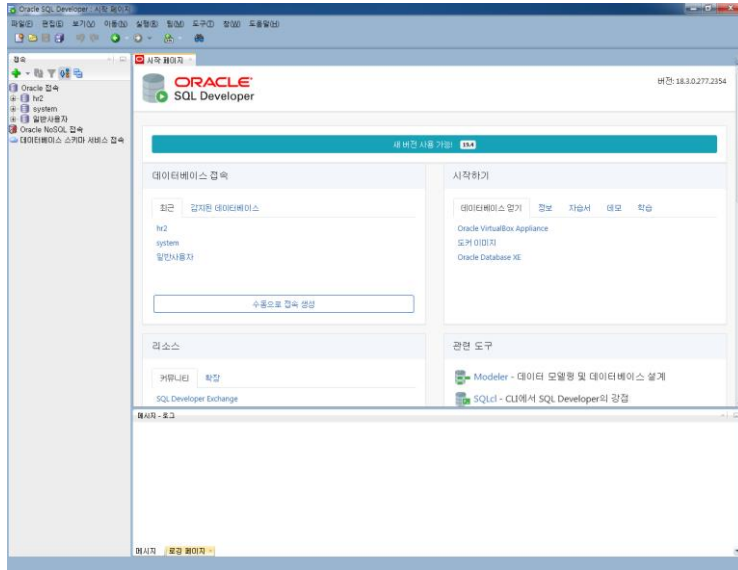


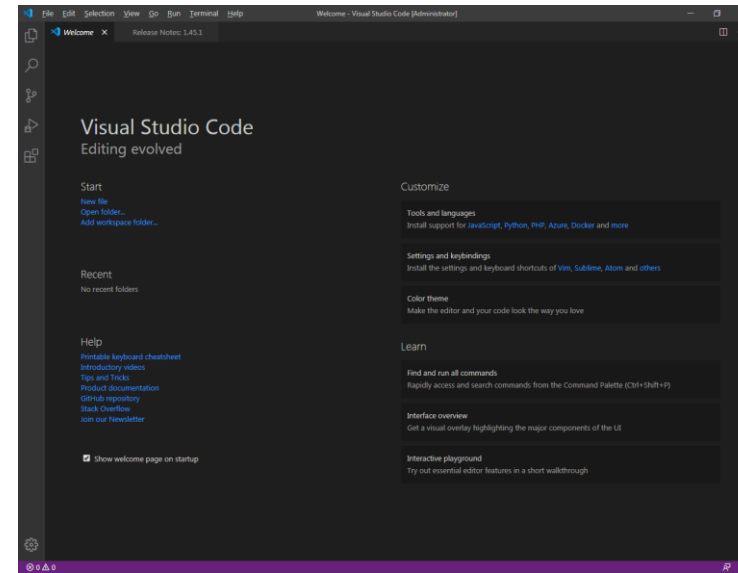
1. 개발 환경에 필요한 내용 명세

항목	내용
포트폴리오 명	MVC 모델2로 개발한 단답형 게시판
개발 언어	Java, JavaScript, Oracle DB, JSP
개발 도구	이클립스, Visual Studio Code, SQL Developer
개발 기간	6월 18일 ~ 6월 22일

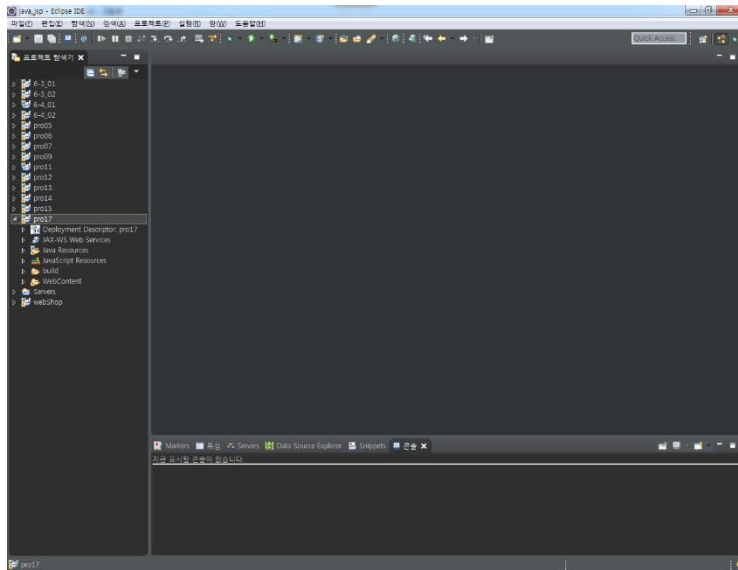
2. 소프트웨어 설치 및 개발환경 구축



Oracle DB



VSC



Eclipse

3. Github 저장소 구성

han6197/06-22-JSP- [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

han6197 / 06-22-JSP- Private [Unwatch](#) 1 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues](#) 0 [Pull requests](#) 0 [Actions](#) [Projects](#) 0 [Security](#) 0 [Insights](#) [Settings](#)

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

1 commit 1 branch 0 packages 0 releases

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

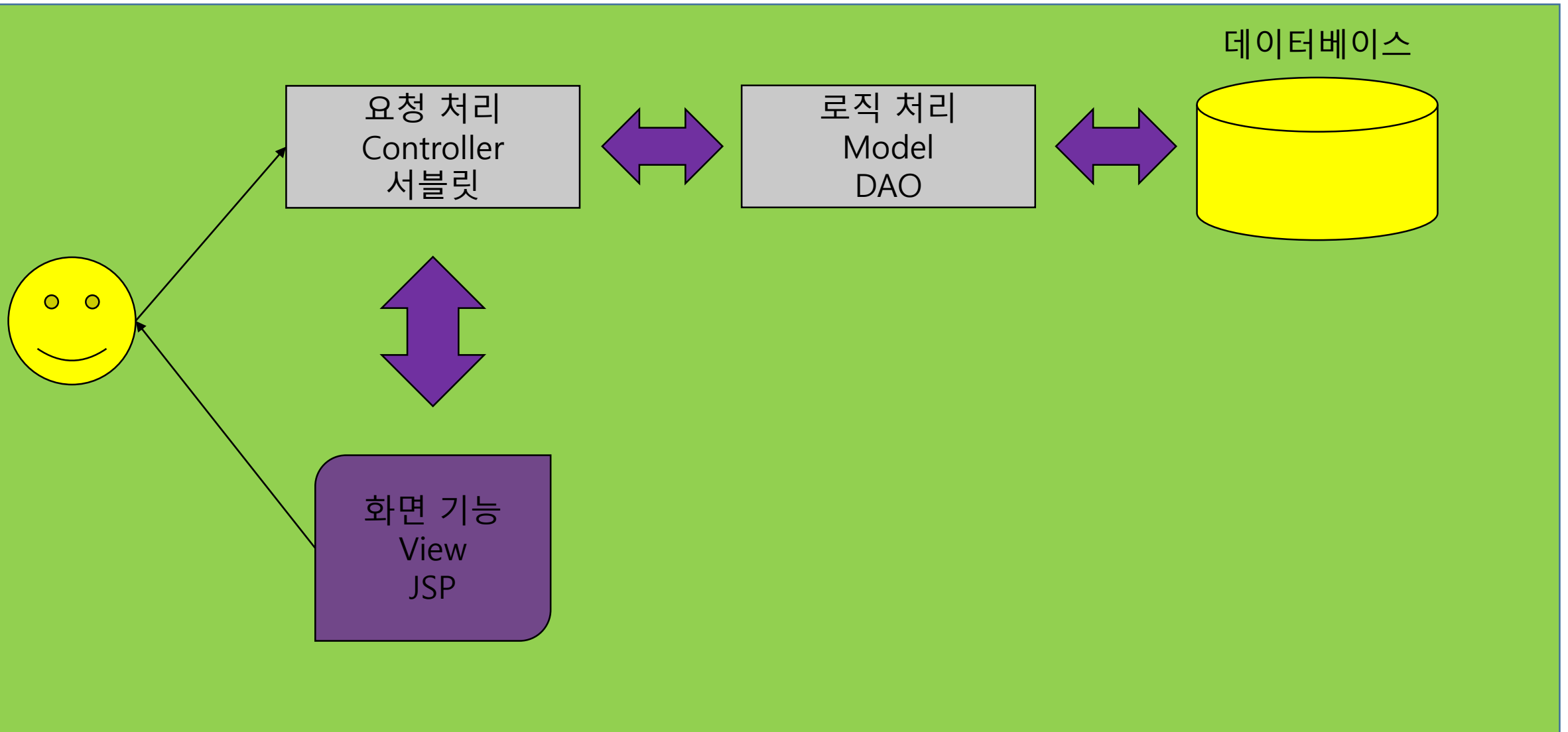
han6197 Add files via upload Latest commit 97e63c3 now

ArticleVO.java	Add files via upload	now
BoardController.java	Add files via upload	now
BoardDAO.java	Add files via upload	now
BoardService.java	Add files via upload	now
articleForm.jsp	Add files via upload	now
listArticles.jsp	Add files via upload	now
replyForm.jsp	Add files via upload	now
viewArticle.jsp	Add files via upload	now

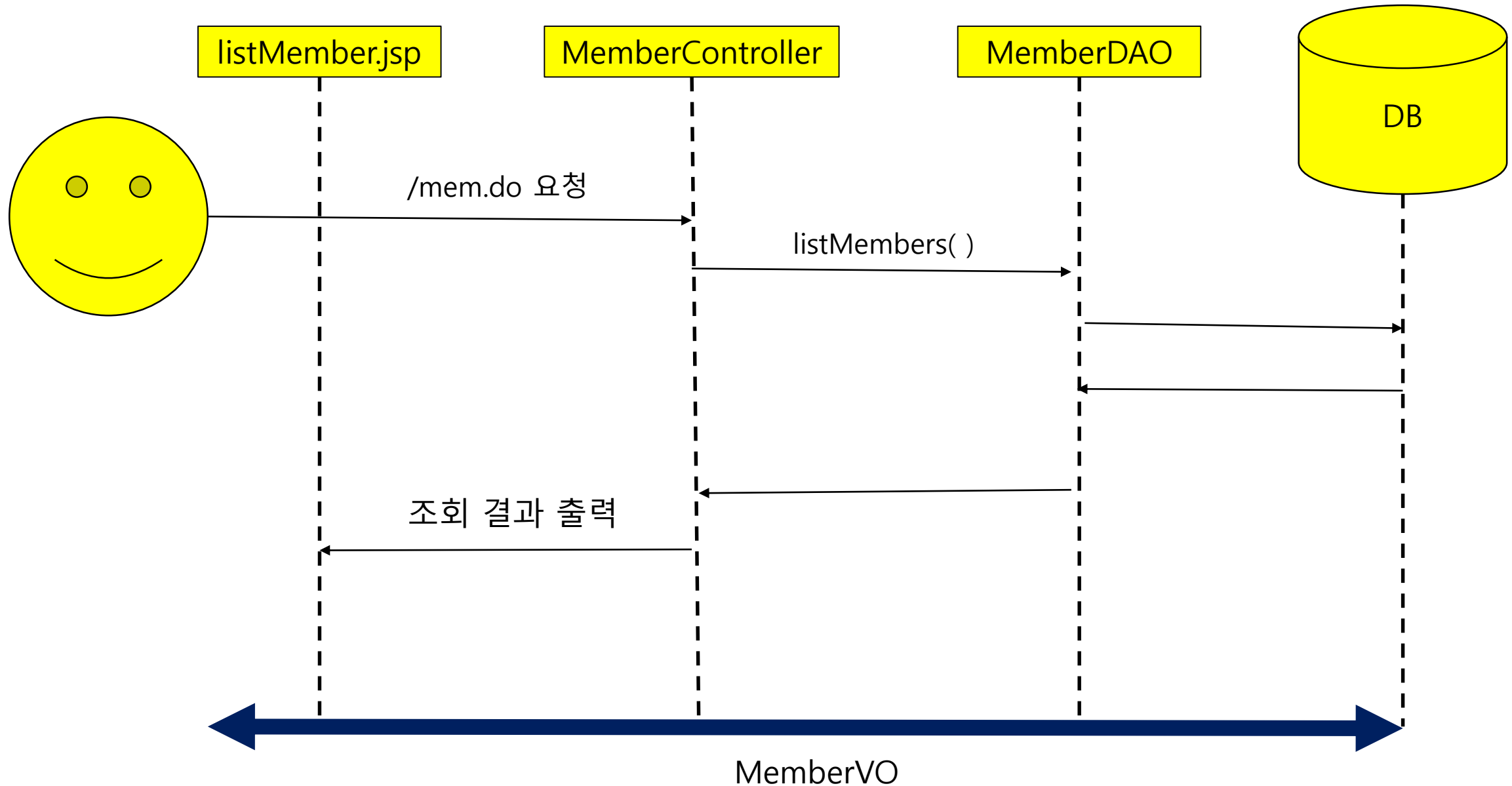
Add a README with an overview of your project. [Add a README](#)

© 2020 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

4. 공통 모듈 구성.



5. 공통 모듈 구현.



6. 컨넥션 풀을 이용하여 JDBC와 연결

Tomcat – context.xml

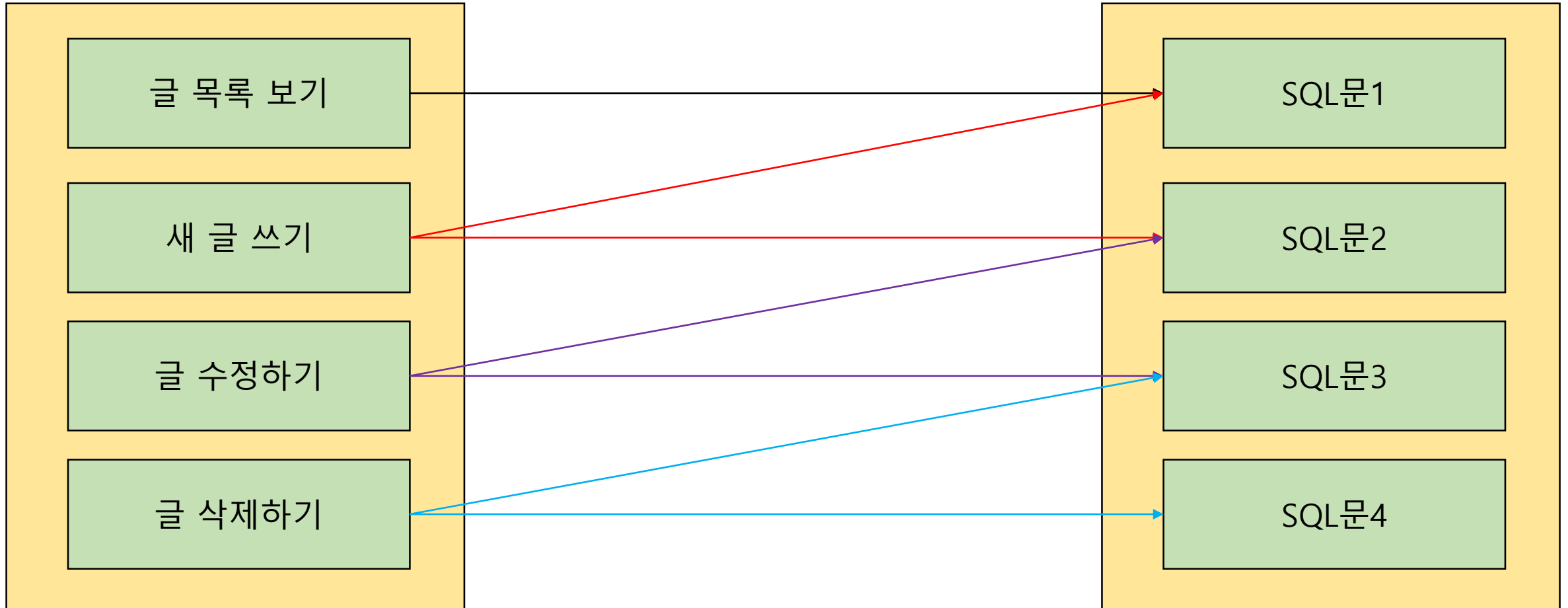
```
<Resource  
  name="jdbc/oracle"  
  auth = "Container"  
  type="javax.sql.DataSource"  
  driverClassName="oracle.jdbc.OracleDriver"  
  url="jdbc:oracle:thin:@localhost:1521:XE"  
  username="hr2"  
  password="12345"  
  maxActive="50"  
  maxWait="-1" />
```

7. 업무프로세스 맵 작성

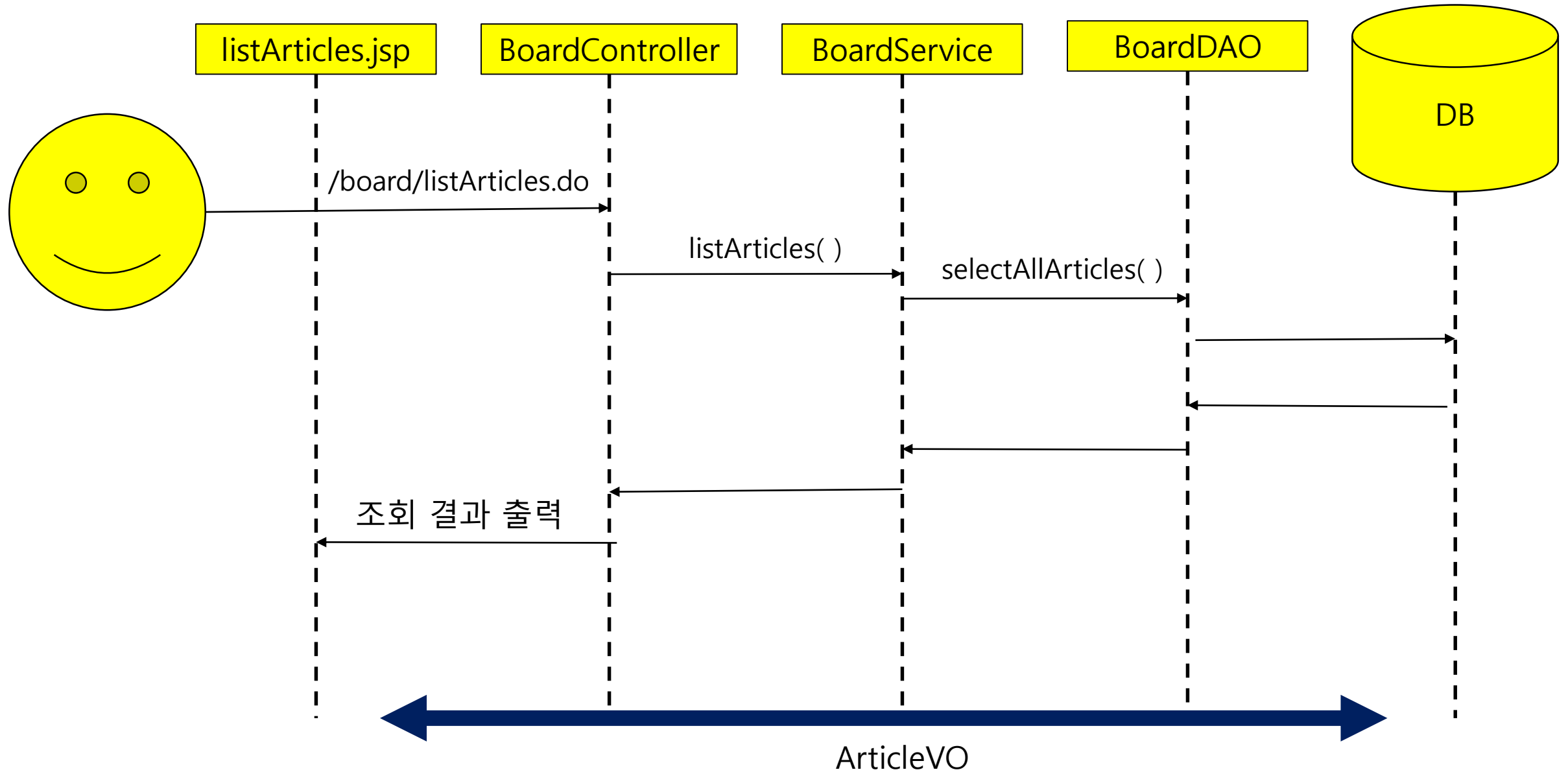
BoardService

답변형 게시판 기능의 모델 구조

BoardDAO



8. 업무프로세스 맵 구현



9. 취약성 제거할 수 있도록 프로그램 구현

```
try {
    List items = upload.parseRequest(request);
    for (int i = 0; i < items.size(); i++) {
        FileItem fileItem = (FileItem) items.get(i);
        if (fileItem.isFormField()) {
            System.out.println(fileItem.getFieldName() + "=" + fileItem.getString(encoding));
            articleMap.put(fileItem.getFieldName(), fileItem.getString(encoding));
        } else {
            System.out.println("파라미터명:" + fileItem.getFieldName());
            // System.out.println("파일명:" + fileItem.getName());
            System.out.println("파일크기:" + fileItem.getSize() + "bytes");
            // articleMap.put(fileItem.getFieldName(), fileItem.getName());
            if (fileItem.getSize() > 0) {
                int idx = fileItem.getName().lastIndexOf("\\");
                if (idx == -1) {
                    idx = fileItem.getName().lastIndexOf("/");
                }

                String fileName = fileItem.getName().substring(idx + 1);
                System.out.println("파일명:" + fileName);
                articleMap.put(fileItem.getFieldName(), fileName); // 익스플로러에서 업로드 파일의 경로 제거 후 r
                File uploadFile = new File(currentDirPath + "\\temp\\" + fileName);
                fileItem.write(uploadFile);

            } // end if
        } // end if
    } // end for
} catch (Exception e) {
    e.printStackTrace();
}
```

10-1. 테스트 수행, 게시물 생성

글목록창

localhost:8090/pro17/board/listArticles.do

글번호	작성자	제목	작성일
1	hong	상품후기입니다	2020. 6. 19
2	lee	[답변] 이용 후기입니다.	2020. 6. 19
3	kim	김유신입니다	2020. 6. 18
4	hong	안녕하세요	2020. 6. 18
5	lee	[답변] 상품 후기입니다	2020. 6. 18
6	hong	[답변] 답변입니다	2020. 6. 18
7	lee	[답변] 답변입니다	2020. 6. 18

1 2 3 4 5 6 7 8 9 10 next

글쓰기

1. 글쓰기 버튼 클릭

글쓰기창

localhost:8090/pro17/board/articleForm.do


새글 쓰기

글제목: 테스트 중입니다. 하나둘 셋, 하나둘 셋

글내용: 테스트 중입니다. 하나 둘 셋

이미지파일 첨부: 파일 선택 Koala.jpg

글쓰기 목록보기



2. 내용 작성 및 글쓰기 버튼 클릭

10-2. 테스트 수행, 게시물 추가



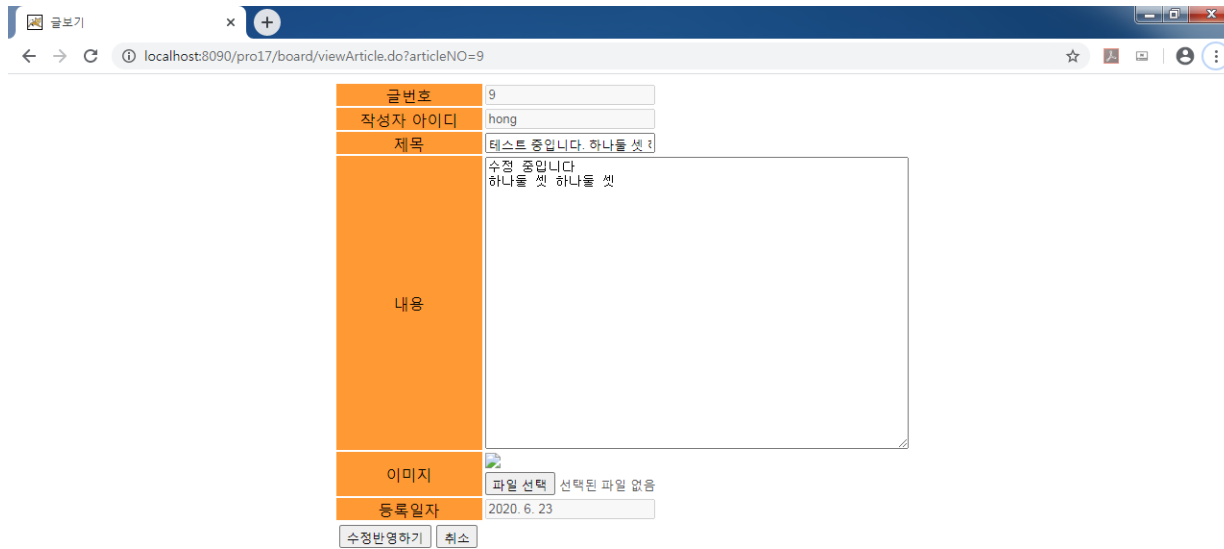
글번호	작성자	제목	작성일
1	hong	테스트 중입니다. 하나둘 셋 하나둘 셋	2020. 6. 22
2	hong	상품후기입니다	2020. 6. 19
3	lee	[답변] 이용 후기입니다.	2020. 6. 19
4	kim	김유신입니다	2020. 6. 18
5	hong	안녕하세요	2020. 6. 18
6	lee	[답변] 상품 후기입니다	2020. 6. 18
7	hong	[답변] 답변입니다	2020. 6. 18
8	lee	[답변] 답변입니다	2020. 6. 18

1 2 3 4 5 6 7 8 9 10 next

글쓰기

3. 게시물 추가 상태 확인

10-3. 테스트 수행, 게시물 수정



글번호: 9

작성자 아이디: hong

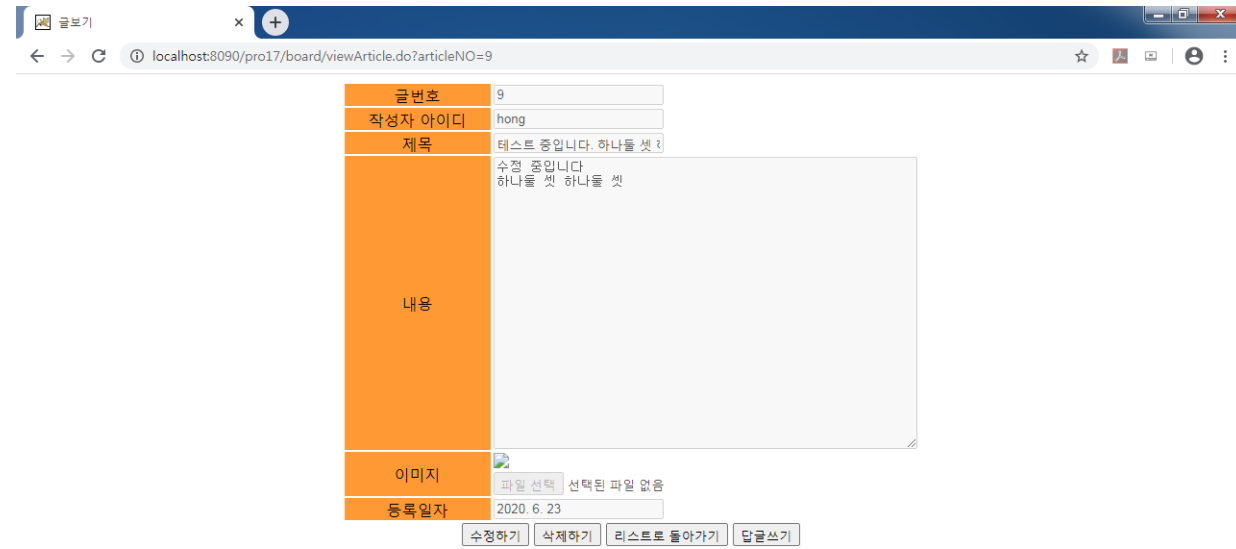
제목: 테스트 중입니다. 하나들 셋

내용: 수정 중입니다. 하나들 셋 하나들 셋

이미지: 선택된 파일 없음

등록일자: 2020. 6. 23

4. 게시물 조회 및 수정



글번호: 9

작성자 아이디: hong

제목: 테스트 중입니다. 하나들 셋

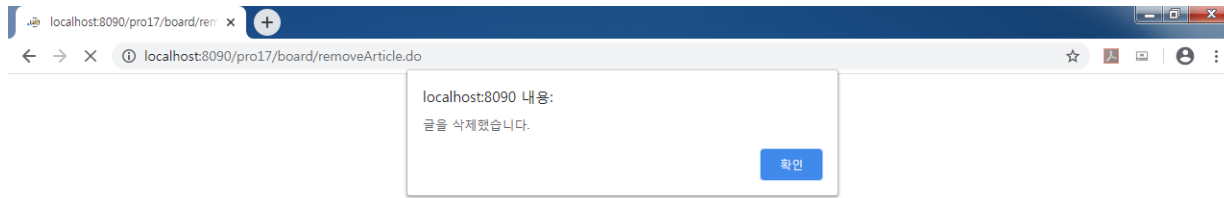
내용: 수정 중입니다. 하나들 셋 하나들 셋

이미지: 선택된 파일 없음

등록일자: 2020. 6. 23

5. 수정된 게시글 확인

10-4. 테스트 수행, 게시물 삭제



A screenshot of a web browser window at localhost:8090/pro17/board/listArticles.do. The page displays a table with 4 columns: 글번호 (Article Number), 작성자 (Author), 제목 (Title), and 작성일 (Date). The table contains 7 rows of data.

글번호	작성자	제목	작성일
1	hong	상품후기입니다	2020. 6. 19
2	lee	[답변] 이용 후기입니다.	2020. 6. 19
3	kim	김유신입니다	2020. 6. 18
4	hong	안녕하세요	2020. 6. 18
5	lee	[답변] 상품 후기입니다	2020. 6. 18
6	hong	[답변] 답변입니다	2020. 6. 18
7	lee	[답변] 답변입니다	2020. 6. 18

1 2 3 4 5 6 7 8 9 10 next

글쓰기

6. 게시물 삭제 버튼 클릭

7. 삭제된 상태 확인

11. 게시글이 많은 순으로 순위를 매기는 프로그램 구현

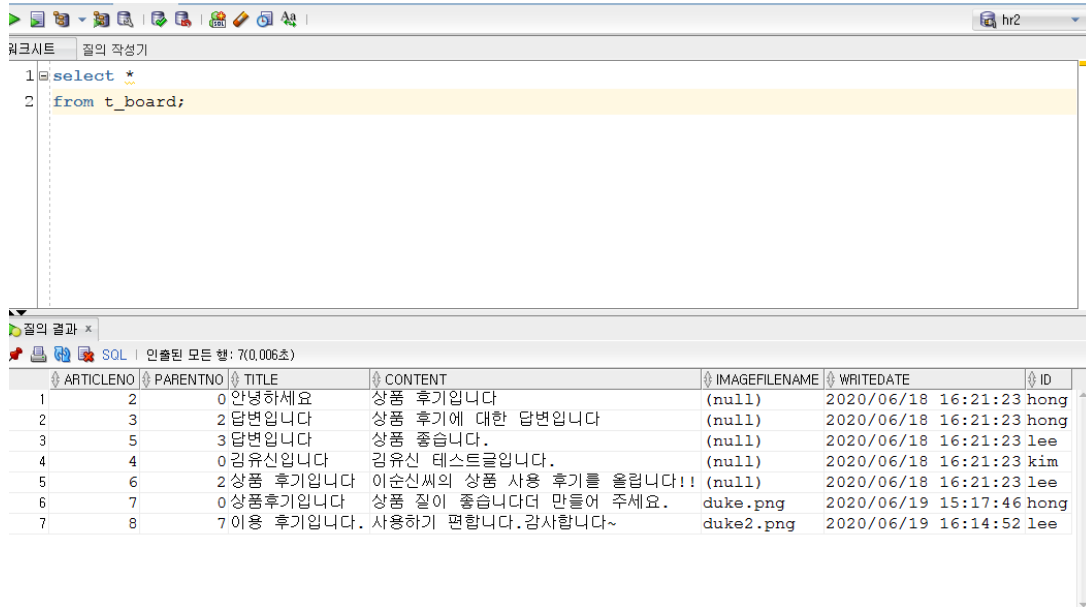
```
<td>작성자</td>
<td>글작성 많은 순서</td>
</br>
<c:choose>
  <c:when test="${articlesList==null}">
    <tr height="10">
      <td colspan="4">
        <p align="center">
          <b><span style="font-size: 9pt;">등록된 글이 없습니다.</span></b>
        </p>
      </td>
    </tr>
  </c:when>
  <c:when test="${articleList !=null}">
    <c:forEach var="article" items="${articlesList}" varStatus="articleNum">
      <tr align="center">
        <td width="10%">${article.id}</td>
        <td width="10%">${article.id}</td>
      </tr>
    </c:forEach>
  </c:when>
</c:choose>
</tr>
</table>
```

```
</body>
</html>
```

```
public List<ArticleVO> selectAllArticles2(){
    List<ArticleVO> articlesList = new ArrayList<ArticleVO>();
    try {
        conn = dataFactory.getConnection();
        String query = "SELECT id, count(id) from t_board group by id" + " order by count (id) desc";
        System.out.println(query);
        pstmt = conn.prepareStatement(query);
        ResultSet rs = pstmt.executeQuery();
        while(rs.next()) {
            String id = rs.getString("id");
            ArticleVO article = new ArticleVO();
            articlesList.add(article);
        }
        rs.close();
        pstmt.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return articlesList;
}
```

```
} else if (action.equals("/test.do")) {
    articlesList = boardService.listArticles();
    request.setAttribute("articlesList", articlesList);
    nextPage = "/board/test.jsp";
}
```

12. 개발된 배치 프로그램의 테스트를 수행



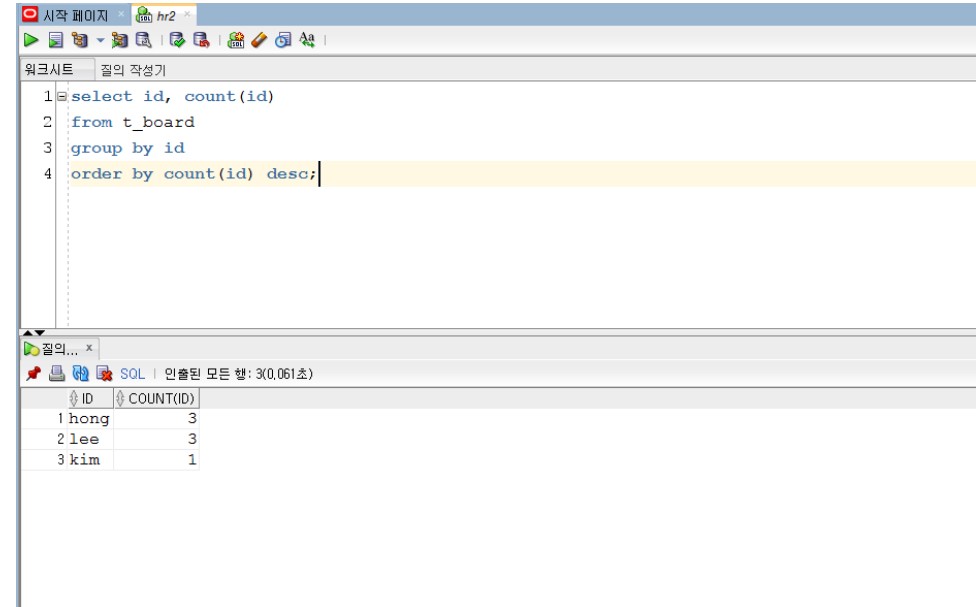
The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL statement:

```
1 select *
2 from t_board;
```

The results pane displays the output of the query, showing a table with 7 rows and 6 columns: ARTICLENO, PARENTNO, TITLE, CONTENT, IMAGEFILENAME, and WRITEDATE. The data is as follows:

ARTICLENO	PARENTNO	TITLE	CONTENT	IMAGEFILENAME	WRITEDATE
1	2	0 안녕하세요	상품 후기입니다	(null)	2020/06/18 16:21:23 hong
2	3	2 답변입니다	상품 후기에 대한 답변입니다	(null)	2020/06/18 16:21:23 hong
3	5	3 답변입니다	상품 좋습니다.	(null)	2020/06/18 16:21:23 lee
4	4	0 김유신입니다	김유신 테스트글입니다.	(null)	2020/06/18 16:21:23 kim
5	6	2 상품 후기입니다	이순신씨의 상품 사용 후기를 올립니다!!	(null)	2020/06/18 16:21:23 lee
6	7	0 상품후기입니다	상품 질이 좋습니다더 만들어 주세요.	duke.png	2020/06/19 15:17:46 hong
7	8	7 이용 후기입니다.	사용하기 편합니다. 감사합니다~	duke2.png	2020/06/19 16:14:52 lee

게시글 입력 현황



The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL statement:

```
1 select id, count(id)
2 from t_board
3 group by id
4 order by count(id) desc;
```

The results pane displays the output of the query, showing a table with 3 rows and 2 columns: ID and COUNT(ID). The data is as follows:

ID	COUNT(ID)
1 hong	3
2 lee	3
3 kim	1

ID당 게시글 입력 개수를 확인한 결과