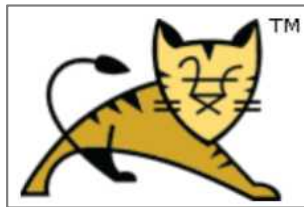


축구 예매 사이트 설계와 구현



소 속	컴퓨터정보계열
과 목 명	Server Side Programming I/ Web Programming
담당교수	김진숙
반	3반
팀 명	EXORDIUM
팀 장	류수빈
팀 원	박선재, 김희준
제 출 일	2019년 6월 26일

최종 요약본

JSP 프로젝트 - I Love Soccer

팀 명 : EXORDIUM

작성일 : 2019. 6. 25 04시

팀 원

박선재	류수빈	김희준
		

■ 사이트를 제작한 이유, 사이트의 목적(개요와 기대효과)

축구를 좋아하거나 흥미가 있는 사람들을 위한 사이트를 제작

축구 최신 뉴스를 알려주고, 유명한 축구 팀들을 알려줘 흥미를 가지게 만들었고

축구 티켓을 판매함으로 좀더 직접적으로 축구를 체험할 수 있도록 함.

또한 회원끼리 소통할 수 있는 게시판을 제작함으로써 축구를 모르는 사람과 축구를 좋아하는 사람 모두 이야기하며 축구의 폭을 넓힐 수 있음

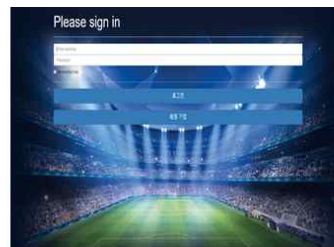
■ 프로젝트 발표 진행 순서

1. 프로젝트 개요.
2. 개발 일정 및 개발 환경.
3. 프로젝트 전체 구성도.
4. 프로젝트 전체 UI 화면.
5. 주요화면 소스 코드 및 DB 테이블 구성.
6. Reference.
7. 프로젝트 시현.

■ 구현 내용(UI 화면)



Home.jsp



Login.jsp



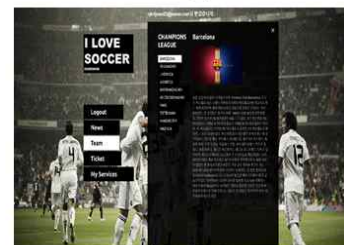
Join.jsp



Home1.jsp



News.jsp



Team.jsp



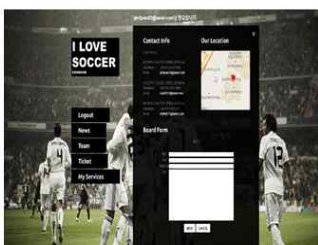
Ticket.jsp



Ticketing.jsp



Board.jsp

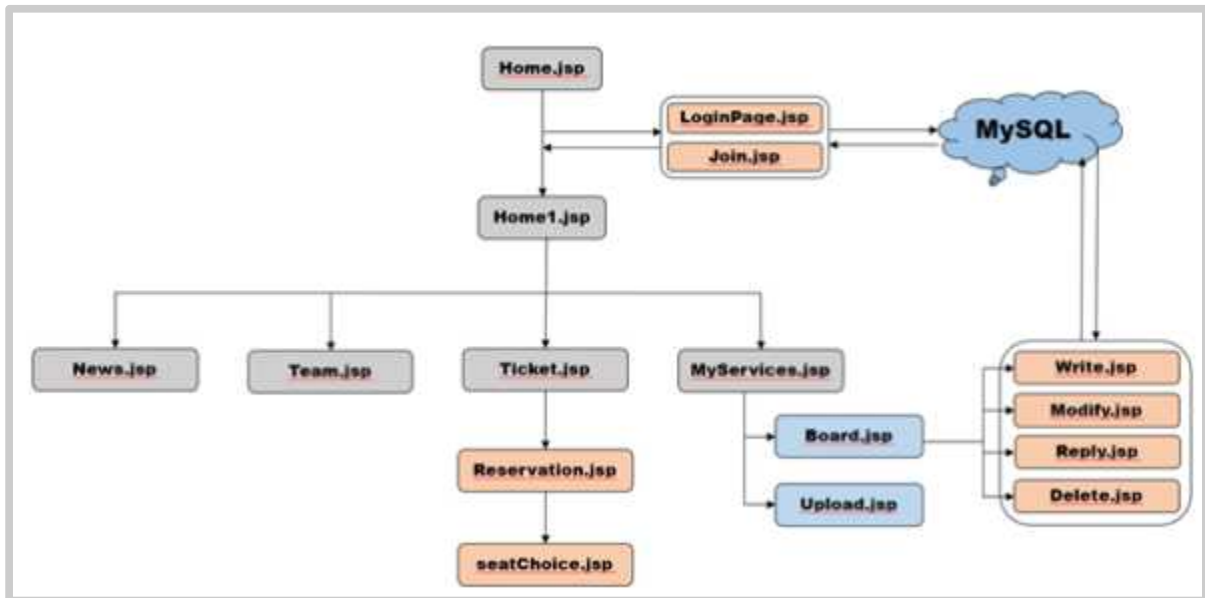


write.jsp



FileUpload.jsp

■ 사이트 맵



■ 참고자료

<http://bootstrapk.com/> - 부트스트랩 공식 홈페이지.

<https://dgblog.tistory.com/114> - MySQL DB 연동 방법.

<https://24getmenot7.tistory.com/273> - JSP 로그인, 회원가입 DB연동.

<https://www.codingfactory.net/10529> - HTML / CSS 적용 방법.

<https://seinarin.tistory.com/3> - JSP 게시판-1.

<https://seinarin.tistory.com/19?category=452459> - JSP 게시판-2.

https://www.slideshare.net/_ce/db-42499372 - 좌석 예매 DB 연동.

<http://webberstudy.com/html-css/html-3/form-2/> - input 정리

<https://loomio.kr/27> - check박스 사용법

<https://neptune297.tistory.com/60> - check박스 배열

<https://www.pngfind.com/freepng/real-madrid/> - 배경없는 이미지

<https://bulkywebdeveloper.tistory.com/m/19?category=683776> - 글자 깨짐 해결

<https://www.everdevel.com/jquery/click.php> - jQuery 클릭 이벤트

<https://ddo-o.tistory.com/63> - DB 조인

<https://yeop-blog.github.io/2017/10/02/2017-10-02-old-blog-post83/> - check 다중 입력

목차

1. 팀원소개	1
2. 프로젝트 개요.....	2
2.1 개요	
2.2 목적 및 필요성	
2.3 개발 일정	
3. 사용자 분석.....	3
3.1 사용자 분석	
3.2 사용자 요구 분석	
4. 사이트 설계.....	4
4.1 개요	
4.2 설계 원칙	
가) 디자인	
나) 콘텐츠	
다) 정보	
4.3 데이터베이스 설계	
가) ERD	
나) 논리적/물리적 데이터베이스 설계	
5. 사이트 구현	9
5.1 개발 환경 및 도구	
가) 개발 환경	
나) 개발 도구	
5.2 구현 프로그램 목록	
5.3 데이터베이스 구성	
5.2 인터페이스 및 기능 구현	
가) 메인	

- 나) 로그인
- 다) 회원가입
- 라) 게시판 글쓰기
- 마) 게시판 글 수정하기.
- 바) 게시판 글 삭제하기.

5.4 데이터베이스 구성

5.5 구현 프로그램 목록

6. 기대효과 및 맺음말.....	16
7. 팀원 소감.....	17
8. 참고서적 및 참고 사이트.....	18

[첨부]

- A. 프로젝트 계획서
- B. 주간 보고서
- C. 발표자료
- D. 소스코드

1. 팀원 소개



팀원이름(사진 포함)	담당역할	세부 수행 내용
 류수빈	팀장	로그인 및 회원가입 세션 기능 수행. 중간 보고서 작성. 바인더 관리.
 박선재	팀원	BootStrap 공부 및 적용. 게시판 기능 및 프로젝트 화면설계. 중간 보고서 작성
 김희준	팀원	BootStrap 공부 및 적용. 티켓 예매하기 및 주식 달기. 중간 보고서 작성.

2. 프로젝트 개요

2.1 개요







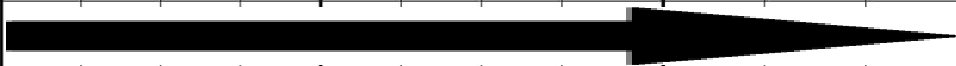
- **주제** : 축구 홈페이지(I Love Soccer)
- **선정 이유** : 팀원들이 축구를 좋아하기도 하고, 좋아하는 주제로 프로젝트를 했을 시 재미있게 과제를 수행 할 수 있을 것 같아서 선정하였음.
- **프로젝트 소개** : 축구를 잘 모르는 사람들이 쉽게 축구에 대한 정보를 알 수 있고 경기 티켓 예매를 할 수 있는 홈페이지를 제작하였습니다.

2.2 목적 및 필요성

목표 : 교우 간의 친목과 신뢰를 형성하여 프로젝트를 수행하여, 완성할 것입니다.

필요성 : 현재 아시아 축구가 월등하게 성장하는 추세이고, 현재 한국에서 축구의 인기가 나날이 증가하고 있습니다. 이 홈페이지를 만듦으로써 한국 사람들이 더욱 쉽게 축구를 접할 수 있다고 생각합니다.

2.3 개발일정

시기 단계별 내용	4월				5월				6월		
	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주
구성 및 기획											
자료수집 및 분석/체계화											
화면 설계 및 구현 - 메인화면 - 팀 소개 화면 - 티켓예매 화면 등											
기능 설계 및 구현 - 로그인 - 회원가입 등											
데이터베이스 설계 및 구현											
테스팅 및 프로젝트 완성											
보고서 작성 발표자료 작성											

3. 사용자 분석

3.1 사용자 분석

- 축구를 좋아하는 모든 사람들과, 축구에 관심이 생기기 시작한 모든 사람들이 사용 하고 정보를 접할 수 있는 축구 홈페이지.

3.2 사용자 요구 분석

- 축구 팀들에 대한 소개.
- 축구 관련 기사를 접할 수 있으면 좋겠다.
- 축구 경기 일정 및 티켓을 예매 할 수 있으면 좋겠다.
- 게시물을 작성하여 공지사항 및 건의사항 등 커뮤니티 공간이 있으면 좋겠다.

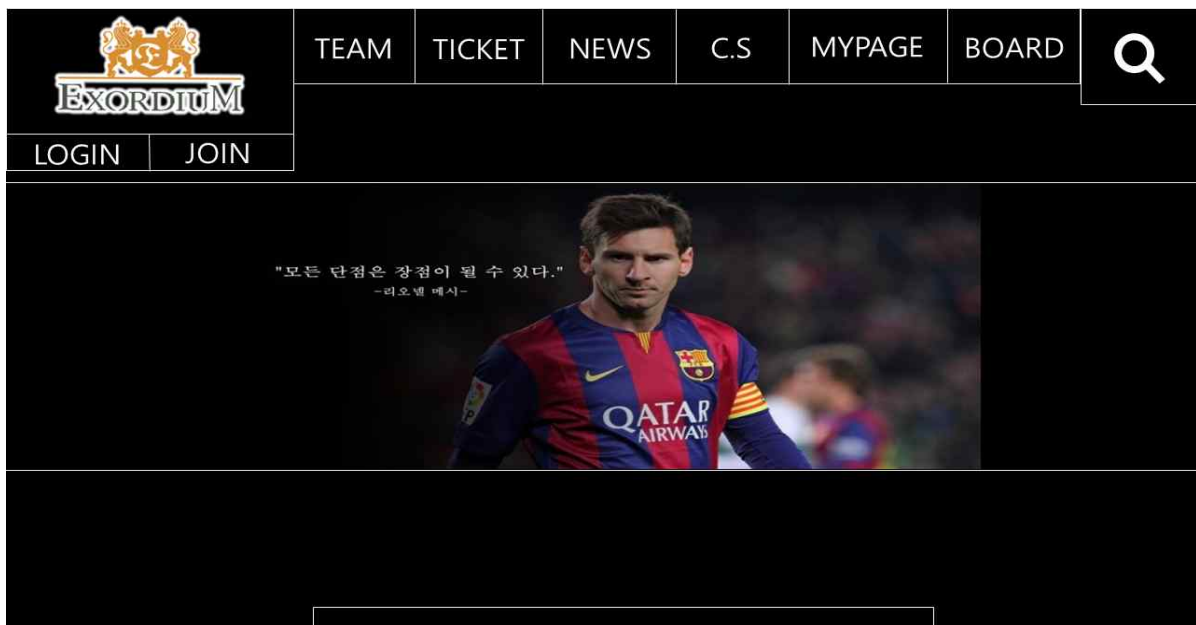
4. 사이트 설계

4.1 개요

- 실제 있는 각 축구 구단의 홈페이지와 축구게임 인 피파 온라인의 레이아웃 과 Contents를 참고하여 만들었음.

4.2 설계

가) 초기 디자인



■ 메인화면



Home.jsp

- 사용자는 회원가입 후 로그인을 해야지만 해당 페이지 내에 기능을 사용 할 수 있다.

■ 로그인 화면



Login.jsp

- 사용자는 회원이라면 로그인을 할 수 있고, 회원이 아니라면 회원가입 버튼을 통해 회원가입 화면으로 이동한다.

■ 회원가입 화면



Join.jsp

- 사용자는 다음 화면에서 회원가입을 할 수 있다.

- 로그인 후 메인화면.



Home1.jsp

- 로그인에 성공한 회원은 홈페이지 내에 기능을 사용할 수 있다.

- News 화면.



News.jsp

- 회원은 축구에 관련된 최신 기사 or 핫한 기사를 받아 볼수 있다.

- Team 화면.



Team.jsp

- 회원은 챔스에 관련된 팀에 대한 정보를 확인할수 있습니다.

- Ticket 예매 화면.



Ticket.jsp

- 로그인 한 회원은 티켓 화면에서 티켓을 예매 할 수 있다. 티켓을 예매 하게 되면 좌석 선택 화면으로 넘어 가게 된다.

- 좌석 예매 화면.



Ticketing.jsp

- 티켓 예매를 마친 회원은 좌석 선택 화면에서 좌석을 선택할 수 있다.

- 좌석 예매 화면.



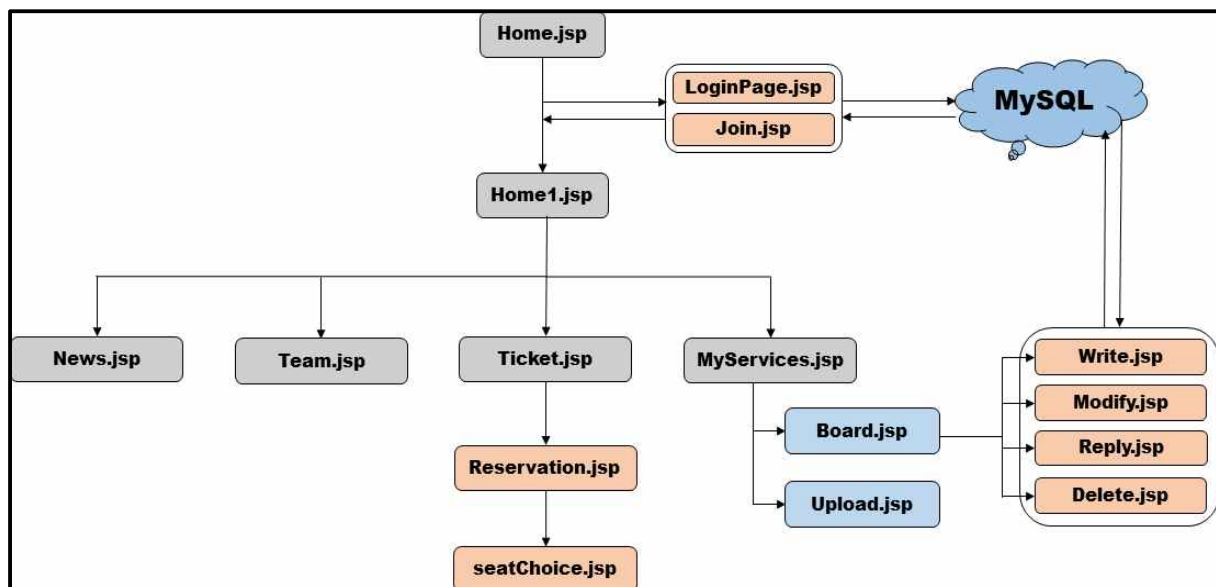
Borad.jsp

- 회원은 My Service 화면에서 게시글을 작성 할 수 있고, 수정, 삭제, 답글을 달 수 있고, 파일을 업로드 할 수 있다.

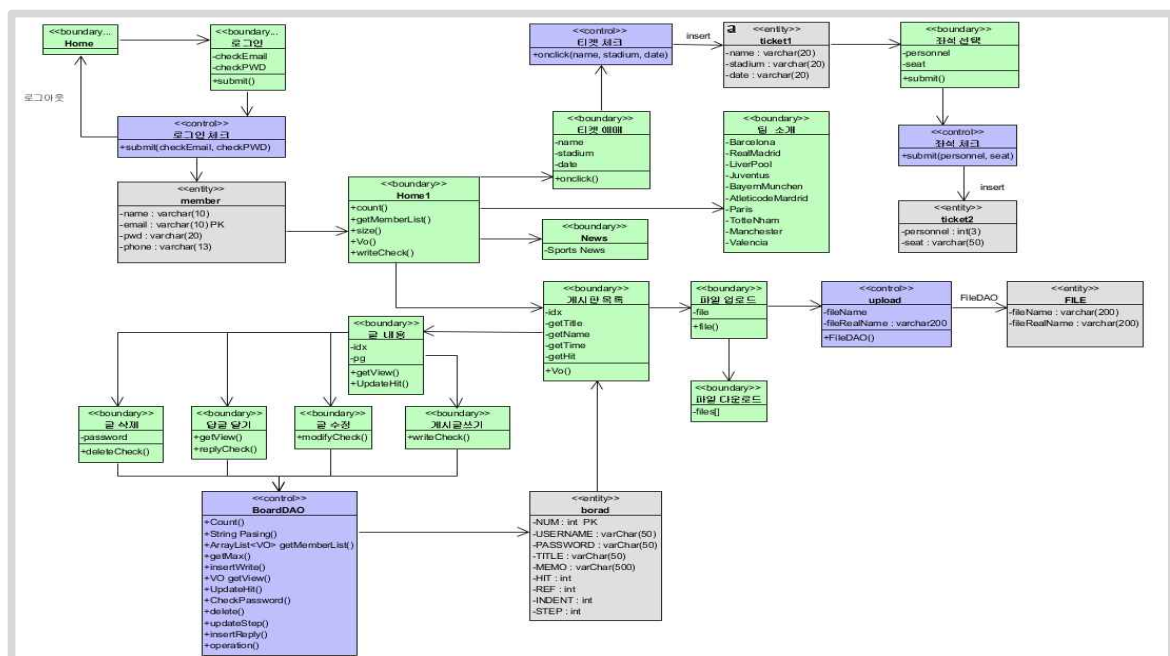
나) 콘텐츠

번호	콘텐츠	구성내용
1	축구 관련 기사	축구에 관련된 최신 기사를 작성한다.
2	축구 팀 소개	챔스에 출전하는 팀에 대한 소개글 을 작성한다.
3	축구 경기 예매	축구 경기를 예매 할 수 있도록 기능을 구현한다.
4	축구 좌석 선택	축구 좌석을 선택 할수 있도록 기능을 구현한다.
5	게시판	게시글 글쓰기, 수정, 삽입, 삭제 기능을 구현한다.

다) 정보(사이트 맵)



4.4 데이터베이스 설계 - ERD



5. 사이트 구현

5.1 개발 환경 및 도구

가) 개발 환경

▪ 시스템

구분	개발환경
서버	CPU : Intel(R) Core(TM) i7-8568U CPU @ 1.80GHz 2.00 GHz RAM : 16.0GB OS : Windows10 Pro
클라이언트	CPU : Intel(R) Core(TM) i7-8568U CPU @ 1.80GHz 2.00 GHz RAM : 16.0GB OS : Windows10 Pro

▪ 웹사이트

구분	내용
URL	http://localhost:8091/%EB%B0%95%EC%84%A0%EC%9E%AC/JSP%EA%B8%B0%EB%A7%90PROJECT/Home.jsp#!/
IP	192.168.0.14
Root Directory	/루트 디렉터리

나) 개발도구

구분	개발도구
Web Server	tomcat 9.0_40
Programming Language	Java8.0/JSP
Database	MySQL
소스 편집	Eclipse
영상 편집	PhotoShop CS3
플래시 제작	Flash

5.2 구현 프로그램 목록

디렉토리	기능	파일명	설명	작성자	비고
/	0. 메인	Home.jsp	메인화면	박선재	
code	1. 로그인	LoginPage.jsp	로그인 화면	류수빈	userDTO 빈에 login 데이터를 입력.
		LoginCheck.jsp	로그인 처리		
		userDTO.java	회원 데이터 저장 빈		userDAO 빈에 회원 정보 데이터 입력.
	2. 회원가입	join.jsp	회원 가입 화면		
	3. 뉴스	News.jsp	축구 관련 기사	박선재	
	4. 팀 소개	Team.jsp	대회에 출전한 팀 소개		
code	5. 티켓예약	Ticket.jsp	티켓 예매	김희준	
		TicketOK.jsp	티켓 예매 처리		
	6. 좌석예약	Ticketing.jsp	좌석 예매		
		TicketingOK.jsp	좌석 예매 처리		
	7. 게시판	Home1.jsp	게시판 메인 화면		
	8.글쓰기	write.jsp	게시글 쓰기	박선재	BoardDAO빈에서 insert 문을 사용하여 DB에 입력.
		writeOK.jsp	게시글 쓰기 처리		
	9.글수정	modify.jsp	게시글 수정		BoardDAO빈에서 update 문을 사용하여 DB정보 수정.
		modifyOK.jsp	게시글 수정 처리		
	10.게시글답글	reply.jsp	게시글 답글		BoardDAO빈에서updateStep을 사용하여 DB 정보 수정.
		replyOK.jsp	게시글 답글 처리		
	11.게시글삭제	delete.jsp	게시글 삭제		BoardDAO빈에서 Delete 문을 사용하여 DB정보 수정.
		deleteOK.jsp	게시글 삭제 처리		
	12.게시글목록	view.jsp	게시글 목록		BoardDAO빈에서 VO getView를 통해 수정,삭제, 삽입,추가한 모든 게시물 확인
	13.파일업로드	FileuploadAction.jsp	파일 업로드 처리		FileDAO빈에서 파일 업로드 처리 (C :드라이브)uplode폴더에 저장
images	이미지 모음			박선재	

5.3 데이터베이스 구성

[테이블명 : member]

번호	컬럼 ID	컬럼 설명	형태	길이	NULL	비 고
1	name	회원 이름.	varchar	10	NOT	PK
2	email	회원 이메일.	varchar	10		
3	pwd	회원 패스워드.	varchar	20		
4	phone	사용자 핸드폰 번호.	varchar	50		

[테이블명 : board]

번호	컬럼 ID	컬럼 설명	형태	길이	NULL	비 고
1	NUM	게시물 번호.	int	auto_increment	NOT	PK
2	USERNAME	사용자 이름.	varchar	50		
3	PASSWORD	사용자 패스워드.	varchar	50		
4	TITLE	게시글 제목.	varchar	50		
5	MEMO	게시글 내용.	varChar	50		
6	TIME	게시글 작성 시간.	varChar			
7	HIT	조회수.	int			
8	REF	글 생성 순서.	int			
9	INDENT	답글과 원글 구분.	int			
10	STEP	답글 출력순서.	int			

[테이블명 : FILE]

번호	컬럼 ID	컬럼 설명	형태	길이	NULL	비 고
1	fileName	파일 이름.	varchar	200		
2	fileRealName	파일 진짜이름.	varchar	200		

[테이블명 : ticket1]

번호	컬럼 ID	컬럼 설명	형태	길이	NULL	비 고
1	name	경기이름	varchar	200		
2	stadium	경기장이름	varchar	200		
3	date	경기일자	varchar	200		

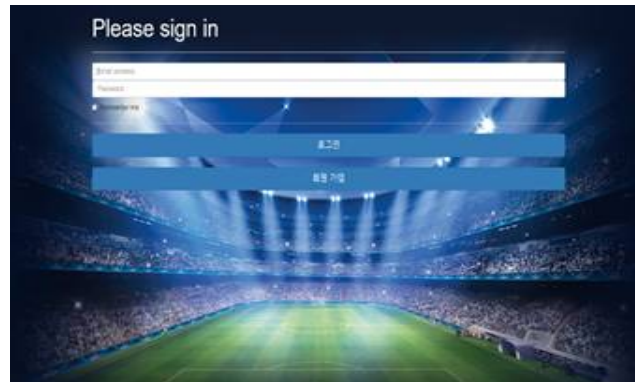
[테이블명 : ticket2]

번호	컬럼 ID	컬럼 설명	형태	길이	NULL	비 고
1	personnel	관람인원	int	3		
2	seat	좌석이름	varchar	50		

5.4 인터페이스 및 기능 구현**가) 메인**

화면명	메인화면	파일명	Home.jsp, Home1.jsp
<p>① 메인 화면에서는 로그인 버튼만 사용 가능하다. 로그인 화면에서 회원정보를 입력 후 로그인에 성공하게 되면 로그인 후 메인화면으로 이동한다. 로그인 후 메인화면에서는 다음 다섯 개의 버튼을 사용할 수 있다.</p> <p>② 입력절차</p> <ul style="list-style-type: none"> - Logout 버튼. - News 버튼. - Team 버튼. - Ticket 버튼. - My Service 버튼. 			

나) 로그인



화면명	로그인 입력	파일명	LoginPage.jsp, LoginCheck.jsp
<p>① 입력사항을 입력 후 “로그인”을 클릭하면 해당 사용자 확인작업 후 로그인 메인화면으로 이동한다. 입력한 정보가 틀린 경우는 아래와 같은 메시지 처리 후 로그인 화면으로 다시 이동한다.</p> <p>② 입력절차</p> <ul style="list-style-type: none"> - 해당 사용자 ID를 입력한다. - 비밀번호를 입력한다. 			

다) 회원가입



화면명	회원가입 화면	파일명	join.jsp, joinOK.jsp
<p>① 입력사항을 입력 후 “입력 확인”을 클릭하면 해당 사용자 확인작업 후 메인화면으로 이동한다. 입력란 정보가 비어있는 경우는 “입력란이 비어있습니다” 라는 메시지 처리를 한다.</p> <p>② 입력절차</p> <ul style="list-style-type: none"> - 해당 사용자 이름, 이메일, 비밀번호, 핸드폰 번호를 입력한다. - 입력확인 버튼을 누르면 회원가입 완료가 된다. - 다시 입력을 누르면 입력한 정보란이 초기화 된다. 			

라) 게시판 글쓰기



화면명	회원가입 화면	파일명	write.jsp, writeOK.jsp
<p>① 입력사항을 입력 후 "SEND"을 클릭하면 해당 입력란을 확인 후 게시글 목록으로 이동한다. 입력란 정보가 비어있는 경우는 "□□입력해주세요" 라는 메시지 처리를 한다.</p> <p>② 입력절차</p> <ul style="list-style-type: none"> - 해당 사용자 제목, 이름, 비밀번호, 내용을 입력한다. - SEND 버튼을 누르면 게시글 작성이 완료가 된다. - Cancel 버튼을 누르면 입력한 정보란이 초기화 된다. 			

마) 게시판 글 수정하기

수정

제목

이름

비밀번호

내용

화면명	회원가입 화면	파일명	modify.jsp, modifyOK.jsp
<p>① 입력사항을 입력 후 "SEND"을 클릭하면 해당 입력란을 확인 후 게시글 목록으로 이동한다. 입력란 정보가 비어있는 경우는 "□□입력해주세요" 라는 메시지 처리를 한다.</p> <p>② 입력절차</p> <ul style="list-style-type: none"> - 해당 사용자 제목, 비밀번호, 내용을 입력한다. - 수정 버튼을 누르면 게시글 수정이 완료가 된다. - 취소 버튼을 누르면 게시글 목록으로 이동한다. 			

바) 게시판 글 삭제하기

내 용	
글번호	3
조회수	1
이름	류수빈
작성일	2019-06-25 14:28:04
제목	컴퓨터정보과
<div>Test</div>	
<div>글쓰기 답글 목록 수정 삭제</div>	

삭제	
비밀번호	<input type="text"/>
<div>삭제 취소</div>	

화면명	회원가입 화면	파일명	delete.jsp, deleteOK.jsp
<p>① 삭제할 게시글을 클릭후 삭제 버튼을 누르면 삭제를 위한 비밀번호 입력창이 나오고, 그 입력 창에 비밀번호 입력 후 삭제 버튼을 누르면 "게시글이 삭제되었습니다." 라는 메시지가 출력하게 된다.</p> <p>② 입력절차</p> <ul style="list-style-type: none"> - 해당 사용자 제목, 비밀번호, 내용을 입력한다. - 삭제 버튼을 누르면 삭제 비밀번호 창이 뜬다. - 비밀번호를 입력후 삭제 버튼을 누르면 게시글이 삭제 되고 Home으로 이동한다. 			

6. 기대효과 및 맺음말

- 이 웹사이트가 구축되었을 때 축구에 대한 정보를 쉽게 접할 수 있다.
- 축구를 잘 모르는 사람들도 팀에 대한 소개가 있으므로 팀의 전통을 이해 할 수 있다.
- 축구 경기 예매하기를 통해 보다 쉽게 축구경기를 예약 할수 있다.

7. 팀원 소감

팀장 : 류수빈

jsp프로젝트를 처음에 받았을 때 어떻게 해야 될지 막막했습니다. HTML 코드에 Java 코드를 넣은 거라 크게 위화감은 없었지만 그래도 새로운 과목이고 하다 보니까 많이 서툴렀습니다. 공부는 jsp, bootstrap등등 공부하였고 화면구현을 다 한 뒤 DB연결하는 부분을 시도하는 중 오류가 생겨서 고치려고 팀원과 얘기도 해보고 인터넷도 찾아본 후 해결을 하였고 팀원과 함께 게시판을 만들며 오류가 생긴 부분 서로 도와주며 게시판도 완성했습니다. 처음에는 막막했지만 이렇게 팀원들과 같이 하다 보니 금방 해결하였고 프로젝트를 진행하면서 모르는 것들을 팀원들과 얘기를 나눠보고 해결을 하여서 더 머리 속에 남을 수 있었던 것 같습니다.

팀원 : 박선재

한 학기를 잘 마무리해서 너무 기분이 좋습니다.

공부하는데 어려움이 있었지만 좋아하는 주제로 프로젝트를 진행하여 완성도 있게 프로젝트를 수행 할 수 있었습니다. 동아리방에서 낮에는 수업 밤에는 프로젝트를 함께 최선을 다하는 팀원들의 노력하는 모습에 뿌듯하면서도 기특했습니다.

JSP 프로젝트를 진행하여 엄청난 양의 공부를 했습니다. 무엇 하나 쉽게 되는 것이 없었으며, 지치고 힘들었지만, 하나하나 해결 해 나가는 과정에 있어서 성취감을 얻었고, 각자 맡은 업무를 충실이 다 해준 팀원들이 있어서 프로젝트를 완성 할 수 있었습니다. 팀 프로젝트를 하며 가장 크게 느낀 것은 인간관계에서 일어나는 어떠한 문제들, 대립, 갈등 들을 어떻게 잘 해 쳐 나가야 되는지에 대한 방법을 터득하고 크게 보면 앞으로의 삶에 있어서 겪을 문제들을 사전에 미리 연습을 한 것이라고 생각합니다. 마지막으로, 이렇게 한 학기를 잘 마무리 하면서 또 한층 성장한 것 같은 생각이 듭니다.

팀원 : 김희준

이번 프로젝트는 JSP를 배워가는데 큰 도움이 되었습니다. 처음 그저 ppt를 보며 수업을 들을 때는 전혀 감이 잡히지 않았습니다. 쉬운 예제하나 만드는 것도 처음엔 어떻게 적고 그것이 어떻게 적용되는지 알 수 없었습니다. 시간을 들여서 완성했을 때 만든 모든 부분을 이해할 순 없었지만 딱 몇 가지 부분만은 어떻게 쓰이는지 이해했습니다. 그 후 JSP팀 프로젝트를 시작했을 때 맡은 티켓 예매 파트 처음엔 많이 막막했습니다. 차근히 품을 만들어 냈고 제일 고난이었던 jQuery, DB파트 초반엔 수업에 풀어봤던 내용들을 활용했었고 후반엔 ppt와 구글링을 많이 이용하여서 코드는 많이 복잡하지만 작동은하는 예매 파트를 만들어 냈습니다. 아직 DAO 자바빈을 이용하는 것이 부족하지만 더 알아내어 더 쉽고 간단하게 코드를 만들어보고 싶습니다.

그리고 잘 도와주었던 팀원들 덕분에 JSP는 다른 과목보다 좀 더 재밌게 프로젝트를 마칠 수 있었습니다

8. 참고서적 및 참고 사이트(예)

- ♦ <http://bootstrapk.com/> - 부트스트랩 공식 홈페이지.
- ♦ <https://dgblog.tistory.com/114> - MySQL DB 연동 방법.
- ♦ <https://24getmenot7.tistory.com/273> - JSP 로그인, 회원가입 DB연동.
- ♦ <https://www.codingfactory.net/10529> - HTML / CSS 적용 방법.
- ♦ <https://seinarin.tistory.com/3> - JSP 게시판-1.
- ♦ <https://seinarin.tistory.com/19?category=452459> - JSP 게시판-2.
- ♦ https://www.slideshare.net/_ce/db-42499372 - 좌석 예매 DB 연동.
- ♦ <http://webberstudy.com/html-css/html-3/form-2/> - input 정리
- ♦ <https://loomio.kr/27> - check박스 사용법
- ♦ <https://neptune297.tistory.com/60> - check박스 배열
- ♦ <https://www.pngfind.com/freepng/real-madrid/> - 배경없는 이미지
- ♦ <https://bulkywebdeveloper.tistory.com/m/19?category=683776> - 글자 깨짐 해결
- ♦ <https://www.everdevel.com/jquery/click.php> - jQuery 클릭 이벤트
- ♦ <https://ddo-o.tistory.com/63> - DB 조인
- ♦ <https://yeop-blog.github.io/2017/10/02/2017-10-02-old-blog-post83/> - check 다중 입력

Web Programming I 프로젝트 계획

1. 팀원 정보

학과	컴퓨터정보과		작성일	2019 년 03 월 27 일
과목	JSP 웹 프로그래밍			
팀명	EXORDIUM			
팀장 및 팀원	성 명	학 번	주 역할	
	류수빈	201868620	팀장 / 관리	
	박선재	201540537	팀원 / 서포트	
	김희준	201865258	팀원 / 서포트	

2. 과제 주제 및 선정 이유

- 주제 : 축구 홈페이지(I Love Soccer)
- 이유 : 다양한 팀에 의견 중 축구 홈페이지라는 의견이 나와 같이 논의해본 결과 구성도 탄탄하고, 아이디어가 신선해 선정하게 되었습니다.

3. 요구사항 명세(정보, 디자인, 구현기능, 구현 우선순위 등)

1. 비회원은 정보를 기재하고 회원가입을 합니다.
2. 회원가입을 한 회원은 로그인 화면으로 넘어갑니다.
3. 로그인한 회원은 메인 기능들을 사용 가능합니다.
4. 주요 기능 구현에서는 팀 소개, 일정, 관련 뉴스, 티켓 예매, 게시판, 검색 창 등을 사용할 수 있다.
5. 축구팀 소개에는 프리미어리그(K리그)에 관한 팀 소개를 넣는다.
6. 관련 뉴스는 게시판 형태 or 배너 형태 제작
7. 메뉴는 위쪽에 일렬로 정렬하고 그 뒤에 검색 창 위치, 로고는 왼쪽 위 모서리. 그 밑에는 관련 뉴스나, 승패를 나타냅니다.
8. 우선순위는 메인화면을 먼저 구현할 것입니다.

[첨부] : B. 주간 보고서

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(6주차)

일자	2019년 4월 6일	시간	12시
장소	웹동아리실	작성자	류수빈 (서명)
팀명(반)	Exordium	참석자	김희준, 박선재

프로젝트 관련 진행사항 및 계획

프로젝트 계획서 작성하고, 레이아웃 설계한 대로 HTML을 이용해 메인을 구현, 로그인화면 구현, Bootstrap을 이용해서 메인 사진 갤러리작성.

주간 학습과 회의 성과물(바인더에 첨부)

로그인 화면, 뉴스 화면(50% 완성), 메인화면

팀원별 학습내용(팀/팀원 학습계획에 근거하여 작성)

팀원명	학습내용/학습계획	확인
류수빈	HTML을 공부하고, 배운 토대로 메인화면 구현(30%)	lu
박선재	수업시간에 배운 로그인 화면&기능 구현	R
김희준	BootStrap을 이용해 뉴스 사진 슬라이드기능 (30%)	B

집단성찰

아직 모르는 게 많지만, 더욱 열심히 공부하여서 부족한 점을 채우도록 할 것입니다.

교수 한줄 평

DB 명칭 부분은 개념이나 Logic 부분이 중요할 것 같음. idea 생각해 보기!
예매도 Makeup

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(6주차)

일자	2019년 4월 19일	시간	4월 19일
장소	607 동아리방	작성자	김희준 (서명)
팀명(반)	EXODIUM	참석자	박선재, 류수빈

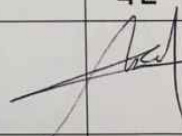
프로젝트 관련 진행사항 및 계획

학습활동 팀 + 개인별 잘못 되었거나 부족한 점들을 수정하고 바인더에 첨부
중간고사 대비

주간 학습과 회의 성과물(바인더에 첨부)

수정된 팀, 개인 학습활동지

팀원별 학습내용(팀/팀원 학습계획에 근거하여 작성)

팀원명	학습내용/학습계획	확인
박선재	팀 학습활동 계획에서 부족한 점들을 수정하여 인쇄	
김희준	개인 학습활동 계획 수정	(2)
류수빈	개인 학습활동 계획 수정	류수빈

집단성찰

학습활동 수정을 제외한 주간 학습을 중간고사 공부로 대체

교수 한줄 평

왜 6주차에 두개였어?!

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(주차)

일자	2019년 4월 19일	시간	5월 10일
장소	607 동아리방	작성자	박선재 (사명)
팀명(반)	EXODIUM	참석자	김희준, 류수빈

프로젝트 관련 진행사항 및 계획

부트스트랩을 이용하여
메인 화면인 Home.jsp와 Team.jsp를 만들어

~~~~~ 제대로 모양을 만들어 개편으로 정리함

## 주간 학습과 회의 성과물(바인더에 첨부)

주간 학습 부트스트랩 심의 학습 and 적용

~~~~~  
마무리

팀원별 학습내용(팀/팀원 학습계획에 근거하여 작성)

팀원명	학습내용/학습계획	확인
박선재	Bootstrap 을 활용한 웹 개발 연습	
김희준	Home.jsp의 뉴스들의 텍스트 채움, 이미지 슬라이드(부트스트랩) 적용	
류수빈	Bootstrap 공부	

집단성찰

Bootstrap 공부를 통한 웹 개발에 대한 자신감 을 얻었습니다.

교수 한줄 평

함께 bootstrap 공부한거니? ~~~~~ 다음 원서에 자세히 설명해주시기 바란다

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(10주차)

일자	2019년 5월 10일	시간	오후 1시
장소	웹동아리실	작성자	류수빈 (서명)
팀명(반)	Exordium	참석자	김희준, 박선재


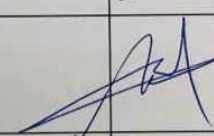

프로젝트 관련 진행사항 및 계획

화면 구성은 티켓예매 제외하고 다 한 상태이고, DB 연동을 해서 sql문으로 회원가입 화면에 기재한 데이터 DB에 삽입 가능, 로그인할 때 회원의 정보를 불러와 일치하는지 확인 후 로그인이 완료되는 것까지 구현.
회원가입 할 때 이메일 중복체크와 티켓예매 화면을 구현할 것이다.

주간 학습과 회의 성과물(바인더에 첨부)

DB연동, 화면 구성, 회원가입, 로그인

팀원별 학습내용(팀/팀원 학습계획에 근거하여 작성)

팀원명	학습내용/학습계획	확인
류수빈	DB 연동과 회원가입, 로그인 기능 구현	
박선재	Bootstrap으로 화면 구현, DB 연동 공부	
김희준	Bootstrap을 공부하고 화면 구성 서포트, 축구 선수 자료 수집, DB 연동 공부	

집단성찰

수업시간에 배운 DB Connection을 토대로 데이터베이스 연동을 이해함으로써 JSP개발 수준이 한 단계 높아진 것 같다.

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(11주차)

일자	2019년 5월 17일	시간	5월 17일
장소	607 동아리 방	작성자	김희준 (서명)
팀 명(반)	EXODIUM	참석자	박선재, 류수빈

프로젝트 관련 진행사항 및 계획

DBCP 과제 완료 및 체육대회

주간 학습과 회의 성과물(바인더에 첨부)

DBCP과제

팀원 별 학습내용(팀/팀원 학습계획에 근거하여 작성)

팀원 명	학습내용/학습계획	확인
박선재	DBCP를 사용자 관리 프로그램(팀 과제)에 연동시키기	Asd
김희준	DBCP를 listdb.jsp 에 연결 시키기 성공	㉠
류수빈	myspl과 연결에 서포팅 및 DBCP 공부	류수빈

집단 성찰

DB를 연동 하는 것에 대해 좀더 많은 것을 깨달음
DBCP로 코드의 양을 줄이는 법을 알았다.

교수 한 줄 평

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(12주차)

일자	2019년 5월 24일	시간	5월 24일
장소	607 웹 동아리실	작성자	박선재 (서명)
팀 명(반)	EXORDIUM	참석자	류수빈, 김희준

프로젝트 관련 진행사항 및 계획

2019.05.22 수요일. JSP 프로젝트 중간 발표를 하였고, 이후 교수님께서 말씀하셨던 프로젝트 파일 정리 와 게시판 만들기(사진 업로드가 가능하도록) 를 향후 어떻게 공부를 하고 만들지 논의 하였음.

주간 학습과 회의 성과물(바인더에 첨부)

교수님의 말씀하셨던 요청사항(보완해야 할 부분)

1. 주석 달기.
2. 게시판 (사진 업로드 기능) 구현
3. 프로젝트 파일 구분하여 정리
4. 코드에 대한 공부.

팀원 별 학습내용(팀/팀원 학습계획에 근거하여 작성)

팀원 명	학습내용/학습계획	확인
박선재	팀 학습활동 계획서에 부족한 점들을 수정하여 인쇄하였음.	(서명)
김희준	개인 학습활동 계획 수정	김희준
류수빈	개인 학습활동 계획 수정	(서명)

집단 성찰

중간 발표를 함으로써 프로젝트의 보완해야 할 부분을 확인 하였음.

교수 한 줄 평

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(13주차)

일자	2019년 5월 29일	시간	5월 29일
장소	607 동아리 방	작성자	박선재 (서명)
팀 명(반)	EXODIUM	참석자	김희준, 류수빈

프로젝트 관련 진행사항 및 계획

작성일: 2019.5.29 현재 프로젝트 진행 사항은 게시판 구현을 하고 있습니다.

팀원 박선재 와 팀장 류수빈 은 MySQL을 이용한 게시판 구현을 진행 중에 있고, 팀원 김희준 은 티켓 예약 구현을 진행 중입니다.

(특이사항 : 팀장 류수빈은 6월 일본어 공부로 인해 바빠질 예정.)

앞으로의 계획은 게시판 구현 완성과 동시에 다시 코드를 완벽히 이해하기 위해 다시 복습할 예정.

주간 학습과 회의 성과물(바인더에 첨부)

게시판 구현 진행 사항

팀원 별 학습내용(팀/팀원 학습계획에 근거하여 작성)

팀원 명	학습내용/학습계획	확인
박선재	JSP 게시판 구현 및 세션 공부.	(서명)
김희준	JSP 티켓 예매 구현 및 공부.	(서명)
류수빈	JSP 게시판 구현 및 세션 공부.	류수빈

집단 성찰

이제 까지 주간 보고서 작성시 부족했던 부분을 고쳐 나가기로 하였고, 과제를 좀 더 꼼꼼하게 공부해 나가기로 다짐 하였습니다.

2019학년도 1학기

웹프로그래밍/Server Side Programming

주간 보고서(14주차)

일자	2019년 6월 7일	팀 명(반)	EXODIUM
장소	607 동아리 방	작성자	김희준 (서명)
주제	K리그 홈페이지 제작		




프로젝트 관련 진행사항 및 계획

Ticket 예매 사이트 만들기 시작
메인 메뉴 리모델링

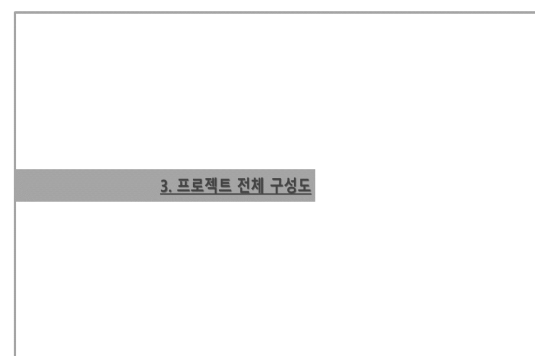
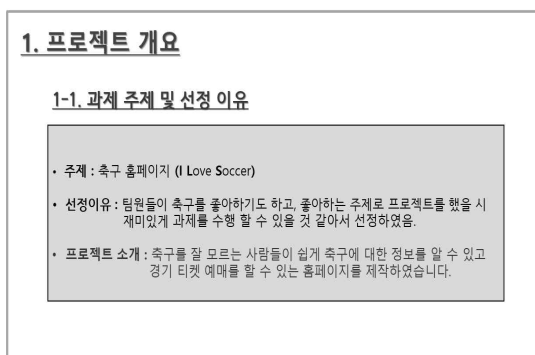
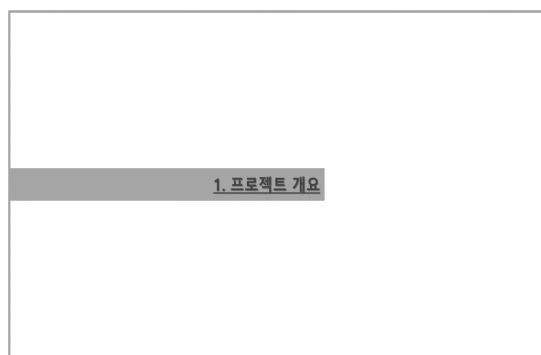
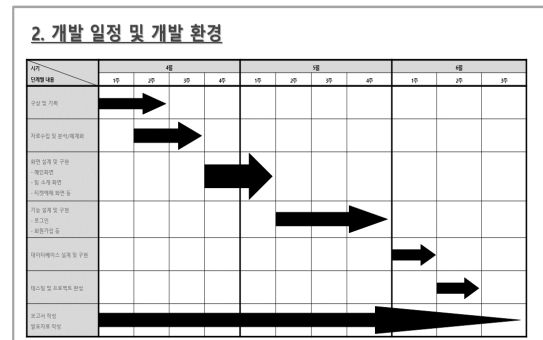
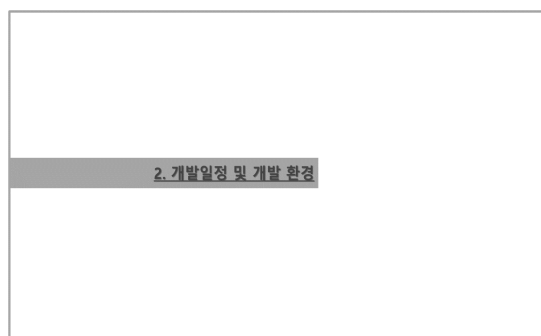
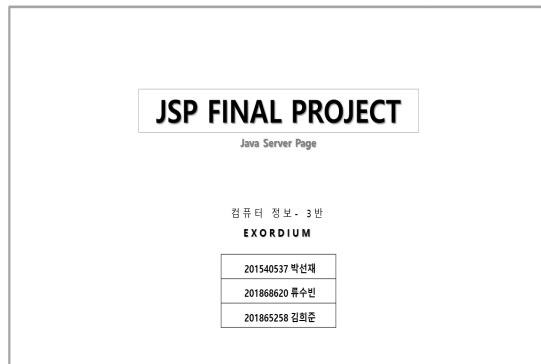
주간 학습 성찰(회의 성과물은 바인더에 첨부)

Main.jsp
Ticket.jsp, TicketOK.jsp

팀원별 학습내용(팀/팀원 학습계획에 근거하여 작성)

참석 팀원명	학습내용/학습계획	확인
박선재	Main.jsp 새롭게 구성	
김희준	티켓 예매 사이트 폼 만들기, DB와 sql문 연동하기위해 공부 중	
류수빈	DBCP 잘 알아볼 수 있도록 공부와 수정	

[첨부] : C. 발표 자료



5. 프로젝트 주요 화면 및 소스코드 설명

```

public int getCnt() { //getCnt() : 관리자 페이지에서 모든 게시물 개수를 반환하는 기능
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
    int cnt = 0;

    try {
        pstmt = con.prepareStatement("SELECT COUNT(*) FROM board");
        pstmt.execute();
        ResultSet rs = pstmt.executeQuery();
        while(rs.next()) {
            cnt = rs.getInt(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt, rs);
    }
    return cnt;
}

//insert, update, delete
public void insertBoard(int no, int cnt) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        pstmt = con.prepareStatement("INSERT INTO board (no, cnt) VALUES (?, ?)");
        pstmt.setInt(1, no);
        pstmt.setInt(2, cnt);
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt);
    }
}

//update
public void updateCnt(int no) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        pstmt = con.prepareStatement("UPDATE board SET cnt=cnt+1 where no=?");
        pstmt.setInt(1, no);
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt);
    }
}

//delete
public void deleteBoard(int no) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        pstmt = con.prepareStatement("DELETE FROM board where no=?");
        pstmt.setInt(1, no);
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt);
    }
}

```

5. 프로젝트 주요 화면 및 소스코드 설명

```

//data utf-8로 인코딩
public String parsing(String data) {
    try {
        data = new String(data.getBytes("UTF-8"), "UTF-8");
    } catch (Exception e) {}
    return data;
}

//home1.jsp에 사용
public int count() {
    //dbconnect
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int cnt = 0;

    try {
        pstmt = con.prepareStatement("SELECT COUNT(*) FROM board");
        pstmt.execute();
        rs = pstmt.executeQuery();
        if(rs.next()) {
            cnt = rs.getInt(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt, rs);
    }
    return cnt;
}

//write, ok.jsp에 사용
public int getCnt() {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int cnt = 0;

    try {
        pstmt = con.prepareStatement("SELECT COUNT(*) FROM board");
        pstmt.execute();
        rs = pstmt.executeQuery();
        if(rs.next()) {
            cnt = rs.getInt(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt, rs);
    }
    return cnt;
}

//board.jsp에 사용
public void insertBoard(int no, int cnt) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        pstmt = con.prepareStatement("INSERT INTO board (no, cnt) VALUES (?, ?)");
        pstmt.setInt(1, no);
        pstmt.setInt(2, cnt);
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt);
    }
}

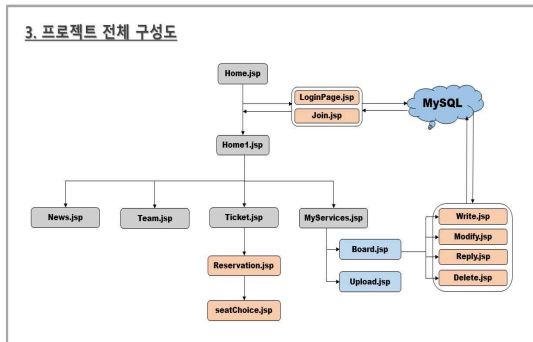
//update
public void updateCnt(int no) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        pstmt = con.prepareStatement("UPDATE board SET cnt=cnt+1 where no=?");
        pstmt.setInt(1, no);
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt);
    }
}

//delete
public void deleteBoard(int no) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        pstmt = con.prepareStatement("DELETE FROM board where no=?");
        pstmt.setInt(1, no);
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        dbclose.close(con, pstmt);
    }
}

```



4. 프로젝트 전체 UI 화면

4. 프로젝트 전체 UI 화면

4. 프로젝트 전체 UI 화면

5. 주요 화면 및 소스코드 설명

5. 프로젝트 주요 화면 및 소스코드 설명

```

request.setCharacterEncoding("UTF-8");
String id = request.getParameter("userEmail");

userDAO userDAO = new userDAO();
int result = userDAO.login(user.getEmail(), user.getUserPw());

//로그인 성공
if(result == 1){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    session.setAttribute("sessionUser", id);
    script.println("alert('로그인 성공.');");
    script.println("location.href='Home1.jsp'");
    script.println("</script>");
}

```

LoginCheck.jsp

5. 프로젝트 주요 화면 및 소스코드 설명



Ticket.jsp



Ticketing.jsp

```

<%
String name;
String stadium;
String date;
String CK;
%>
<%
//1. 데이터 베이스 연결
Connection con = DriverManager.getConnection();

//2. DBCP를 이용한 풀링
Context initCtx = new InitialContext();
Context envCtx = (Context) initCtx.lookup("java:comp/env");
DataSource ds = (DataSource) envCtx.lookup("jdbc/test");
con = ds.getConnection();

//3. insertForm에 의해
name = request.getParameter("name");
stadium = request.getParameter("stadium");
date = request.getParameter("date");
CK = request.getParameter("CK");

//4. insertForm에 의해(값을)
String sql = "insert into ticket(name, stadium, date) values('" + name + "','" + stadium + "','" + date + "')";
Statement stat = con.createStatement();

//5. 실행
PreparedStatement ps = con.prepareStatement(sql);
ps.executeUpdate();
%>

```

TicketingOK.jsp

6. DataBase 테이블 구성

6. DataBase 테이블 구성

```

create table Board(
NUM int NOT NULL auto_increment,
USERNAME varChar(50),
PASSWORD varChar(50),
TITLE varChar(50),
MEMO varChar(500),
TIME TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
HIT int, // Hit : 조회수,
REF int, // REF : 글이 생성되는 순서,
INDENT int, // INDENT : 답글과 댓글을 구분하기 위한,
STEP int, // STEP : 답글 순서를 구분하기 위한,
PRIMARY KEY ( NUM )
);

```

```

create table ticket1(
name varchar(20), /* 경기이름 */
stadium varchar(20), /* 경기장이름 */
date varchar(20) /* 경기일자 */
);

```

```

create table ticket2(
personnel int(3), /* 관할인원 */
seat varchar(50) /* 좌석이름 */
);

```

```

create table member(
name varchar(10),
email varchar(50) primary key,
pwd varchar(12),
phone varchar(13)
);

```

```

create table FILE (
fileName varchar(200),
fileRealName varchar(200)
);

```

6. 프로젝트 전체 로직 분리

```

board
  > DAO.java
  > DBClose.java
  > DBConnect.java
  > VO.java
DBUtil
  > DBUtil.java
file
  > downloadAction.java
  > FileDAO.java
  > FileDTO.java
user
  > userDAO.java
  > userDTO.java

```

```

JSP기밀PROJECT
> css
> fonts
> Images
> img
> js
  > delete_ok.jsp
  > fileDownload.jsp
  > Home.jsp
  > Join.jsp
  > JoinOK.jsp
  > LoginCheck.jsp
  > LoginPage.jsp
  > Logout.jsp
  > modify_ok.jsp
  > reply_ok.jsp
  > reply.jsp
  > TicketingOK.jsp
  > TicketCK.jsp
  > TicketPICK.jsp
  > uploadAction.jsp
  > view.jsp
  > write_ok.jsp
  > write.jsp
  > write.jsp
> META-INF
> WEB-INF
  > lib
    > cosjar
    > mysql-connector-java-8.0.
    > tomcat-dbcp.jar

```

7. Reference

7. Reference

- ✓ <http://bootstrapk.com/> - 부트스트랩 공식 홈페이지.
- ✓ <https://dgblog.tistory.com/114> - MySQL DB 연동 방법.
- ✓ <https://24getmenot7.tistory.com/273> - JSP 로그인, 회원가입 DB연동.
- ✓ <https://www.codingfactory.net/10529> - HTML / CSS 적용 방법.
- ✓ <https://seinarin.tistory.com/3> - JSP 게시판-1.
- ✓ <https://seinarin.tistory.com/19?category=452459> - JSP 게시판-2.
- ✓ <https://www.slideshare.net/ce/db-42499372> - 좌석 예매 DB 연동.
- ✓ <http://webberstudy.com/html-css/html-3/form-2/> - input 정리
- ✓ <https://loomio.kr/27> - check박스 사용법
- ✓ <https://neptune297.tistory.com/60> - check박스 배열

7. Reference

- ✓ <https://www.pngfind.com/freepng/real-madrid/> - 배경없는 이미지
- ✓ <https://bulkywebdevelopment.tistory.com/m/19?category=683776> - 글자 깨짐 해결
- ✓ <https://www.everdevel.com/Query/click.php> - jQuery 클릭 이벤트
- ✓ <https://ddo-o.tistory.com/63> - DB 조인
- ✓ <https://yeop-blog.github.io/2017/10/02/2017-10-02-old-blog-post83/> - check 다중 입력

Thank You

[첨부] : D. 소스 코드

```
package board;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import javax.security.auth.message.callback.PrivateKeyCallback.Request;

public class DAO {
    DBConnect dbconnect = null;
    String sql="";

    public DAO() {
        dbconnect = new DBConnect();
    }

    //Home1.jsp에서 사용
    // board테이블 찾아서 게시판 창에 입력
    public int count() {
        //드라이버를 로드
        Connection con = dbconnect.getConnection();
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        int cnt = 0;

        try {
            // sql문 입력하고 사용
            sql = "SELECT COUNT(*) FROM board";
            pstmt = con.prepareStatement(sql);
            rs = pstmt.executeQuery();

            if(rs.next()) {
                cnt=rs.getInt(1);
            }

        }catch(Exception e) {

        }finally {
            DBClose.close(con,pstmt,rs);
        }
        return cnt;
    }

    // data를 UTF-8로 인코딩
```

```

public String pasing(String data) {
    try {
        data = new String(data.getBytes("8859_1"), "UTF-8");
    } catch (Exception e) { }
    return data;
}

//Home1.jsp에서 사용
//게시판에 글 목록에 글 표시
public ArrayList<VO> getMemberList() {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    ArrayList<VO> alist = new ArrayList<VO>();

    try {
        sql = "SELECT NUM, USERNAME, TITLE, TIME, HIT, INDENT from board order by ref
desc, step asc";

        pstmt = con.prepareStatement(sql);
        rs = pstmt.executeQuery();

        while(rs.next()) {
            VO vo = new VO();
            boolean dayNew = false;
            vo.setNum(rs.getInt(1));
            vo.setName(rs.getString(2));
            vo.setTitle(rs.getString(3));

            Date date = new Date();
            SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd");
            String year = (String)simpleDate.format(date);
            String yea = rs.getString(4).substring(0,10);

            if(year.equals(yea)){
                dayNew = true;
            }

            vo.setTime(yea);
            vo.setHit(rs.getInt(5));
            vo.setIndent(rs.getInt(6));
            vo.setDayNew(dayNew);
            alist.add(vo);
        }

    } catch (Exception e) {

    } finally {
        DBClose.close(con,pstmt,rs);
    }
    return alist;
}

```

```
}
```

```
//write_ok.jsp에서 사용
```

```
//글쓸때마다 NUM의 카운터 1씩 증가시킴
```

```
public int getMax() {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int max = 0;

    try {
        sql = "SELECT MAX(NUM) FROM board";
        pstmt = con.prepareStatement(sql);
        rs = pstmt.executeQuery();

        if(rs.next()) {
            max=rs.getInt(1);
        }

    }catch(Exception e) {

    }finally {
        DBClose.close(con,pstmt,rs);
    }
    return max;
}
```

```
//write_ok.jsp에서 사용
```

```
//글쓰기창에서 글쓰면 board테이블에 insert
```

```
public void insertWrite(VO vo, int max) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {

        String sql = "INSERT INTO board(USERNAME,PASSWORD,TITLE,MEMO,REF)
VALUES(?,?,?,?,?)";

        pstmt = con.prepareStatement(sql);

        pstmt.setString(1, pasing(vo.getName()));
        pstmt.setString(2, pasing(vo.getPassword()));
        pstmt.setString(3, pasing(vo.getTitle()));
        pstmt.setString(4, pasing(vo.getMemo()));
        pstmt.setInt(5, max+1);

        pstmt.execute();
    }catch(Exception e) {

    }finally {
        DBClose.close(con,pstmt);
    }
}
```

```

    }
}

//reply_ok.jsp, reply.jsp, modify.jsp, view.jsp 에서 사용
//VO객체에 board테이블의 모든 데이터를 넣음
public VO getView(int idx) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    VO vo = null;

    try {
        sql = "SELECT USERNAME, TITLE, MEMO, TIME, HIT, PASSWORD, REF, INDENT, STEP
FROM board WHERE NUM=?";
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, idx);
        rs = pstmt.executeQuery();

        if(rs.next()) {
            vo = new VO();
            vo.setName(rs.getString(1));
            vo.setTitle(rs.getString(2));
            vo.setMemo(rs.getString(3));
            vo.setTime(rs.getString(4));
            vo.setHit(rs.getInt(5)+1);
            vo.setPassword(rs.getString(6));
            vo.setRef(rs.getInt(7));
            vo.setIndent(rs.getInt(8));
            vo.setStep(rs.getInt(9));
        }

    }catch(Exception e) {

    }finally {
        DBClose.close(con,pstmt,rs);
    }
    return vo;
}

//view.jsp에서 사용
//게시물 본 횟수(HIT) 표시
public void UpdateHit(int idx) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        sql = "UPDATE board SET HIT=HIT+1 where NUM=?";
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, idx);
        pstmt.executeUpdate();
    }
}

```

```
    }catch(Exception e) {

    }finally {
        DBClose.close(con,pstmt);
    }
}

//modify_ok.jsp에서 사용
//글 수정할때 패스워드 맞는지 확인
public boolean checkPassword(VO vo, int idx) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    boolean ch = false;

    try {
        sql = "SELECT NUM FROM board where NUM=? and PASSWORD=?";
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, idx);
        pstmt.setString(2, vo.getPassword());
        rs = pstmt.executeQuery();

        if(rs.next()) {
            ch = true;
        } else {
            ch = false;
        }
    }

    }catch(Exception e) {

    }finally {
        DBClose.close(con,pstmt,rs);
    }
    return ch;
}
```

```
//delete_ok.jsp에서 사용
//게시판 글의 NUM받아서 알맞은 DB삭제
public void delete(int idx) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        sql = "DELETE FROM board WHERE NUM=?";
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, idx);
        pstmt.executeUpdate();

    }catch(Exception e) {

    }finally {

    }
```



```
        DBClose.close(con,pstmt);
    }
}

//modify.ok.jsp에서 사용
//게시판 글의 NUM을 찾아와 알맞은 DB를 수정
public void modify(VO vo, int idx) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        sql = "UPDATE board SET TITLE=?, MEMO=? where NUM=?";
        pstmt = con.prepareStatement(sql);
        pstmt.setString(1, pasing(vo.getTitle()));
        pstmt.setString(2, pasing(vo.getMemo()));
        pstmt.setInt(3, idx);
        pstmt.executeUpdate();

    }catch(Exception e) {

    }finally {
        DBClose.close(con,pstmt);
    }
}

//reply_ok.jsp에서 사용
//답글 갯수(STEP) 표시
public void UpdateStep(int ref, int step) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;

    try {
        sql = "UPDATE board SET STEP=STEP+1 where REF=? and STEP>?";
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, ref);
        pstmt.setInt(1, step);
        pstmt.executeUpdate();

    }catch(Exception e) {

    }finally {
        DBClose.close(con,pstmt);
    }
}

//reply_ok.jsp에서 사용
//입력한 답글을 DB(board테이블)에 저장
public void insertReply(VO vo, int ref, int indent, int step) {
    Connection con = dbconnect.getConnection();
    PreparedStatement pstmt = null;
```

```
try {
    sql = "INSERT INTO board(USERNAME, PASSWORD, TITLE, MEMO, REF, INDENT, STEP)
VALUES(?,?,?,?,?,?,?)";
    pstmt = con.prepareStatement(sql);

    pstmt.setString(1, pasing(vo.getName()));
    pstmt.setString(2, pasing(vo.getPassword()));
    pstmt.setString(3, pasing(vo.getTitle()));
    pstmt.setString(4, pasing(vo.getMemo()));
    pstmt.setInt(5, ref);
    pstmt.setInt(6, indent+1);
    pstmt.setInt(7, step+1);

    pstmt.execute();
}catch(Exception e) {

}finally {
    DBClose.close(con,pstmt);
}
}
```

```
package board;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class DBClose {
    public static void close(Connection con, PreparedStatement pstmt, ResultSet rs) {
        try {
            try {
                if(rs!=null) { rs.close(); rs=null;}
            }catch(Exception e) {}

            try {
                if(pstmt!=null) { pstmt.close(); pstmt=null; }
            }catch(Exception e) {}

            try {
                if(con!=null) { con.close(); con=null; }
            }catch(Exception e) {}
        }catch(Exception e) {}
    }

    public static void close(Connection con, PreparedStatement pstmt) {
        try {
            try {
                if(pstmt!=null) { pstmt.close(); pstmt=null; }
            }catch(Exception e) {}

            try {
                if(con!=null) { con.close(); con=null; }
            }catch(Exception e) {}
        }catch(Exception e) {}
    }

    public static void close(Connection con, Statement stmt) {
        try {
            try {
                if(stmt!=null) { stmt.close(); stmt=null; }
            }catch(Exception e) {}

            try {
                if(con!=null) { con.close(); con=null; }
            }catch(Exception e) {}
        }catch(Exception e) {}
    }

    public static void close(Connection con) {
        try {
```

```
        try {
            if(con!=null) { con.close(); con=null; }
        }catch(Exception e) {}
    }catch(Exception e) {}
}

public static void close(PreparedStatement pstmt) {
    try {
        try {
            if(pstmt!=null) { pstmt.close(); pstmt=null; }
        }catch(Exception e) {}

    }catch(Exception e) {}
}

public static void close(Statement stmt) {
    try {
        try {
            if(stmt!=null) { stmt.close(); stmt=null; }
        }catch(Exception e) {}

    }catch(Exception e) {}
}

public static void close(ResultSet rs) {
    try {
        try {
            if(rs!=null) { rs.close(); rs=null; }
        }catch(Exception e) {}

    }catch(Exception e) {}
}

public static void close(PreparedStatement pstmt, ResultSet rs) {
    try {
        try {
            if(rs!=null) { rs.close(); rs=null; }
        }catch(Exception e) {}

        try {
            if(pstmt!=null) { pstmt.close(); pstmt=null; }
        }catch(Exception e) {}

    }catch(Exception e) {}
}
}
```

```
package board;
import java.sql.Connection;
import java.sql.DriverManager;

import java.sql.Connection;
import java.sql.DriverManager;

public class DBConnect {
    public static Connection getConnection() {
        Connection con = null;
        try {
            String url = "jdbc:mysql://localhost:3306/test?serverTimezone=UTC";
            String user = "root";
            String pwd = "1111";
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, user, pwd);
            return con;
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
        return null;
    }
}
```

```
package user;

import java.sql.Connection;
import java.sql.PreparedStatement;

import DBUtil.DBUtil;

public class userDAO {
    public int join(String userName, String userEmail, String userPWD, String userPhone) {
        String sql = "insert into member Values(?,?,?,?)";
        try {
            Connection con = DBUtil.getConnection();
            PreparedStatement psmt = con.prepareStatement(sql);
            psmt.setString(1, userName);
            System.out.println(userName);
            psmt.setString(2, userEmail);
            System.out.println(userEmail);
            psmt.setString(3, userPWD);
            System.out.println(userPWD);
            psmt.setString(4, userPhone);
            System.out.println(userPhone);

            return psmt.executeUpdate();
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
        return -1;
    }
}
```

```
package user;

public class userDTO {

    String userName;
    String userEmail;
    String userPWD;
    String userPhone;

    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getUserEmail() {
        return userEmail;
    }
    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail;
    }
    public String getUserPWD() {
        return userPWD;
    }
    public void setUserPWD(String userPWD) {
        this.userPWD = userPWD;
    }
    public String getUserPhone() {
        return userPhone;
    }
    public void setUserPhone(String userPhone) {
        this.userPhone = userPhone;
    }
}
```

```
package file;

public class FileDTO {

    String fileName;
    String fileRealName;

    public String getFileName() {
        return fileName;
    }
    public void setFileName(String fileName) {
        this.fileName = fileName;
    }
    public String getFileRealName() {
        return fileRealName;
    }
    public void setFileRealName(String fileRealName) {
        this.fileRealName = fileRealName;
    }
    public FileDTO(String fileName, String fileRealName) {
        super();

        this.fileName = fileName;
        this.fileRealName = fileRealName;
    }
}
```



```
package file;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class FileDAO {

    private Connection con;

    public FileDAO() {
        try {
            String url = "jdbc:mysql://localhost:3306/test?serverTimezone=UTC";
            String user = "root";
            String pwd = "1111";
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, user, pwd);

        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }

    public int upload(String fileName, String fileRealName) {
        String SQL = "INSERT INTO FILE VALUES (?,?)";
        try {
            PreparedStatement pstmt = con.prepareStatement(SQL);
            pstmt.setString(1, fileName);
            pstmt.setString(2, fileRealName);
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return -1;
    }
}
```

```

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/downloadAction")
public class downloadAction extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String fileName = request.getParameter("file");

        String directory = this.getServletContext().getRealPath("/upload/");
        File file = new File(directory + "/" + fileName);

        String mimeType = getServletContext().getMimeType(file.toString());
        if (mimeType == null) {
            response.setContentType("application/octet-stream");
        }

        String downloadName = null;
        if ( request.getHeader("user-agent").indexOf("MSIE") == -1) {
            downloadName = new String (fileName.getBytes("UTF-8"), "8859_1");
        } else {
            downloadName = new String(fileName.getBytes("EUC-KR"), "8859_1");
        }

        response.setHeader("Content-Disposition", "attachment;filename=" + downloadName + ".W");

        FileInputStream fileInputStream = new FileInputStream(file);
        ServletOutputStream servletOutputStream = response.getOutputStream();

        byte b[] = new byte[1024];
        int data = 0;

        while ((data = fileInputStream.read(b,0,b.length)) != -1) {
            servletOutputStream.write(b, 0, data);
        }

        servletOutputStream.flush();
        servletOutputStream.flush();
        fileInputStream.close();
    }
}

```

```

<%@page import="DBUtil.DBUtil"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="javax.naming.*,javax.sql.DataSource"
    import="java.sql.*"%>
<!DOCTYPE html>
<html>
<head>
<!-- content를 사용하여서 페이지에 들어왔을때 0초뒤 지정해둔 url로 가게 설정 -->
<meta charset="UTF-8" http-equiv=refresh content="0;url=Home1.jsp#!/page_MoreC">
<title>예약하기</title>
</head>
<body>
    <%
        String name;
        String stadium;
        String date;
        String Cck;

    %>
    <%
        request.setCharacterEncoding("UTF-8");
        //1. 드라이버 로드
        Connection con = DBUtil.getConnection();

        //2. DBCP를 사용하여 커넥션 객체 얻기
        Context initCtx = new InitialContext();
        Context envCtx = (Context) initCtx.lookup("java:comp/env");

        con = DBUtil.getConnection();

        //4. insertForm에서 값 전송 받기
        name = request.getParameter("name");
        stadium = request.getParameter("stadium");
        date = request.getParameter("date");
        Cck = request.getParameter("Cck");

        //4. 받은 값으로 sql문 사용(삽입 문)
        String sql = "insert into ticket1(name, stadium, date) values('" + name + "','" + stadium + "','" +
date + "')";

        Statement stmt = con.createStatement();

        //5. 질의문 실행
        PreparedStatement ps = con.prepareStatement(sql);
        ps.execute();

    %>

</body>
</html>

```

```

<%@page import="DBUtil.DBUtil"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="javax.naming.*,javax.sql.DataSource"
    import="java.sql.*"%>
<!DOCTYPE html>
<html>
<head>
<!-- content를 사용하여서 페이지에 들어왔을때 0초뒤 지정해둔 url로 가게 설정 -->
<meta charset="UTF-8" http-equiv=refresh content="0;url=Home1.jsp#!">
<title>Insert title here</title>
</head>
<body>
    <%
        String personnel;
        String seat;
    %>
    <%
        request.setCharacterEncoding("UTF-8");
        //1. 드라이버 로드
        Connection con = null;

        //2. DBCP를 사용하여 커넥션 객체 얻기
        Context initCtx = new InitialContext();
        Context envCtx = (Context) initCtx.lookup("java:comp/env");

        con = DBUtil.getConnection();

        //4. insertForm에서 값 전송 받기
        personnel = request.getParameter("output");
        seat = request.getParameter("input");

        //4. 받은 값으로 sql문 사용(삽입 문)
        String sql = "insert into ticket2(personnel, seat) values('" + personnel + "','" + seat + "')";
        Statement stmt = con.createStatement();

        //5. 질의문 실행
        PreparedStatement ps = con.prepareStatement(sql);
        ps.execute();
    %>

    <!--
    <form action="Home1.jsp#!/page_MoreC" name="mmyForm" method="post">
        <input type="text" name="name" value="<%=seat%>" />
        <input type="submit" value="이 페이지 없애고 바로가고싶어요 π" />
    </form>
    -->

</body>
</html>

```