 때 나오는 화면이다. text 상자에 현재 사용자들의 정보가 적혀있고, 이를 수정할 수 있다. 아이디와 지역 정보는 수정이 불가능하다.

7. Implementation requirements

H/W platform requirements

Web

- CPU: 13th Gen Intel(R) Core(TM) i5-13500
- RAM: 16GB
- Storage: SAMSUNG MZVL21T0HCLR-00B00
- Network

DB Server

-

- CPU: 13th Gen Intel(R) Core(TM) i5-13500
- RAM: 16GB
- Storage: SAMSUNG MZVL21T0HCLR-00B00

S/W

WebServer

- OS: Window
- Implementation
- DBMS: MySQL

Nonfunctional requirements

AI

- Visual code

- tensorflow

-anaconda

-package

- tensorflow
- sklearn
- numpy
- pandas
- PLT

8. Glossary

이름	설명
인터페이스	서로다른 두 시스템, 장치, 소프트웨어 따위를 서로 이어주는 부분.
AI	Artificial intelligence의 준말
DB	공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합
인스턴스	클래스의 현재 생성된 객체

9. References

- [1] Alistair Cockburn, Writing Effective Use Cases (n.p.: Addison-Wesley Longman, 1999), 46-52, 125-127, 170, 174-177.