

## **‘축구인들의 모임’ 프로젝트**

# 목차

## 1. 서론

### 1.1 개요

### 1.2 Software Architecture

## 2. 본론

### 2.1 Use Case Diagram

### 2.2 Activity Diagram

### 2.3 UI Prototype & Completed UI

### 2.4 Data Modeling

### 2.5 DAO/VO

### 2.6 Class Diagram

### 2.7 Task List

## 3. 결론

# 1. 서론

## 1.1 개요

축구를 선호하는 회원들에게 소통할 수 있는 공간이다. 로그인이 필요한 서비스이며, 회원은 자유롭게 게시글을 남길 수 있고, 리그의 구단 정보 확인 등을 할 수 있다. 글을 수정 및 삭제할 수 있으며, 다른 회원들이 남긴 게시글을 확인할 수 있다. 국내외 축구 팬 간 소통과, 구단 정보의 용이한 접근성을 기대한다.

## 1.2 Software Architecture

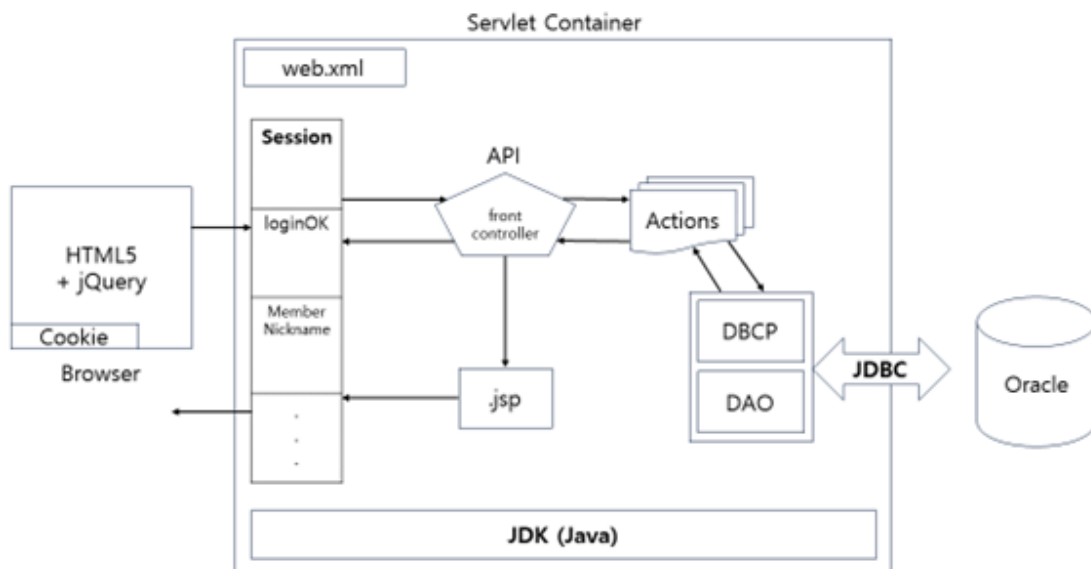


Figure 1 Architecture

## 2. 본론

### 2.1 Use Case Diagram

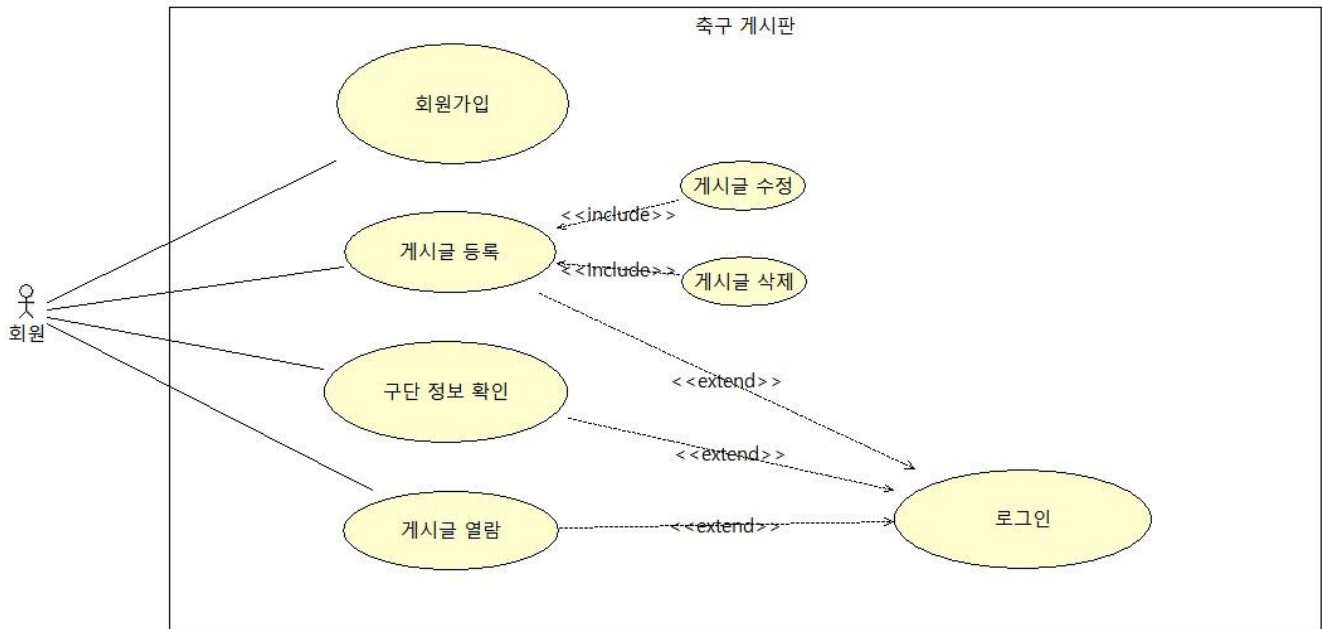


Figure 2 Use Case Diagram

### 2.2 Activity Diagram

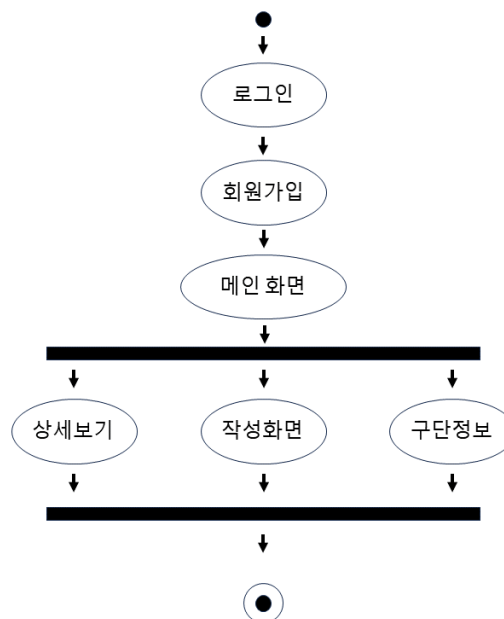
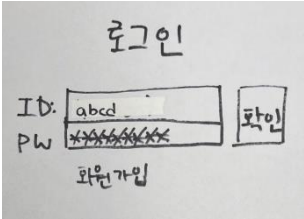
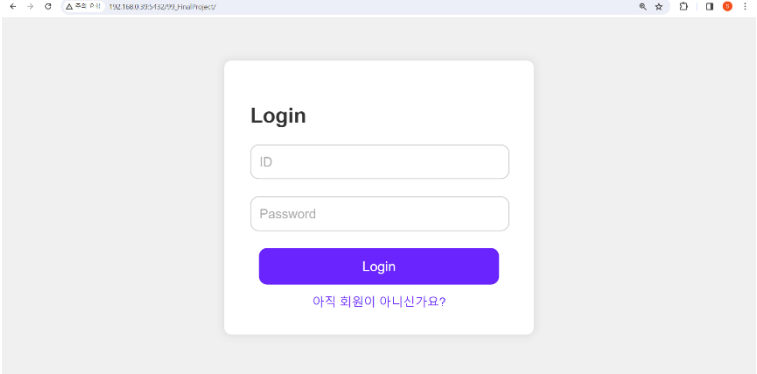
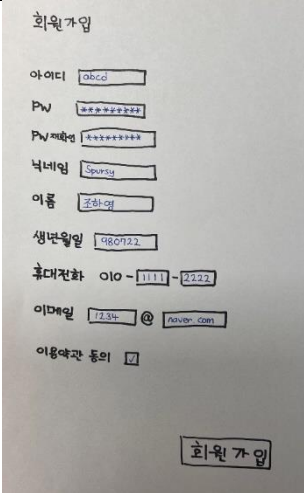
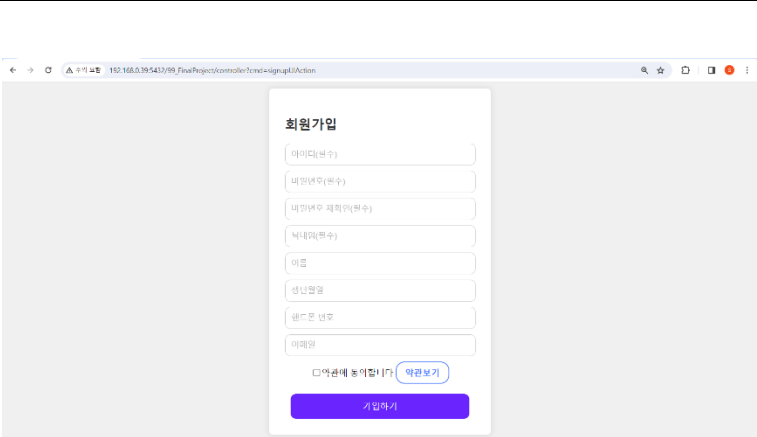
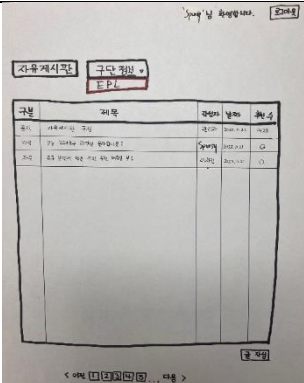
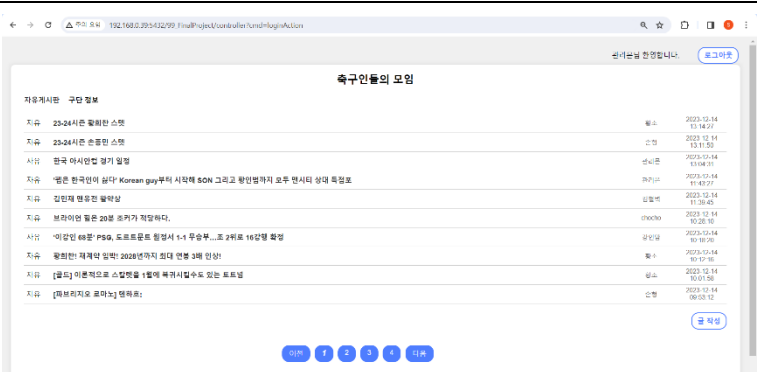
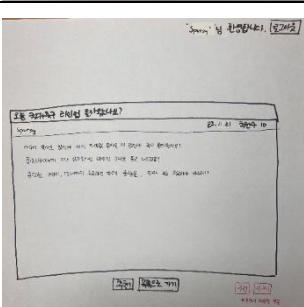
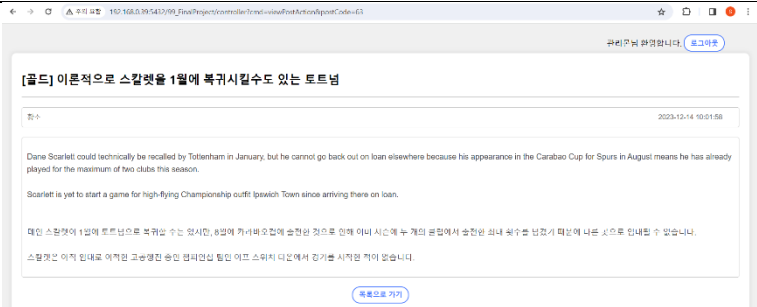


Figure 3 Activity Diagram (WorkFlow)

## 2.3 UI Prototype & Completed UI

	Prototype	Completed UI
로그인		
회원가입		
메인 화면		
상세보기		

글 작성성

구단 선택

구단 정보

토트넘 핫스퍼	
창단연도	1882년
연고지	런던
홈구장	토티엄 핫스퍼 스타디움
역대입	Spurs
EPL 우승	2회
FA컵 우승	8회
리그컵 우승	4회
UEFA 챔피언스리그 우승	0회
유러피언컵 우승	2회

Table 1 Prototype & Completed UI

## 2.4 Data Modeling

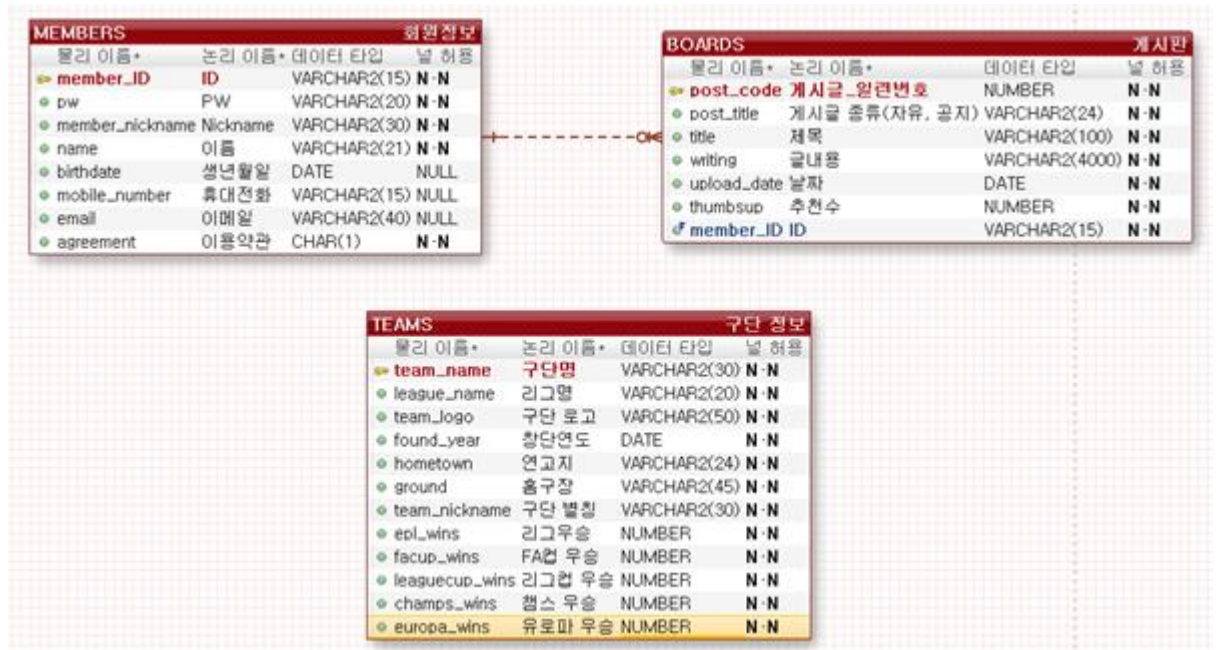


Figure 4 Data Modeling

## 2.5 DAO/VO

### BoardDAO

```
package com.oopsv.model;

//회원정보 VO
//게시판 post_code, post_title, title, writing, upload_date, thumbsup
public class BoardVO {
    //member data
    private int postCode;
    private String postTitle;
    private String title;
    private String writing;
    private String memberNickname;
    private String uploadDate;
    private int thumbsUp;

    //constructors
    public BoardVO() {}

    public BoardVO(int postCode, String postTitle, String title, String writing, String memberNickname,
        String uploadDate, int thumbsUp) {
        super();
        this.postCode = postCode;
        this.postTitle = postTitle;
        this.title = title;
        this.writing = writing;
        this.memberNickname = memberNickname;
        this.uploadDate = uploadDate;
        this.thumbsUp = thumbsUp;
    }
}

//회원정보 DAO
public class BoardDAO {
    private Connection conn;
    private PreparedStatement pstmt;
    private ResultSet rs;

    //회원정보 등록
    public boolean addMember(MemberVO vo) {
        boolean result = false;
        String sql = "insert into members(member_id, pw, member_nickname, name, birthdate, mobile_number, email, agreement) "
            + "values(?, ?, ?, ?, ?, ?, ?, ?)";

        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, vo.getMemberID());
            pstmt.setString(2, vo.getPw());
            pstmt.setString(3, vo.getMemberNickname());
            pstmt.setString(4, vo.getName());
            pstmt.setString(5, vo.getBirthdate());
            pstmt.setString(6, vo.getMobileNumber());
            pstmt.setString(7, vo.getEmail());
            pstmt.setInt(8, vo.getAgreement());
            if(pstmt.executeUpdate() > 0) {
                result = true;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    //회원정보 조회
    public int getPostThumbers(int postCode) {
        int result = 0;
        String sql = "select count(*) from boards "
            + "where post_code = ?";

        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, postCode);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                result = rs.getInt(1);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    //게시판 조회
    public BoardVO getPost(int postCode) {
        BoardVO vo = null;
        String sql = "select post_code, post_title, title, writing, member_nickname, upload_date, thumbsup from boards b, members m "
            + "where b.member_id = m.member_id and b.post_code = ?";

        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, postCode);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                vo = new BoardVO(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getInt(7));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return vo;
    }
}
```

Figure 5 BoardDAO 1

```

//회원 삭제
public boolean deletePost(String postTitle, String title, String writing, String memberID){
    boolean result=false;
    String sql="delete into boards(post_code, post_title, title, writing, upload_date, thumbsup, member_id) values(boards_seq.nextval, ?, ?, ?, ?, ?, ?)";
    try{
        PreparedStatement pstmt=conn.prepareStatement(sql);
        pstmt.setString(1, postTitle);
        pstmt.setString(2, title);
        pstmt.setString(3, writing);
        pstmt.setString(4, memberID);
        if(pstmt.executeUpdate()>0){
            result=true;
        }
        pstmt.close();
        conn.close();
    }catch(SQLException e){
        e.printStackTrace();
    }
    return result;
}

//회원 수정
public boolean updatePost(String postTitle, String title, String writing, int postCode, String memberID){
    boolean result=false;
    String sql="update boards set post_title=?, title=?, writing=? where post_code=? and member_id=?";
    try{
        PreparedStatement pstmt=conn.prepareStatement(sql);
        pstmt.setString(1, postTitle);
        pstmt.setString(2, title);
        pstmt.setString(3, writing);
        pstmt.setInt(4, postCode);
        pstmt.setString(5, memberID);
        if(pstmt.executeUpdate()>0){
            result=true;
        }
        pstmt.close();
        conn.close();
    }catch(SQLException e){
        e.printStackTrace();
    }
    return result;
}

//회원 삭제
public boolean removePost(int postCode, String memberID){
    boolean result=false;
    String sql="delete from boards where post_code=? and member_id=?";
    try{
        PreparedStatement pstmt=conn.prepareStatement(sql);
        pstmt.setInt(1, postCode);
        pstmt.setString(2, memberID);
        if(pstmt.executeUpdate()>0){
            result=true;
        }
        pstmt.close();
        conn.close();
    }catch(SQLException e){
        e.printStackTrace();
    }
    return result;
}

//팀명 반환
public TeamVO getTeamInfo(String teamName){
    TeamVO vo=null;
    String sql="select team_name, team_logo, found_year, hometown, ground, team_nickname, "
        + "top_wins, fup_wins, leaguecup_wins, champs_wins, europa_wins "
        + "from TEAM "
        + "where team_name=?";
    try{
        PreparedStatement pstmt=conn.prepareStatement(sql);
        pstmt.setString(1, teamName);
        ResultSet rs=pstmt.executeQuery();
        if(rs.next()){
            vo=new TeamVO(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8),
                rs.getString(9), rs.getString(10), rs.getString(11));
        }
    }catch(SQLException e){
        e.printStackTrace();
    }
    return vo;
}

//ID 중복 체크
public boolean idCheck(String memberID){
    boolean result=false;
    String sql="select * from members where member_id=?";
    try {
        PreparedStatement pstmt=conn.prepareStatement(sql);
        pstmt.setString(1, memberID);
        ResultSet rs=pstmt.executeQuery();

        if(rs.next()){
            result=true;
        }

        rs.close();
        pstmt.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return result;
}

//nickname 중복 체크
public boolean nicknameCheck(String memberNickname){
    boolean result=false;
    String sql="select * from members where member_nickname=?";
    try {
        PreparedStatement pstmt=conn.prepareStatement(sql);
        pstmt.setString(1, memberNickname);
        ResultSet rs=pstmt.executeQuery();

        if(rs.next()){
            result=true;
        }

        rs.close();
        pstmt.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return result;
}

```

Figure 6 BoardDAO 2

## BoardVO

```

package com.oopsw.model;

//회원정보 memberID, pw, member nickname, name, birthdate, mobile number, email, agreement
//게시판 post_code, post title, title, writing, upload_date, thumbsup
public class BoardVO {
    //member data
    private int postCode;
    private String postTitle;
    private String title;
    private String writing;
    private String memberNickname;
    private String uploadDate;
    private int thumbsUp;

    //constructors
    public BoardVO() {}

    public BoardVO(int postCode, String postTitle, String title, String writing, String memberNickname,
        String uploadDate, int thumbsUp) {
        super();
        this.postCode = postCode;
        this.postTitle = postTitle;
        this.title = title;
        this.writing = writing;
        this.memberNickname = memberNickname;
        this.uploadDate = uploadDate;
        this.thumbsUp = thumbsUp;
    }

    public void setUploadDate(String uploadDate) {
        this.uploadDate = uploadDate;
    }

    public int getThumbsUp() {
        return thumbsUp;
    }

    public void setThumbsUp(int thumbsUp) {
        this.thumbsUp = thumbsUp;
    }

    @Override
    public String toString() {
        return "\n postCode=" + postCode + ", postTitle=" + postTitle + ", title=" + title + ", writing=" + writing + ", memberNickname="
            + memberNickname + ", uploadDate=" + uploadDate + ", thumbsUp=" + thumbsUp;
    }
}

public int getPostCode() {
    return postCode;
}

public void setPostCode(int postCode) {
    this.postCode = postCode;
}

public String getPostTitle() {
    return postTitle;
}

public void setPostTitle(String postTitle) {
    this.postTitle = postTitle;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getWriting() {
    return writing;
}

public void setWriting(String writing) {
    this.writing = writing;
}

public String getMemberNickname() {
    return memberNickname;
}

public void setMemberNickname(String memberNickname) {
    this.memberNickname = memberNickname;
}

public String getUploadDate() {
    return uploadDate;
}

```

Figure 7 BoardVO



## MemberVO

```
package com.oopsw.model;
//회원정보 memberID, pw, member_nickname, name, birthdate, mobile_number, email, agreement
public class MemberVO {
    private String memberID;
    private String pw;
    private String memberNickname;
    private String name;
    private String birthdate;
    private String mobileNumber;
    private String email;
    private int agreement;

    public MemberVO() {}
    public MemberVO(String memberID, String pw, String memberNickname, String name, String birthdate,
        String mobileNumber, String email, int agreement) {
        super();
        this.memberID = memberID;
        this.pw = pw;
        this.memberNickname = memberNickname;
        this.name = name;
        this.birthdate = birthdate;
        this.mobileNumber = mobileNumber;
        this.email = email;
        this.agreement = agreement;
    }

    public void setUploadDate(String uploadDate) {
        this.uploadDate = uploadDate;
    }

    public int getThumbsUp() {
        return thumbsUp;
    }

    public void setThumbsUp(int thumbsUp) {
        this.thumbsUp = thumbsUp;
    }

    @Override
    public String toString() {
        return "\n postCode=" + postCode + ", postTitle=" + postTitle + ", title=" + title + ", writings=" + writing + ", memberNickname="
            + memberNickname + ", uploadDate=" + uploadDate + ", thumbsUp=" + thumbsUp;
    }
}

public int getPostCode() {
    return postCode;
}

public void setPostCode(int postCode) {
    this.postCode = postCode;
}

public String getPostTitle() {
    return postTitle;
}

public void setPostTitle(String postTitle) {
    this.postTitle = postTitle;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getWriting() {
    return writing;
}

public void setWriting(String writing) {
    this.writing = writing;
}

public String getMemberNickname() {
    return memberNickname;
}

public void setMemberNickname(String memberNickname) {
    this.memberNickname = memberNickname;
}

public String getUploadDate() {
    return uploadDate;
}
```

Figure 8 MemberVO

## TeamsVO

```
package com.oopsw.model;

public class TeamsVO {
    private String teamName;
    private String teamLogo;
    private String foundYear;
    private String hometown;
    private String ground;
    private String teamNickname;
    private int eplWins;
    private int facupWins;
    private int leaguecupWins;
    private int champsWins;
    private int europaWins;

    public TeamsVO() {}
    public TeamsVO(String teamName, String teamLogo, String foundYear, String hometown,
        String ground, String teamNickname, int eplWins, int facupWins, int leaguecupWins, int champsWins,
        int europaWins) {
        super();
        this.teamName = teamName;
        this.teamLogo = teamLogo;
        this.foundYear = foundYear;
        this.hometown = hometown;
        this.ground = ground;
        this.teamNickname = teamNickname;
        this.eplWins = eplWins;
        this.facupWins = facupWins;
        this.leaguecupWins = leaguecupWins;
        this.champsWins = champsWins;
        this.europaWins = europaWins;
    }

    public int getFacupWins() {
        return facupWins;
    }

    public void setFacupWins(int facupWins) {
        this.facupWins = facupWins;
    }

    public int getLeaguecupWins() {
        return leaguecupWins;
    }

    public void setLeaguecupWins(int leaguecupWins) {
        this.leaguecupWins = leaguecupWins;
    }

    public int getChampsWins() {
        return champsWins;
    }

    public void setChampsWins(int champsWins) {
        this.champsWins = champsWins;
    }

    public int getEuropaWins() {
        return europaWins;
    }

    public void setEuropaWins(int europaWins) {
        this.europaWins = europaWins;
    }

    @Override
    public String toString() {
        return "teamName=" + teamName + ", teamLogo=" + teamLogo
            + ", foundYear=" + foundYear + ", hometown=" + hometown + ", ground=" + ground + ", teamNickname="
            + teamNickname + ", eplWins=" + eplWins + ", facupWins=" + facupWins + ", leaguecupWins="
            + leaguecupWins + ", champsWins=" + champsWins + ", europaWins=" + europaWins;
    }
}

public String getTeamName() {
    return teamName;
}

public void setTeamName(String teamName) {
    this.teamName = teamName;
}

public String getTeamLogo() {
    return teamLogo;
}

public void setTeamLogo(String teamLogo) {
    this.teamLogo = teamLogo;
}

public String getFoundYear() {
    return foundYear;
}

public void setFoundYear(String foundYear) {
    this.foundYear = foundYear;
}

public String getHometown() {
    return hometown;
}

public void setHometown(String hometown) {
    this.hometown = hometown;
}

public String getGround() {
    return ground;
}

public void setGround(String ground) {
    this.ground = ground;
}

public String getTeamNickname() {
    return teamNickname;
}

public void setTeamNickname(String teamNickname) {
    this.teamNickname = teamNickname;
}

public int getEplWins() {
    return eplWins;
}

public void setEplWins(int eplWins) {
    this.eplWins = eplWins;
}
```

Figure 9 TeamsVO

## 2.6 Class Diagram

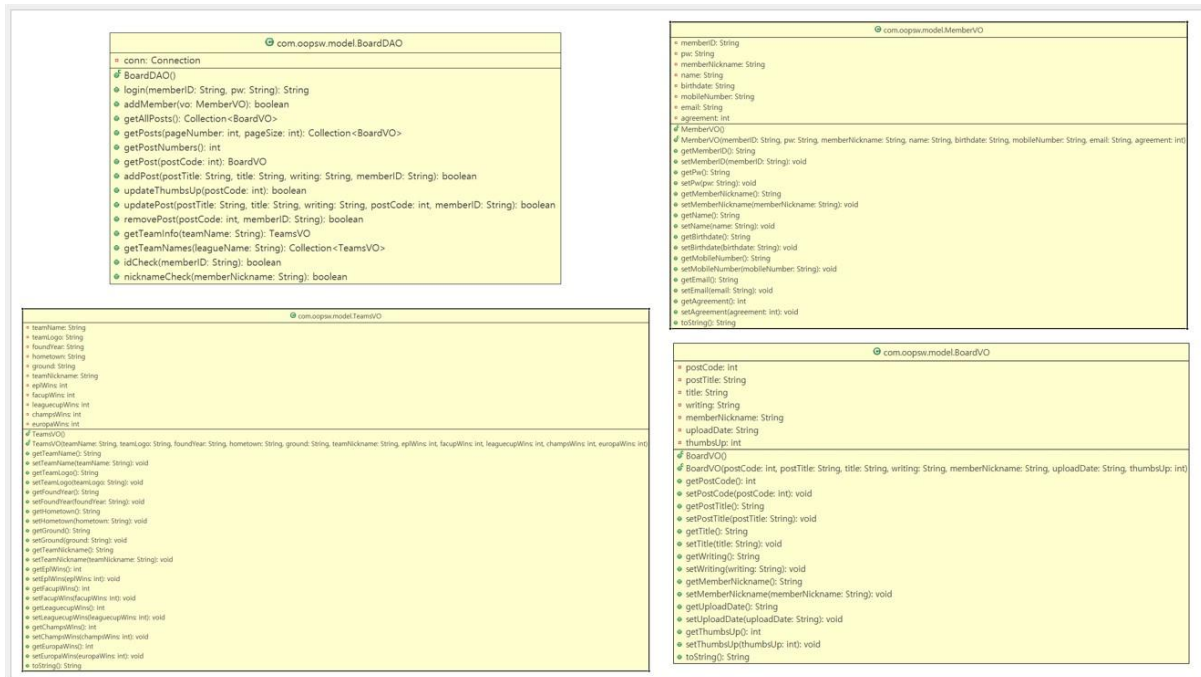


Figure 10 Class Diagram

## 2.7 Task List

Task	SQL
Log In	select member_nickname from members where member_id=? and pw=?
Sign up	insert into members(member_id, pw, member_Nickname, name, birthdate, mobile_number, eMail, agreement)"
Post Lists	select post_code, post_title, title, member_nickname, upload_date, thumbsup from (" select post_code, post_title, title, member_nickname, upload_date, thumbsup, row_number() over (order by post_code desc) as row_num from boards b, members m where m.member_id=b.member_id) where row_num > (?- 1)*? and row_num<=?*?
Get number of posts (boards)	select count(*) from boards
View post	select post_code, post_title, title, writing, member_nickname, upload_date, thumbsup from boards b, members m where m.member_id=b.member_id and post_code=?
Upload post	insert into boards(post_code, post_title, title, writing, upload_date, thumbsup, member_id) values(boards_seq.nextval, ?, ?, ?, sysdate, 0, ?)
Update post	update boards set post_title=?, title=?, writing=? where post_code=? and member_id=?
Delete post	delete from boards where post_code=? and member_id=?
Get team information	select team_name, team_logo, found_year, hometown, ground, team_nickname, epl_wins, facup_wins, leaguecup_wins, champs_wins, europa_wins from TEAMS where team_name=?
Check ID	select * from members where member_id=?
Check nickname	select * from members where member_nickname=?

**Table 2 Task List (Task, List)**

	HTML/JSP	Servlet	cmd	Action
Log In	login.html	Front Controller	loginAction	LoginAction()
Sign Up	signup.jsp		signupUIAction	SignupUIAction()
			signupAction	SignupAction()
Boards	boards.jsp		getPostAction	getPostAction()
View Post	posts.jsp		viewPostAction	viewPostAction()
	postsUpdate.jsp			
Upload Post	addPost.jsp		addPostUIAction	addPostUIAction()
			addPostAction	addPostAction()
Update Post	updatePosts.jsp		updatePostUIAction	updatePostUIAction()
			updatePostAction	updatePostAction()
Select Team	selectTeam.jsp		viewTeamsUIAction	viewTeamsUIAction()
Search Team's information	teamInfo.jsp		-	-
Delete Post	-		deletePostAction	deletePostAction()
Check ID	signup_id_check.jsp (using jQuery)		idCheckAction	IdCheckAction()
Check Nickname	signup_nickname_check.jsp (using jQuery)		nicknameCheckAction	NicknameCheckAction()

Table 3 Task List (HTML, Servlet, cmd, Action)

The screenshot shows a web application with a header bar containing navigation links and a user login status. The main content area is titled '축구인들의 모임' (Footballers' Gathering) and displays a list of posts. Each post entry includes a title, a brief description, the author's name, and the posting date. The posts are listed in a table-like format with alternating row colors. At the bottom of the page, there is a pagination bar with buttons for '이전' (Previous), '1', '2', '3', '4', and '다음' (Next).

Figure 11 Main UI\_Board

192.168.0.39:5432/99\_FinalProject/

## Login

[아직 회원이 아니신가요?](#)

Figure 12 UI\_Log In

192.168.0.39:5432/99\_FinalProject/controller?cmd=signupUIAction

## 회원가입

☐ 약관에 동의합니다

Figure 13 UI\_Sign Up

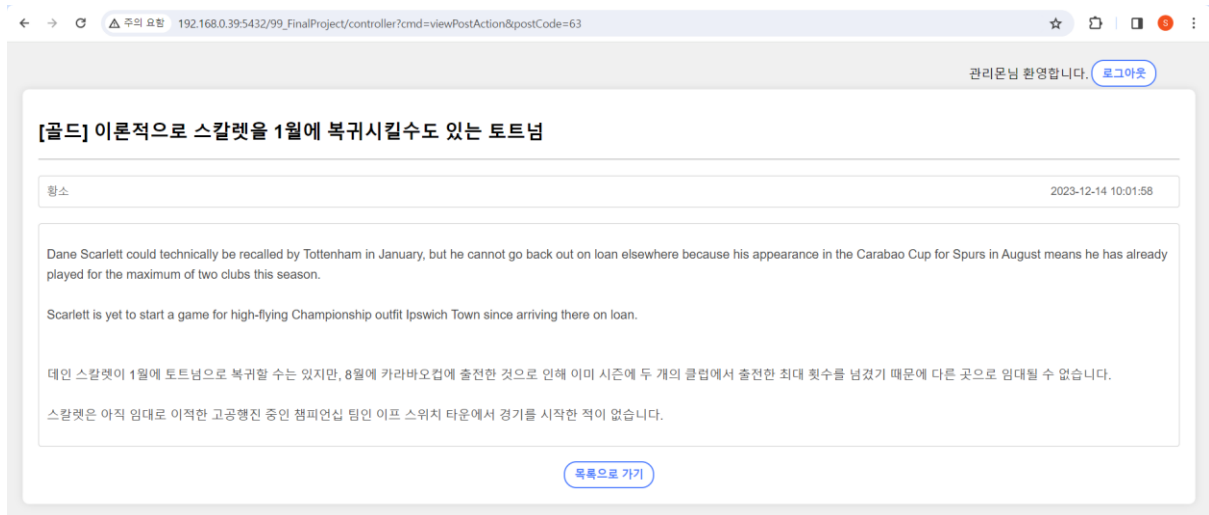


Figure 14 UI\_Post



Figure 15 UI\_Upload

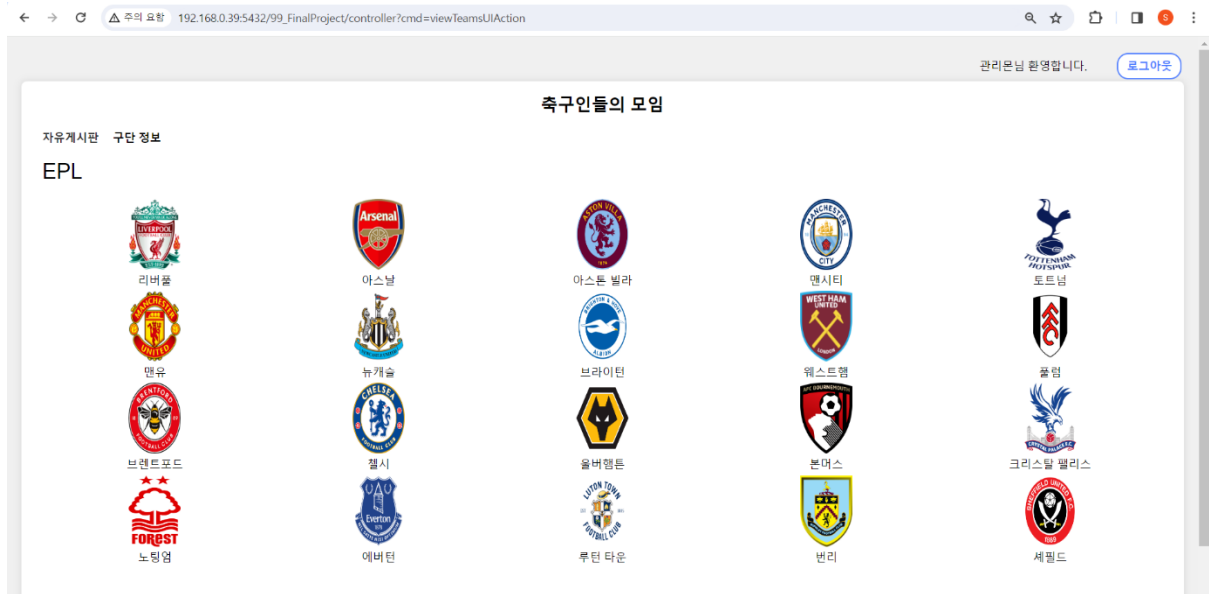


Figure 16 UI\_Select Team

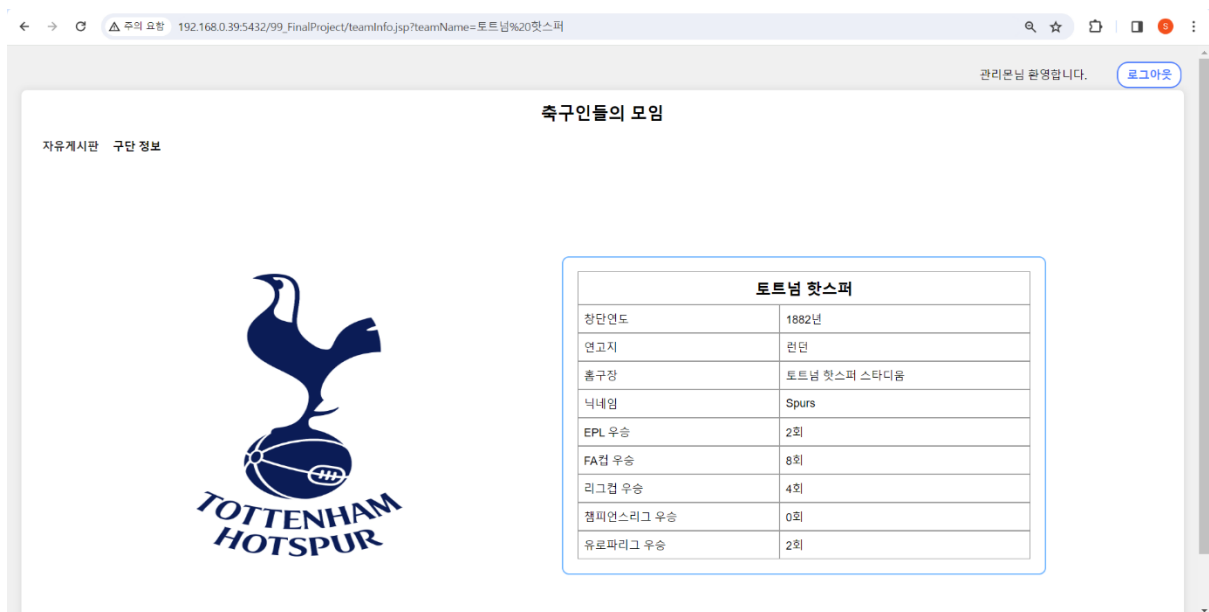


Figure 17 UI\_Team Information

### 3. 결론

#### 이상현:

'회원을 위한 방명록을 만들 것'이라는 요구사항을 바탕으로 프로젝트를 시작했습니다. 우선 Usecase Diagram, Class Diagram 그리고 Activity Diagram을 요구사항을 분석하며 작성했습니다. 이 과정으로 전체적인 시스템의 구조와 동작을 시각화하여 가독성 높은 분석 설계를 할 수 있었습니다.

그리고 UI명세서를 페이퍼 프로토타입으로 만들어 사용자가 상호작용할 화면을 시각화했습니다. 시각화하고 보니 어떤 데이터가 필요하고 어떻게 구성할지가 보였고 이를 데이터 모델링으로 정리하였습니다. 이렇게 정리한 것들을 바탕으로 SQL문을 작성하여 데이터 베이스를 구축하고 DAO와 VO파일을 작성함으로써 데이터베이스와의 상호작용을 위한 코드를 작성했습니다.

소프트웨어 아키텍처는 MVC패턴에 Action을 추가한 방향으로 잡음으로써 코드 구현에 방향성을 정했습니다. MVC 패턴을 사용하여 방명록을 만드는 것은 정말 어려운 일이었습니다. Model, View, Controller로 코드를 나누어야 한다는 개념을 알고 있었지만, 실제로 이를 적용하는 것은 예상보다 훨씬 어려웠습니다. 처음에는 모든 것을 한 번에 해결하려는 유혹에 빠졌는데, MVC 패턴의 핵심 역할을 되새기며 코드를 모듈화하고 각 부분을 분리하는 데 집중했습니다.

서버와의 연결에 대한 지식이 부족했고, 동기와 비동기 연결의 차이를 이해하지 못한 채 코드를 작성하는 데 많은 어려움을 겪었습니다. 그러나 나중에 구현하려던 기능이 비동기적 처리를 요구한다는 것을 깨달았고, 이에 대해 추가적인 학습을 통해 Ajax와 jQuery를 적용했습니다.

오류 메시지를 해석하고 디버깅하는 과정에서는 정말 많은 시간과 노력이 필요했습니다. 하지만 이를 통해 뿌듯함을 느끼고, 문제를 해결하는 데 필요한 기술을 익히며 큰 성장을 이루었습니다. 두 달 동안 배우고 프로젝트를 진행한 한 주간이 정말 소중한 경험이었습니다. 이제 코드가 완성되어 보니 각 부분의 역할이 더 명확하게 다가오는 것을 느끼고 있습니다. 이런 경험을 통해 더 나은 프로그래머로 성장하는 데 큰 도움이 되었습니다.



**소스코드:**

<https://github.com/SanghyeonLee97/First-Guestbook-Project>