

운동, 여행, 취미 매칭 플랫폼

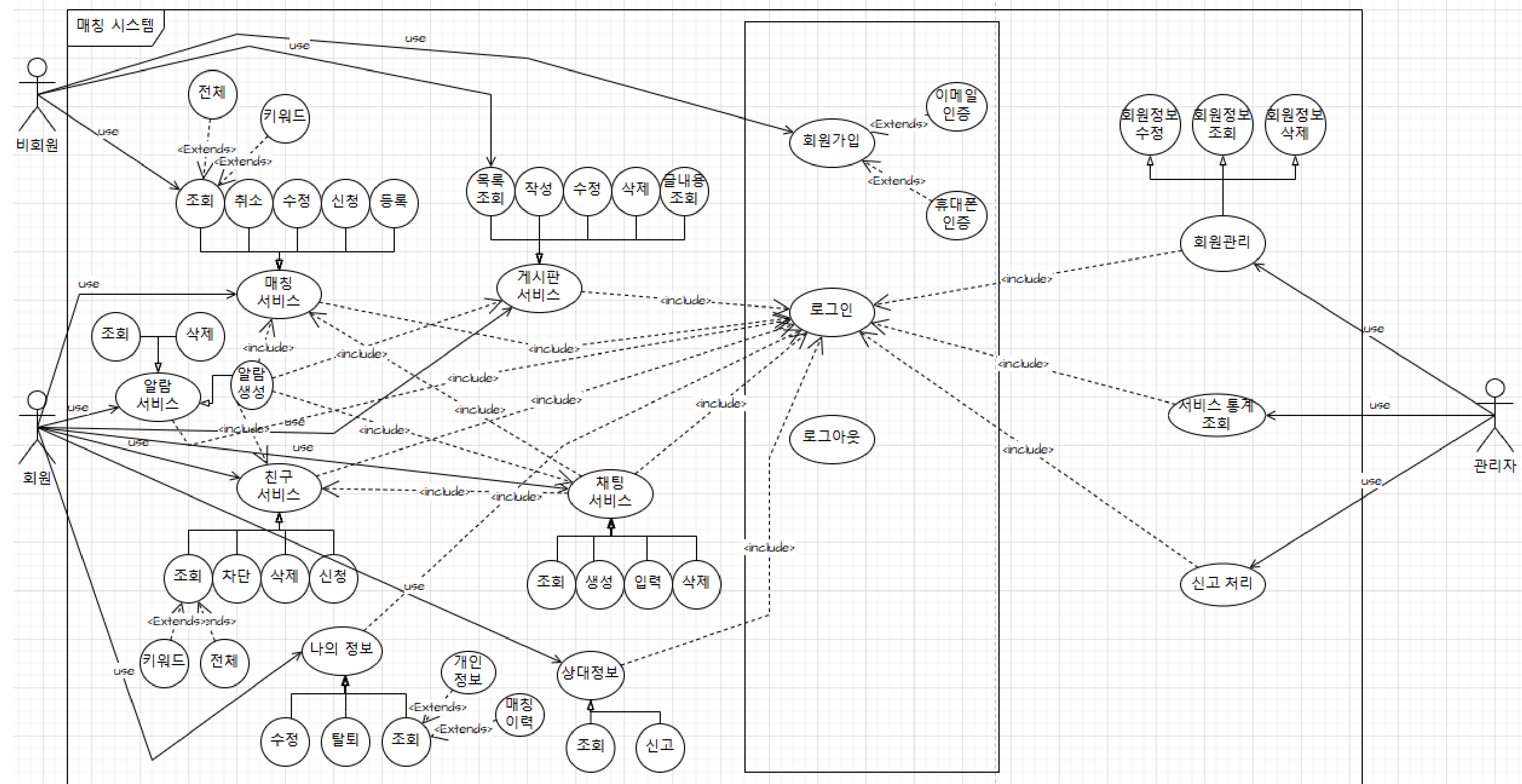
PlayMatch

통합 구현

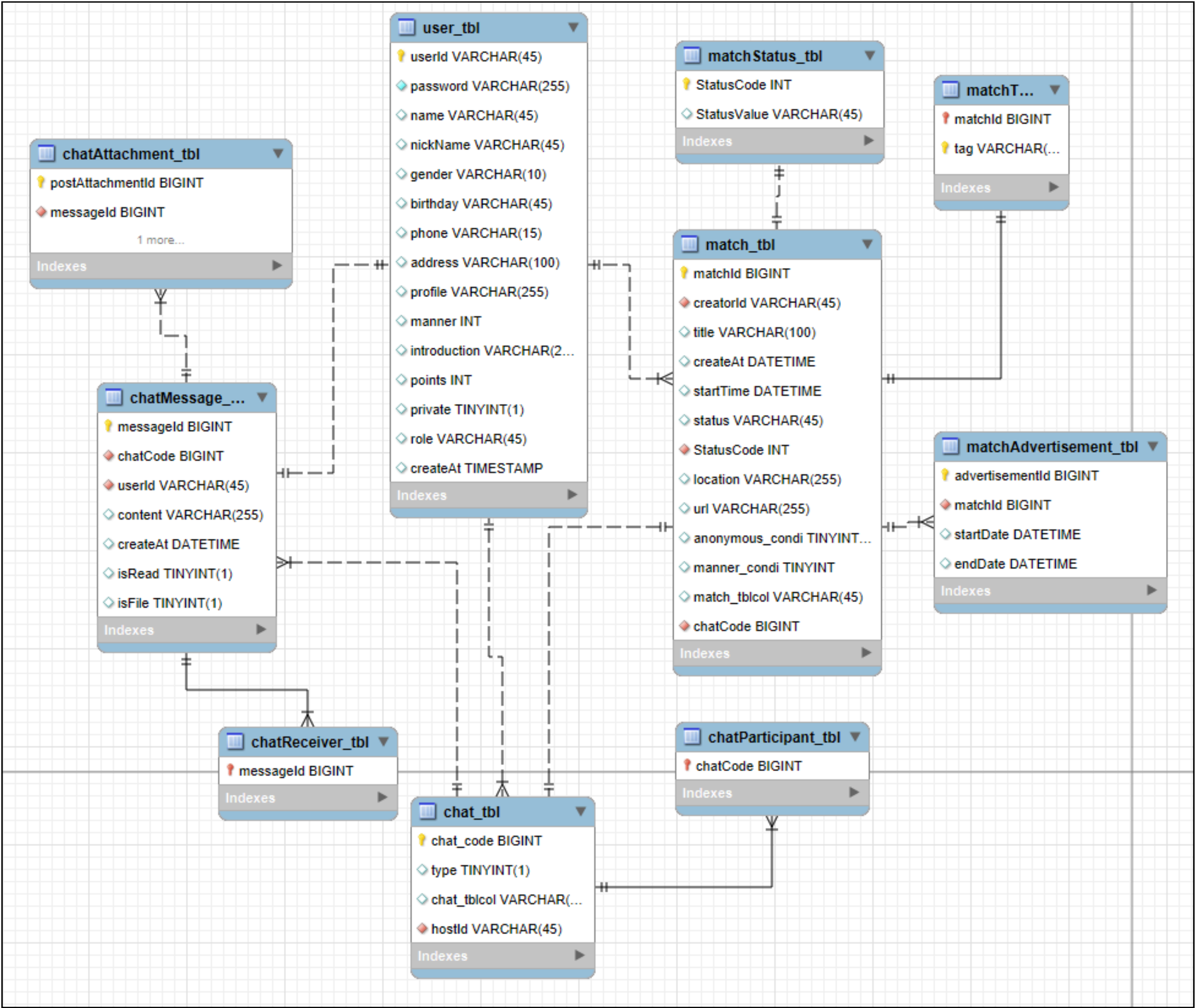
2팀

서은빈, 문상희, 신현우, 최서연

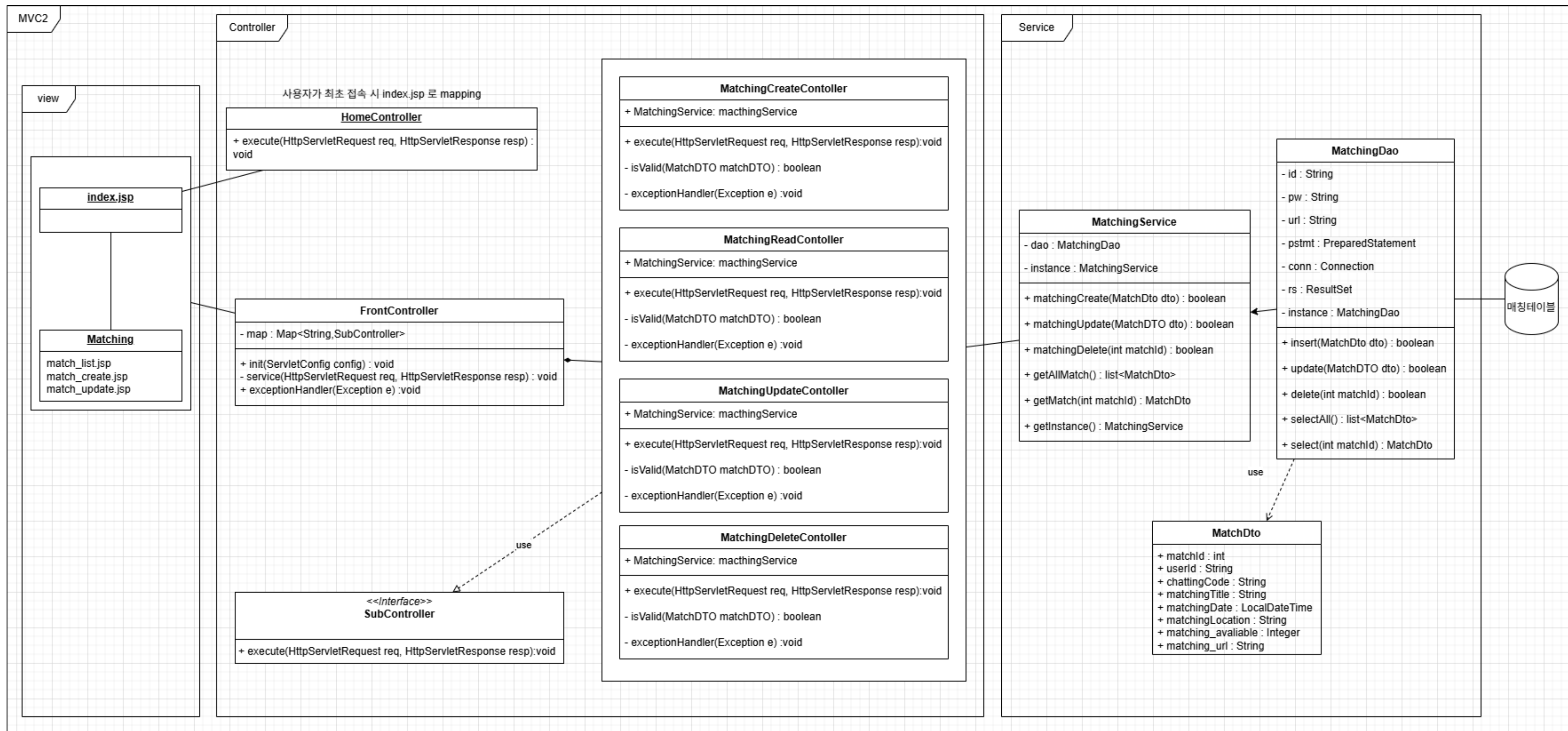
1. 요구사항 유스케이스



2. ERD



3. MVC기반 클래스 다이어그램 - 매칭



4. API 명세서

/api/auth : 로그인에 관한 프로세스

- post/login : 로그인
 - request : method - post

파라미터	타입	필수여부	설명
userId	String	필수	유저 아이디
password	String	필수	유저 비밀번호

- response

필드	타입	필수여부	설명
header			
jwtToken	Map<String, Object>	필수	로그인 시 필요한 토큰
cookie	cookie	필수	로그인시 필요한 쿠키
body			
userDto	UserDto	필수	유저 정보

- 응답 - 성공시

```
{
  "jwtToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWUiOiJ1c2VyMiIsImF1dGgiOiJST0xFX1VTRVIlLCJ1c2VySWQiOiJ1c2VyMiIsImhhbmRlc0NzYz0Tk0NSwiZXhwIjoxNzQ3NjQxNzQ1fQ.pWlpiHCpj51a0q2uE5gEXsmy6036xo2I73Dj8Jvqx3c",
  "userDTO": {
    "id": 0,
    "userId": "user2",
    "password": null,
    "name": "문상희",
    "nickName": "user2",
    "gender": "male",
    "birthday": "1994-06-24",
    "phone": "010-1234-5678",
    "address": "지역을 설정해주세요",
    "profile": "http://localhost:8100/profile/user2",
    "manner": 100,
    "introduction": "기본 소개",
    "points": 0,
    "createAt": "2025-05-19",
    "platform": "0",
    "role": null,
    "private": false
  }
}
```

- 응답 - 실패시

아이디 혹은 비밀번호가 일치하지 않습니다.

4. API 명세서

/api/auth : 인증관련 API

- post/sign : 회원가입
- request : method - post

파라미터	타입	필수 여부	설명
authCheck	boolean	필수	본인인증 유무
idCheck	boolean	필수	동일 계정 유무
userId	String	필수	사용자 id (고유값)
nickName	String	필수	사용자 닉네임
password	String	필수	사용자 비밀번호
repassword	String	필수	비밀번호 확인
imp_uid	String	필수	본인인증 uid

- response

파라미터	타입	필수 여부	설명
status_code	int	필수	가입 성공 실패 여부 (HTTP 상태 코드)
fail	boolean	필수	오류에 대한 설명 메시지

- 응답 예시 - 회원가입 요청

```
{
  "authCheck" : true
  "idCheck" : true
  "userID" : "ididid"
  "nickName" : "닉네임"
  "password" : "1234"
  "repassword" : "1234"
  "imp_uid" : "imp_967084524038"
}
```

- 응답 예시 - 성공/실패시

```
{
  "status_code" : 200
}
```

```
{
  "status_code" : 400
  "fail" : "같은 명의로 등록된 계정이 있습니다."
}
```

4. API 명세서

api/chat : 채팅에 관한 프로세스

- post/creatMatchChat: 그룹 채팅방 생성
 - request : method - post

파라미터	타입	필수 여부	설명
chatCode	int	필수	채팅코드
hostID	String	필수	호스트 아이디
<header>Authorization	String	필수	JWT_ACCESS 토큰
<header>Content-Type	String	필수	컨텐츠 타입

- response

필드	타입	필수 여부	설명
is_success	Boolean	필수	성공 여부
message	String	필수	내용

- 응답 예시

Response(성공):

```
{
  "is_success" : true
  "message": "채팅방이 생성되었습니다"
}
```

Response(실패):

```
{
  "is_success" : false
  "message": "채팅방 생성에 실패하였습니다"
}
```

4. API 명세서

/api/user : 유저관련 API

- post/infoUpdate : 유저 정보 업데이트
 - request : method - post

파라미터	타입	필수 여부	설명
userId	String	필수	수정 계정 아이디
value	String	필수	수정 값
type	String	필수	수정 컬럼
<header>Authorization	String	필수	JWT_ACCESS 토큰
<header>Content-Type	String	필수	컨텐츠 타입

- response

파라미터	타입	필수 여부	설명
status_code	int	필수	수정 성공 실패 여부 (HTTP 상태 코드)
userDTO	Map<Strimg,String>	필수	수정된 userDTO 정보

- 응답예시 - 수정 요청

```
{
  "userId": "admin",
  "value": "수정결과",
  "type": "introduction"
}
```

- 응답예시 - 성공/ 실패

```
{
  "status_code" : 200
  "userDTO": {
    "id": 0,
    "userId": "admin",
    "password": "",
    "name": "admin",
    "nickName": "6236236",
    "gender": "male",
    "birthday": "1986-01-01",
    "phone": "01012345678",
    "address": "대구",
    "profile": "http://localhost:8100/profile/admin",
    "manner": 300,
    "introduction": "수정결과",
    "points": 100,
    "createAt": "2025-05-10",
    "platform": null,
    "role": "",
    "private": true
  }
}
```

```
{
  "status_code" : 400
  "fail" : "수정값이 공백입니다."
}
```


5. 인터페이스 정의서

날짜	버전	작성자	승인자	내용
2025-05-19	1	2조		인터페이스 정의서 1.0 작성

시스템명	Play_Match	서브시스템 명	AuthService, ChatService, UserService		
단계	설계	작성일자	2025년 05월 19일	버전	1

인터페이스 목록

송신				전달			수신				관련요구사항 ID	비고
인터페이스 번호	일련번호	송신시스템명	프로그램ID	처리형태	인터페이스 방식	발생빈도	상대 담당자 확인	프로그램ID	수신 시스템명	수신번호		
AUTH_001	1	Play_Match	USR_SIGN	ON-LINE	URL 호출	수시	담당자 정보	USR_SIGN_DB	Authentication	1	REQ_USR_SIGN	회원가입
CHAT_001	2	Play_Match	CHAT_PROC	ON-LINE	URL 호출	수시	담당자 정보	CHAT_PROC_DB	Chating	2	REQ_CHAT_PROC	채팅방 개설
LOGIN_001	3	Play_Match	USR_LOGIN	ON-LINE	URL 호출	수시	담당자 정보	USR_LOGIN_DB	Authentication	3	REQ_USR_LOGIN	로그인
MYPAGE_001	4	Play_Match	MYPAGE_PROC	ON-LINE	URL 호출	수시	담당자 정보	MYPAGE_PROC_DB	User	4	REQ_MYPAGE_PR OC	내정보 수정

5. 인터페이스 정의서 - 인터페이스 명세

인터페이스 번호	데이터 송신 시스템					송신 프로그램 ID	데이터 수신 시스템					수신 프로그램 ID	
	시스템명	데이터 저장소명	속성명	데이터타입	길이		데이터 저장소명	속성명	데이터타입	길이	시스템명		
AUTH_001	AuthController	SignDto	authCheck	boolean	1	Play_Match	회원가입 응답 데이터	status_code	int	3	AuthService	USR_SIGN_DB	
			idCheck	boolean	1			fail	boolean	1			
			userId	string	255								
			nickName	string	255								
			password	string	255								
			imp_uid	string	255								
CHAT_001	Chat Controller	채팅방 개설 요청 데이터	chatCode	int	225	Play_Match	채팅방 개설 요청 데이터	is_success	boolean	1	ChatService	CHAT_PROC_DB	
			Content-Type	String	255			message	String	225			
			hostID	String	225								
			Authorization	String	255								
LOGIN_001	AuthController	UserDto	userId	string	255	Play_Match	UserDTO	userId	String	255	UserService	USR_LOGIN_DB	
			password	string	255			nickName	String	255			
					phone			String	255				
					address			String	255				
					profile			String	255				
					introduction			String	255				
					isPrivate			boolean	1				
					JwtTokenDTO		jwtToken	String	255				
		refreshToken	String	255									
MYPAGE_001	UserController	수정 요청 데이터	userId	string	255	Play_Match	수정 응답 데이터	status_code	int	3	UserService	MYPAGE_PROC_DB	
			type	string	255		UserDTO	userId	String	255			
			value	string	255			nickName	String	255			
			Authorization	String	255			phone	String	255			
			Content-Type	String	255			address	String	255			
					profile			String	255				
					introduction			String	255				
					isPrivate			boolean	1				

회원가입 시스템 구현

1. 회원가입

- 아이디, 닉네임, 비밀번호, 비밀번호 확인, 본인인증 성공시 회원가입 가능
- 우측 사용 테이블



















Table Name:

Charset/Collation:

utf8mb4

utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 userId	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 password	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 nickName	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 gender	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 birthday	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 phone	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 address	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 profile	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 manner	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 introduction	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 points	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 isPrivate	TINYINT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 role	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'USER'
 createAt	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP

회원 로그인

회원가입 성공

☐ 아이디 저장하기

제출

[회원가입](#) | [아이디찾기](#) | [비밀번호찾기](#)

로그인 시스템 구현

2. 로그인

회원 로그인

user1

....

☐ 아이디 저장하기

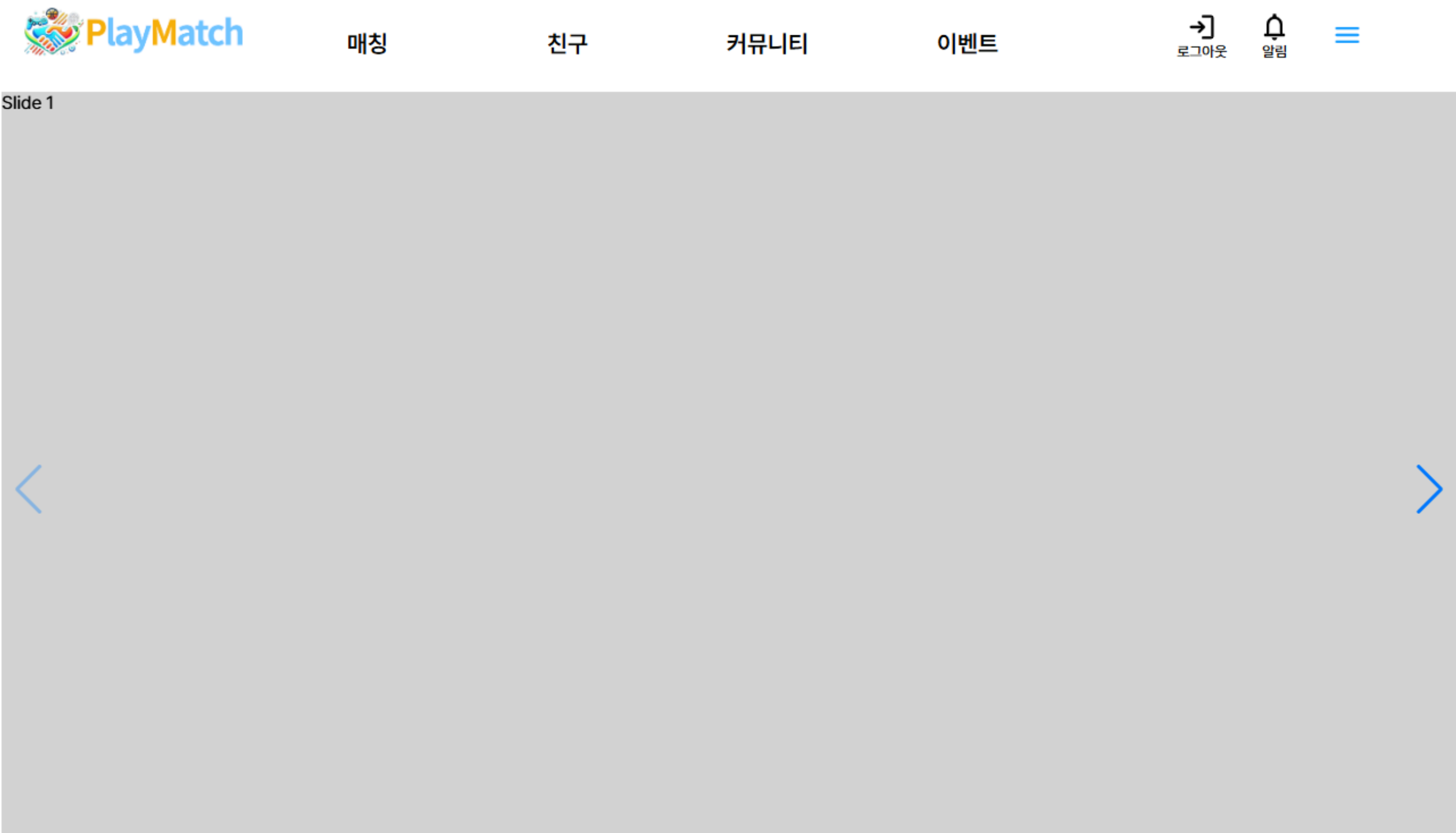
로그인

회원가입 | 아이디찾기 | 비밀번호찾기

SNS 계정으로 간편하게 로그인하세요



- 로그인 성공 시 메인 페이지로 이동



	userId	password	name	nickName	gender	birthday	phone	address	profile	manner	introduction	points	isPrivate	role	createAt
▶	user 1	\$2a\$10\$pfOyZC4lATGruBbs1TaF1OD2DV3a2mk...	NULL	user 1	NULL	NULL	NULL	지역을 설정해주세요	http://localhost:8100/profile/user1	100	기본 소개	0	0	USER	2025-05-19 16:10:21
	user 2	\$2a\$10\$1v0tZvTw.DPusiFJ2Ffu/yG6BUEyZry...	문상희	user 2	male	1994-06-24	010-1234-5678	지역을 설정해주세요	http://localhost:8100/profile/user2	100	기본 소개	0	0	USER	2025-05-19 16:31:56

로그인 시스템 구현

3. 정보 수정

- 회원 정보 수정시 반영 된 정보 출력

My매너

60단위

6236236

admin

내 정보

내 정보 관리

비밀번호 변경

내정보관리

아이디

admin

이름

admin

성별

남자

생일

1986-01-01

소개

수정 전 소개

수정하기

닉네임

6236236

수정하기

연락처

01012345678

수정하기

활동지역

대구

수정하기

프로필 공개

공개

수정하기

id	introduction	points	isPrivate	role
200	테스트ID	100	1	USER
300	수정 전 소개	100	1	ADMIN
100	기본 소개	0	0	USER
20	테스트ID	100	1	USER

My매너

60단위

6236236

admin

내 정보

내 정보 관리

비밀번호 변경

수정하기 누른 후 수정 내용 입력 시

소개

수정 전 소개

수정 후 소개

완료

내정보관리

아이디

admin

이름

admin

성별

남자

생일

1986-01-01

소개

수정 후 소개

수정하기

닉네임

6236236

수정하기

연락처

01012345678

수정하기

활동지역

대구

수정하기

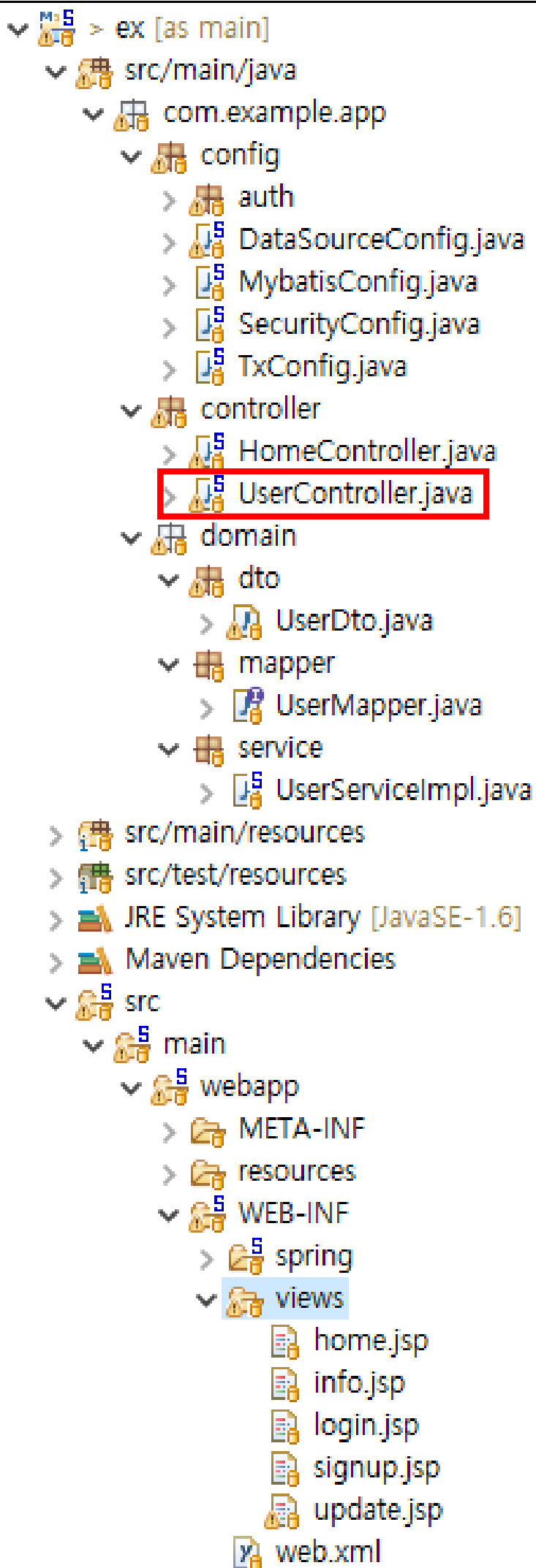
프로필 공개

공개

수정하기

manner	introduction	points	isPrivate	role
200	테스트ID	100	1	USER
300	수정 후 소개	100	1	ADMIN
100	기본 소개	0	0	USER
20	테스트ID	100	1	USER

코드



• UserController.java

```
package com.example.app.controller;

import javax.validation.Valid;

@Controller
@Slf4j
public class UserController {

    @Autowired
    UserServiceImpl userServiceImpl;

    @GetMapping("/signup")
    public void join() {
        log.info("GET /signup..");
    }

    @PostMapping("/signup")
    public String join_post(@Valid UserDto dto, BindingResult bindingResult, Model model,
        RedirectAttributes redirectAttributes) {
        log.info("POST /signup.." + dto);

        for (FieldError error : bindingResult.getFieldErrors()) {
            log.info("Error Field : " + error.getField() + " Error Msg : " + error.getDefaultMessage());
            model.addAttribute(error.getField(), error.getDefaultMessage());
            return "/signup";
        }

        boolean isJoin = userServiceImpl.userJoin(dto);

        if (isJoin) {
            redirectAttributes.addFlashAttribute("message", "회원가입 성공");
            return "redirect:/login";
        } else {
            return "/signup";
        }
    }
}
```

코드

```
▼ > ex [as main]
  ▼ src/main/java
    ▼ com.example.app
      ▼ config
        > auth
        > DataSourceConfig.java
        > MybatisConfig.java
        > SecurityConfig.java
        > TxConfig.java
      ▼ controller
        > HomeController.java
        > UserController.java
      ▼ domain
        ▼ dto
          > UserDto.java
        ▼ mapper
          > UserMapper.java
        ▼ service
          > UserServiceImpl.java
      > src/main/resources
      > src/test/resources
      > JRE System Library [JavaSE-1.6]
      > Maven Dependencies
    ▼ src
      ▼ main
        ▼ webapp
          > META-INF
          > resources
          ▼ WEB-INF
            > spring
            ▼ views
              home.jsp
              info.jsp
              login.jsp
              signup.jsp
              update.jsp
              web.xml
```

```
@GetMapping("/login")
public void login() {
    log.info("GET /login..");
}

@GetMapping("/remove")
public String remove(@RequestParam(value = "email") String email, RedirectAttributes redirectAttributes) {
    log.info("GET /remove..");

    boolean isLeave = userServiceImpl.userLeave(email);

    if (isLeave) {
        redirectAttributes.addFlashAttribute("message", "회원탈퇴 성공");
        return "redirect:/login";
    } else {
        return "/";
    }
}

@GetMapping("/update")
public void update() {
    log.info("GET /update..");
}

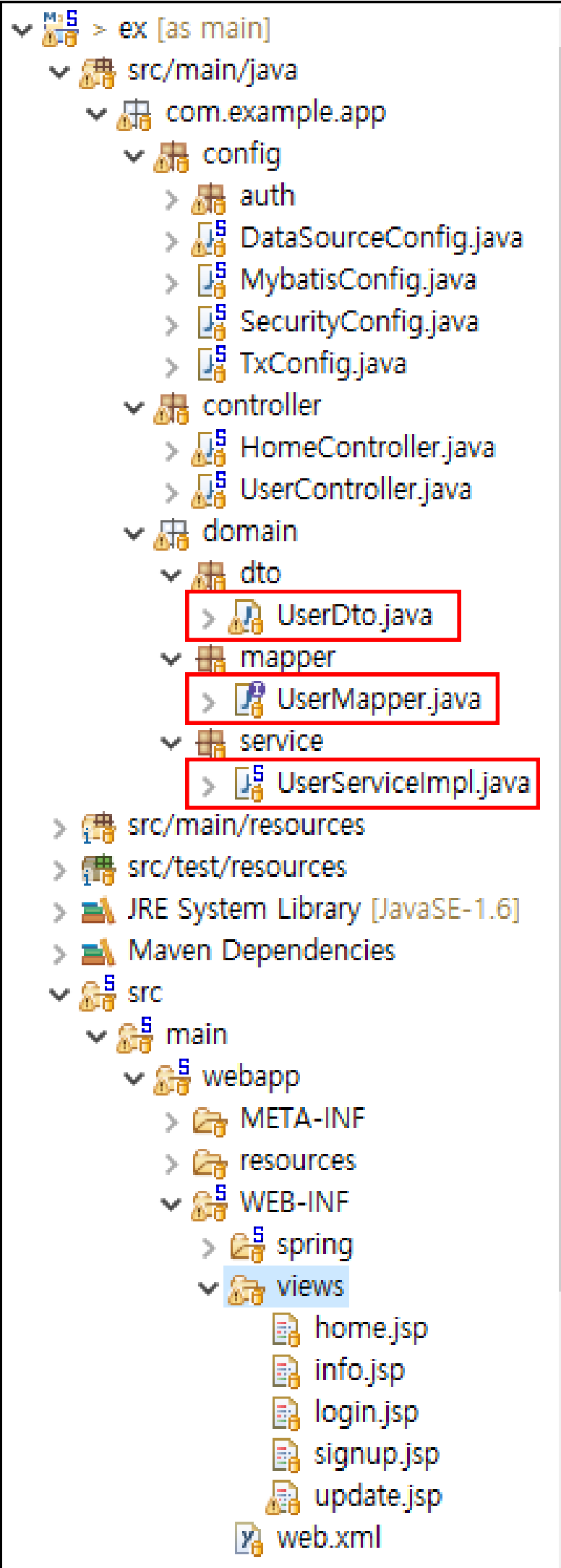
@PostMapping("/update")
public String update_post(@Valid UserDto dto, BindingResult bindingResult, Model model, RedirectAttributes redirectAttributes) {
    log.info("POST /update.." + dto);

    boolean isUpdate = userServiceImpl.userUpdate(dto);
    UserDto userDto = userServiceImpl.userSelect(dto.getEmail());

    if (isUpdate) {
        model.addAttribute("userDTO", userDto);
        return "/info";
    } else {
        return "/";
    }
}

@GetMapping("/user")
public void user(Model model) {
    Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
    log.info("Value : " + authentication.getName());
    log.info("Value : " + authentication.getAuthorities());
    model.addAttribute("username", authentication.getName());
    model.addAttribute("auth", authentication);
}
}
```

코드



• UserDto

```
1 package com.example.app.domain.dto;
2
3 import java.time.LocalDate;
4
14 @Data
15 @NoArgsConstructor
16 @AllArgsConstructor
17 @Builder
18 public class UserDto {
19     private String email;
20     private String name;
21     private String nickname;
22     @NotBlank(message="password 를 입력하세요")
23     private String password;
24     private String role;
25 }
26
27
```

• UserMapper

```
1 package com.example.app.domain.mapper;
2
3 import org.apache.ibatis.annotations.Delete;
4
10 @Mapper
11 public interface UserMapper {
12
13     @Insert(value = "insert into user_tbl values(#{email},#{name},#{nickname},#{password},#{role})")
14     public int insert(UserDto userDto);
15
16     @Select(value = "select * from user_tbl where email = #{email}")
17     public UserDto selectAt(String username);
18
19     @Delete(value = "delete from user_tbl where email = #{email}")
20     public int delete(String email);
21
22     @Update(value = "update user_tbl set name=#{name}, nickname=#{nickname} where email = #{email}")
23     public int update(UserDto userDto);
24 }
25
```

• UserServiceImpl

```
1 package com.example.app.domain.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Service
6 @Slf4j
7 public class UserServiceImpl {
8     @Autowired
9     private PasswordEncoder passwordEncoder;
10
11     @Autowired
12     private UserMapper userMapper;
13
14     @Transactional(noRollbackFor = Exception.class)
15     public boolean userJoin(UserDto userDto) {
16
17         userDto.setPassword(passwordEncoder.encode(userDto.getPassword()));
18         userDto.setRole("ROLE_USER");
19         int result = userMapper.insert(userDto);
20         return result > 0;
21     }
22
23     @Transactional(noRollbackFor = Exception.class)
24     public boolean userUpdate(UserDto userDto) {
25         System.out.println(userDto.getEmail());
26         int result = userMapper.update(userDto);
27         return result > 0;
28     }
29
30     @Transactional(noRollbackFor = Exception.class)
31     public boolean userLeave(String email) {
32
33         int result = userMapper.delete(email);
34         return result > 0;
35     }
36
37     @Transactional(noRollbackFor = Exception.class)
38     public UserDto userSelect(String email) {
39
40         UserDto userDto = userMapper.selectAt(email);
41         return userDto;
42     }
43 }
44
```