

# 애니모어 게시판



개발자 김주희  
제가 맡은 게시판 구현에 대하여  
설명드리겠습니다.

## CONTENS



동기



개발환경



개발일정



설계와 구조



후기

## 1. 동기

# ‘게시판’을 맡은 이유

### ☑ Check Point 01

제일 기본적인 기능이 들어가는 곳이 어디냐고 묻는다면 전 '게시판'이라고 답하겠습니다.

그만큼 제일 기본적이면서도 중요한 기능이라고 생각하여 기초를 탄탄히 하고자 게시판을 맡게 되었습니다.

### ☑ Check Point 02

게시판의 검색부분을 구현하기 위해 Ajax에 대하여 더 자세히 공부하고 싶었기 때문에 맡게 되었습니다.

### ☑ Check Point 03

수업시간 배웠던 내용을 직접 적용을 통하여 감각과 경험을 쌓고자 선택하였습니다.

## 2.개발환경

### ‘ 백엔드 개발환경 ’

JAVA11

SQL

STS4

Spring JDBC

Gradle

왜? 실전에서 주로 사용하는 툴을 선택하여  
언제든 상황에 적응하고 효율성을 높이기 위하여 선택하  
였습니다.

### ‘ 프론트엔드 개발환경 ’

HTML

CSS

JS

Bootstrap

Ajax

왜? 보편화 된 HTML과 CSS를 꾸며 웹표준을 다루었고 페이징  
버튼은 Bootstrap을 선택하여 디자인을 깔끔하게 마무리하  
였습니다.

### 3.개발일정



6/18~7/13(26일) 프로젝트 기획 & 설계

7/21~8/2(13일) 회원관리 구현(팀원전체참여)

8/2~9/10(40일) 커뮤니티 서버구현&view구현

9/10~9/17(8일) 단위&수정테스트&최종배포

## 4.설계와구조

### 1.데이터베이스

#### 게시판

C##ANIMORE.BOARD		
P	* BNUM	NUMBER (8)
	* BCATEGORY	CHAR (1 BYTE)
	* BTITLE	VARCHAR2 (150 BYTE)
	* ID	VARCHAR2 (40 BYTE)
	* BCDATE	TIMESTAMP
	* BUDATE	TIMESTAMP
	* BHIT	NUMBER (5)
	* BGOOD	NUMBER (5)
	* BREPLY	NUMBER (5)
	* BCONTENT	CLOB
	PBNUM	NUMBER (10)
	BGROUP	NUMBER (5)
	BSTEP	NUMBER (3)
	BINDENT	NUMBER (3)
	BSTATUS	CHAR (1 BYTE)
	BNDATE	TIMESTAMP
BOARD_BNUM_PK (BNUM)		
BOARD_BNUM_PK (BNUM)		

#### 게시글 첨부파일

C##ANIMORE.BOARDFILE		
P	* FNUM	NUMBER (10)
	BNUM	NUMBER (8)
	STORE_FNAME	VARCHAR2 (50 BYTE)
	UPLOAD_FNAME	VARCHAR2 (50 BYTE)
	Fsize	VARCHAR2 (45 BYTE)
	Ftype	VARCHAR2 (50 BYTE)
	Cdate	TIMESTAMP
	Udate	TIMESTAMP
BOARDFILE_FNUM_PK (FNUM)		
BOARDFILE_FNUM_PK (FNUM)		

#### 좋아요 게시글

C##ANIMORE.GOODBOARD		
P	* GNUM	NUMBER (8)
F	* ID	VARCHAR2 (40 BYTE)
	* BNUM	NUMBER (8)
GOODBOARD_GNUM_PK (GNUM)		
GOODBOARD_ID_FK (ID)		
GOODBOARD_GNUM_PK (GNUM)		

#### 댓글

C##ANIMORE.RBOARD		
P	* RNUM	NUMBER (10)
	* BNUM	NUMBER (8)
	* ID	VARCHAR2 (40 BYTE)
	* RCDATE	TIMESTAMP
	* RUDATE	TIMESTAMP
	* RCONTENT	VARCHAR2 (800 BYTE)
	* RGOOD	NUMBER (5)
	* RGROUP	NUMBER (5)
	* RSTEP	NUMBER (5)
	* RINDENT	NUMBER (5)
	PRNUM	NUMBER (10)
	* STATUS	CHAR (1 BYTE)
RBOARD_RNUM_PK (RNUM)		
RBOARD_RNUM_PK (RNUM)		

## 4.설계와구조

### 2.패키지구조

#### 게시판

- > BoardDAO.java
- > BoardDAOImpl.java

- > BoardDTO.java
- > BoardReqDTO.java
- > SearchDTO.java

- > BoardSVC.java
- > BoardSVCImpl.java

#### 게시글 첨부파일

- > BoardUploadFileDAO.java
- > BoardUploadFileDAOImpl.java

- > BoardUploadFileDTO.java

#### 좋아요 게시글

- > GoodBoardDAO.java
- > GoodBoardDAOImpl.java

- > GoodBoardDTO.java

- > GoodBoardSVC.java
- > GoodBoardSVCImpl.java

#### 컨트롤러

- > BoardController.java
- > APIRboardController.java

#### 댓글

- > RboardDAO.java
- > RboardDAOImpl.java

- > RboardDTO.java
- > RboardListReqDTO.java

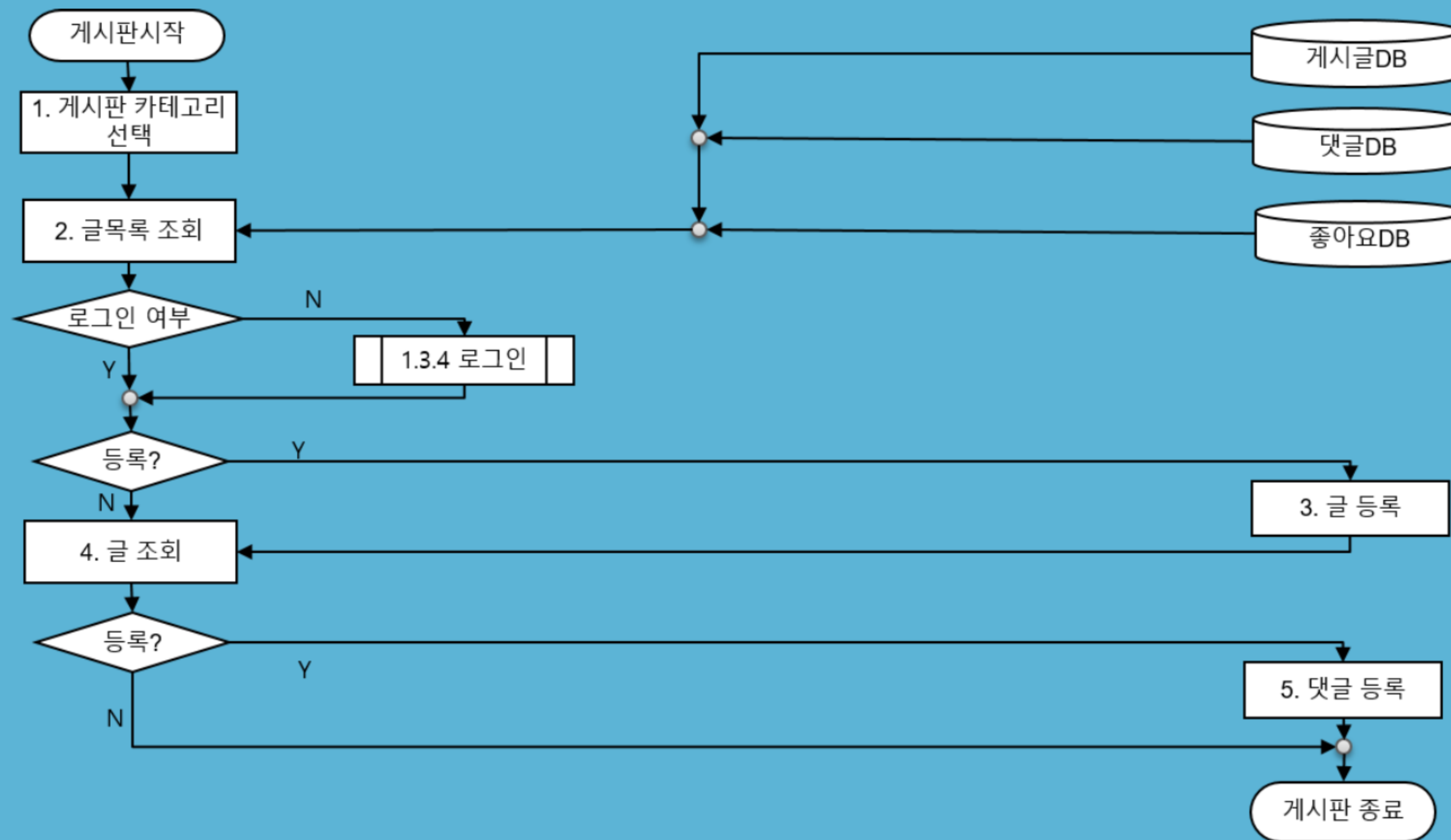
- > RboardSVC.java
- > RboardSVCImpl.java

#### View

- board
  - addBoardForm.html
  - board\_health.html
  - board.html
  - boardDetail.html
  - modifyBoardForm.html
  - replyForm.html

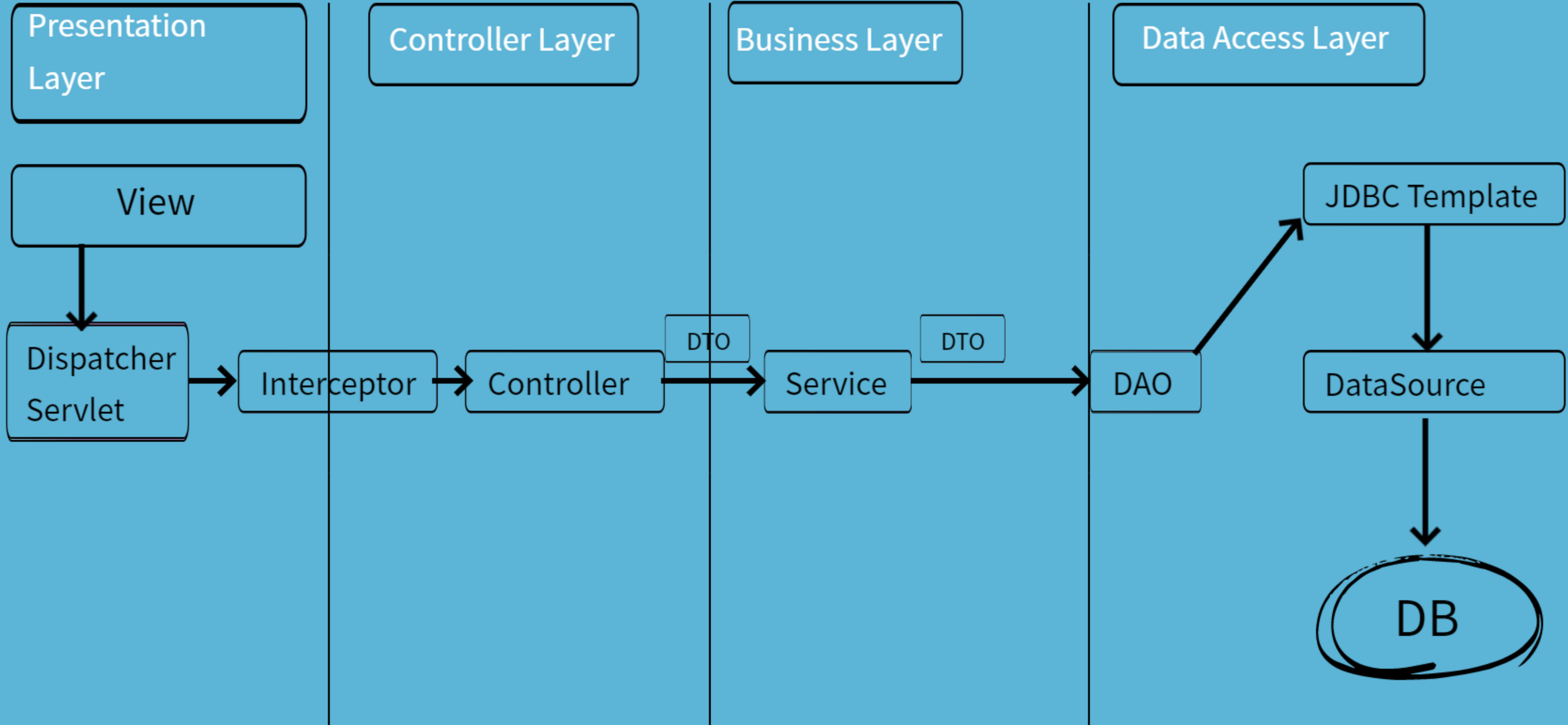
## 4.설계와구조

### 3. 프로세스





## 4. 아키텍처



## 4. 설계와 구조

## 5.back - 게시글

새끼보세요

제목	작성자	조회	좋아요
꺼지지 않나요?	저사람	2	1
	휴먼	0	0

글쓰기

Q&A

제목을 입력하세요.

## 4.설계와구조

```
//게시글 등록처리
@PostMapping("")
@Transactional
public String addpost(@Valid @ModelAttribute BoardForm boardForm,
                      BindingResult bindingResult,
                      HttpServletRequest request,
                      RedirectAttributes redirectAttributes) throws IllegalStateException, IOException {

    HttpSession session = request.getSession(false);

    LoginMember loginMember = (LoginMember)session.getAttribute("loginMember");
    String loginMemberId = loginMember.getId();

    if(bindingResult.hasErrors()) {return "board/addBoardForm";}

    //boardSVC.addBoard(loginMemberId,boardDTO);
    Log.info("boardForm:{}",boardForm);

    fileStore.setFilePath("D:/animore/src/main/resources/static/img/upload/board/");

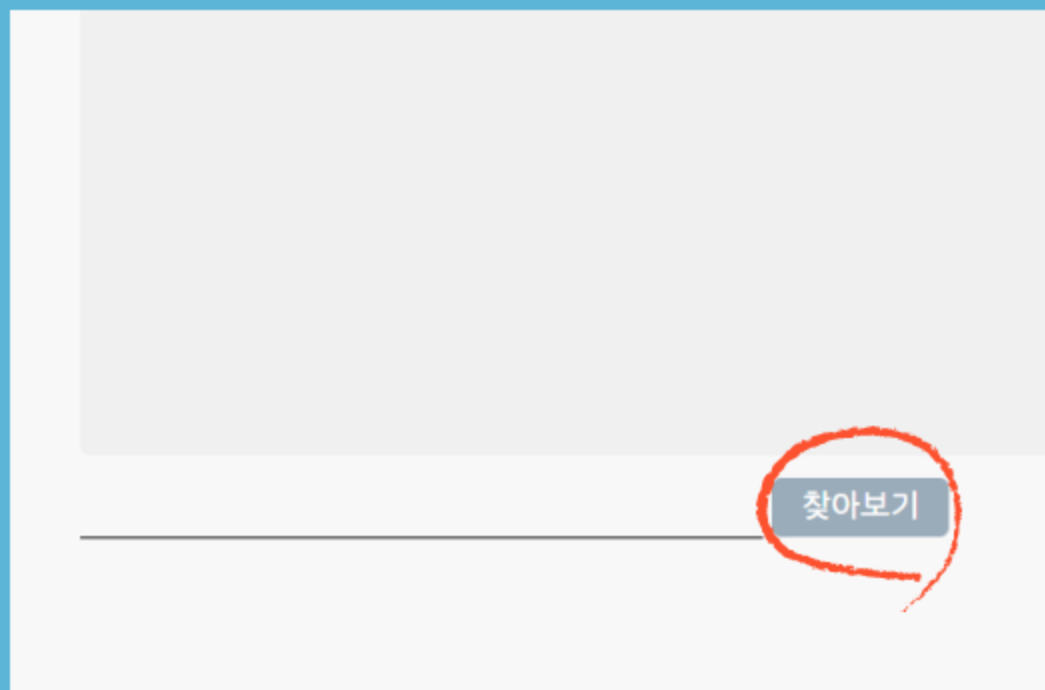
    BoardDTO boardDTO = new BoardDTO();
    //boardForm 의 값이 boardDTO에 복사됨
    BeanUtils.copyProperties(boardForm, boardDTO);
    //첨부파일 메타정보 추출
    List<MetaOfUploadFile> storedFiles = fileStore.storeFiles(boardForm.getFiles());

    boardDTO.setFiles(convert(storedFiles));

    BoardReqDTO stored = boardSVC.addBoard(loginMemberId,boardDTO);
    stored.setFiles(boardDTO.getFiles());
    Log.info("stored:{}",stored);
    redirectAttributes.addAttribute("bnum",stored.getBnum());

    return "redirect:/board/post/{bnum}";
}
```

## 5.back - 첨부파일



## 4.설계와구조

```
private BoardUploadFileDTO convert(MetaOfUploadFile attatchFile) {
    BoardUploadFileDTO boardUploadFileDTO = new BoardUploadFileDTO();
    BeanUtils.copyProperties(attatchFile, boardUploadFileDTO);
    return boardUploadFileDTO;
}

private List<BoardUploadFileDTO> convert(List<MetaOfUploadFile> uploadFiles){
    List<BoardUploadFileDTO> list = new ArrayList<>();

    for(MetaOfUploadFile file : uploadFiles) {
        BoardUploadFileDTO uploadFileDTO = convert(file);
        list.add(uploadFileDTO);
    }
    return list;
}

/**
 * 멀티파트 파일 1개 저장
 * @param multipartFile
 * @return
 * @throws IllegalStateException
 * @throws IOException
 */
public MetaOfUploadFile storeFile(MultipartFile multipartFile) throws IllegalStateException, IOException {

    if(multipartFile.isEmpty()) return null;

    //클라이언트가 업로드한 파일명
    String originalFileName = multipartFile.getOriginalFilename();
    //서버에 저장할 파일명
    String storeFileName = createStoreFileName(originalFileName);
    String fsize = String.valueOf(multipartFile.getSize());
    String ftype = multipartFile.getContentType();

    //파일시스템에 저장
    multipartFile.transferTo(Path.of(getFullPath(storeFileName)));

    return new MetaOfUploadFile(storeFileName,originalFileName, fsize, ftype);
}
```

## 4.설계와구조

### 5.back - 검색

제목 ▼ 애니모어

검색

1

```
//게시글 검색결과 목록
@Override
public List<BoardReqDTO> list(SearchDTO searchDTO) {
    StringBuffer sql = new StringBuffer();
    sql.append(" select t1.* ");
    sql.append(" from(select row_number() over (order by bgroup desc, bstep asc) num, ");
    sql.append("         b.bnum,b.bhit,b.bgood,b.btitle,b.id,m.nickname,b.bdate,b.bcategory,b.breply,b.bcontent,b.bgroup,b.bstep,b.bindent ");
    sql.append("        from board b, member m ");
    sql.append("        where b.id = m.id ");
    sql.append("        and bcategory=? ");

    switch(searchDTO.getSearchType()) {
    case "btitle":
        sql.append(" and b.btitle like '%" + searchDTO.getKeyword() + "%' ");
        break;
    case "nickname":
        sql.append(" and m.nickname like '%" + searchDTO.getKeyword() + "%' ");
        break;
    case "bcontent":
        sql.append(" and b.bcontent like '%" + searchDTO.getKeyword() + "%' ");
        break;

        default:
            break;
    }
    sql.append(") t1 ");
    sql.append(" where num between ? and ? ");
    log.info(sql.toString());
    List<BoardReqDTO> list = jt.query(sql.toString(),
                                     new BeanPropertyRowMapper<>(BoardReqDTO.class),
                                     searchDTO.getBcategory(),searchDTO.getStartRec(),searchDTO.getEndRec());

    return list;
}
```

## 4.설계와구조

### 5.back - 페이징

```
public PageCriteria(RecordCriteria rc, int pageCount) {
    this.rc = rc;
    this.PAGE_COUNT_PER_PAGE = pageCount;
    calculatePaging();
}

private void calculatePaging() {
    //요청페이지의 시작페이지
    //(올림(요청페이지/요청페이지에 보여줄 페이지수)-1) * 요청페이지에 보여줄 페이지수+1
    startPage = (int)(Math.ceil(rc.getReqPage()/((double)PAGE_COUNT_PER_PAGE)-1)*PAGE_COUNT_PER_PAGE+1;

    //요청페이지의 종료페이지
    //올림(요청페이지/요청페이지에 보여줄 페이지수) * 요청페이지에 보여줄 페이지수
    endPage = (int)Math.ceil(rc.getReqPage()/((double)PAGE_COUNT_PER_PAGE)*PAGE_COUNT_PER_PAGE;

    //최종페이지
    //올림(전체레코드수 / 요청페이지에 보여줄 레코드수)
    finalEndPage = (int)Math.ceil(totalRec/((double)rc.getREC_COUNT_PER_PAGE()));

    //최종페이지가 요청페이지의 종료페이지보다 작으면 최종페이지가 종료페이지가 된다.
    if(finalEndPage < endPage) {
        endPage = finalEndPage;
    }
}

public RecordCriteria getRc() {
    return rc;
}
```

```
@Autowired
@Qualifier("fc10")
private FindCriteria fc;
@Autowired
@Qualifier("fc8")
private FindCriteria fc2;
```

```
//게시판 전체레코드수
fc2.setTotalRec(boardSVC.totalRecordCount(cate));
```

```
list = boardSVC.list(cate,
                    fc2.getRc().getStartRec(),
                    fc2.getRc().getEndRec());
```



## 4.설계와구조

### 5.back - 좋아요게시글

```
//좋아요클릭시
@ResponseBody
@GetMapping("/good/{bnum}")
public Result addGoodBoard(@PathVariable int bnum,
                            HttpServletRequest request) {
    HttpSession session = request.getSession(false);
    LoginMember loginMember = (LoginMember)session.getAttribute("loginMember");
    String id = loginMember.getId();
    int res= goodBoardSVC.isGoodBoard(bnum,id);
    if(res==0) {
        goodBoardSVC.addGoodBoard(id, bnum);
        goodBoardSVC.upGoodBoardCnt(bnum);
    }else {
        goodBoardSVC.delGoodBoard(bnum, id);
        goodBoardSVC.downGoodBoardCnt(bnum);
    }
    Integer bgoodCnt= goodBoardSVC.GoodBoardCnt(bnum);
    Result result = new Result();
    if (res == 0) {
        result.setRtcd("01");
        result.setRtmsg("좋아요 추가되었습니다.");
        result.setData(bgoodCnt);
    } else {
        result.setRtcd("00");
        result.setRtmsg("좋아요 제거되었습니다.");
        result.setData(bgoodCnt);
    }

    return result;
}
```

```
//좋아요버튼클릭시
$heart.addEventListener('click',e=>{
    const URL = `/board/good/${e.target.dataset.bnum}`;
    console.log(URL);
    console.log('하트버튼클릭');
    request.get(URL)
        .then(res=>res.json())
        .then(res=>{
            console.log(res);
            if(res.rtcd=='00'){
                $heart.classList.toggle('toggleColorChange');
                $heart.classList.toggle('far');
                $heart.classList.toggle('fas');
            } else if(res.rtcd=='01'){
                $heart.classList.toggle('toggleColorChange');
                $heart.classList.toggle('far');
                $heart.classList.toggle('fas');
            } else{
                throw new Error(res.rtmsg);
            }
            $likeHits.textContent = res.data;
        })
        .catch(err=>{
            //오류로직 처리
            console.log (err.message);
        });
});
```

## 5.후기

홈페이지에서 기본적인 기능을 구현해볼수 있는 기회였던 프로젝트였습니다.

게시판의 CRUD와 좋아요기능,공지기능,첨부파일을 구현하면서 많은 노력과 시간이 투자되어야 한다는 것을 느끼게 되었습니다.

게시글목록에서 ThymeLeaf를 사용하여 서버에서 동적으로 html을 구현하였습니다.

효율성을 높히고 제가 필요한 부분을 깔끔하게 구현할수 있어 만족하였습니다.

프로젝트를 하며 막히는 부분에선 질문을 하고 조원들과 머리를 돌려싸고 고민하는 시간이 저에겐 기본기를 탄탄히 할수있는 기회였습니다.

앞으로 늘 궁금해하고 찾아내고 성장할수있는 능동적인 개발자가 되겠습니다.

읽어주셔서 감사합니다.

**감사합니다**

*Thank You!*