

PPT PRESENTATION

Enjoy your stylish business and campus life with BIZCAM

3조 미니 프로젝트

여행 리뷰

프로젝트 주제	여행 리뷰 ?
주제 선정이유	가장 자신있는 부분들 합쳐서 협의함.
프로젝트 기간	2021/11/22 ~ 2021/12/03 약 2주일간.
개발 환경	OS = Windows 10
	Language = JAVA 11, HTML, CSS, Ajax, Vue.js, JQuery
	Tool = IntelliJ IDEA 2021.2.3 / Community / Ultimate
	SpringBoot 2.5.6 / Gradle Project / Thymeleaf / sockjs / websocket / Lombok / devtools ..등등
	DB = Oracle Database 10g

팀장 정광수 (프론트, 백)

팀원 권기철 (프론트, 백)

팀원 이성찬 (프론트)

팀원 김지영 (프론트)

팀원 전재연 (프론트)

기획

주제 선정 및 정보 수집 팀원들의 역할 분배

디자인

구현 기술, 웹 사이트 디자인 구상

개발



권기철, 정광수 기반 DB, Controller 구현, 나머지 프론트

수정 및 보완

버그, 예외처리, 보안 강화

제출

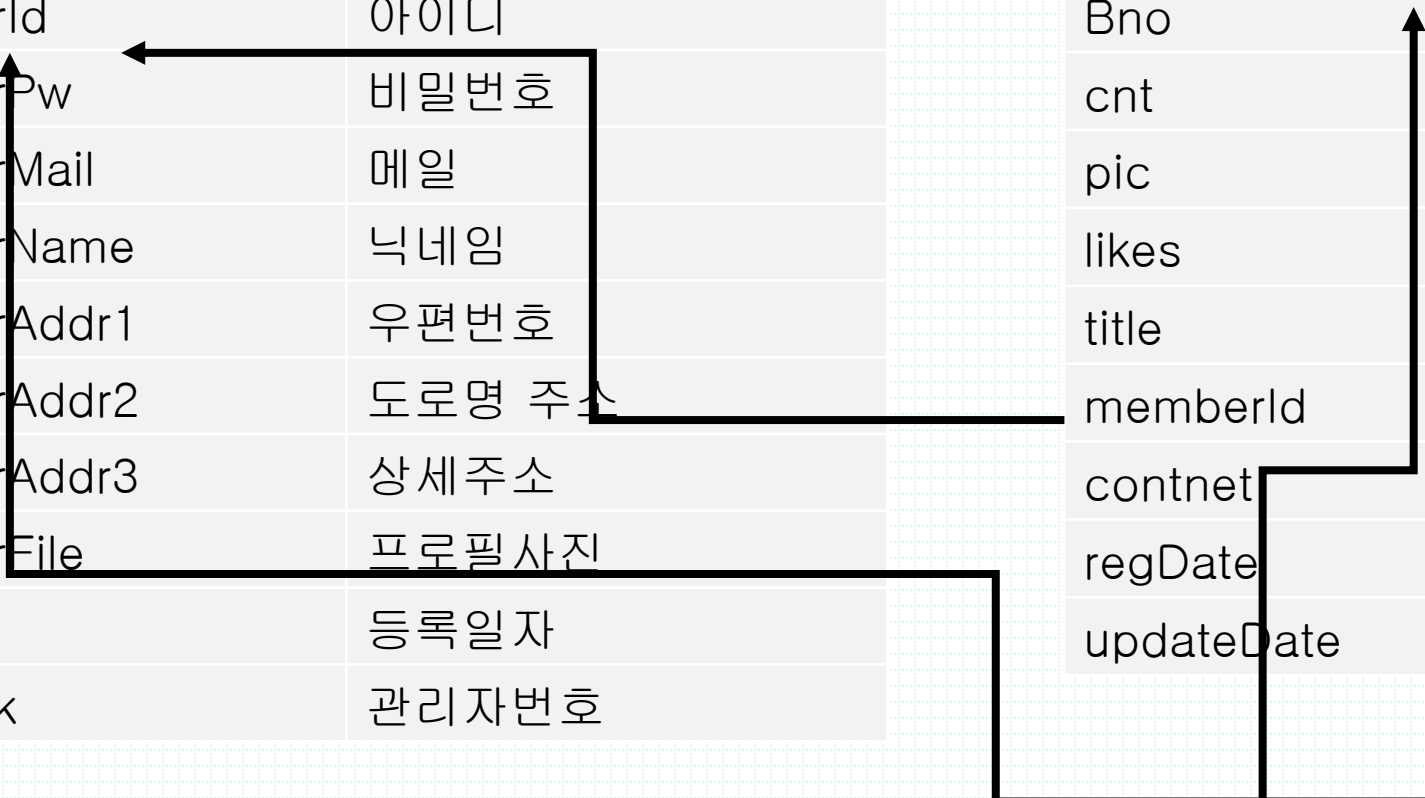
제출

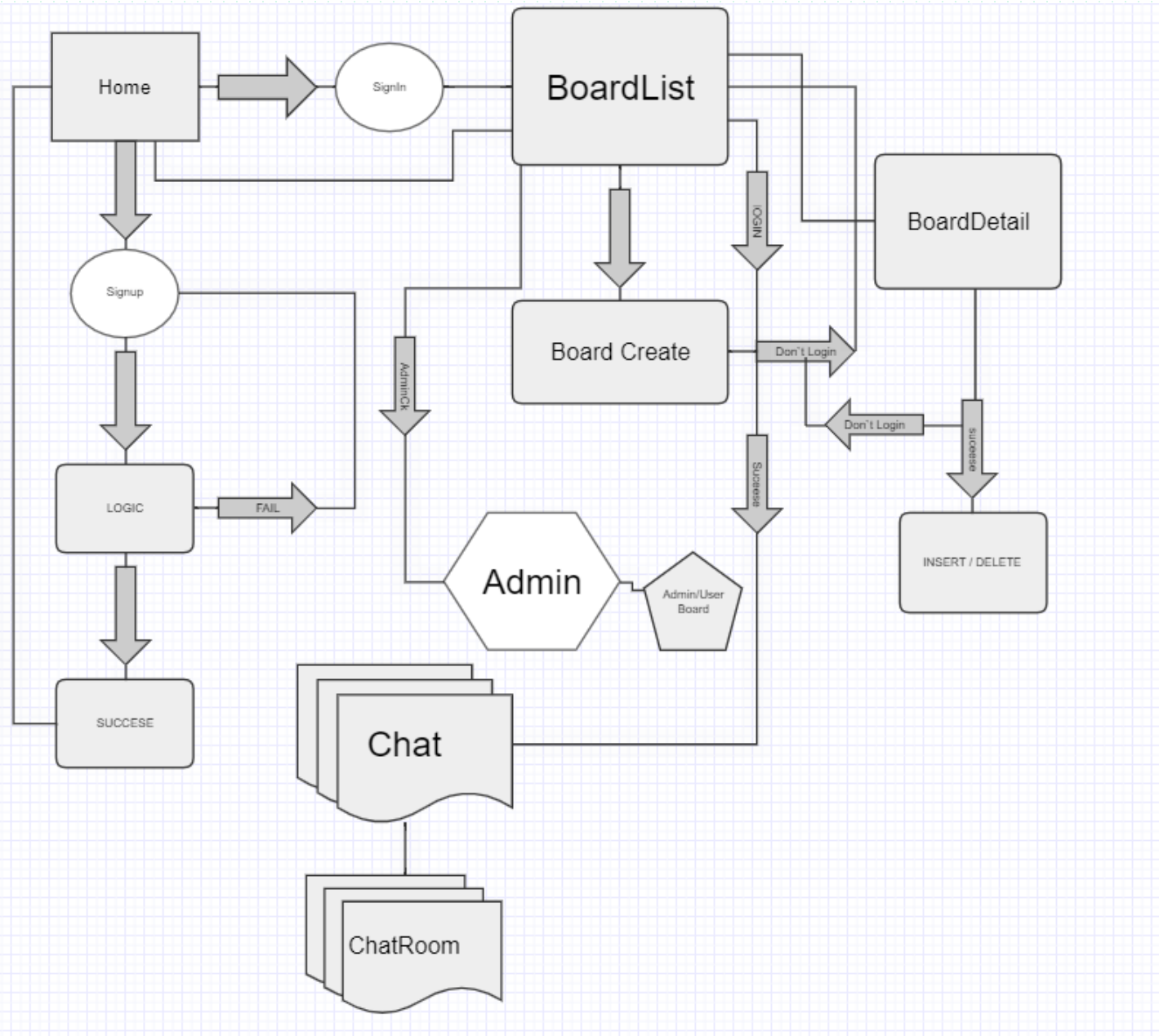
구분		진행내용	11월 21일 ~ 12월 3일			
			11/23	11/27	12/1	12/3
기획		• 컨텐츠 구상 및 정의				
디자인	웹 설계	• 웹페이지 구상 및 UI 상세 설계				
	DB 설계	• 데이터베이스 상세 설계				
	데이터베이스 구현	• 유저 생성 및 데이터베이스 테이블 설계				
개발	UI구현	• UI 레이아웃 구현				
		• 디자인 구현				
	컨트롤러 구현	• DAO 구현				
		• 데이터 검증 메서드 구현				
		• 페이지 연동				
수정 및 보완		• 예외 처리 및 버그 수정				
제출		• 최종 검토 및 제출				

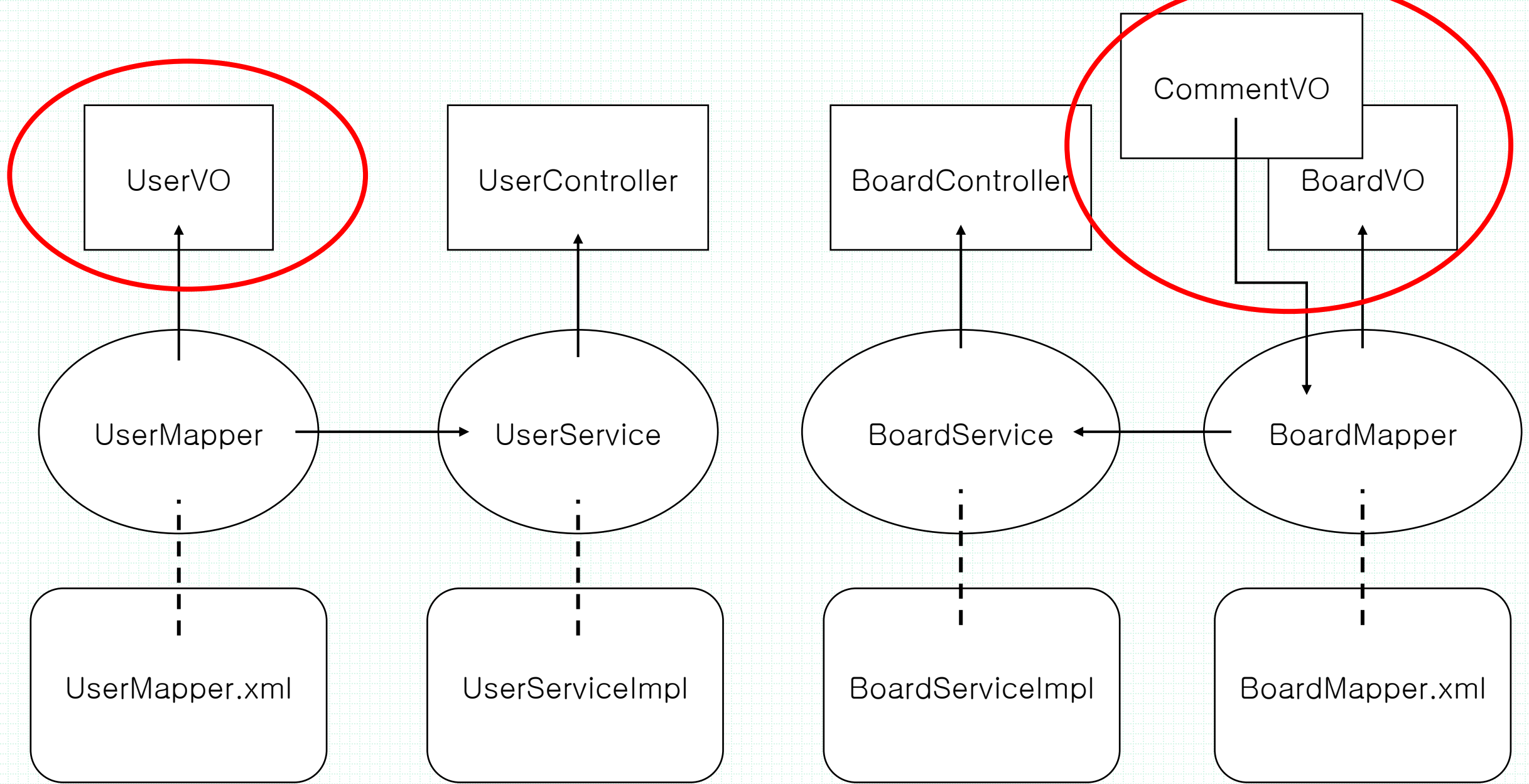
UserVO	
memberId	아이디
memberPw	비밀번호
memberMail	메일
memberName	닉네임
memberAddr1	우편번호
memberAddr2	도로명 주소
memberAddr3	상세주소
memberFile	프로필사진
regdate	등록일자
Adminck	관리자번호

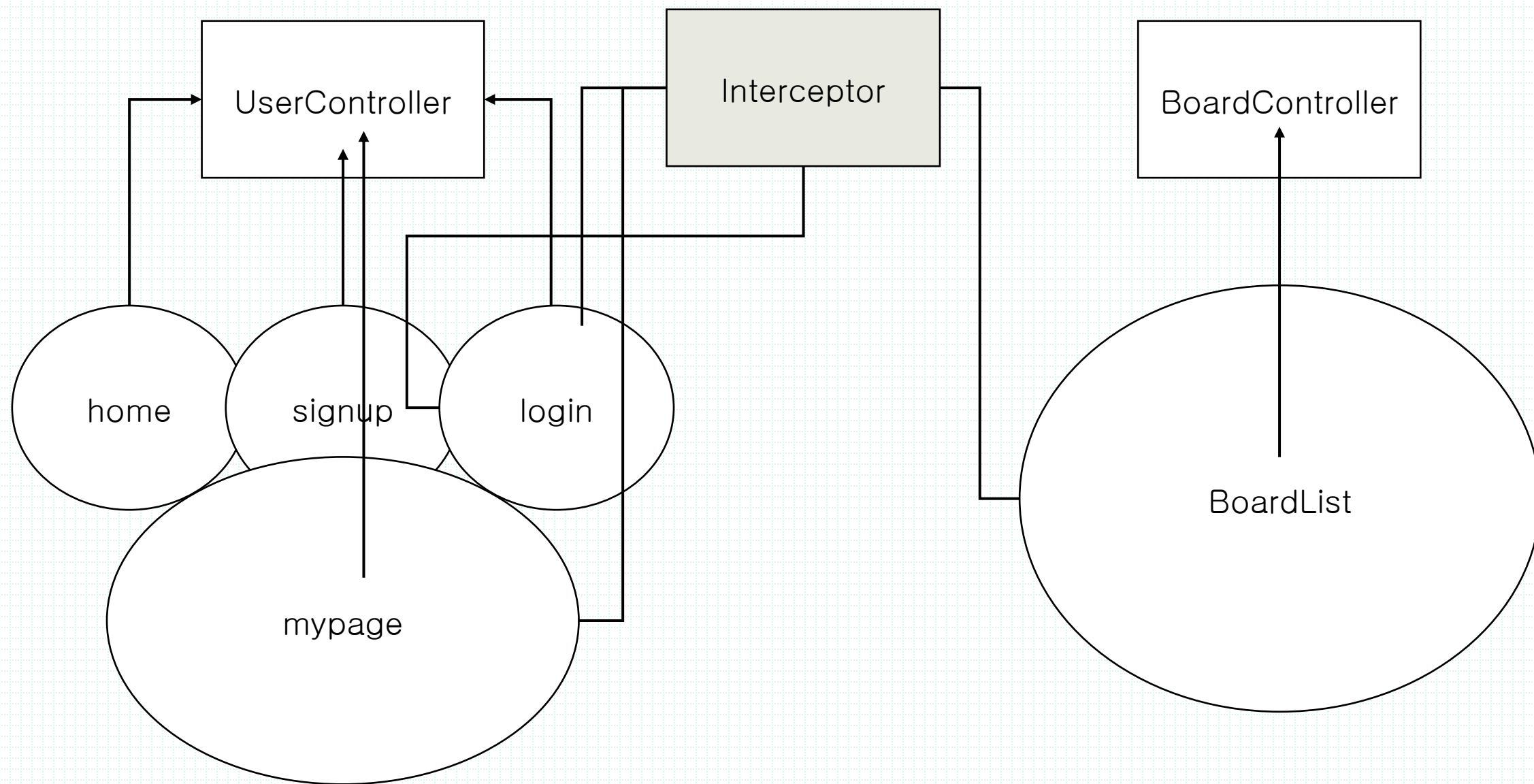
Board	
Bno	게시물 pk번호
cnt	조회수
pic	사진
likes	좋아요
title	제목
memberId	작성자
contnet	내용
regDate	등록일자
updateDate	업데이트 일자

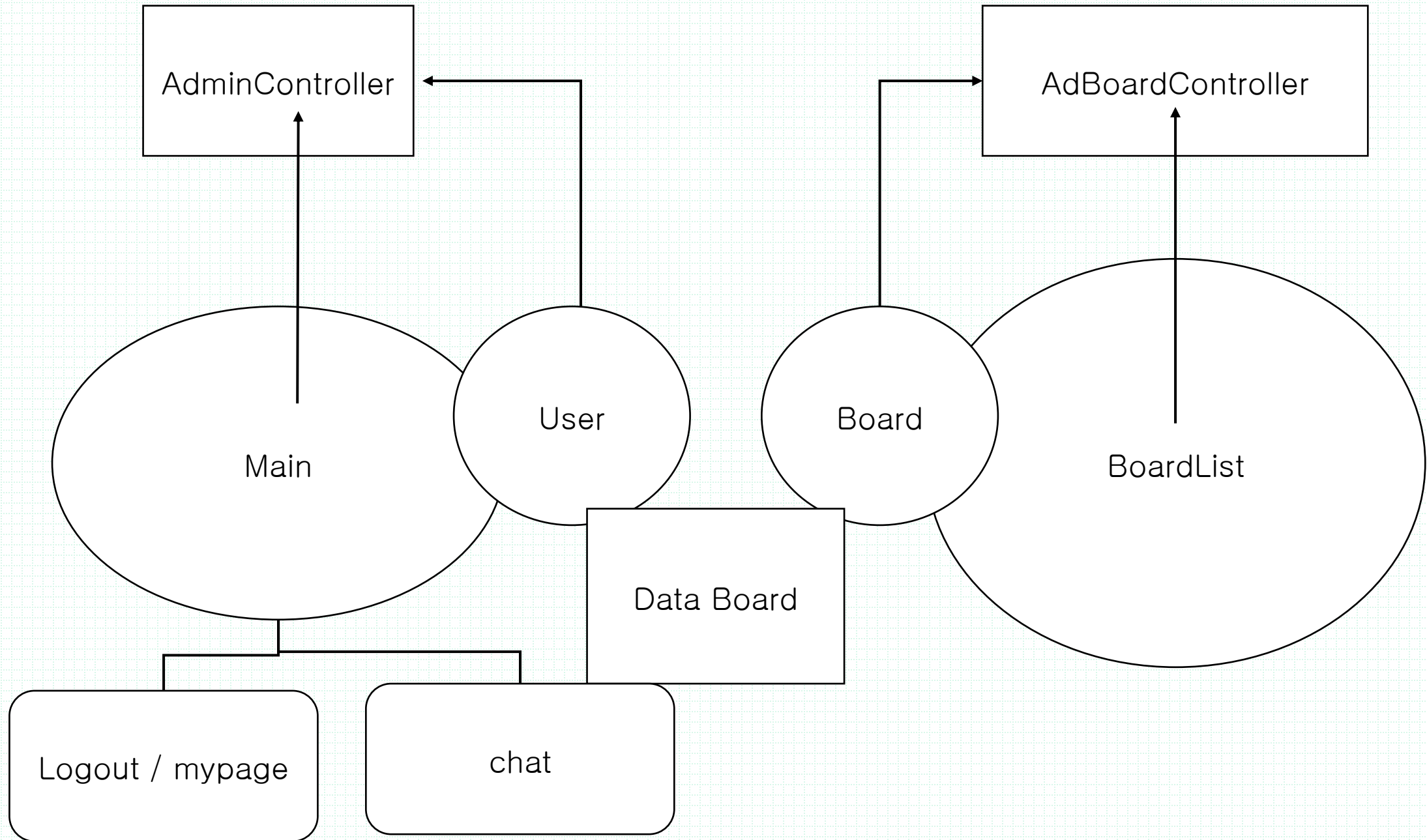
Comments	
Bno	게시물 pk번호
cno	댓글 pk번호
memberId	댓글 작성자
context	댓글 내용











```

// 회원가입;
public void memberJoin(UserVO user);

// 아이디 중복 검사 Check
public int idCheck(String memberId);

// 로그인
public UserVO UserLogin(UserVO user);

// 유저 업데이트
public int UserUpdate(UserVO user);

// 유저 삭제
public void UserDelete(UserVO user);

// 유저 리스트
public List<UserVO> GetList(Criteria cri);

// 상세 유저 정보
public UserVO GetUser(UserVO user);

//관리자 사이트에서 유저 삭제
public void adminUserdelete(String memberId);

// 총 유저수
public int CountUser(Criteria cri);

// 특정 유저 id
public int CountUserId(String memberId);

```

```

<![CDATA[
    select memberId, memberName, memberMail, memberAddr1, memberAddr2
    from (
        select rownum as rn, memberId, memberName, memberMail, memberAddr1, memberAddr2
        from mini_member
        where rownum <= #{pageNum} * #{amount}
    )
    <if test="keyword != null">
        and memberId like '%'||#{keyword}||'%'
    </if>
    <![CDATA[
    )
    where rn > (#{pageNum} - 1) * #{amount}
    ]]>
</select>

```

```

<select id="CountUser" resultType="int">

    select count(*) from mini_member

    <if test="keyword != null">

        where memberId like '%'||#{keyword}||'%'

    </if>

</select>

```

```

01 public interface UserService {
02
03     public void memberJoin(UserVO vo);
04
05     public int idCheck(String memberId);
06
07     public UserVO memberLogin(UserVO user);
08
09     public void memberUpdate(UserVO user);
10
11     public void memberDelete(UserVO user);
12
13     public List<UserVO> getUserList(Criteria cri);
14
15     public UserVO getUser(UserVO user);
16
17     public void UserDelete(String memberId);
18
19     public int CountUser(Criteria cri);
20
21     public int CountUserId(String memberId);
22
23     public int CountBoard();
24
25     public int CountComment();
26
27     // 마이페이지 관련
28     List<BoardVO> getMyBoard(Criteria cri);
29     //
30     int selectBoardListCnt(Criteria cri);

```

```

@Override
public int CountUserId(String memberId) { return userMapper.CountUserId(memberId); }

@Override
public int CountBoard() { return userMapper.CountBoard(); }

@Override
public int CountComment() { return userMapper.CountComment(); }

@Override
public List<BoardVO> getMyBoard(Criteria cri) { return userMapper.getMyBoard(cri); }
@Override
public int selectBoardListCnt(Criteria cri) { return userMapper.selectBoardListCnt(cri); }

```

Travel

#여행#리뷰#소통

시작하기

로그인

회원 가입



```
<a href="/board/getBoardList"
  <span> 시작하기</span>
</a>
<a href="/user/login" class="l
  <span> 로그인</span>
</a>
<a href="/user/signup" class="
  <span>회원 가입</span>
</a>
```

```
// 홈 화면
@GetMapping("/")
public String hello(){

    return "hello";
}
```

회원가입

아이디

s

영문자 또는 숫자 6~20

비밀번호

as

제대로 적으시오 비밀번호 정규식
자 포함 형태의 8~20

비밀번호 확인

as

이름

d

이메일

sss

올바르지 못한 이메일 형식입니다

인증번호 전송

주소

주소 찾기

```
@PostMapping("/user/signup")
public String UserSignupPOST(UserV0 user){

    String noHashPw = "";
    String HashPw = "";

    noHashPw = user.getMemberPw();
    HashPw = passwordEncoder.encode(noHashPw);
    user.setMemberPw(HashPw);

    userService.memberJoin(user);

    return "redirect:/";
}
```

```
@Autowired
private BCryptPasswordEncoder passwordEncoder;
```

MEMBERPW
\$2a\$10\$QLpUytjd1K03zUaAlv/a/.c/azG0lzZOmss/zaSvH1sOVOWKcoi
\$2a\$10\$yKFBHCUprY1/Zc3AKMhJ.wsFKOmoOq6m5T2Ybx00iu5/s5Ej86ly

```
@PostMapping("/user/userIdCheck")
@ResponseBody
public String userIdCheck(String memberId){

    int result = userService.idCheck(memberId);

    if (result != 0){
        return "fail";
    }else {
        return "success";
    }
}
```

```
@GetMapping("/user/mailCheck")
@ResponseBody
public String mailCheck(String email){

    log.info("이메일 데이터 전송 확인");
    log.info("인증번호 : " + email );

    Random random = new Random();
    int Num = random.nextInt( bound: 888888)+111111;
    log.info("인증번호 : " + Num);
}
```

```
@Autowired
private JavaMailSender mailSender;
```

```
// 유효성 검사
```

```
let idCheck = false;
let idckCheck = false;
let pwCheck = false;
let pwckCheck = false;
let pwckckCheck = false;
let nameCheck = false;
let mailCheck = false;
let mailnumCheck = false;
let addressCheck = false;
```

```
$.ajax({
  type : "post",
  url : "/user/userIdCheck",
  data : data,
  success : function (result){
    if(result != 'fail'){
      $('#id_input_re_1').css("display","inline-block") //
      $('#id_input_re_2').css("display","none") // 실패
      idckCheck = true;
    } else{
      $('#id_input_re_2').css("display","inline-block")
      $('#id_input_re_1').css("display","none")
      idckCheck = false;
    }
  }
});
```

```
$.ajax({
  type: "GET",
  url: "/user/mailCheck?email="+email,
  success: function (data){
    checkBox.attr("disabled", false);
    boxWrap.attr("id", "mail_check_input_box_true");
    code = data;
  }
});
```

```
if(idCheck&&idckCheck&&pwCheck&&pwckCheck&&pwckckCheck&&nameCheck&&mailCheck&&mailnumCheck&&addressCheck){
  $("#signup_form").attr("action","/user/signup");
  $("#signup_form").submit();
}
```

Welcome Aboard

지금 로그인하셔서 사람들과 함께 즐거운 여행을 떠나세요!


 ID를 입력하세요

 PW를 입력하세요

로그인



```
@PostMapping("/user/login.do")
public String UserLoginForm(HttpServletRequest request, UserVO user, RedirectAttributes rttr){
    // HttpSession 인터페이스는 둘 이상의 page request에서
    // 사용자를 식별하거나,
    // 웹 사이트를 방문하고 해당 사용자에 대한 정보를
    // 저장하는 방법을 제공한다.

    HttpSession session = request.getSession();
    String noHashPw = "";
    String HashPw = "";

    UserVO vo = userService.memberLogin(user);

    if (vo == null){
        // 실패
        int result = 0;
        rttr.addFlashAttribute( attributeName: "result" ,result);
        return "redirect:/user/login";
    } else {

        noHashPw = user.getMemberPw();
        HashPw = vo.getMemberPw();

        if(true == passwordEncoder.matches(noHashPw,HashPw)){
            // 성공
            vo.setMemberPw("");
            session.setAttribute( name: "member",vo);
            session.setAttribute( name: "id",vo.getMemberId());

            return "redirect:/board/getBoardList";
        } else {
            // 실패

```



```

@Mapper
public interface BoardMapper {

    // 글 등록
    void insertBoard(BoardVO board);
    // 글 수정 bno를 이용 title.content 수정가능
    void updateBoard(BoardVO board);
    // 글 삭제
    void deleteBoard(BoardVO board);
    // 상세보기
    BoardVO getBoard(BoardVO board);
    // 전체 |보기
    List<BoardVO> getBoardList(BoardVO board);
    // 페이지징 용도
    List<BoardVO> getPage(Criteria cri);
    // 게시글 갯수
    public int selectBoardListCnt(Criteria cri);
    // 조회수 증가
    void read(BoardVO board);
    // 댓글 등록
    void insertComment(CommentVO comment);
    // 댓글 조회
    List<CommentVO> getCommentList(BoardVO board);
    // 댓글 삭제
    void deleteComment(CommentVO comment);
}

```

```

@Service("boardService")
public class BoardServiceImpl implements BoardService{

    @Autowired
    private BoardMapper boardMapper;

    @Override
    public List<BoardVO> getBoardList(BoardVO board) { return boardMapper.getBoardList(board); }
    @Override
    public BoardVO getBoard(BoardVO board) {return this.boardMapper.getBoard(board);}
    @Override
    public void insertBoard(BoardVO board) { this.boardMapper.insertBoard(board); }
    @Override
    public void deleteBoard(BoardVO board) { this.boardMapper.deleteBoard(board); }
    @Override
    public void updateBoard(BoardVO board) { this.boardMapper.updateBoard(board); }
    @Override
    public List<BoardVO> getPage(Criteria cri) { return this.boardMapper.getPage(cri); }
    @Override
    public int selectBoardListCnt(Criteria cri) { return this.boardMapper.selectBoardListCnt(cri); }
    @Override
    public void read(BoardVO board) { this.boardMapper.read(board); }
    @Override
    public void insertComment(CommentVO comment) { this.boardMapper.insertComment(comment); }
    @Override
    public List<CommentVO> getCommentList(BoardVO board) { return this.boardMapper.getCommentList(board); }
    @Override
    public void deleteComment(CommentVO comment) { this.boardMapper.deleteComment(comment); }
}

```


제목

Search

글쓰기

```
@RequestMapping(value = "/getBoardList")
public String getBoardList( Criteria c, Model model){
    int total = boardService.selectBoardListCnt(c);
    PageMakerVO pMake=new PageMakerVO(c,total);
    model.addAttribute( attributeName: "pageMaker",pMake);
    model.addAttribute( attributeName: "boardList",boardService.getPage(c));
    return "board/getList";
}
```

admin1 거기 가봄? 성심당 2021-12-02 10:45 조회수 15 · 추천 0	admin1 1 1 2021-12-01 16:24 조회수 62 · 추천 0	admin1 20 20 2021-12-01 10:32 조회수 5 · 추천 0	admin1 14 14 2021-12-01 09:43 조회수 1 · 추천 0	admin1 13 13 2021-12-01 09:43 조회수 1 · 추천 0
admin1 12 12 2021-12-01 09:43 조회수 2 · 추천 0	admin1 11 11 2021-12-01 09:42 조회수 0 · 추천 0	admin1 10 10 2021-12-01 09:42 조회수 0 · 추천 0	admin1 9 9 2021-12-01 09:42 조회수 0 · 추천 0	admin1 8 8 2021-12-01 09:42 조회수 0 · 추천 0

작성자	admin1
제목	<input type="text"/>
내용	<div></div>
첨부파일	<div>파일 선택</div> <div>선택된 파일 없음</div>

등록

```
@RequestMapping(value = "/insertBoard")
public String insertBoard(
    HttpServletRequest forSession, MultipartFile file, BoardVO vo) throws IOException {
    if(file != null){
        String projectPath = System.getProperty("user.dir") + "\\src\\main\\resources\\static\\imgs";
        String basePath = projectPath + "/";
        UUID uuid = UUID.randomUUID();
        String fileName = uuid + "_" + file.getOriginalFilename();
        File saveFile = new File(pathname: basePath + fileName);
        file.transferTo(saveFile);
        vo.setPic(fileName);
    }

    HttpSession session = forSession.getSession();
    String id = (String) session.getAttribute(name: "id");
    vo.setMemberId(id);

    boardService.insertBoard(vo);

    return "redirect:/board/getBoardList";
}
```

거기 가봄?

admin1 / 수정 / 삭제

2021.12.02 10:45 조회수 15

성심당

2개의 댓글

댓글을 남겨주세요

작성


admin1

삭제

Windows 정보 인증

```
// 댓글 등록 메소드
@RequestMapping(value="/insertComment")
public String insertBoard(CommentVO comment,
                          HttpServletRequest forSession){
    HttpSession session = forSession.getSession();
    String id = (String) session.getAttribute("id");
    comment.setMemberId(id);
    boardService.insertComment(comment);
    return "redirect:/board/getBoard?bno="+comment.getBno();
}

// 댓글 삭제 메소드
@RequestMapping(value="/deleteComment")
public String deleteComment(CommentVO comment){
    this.boardService.deleteComment(comment);
    return "redirect:/board/getBoard?bno="+comment.getBno();
}
```

작성자	admin1
제목	얼마전에 다녀온 곳
내용	<div> <div>노을보세요</div> </div>
첨부파일 변경	<div>  <div> <div>파일 선택</div> <div>선택된 파일 없음</div> </div> </div>

```

@RequestMapping(value = "/updateBoard")
public String updateBoard(
    MultipartFile file, BoardVO vo) throws IOException {

    String projectPath = System.getProperty("user.dir") + "\\src\\main\\resources\\static\\imgs";
    String basePath = projectPath + "/";
    UUID uuid = UUID.randomUUID();
    String fileName = uuid + "_" + file.getOriginalFilename();
    File saveFile = new File(pathname: basePath + fileName);
    file.transferTo(saveFile);
    vo.setPic(fileName);

    boardService.updateBoard(vo);
    return "redirect:/board/getBoardList";
}

@RequestMapping(value = "/deleteBoard")
public String deleteBoard(BoardVO vo){
    boardService.deleteBoard(vo);
    return "redirect:/board/getBoardList";
}

```

관리자 페이지

현재 가입한 유저의 수

2

현재 등록된 글의 수

3

현재 등록된 댓글의 수

0

의 삭제가 가능합니다.

확인해 보세요

Mini_Project

홈으로

채팅

데이터 관리

관리자 계정 관리

로그아웃

```
@Autowired
private UserService userService;

@GetMapping("/admin/main")
public String main(Model model, Criteria cri){

    int CommentCnt = userService.CountComment();
    int BoardCnt = userService.CountBoard();
    int usercnt = userService.CountUser(cri);
    model.addAttribute(attributeName: "usercnt", usercnt);
    model.addAttribute(attributeName: "BoardCnt", BoardCnt);
    model.addAttribute(attributeName: "CommentCnt", CommentCnt);

    return "/admin/main";
}
```

보드 리스트

목록 Search 검색하기

보드번호	제목	작성자	내용	작성날짜	조회수
242	얼마전에 다녀온 곳	admin1	느음보세요 봤어요?	2021-12-02 16:54	7
230	1	admin1	1	2021-12-01 16:24	62
229	20	admin1	20	2021-12-01 10:32	5
228	14	admin1	14	2021-12-01 09:43	1
227	13	admin1	13	2021-12-01 09:43	1
226	12	admin1	12	2021-12-01 09:43	3
225	11	admin1	11	2021-12-01 09:42	0
224	10	admin1	10	2021-12-01 09:42	0
223	9	admin1	9	2021-12-01 09:42	0
222	8	admin1	8	2021-12-01 09:42	0

처음 1 2 마지막

```
@RequestMapping(value = "/admin/boardList")
public String adminBoardList(Model model, Criteria c){
    int total = boardService.selectBoardListCnt(c);
    PageMakerVO pMake=new PageMakerVO(c,total);
    model.addAttribute( attributeName: "pageMaker",pMake);
    model.addAttribute( attributeName: "boardList",boardService.getPage(c));
    return "admin/boardList";
}
```



게시판 상세 정보

보드번호	242
제목	얼마전에 다녀온 곳
작성자	admin1
내용	노을보세요 봤어요?
작성날짜	2021-12-02 16:54
조회수	7
파일	
게시글 댓글 내용, 댓글 숫자 : 2개	댓글작성자
아노 댓글 삭제	admin1
네 댓글 삭제	admin1

[게시판 목록으로](#)
[해당게시글 삭제](#)

```

@RequestMapping(value="/admin/getBoard")
public String adminGetBoard(Model model, BoardVO vo){
    model.addAttribute( attributeName: "commentList",boardService.getCommentList(vo));
    model.addAttribute( attributeName: "board",boardService.getBoard(vo));
    return "admin/getBoard";
}

@RequestMapping(value = "/admin/boardDelete")
public String deleteBoard(BoardVO vo){
    boardService.deleteBoard(vo);
    return "redirect:/admin/boardList";
}

// 댓글 삭제 메소드
@RequestMapping(value="/admin/deleteComment")
public String deleteComment(CommentVO comment){
    this.boardService.deleteComment(comment);
    return "redirect:/admin/getBoard?bno="+comment.getBno();
}
    
```

당신의 프로필



정광수



당신이 등록한 게시물

제목	페이지
o	이동
o L o o ㄹ	이동
d	이동

처음 1 마지막

```
@GetMapping("/user/mypage")
public String mypageGET(HttpServletRequest forSession,
                        Model model,
                        Criteria cri
                        ){

    HttpSession session = forSession.getSession();
    String id = (String) session.getAttribute("id");
    cri.setMemberId(id);

    int total = userService.selectBoardListCnt(cri);
    PageMakerVO pMake=new PageMakerVO(cri,total);

    model.addAttribute("pageMaker",pMake);
    model.addAttribute("boardList",userService.getM

    return "user/mypage";
}
```


회원 정보 수정

닉네임 변경

정광수

이메일

slwhswk@naver.com

파일 선택 선택된 파일 없음

제출

```
@PostMapping("/user/mypage/modify")
public String mypagePOST(UserVO user, HttpServletRequest request, MultipartFile file) throws IOException {
    HttpSession session = request.getSession();

    String projectPath = System.getProperty("user.dir") + "\\src\\main\\resources\\static\\img";

    String basePath = projectPath + "/";

    UUID uuid = UUID.randomUUID();
    String fileName = uuid + "_" + file.getOriginalFilename();
    File saveFile = new File(pathname: basePath + fileName);
    file.transferTo(saveFile);

    user.setMemberFile(fileName);

    // session MemberId값 넣기
    String value = (String) session.getAttribute(name: "id");
    user.setMemberId(value);

    userService.memberUpdate(user);

    session.setAttribute(name: "member", user);
    session.setAttribute(name: "id", user.getMemberId());

    return "redirect:/board/getBoardList";
}
```



회원 삭제

회원 삭제시 게시물, 댓글이 전부 삭제

또한 게시물의 복원 또한 할 수 없습

그래도 삭제하시겠습니까?

제출

```
@GetMapping("/user/mypage/delete")
```

```
public String deleteGET(){
```

```
    return "/user/delete";
```

```
}
```

```
@PostMapping("/user/mypage/delete")
```

```
public String deletePOST(UserVO user, HttpServletRequest request){
```

```
    HttpSession session = request.getSession();
```

```
    String value = (String) session.getAttribute("id");
```

```
    user.setMemberId(value);
```

```
    userService.memberDelete(user);
```

```
    session.invalidate();
```

```
    return "redirect:/";
```

```
}
```

```
}
```

```

@GetMapping("/admin/userList")
public String userList(Model model, Criteria cri){

    int CntUser = userService.CountUser(cri);
    PageMakerVO pageMake = new PageMakerVO(cri, CntUser);

    model.addAttribute("list", userService.getUserList(cri));
    model.addAttribute("pageMaker", pageMake);

    return "/admin/userList";
}

@GetMapping("admin/getuser")
public String getUserPageGET(UserVO user, Model model, Criteria cri){
    int CntUser = userService.CountUser(cri);
    PageMakerVO pageMake = new PageMakerVO(cri, CntUser);

    model.addAttribute("countboard", userService.CountUserId(user.getMemberId()));
    model.addAttribute("userget", userService.getUser(user));
    model.addAttribute("pageMaker", pageMake);

    return "/admin/getuser";
}

```

SockJs

WebSocket의 불안정성

- 1.WebSocket
- 2.Http Streaming
- 3.Http Long Polling

Reference

<https://dev-gorany.tistory.com/224>

Stomp

Stomp는 메시지 전송을 효율적으로

Publisher / Subscribe
개발자가 명확히 구분할 수 있음

STOMP 프로토콜은 WebSocket 위에서 동작하는 프로토콜로써 클라이언트와 서버가 전송할 메시지의 유형, 형식, 내용들을 정의하는 매커니즘

Reference

<https://dev-gorany.tistory.com/235>



시스템 메모리를 사용하는 키-값 데이터 스토어

채팅 Publisher와 Subscribe를 따로 구분하고
서버를 꺼도 유지가 될 수 있게
Redis ,Stomp,SockJs를 사용해서 구현.

Reference

<https://brunch.co.kr/@skykamja24/575>

특징

- 처리 속도가 매우 빠름
- 데이터가 메모리에만 저장
- 만료일 지정해서 삭제 가능
- 저장소 메모리 재사용 가능

```
import ...

@Slf4j //lombok
@Profile("localhost") // profile이 local일때만 활성화
@Configuration
public class EmbeddedRedisConfig {

    @Value("${spring.redis.port}")
    private int redisPort;

    private RedisServer redisServer;

    @PostConstruct
    public void redisServer() throws IOException {
        redisServer = new RedisServer(redisPort);
        redisServer.start();
    }

    @PreDestroy
    public void stopRedis() {
        if (redisServer != null) {
            redisServer.stop();
        }
    }
}
```

```
@Configuration
public class RedisConfig {

    @Bean
    public RedisMessageListenerContainer redisMessageListener(RedisConnectionFactory connectionFactory) {

        RedisMessageListenerContainer container = new RedisMessageListenerContainer();
        container.setConnectionFactory(connectionFactory);
        return container;
    }

    @Bean
    public RedisTemplate<String, Object> redisTemplate(RedisConnectionFactory connectionFactory) {
        RedisTemplate<String, Object> redisTemplate = new RedisTemplate<>();

        redisTemplate.setConnectionFactory(connectionFactory);
        redisTemplate.setKeySerializer(new StringRedisSerializer());
        redisTemplate.setValueSerializer(new Jackson2JsonRedisSerializer<>(String.class));

        return redisTemplate;
    }
}
```

```
spring.profiles.active=local
spring.redis.host=localhost
spring.redis.port=6379
```

채팅방 생성 : pub / sub 구현을 위한 Topic이 생성됨

채팅방 입장 : Topic 구독

채팅방에서 메시지를 송수신 : 해당 Topic으로 메시지를 송신(pub), 메시지를 수신(sub)

```
@Getter
@Setter
public class ChatRoom implements Serializable {

    private static final long serialVersionUID = 6754389567348905L;

    private String roomId;
    private String name;

    public static ChatRoom create(String name){
        ChatRoom chatRoom = new ChatRoom();
        chatRoom.roomId = UUID.randomUUID().toString();
        chatRoom.name = name;
        return chatRoom;
    }
}
```

```
@Getter
@Setter
public class ChatMessage {

    public enum MessageType{
        ENTER, TALK
    }

    private MessageType type;
    private String roomId;
    private String sender;
    private String message;
}
```

채팅방 생성 : pub / sub 구현을 위한 Topic이 생성됨

채팅방 입장 : Topic 구독

채팅방에서 메시지를 송수신 : 해당 Topic으로 메시지를 송신(pub), 메시지를 수신(sub)

```
@Service
@RequiredArgsConstructor
public class RedisPublisher {
    private final RedisTemplate<String, Object> redisTemplate;

    public void publish(ChannelTopic topic, ChatMessage message){

        redisTemplate.convertAndSend(topic.getTopic(), message);
    }
}
```

```
@Slf4j
@Service
@RequiredArgsConstructor
public class RedisSubscriber implements MessageListener {

    private final ObjectMapper objectMapper;
    private final RedisTemplate redisTemplate;
    private final SimpMessageSendingOperations messagingTemplate;

    @Override
    public void onMessage(Message message, byte[] pattern) {
        try {
            // redis에서 발행된 데이터를 받아 deserialize
            String publishMessage = (String) redisTemplate.getStringSerializer().deserialize(message.getBody());
            // ChatMessage 객체로 맵핑
            ChatMessage roomMessage = objectMapper.readValue(publishMessage, ChatMessage.class);
            // Websocket 구독자에게 채팅 메시지 Send
            messagingTemplate.convertAndSend(destination: "/sub/chat/room/" + roomMessage.getRoomId(), roomMessage);
        } catch (Exception e) {
            log.error(e.getMessage());
        }
    }
}
```



```
import ...

@RequiredArgsConstructor
@Repository
public class ChatRoomRepository {

    private final RedisMessageListenerContainer redisMessageListenerContainer;

    private final RedisSubscriber redisSubscriber;

    private static final String CHAT_ROOMS = "CHAT_ROOMS";
    private final RedisTemplate<String, Object> redisTemplate;
    private HashOperations<String, String, String> hashOperations;

    private Map<String, ChannelTopic> topicMap;

    @RequiredArgsConstructor
    @Controller
    public class ChatController {

        private final RedisPublisher redisPublisher;
        private final ChatRoomRepository chatRoomRepository;

        @PostMapping("/chat/message")
        public void message(ChatMessage message) {
            if (ChatMessage.MessageType.ENTER.equals(message.getType())){
                chatRoomRepository.enterChatRoom(message.getRoomId());

                message.setMessage(message.getSender() + "님이 입장하셨습니다.");
            }

            redisPublisher.publish(chatRoomRepository.getTopic(message.getRoomId()), message);
        }
    }
}
```

Pub/chat/message

```

@Slf4j
@Component
public class MypageInterceptor implements HandlerInterceptor {

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {

        HttpSession session = request.getSession();

        log.info("session 값 없으면 바로 내쫓아내버리기~");

        UserVO user = (UserVO) session.getAttribute( name: "member");

        if (user == null ){

            response.sendRedirect( location: "/board/getBoardList");

            return false;

        }

        return true;

    }
}

```

```

@Configuration
public class WebConfig implements WebMvcConfigurer {

    private final static String adminHttp = "/admin/**";
    private final static String LoginHttp = "/user/login.do";
    private final static String MypageHttp = "/user/mypage/**";
    private final static String insertBoardHttp = "/board/goInsert";
    private final static String UserHttp = "/user/**";
    private final static String commentHttp = "/board/insertComment";

    @Override
    public void addInterceptors(InterceptorRegistry registry) {

        registry.addInterceptor(new AdminInterceptor()).addPathPatterns(adminHttp);

        registry.addInterceptor(new LoginInterceptor()).addPathPatterns(LoginHttp);

        registry.addInterceptor(new MypageInterceptor())
            .addPathPatterns(MypageHttp)
            .addPathPatterns("/chat/**")
            .addPathPatterns(insertBoardHttp);

        registry.addInterceptor(new BoardInterceptor()).addPathPatterns(commentHttp);

        registry.addInterceptor(new UserInterceptor())
            .addPathPatterns("/user/login")
            .addPathPatterns("/user/signup");

    }
}

```