

# 1조 파일 프로젝트

코딩

Do it!

팀장 : 이명진

팀원 : 김고운, 신윤경, 유근형, 조영민, 황은선

프로젝트 기간 : 2023.02.15 ~ 2023.03.07



1. 주제선정 배경 및 기획의도
2. 벤치마킹 및 차별화 전략
3. 요구사항 명세서
4. 역할분담 및 작업일정
5. 프로젝트 프로파일러
6. 사용자 흐름도
7. 사용자 화면 정의서
8. ER-다이어그램, 테이블 명세서
9. 클래스 명세서, 서비스 명세서
10. SQL, 인스턴스 명세서
11. 주요 코드
12. 프로젝트를 마치며

# 1. 주제선정 배경 및 기획의도

주제 티켓 예매 사이트

사이트 로고



서비스 이름 뜻

티켓(표)

TICKET +

CATCH

잡다

# 1. 주제선정 배경 및 기획의도

---

## 주제선정 배경

: “자바를 활용한 AI 기반 지능형 서비스 개발자 과정”을 통해 배운 Java와 Spring을 기반으로 프로젝트를 진행 실제로 사람들이 일상에서 많이 사용하는 사이트를 만들어 보고자 하였고 그 결과 2차 때 진행한 프로젝트를 확장 및 보완하여 프로젝트의 완성도를 높이고자 주제를 선정

## 기획의도

: 비슷한 서비스를 제공 중인 사이트들을 실제로 이용하며 아쉬웠던 점들을 보완하고 새로운 기능들을 추가하여 차별화 전략으로 내세워 보고자 사이트를 기획하고 제작함

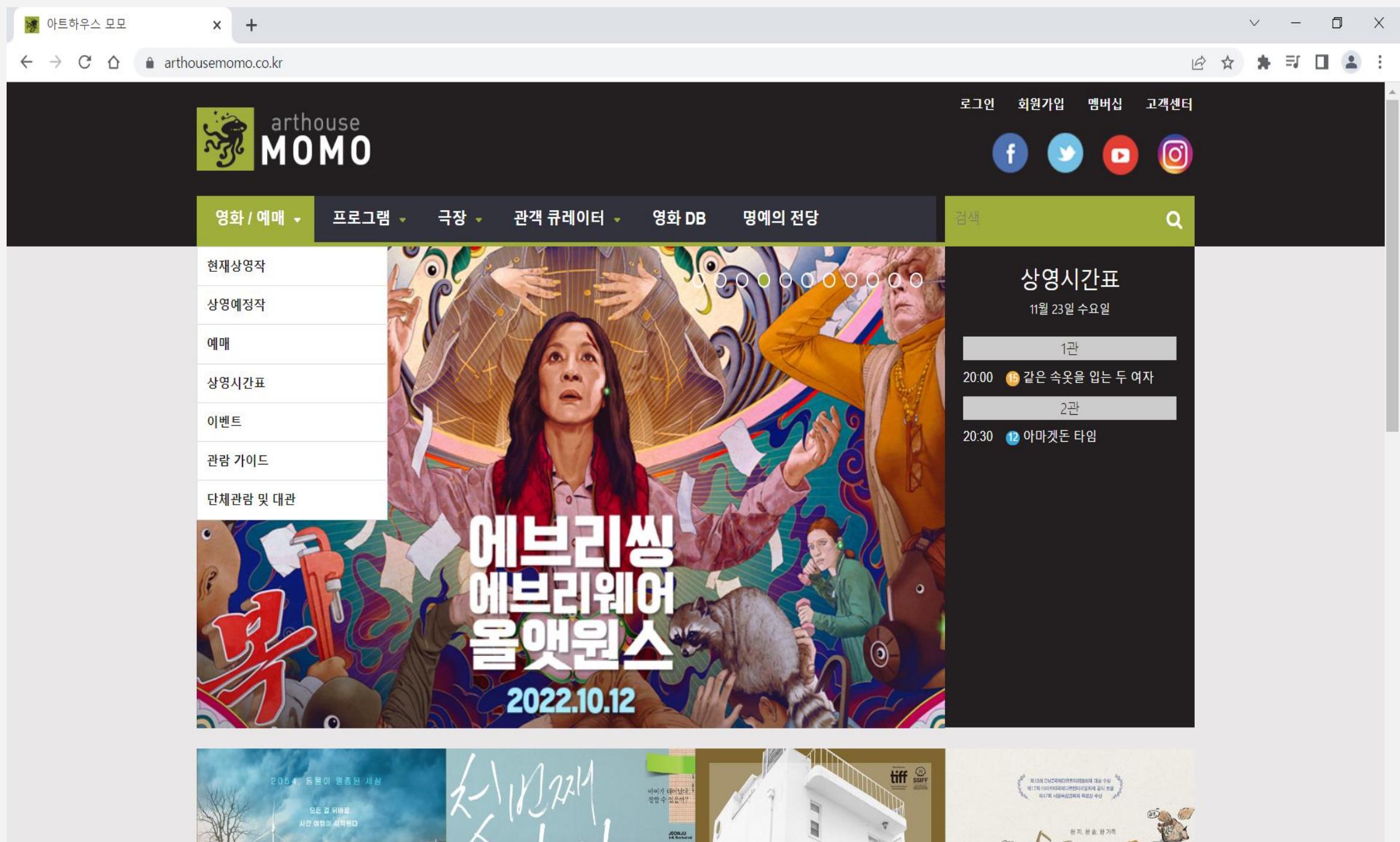
## 2. 벤치마킹 및 차별화 전략

### 벤치마킹 사이트

#### - 아트하우스 모모 (영화 예매 사이트)

: 사이트의 UI/UX가 깔끔하고 기능들이 군더더기 없어

전반적인 사이트의 분위기와 기능을 참고



## 2. 벤치마킹 및 차별화 전략

### - YES24 티켓 (통합 예매 사이트)

: 메인 페이지 - 장르별 랭킹, 현재상영작, 개봉예정작 부분 참고

The screenshot shows the YES24 Ticket website's ranking section for musicals. The page title is 'RANKING' with filters: #뮤지컬, #콘서트, #연극, #클래식, #전시. The ranking table includes:

순위	공연	개봉일	장소
1위	미수 [뮤지컬]	2022.12.03 ~ 2023.02.19	대학로 자유극장
2위	태양의서커스 <뉴 알레그리아>	2022.10.20 ~ 2023.01.01	잠실종합운동장 내 빅탑
3위	로빈	2023.01.09 ~ 2023.04.09	대학로 IOM 1관

A circular badge on the right says '조용필 뮤지컬대본 공모전'.

The screenshot shows the YES24 Ticket website's 'MUSICAL & PLAY' section. The page title is 'MUSICAL & PLAY' with filters: #뮤지컬, #연극, #클래식, #전시, #아동. The section displays various show posters:

- 뮤지컬 화가 시리즈
- 뮤지컬 볼룸업
- 뮤지컬 클립트
- 연극 아트 (ART)
- 연극 <레드>
- 장진의 연극 [서툰 사람들]

## 2. 벤치마킹 및 차별화 전략

### - 인터파크 티켓 (통합 예매 사이트)

: 티켓 예매 사이트 시장에서 점유율 70% 차지하는 곳

상세페이지, 예매 기능 등이 다른 사이트에 비해 깔끔하여 참고

The screenshot shows the homepage of the Interpark Tickets website. At the top, there's a search bar and navigation links for categories like Musical, Concert, Drama, etc. A banner for '42ND STREET' is visible. The main content area displays a show listing for '태양의서커스<뉴 알레그리아>'. It includes a thumbnail image of the show, a rating of 9.1, and details about the venue (Jamsil Sports Complex Indoor Stadium), dates (Oct 20, 2022 - Jan 1, 2023), and ticket prices (VIP: 290,000원, SR: 190,000원, R: 150,000원, S: 120,000원, A: 90,000원, B: 70,000원). To the right, there's a calendar for November 2022 with the 24th highlighted in red, indicating the date of the performance. Below the calendar, it says '회차 1회 20:00'. At the bottom right, there's a large red button labeled '예매하기' (Buy Now).

## 2. 벤치마킹 및 차별화 전략

### - 티켓링크 (통합 예매 사이트)

: 후기 작성 페이지 등을 참고

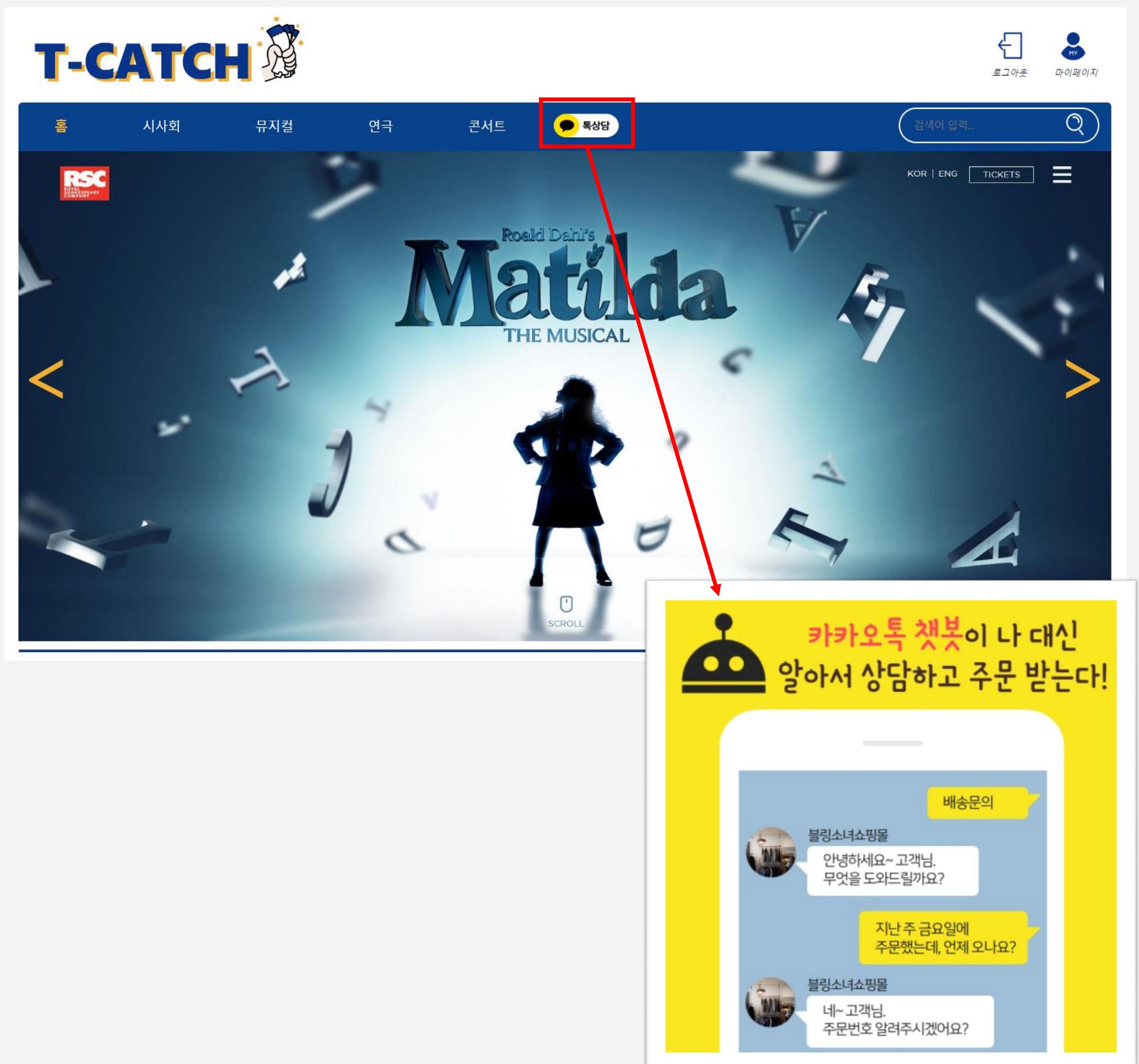
The screenshot shows a web browser window for TicketLink. The URL in the address bar is ticketlink.co.kr/product/41021. The main content area displays a product listing for '장진의 연극 <서툰 사람들>' at '예스24스테이지 3관' from '2022.11.26' to '2023.02.19'. It includes dropdown menus for selecting '2022.11.26' and '14시 00분'. A large red button labeled '예매하기' (Buy) is visible. Below the product info, there are five tabs: '상세정보' (Detailed Information), '관람후기' (Review), '기대평' (Expectation Review), '장소정보' (Venue Information), and '예매/취소안내' (Purchase/Cancellation Instructions). The '관람후기' tab is selected, showing '0' reviews and a 5-star rating of '0.0 / 5'. A note below states: '게시판 운영규정에 맞지 않는 글은 사전 통보없이 삭제될 수 있습니다.' (Posts that do not match the board operating rules will be deleted without prior notice.) To the right is a link to '게시판 운영규정' (Board Operating Rules). A text input field for reviews is present with placeholder text '별점을 선택해주세요.' (Please select a rating) and a note '관람후기를 남겨보세요!' (Leave a review!). A character limit indicator '0/1000' and a dark blue '등록' (Register) button are at the bottom of the input field. At the very bottom of the page, a small note says '관람후기를 등록해주세요.' (Please register your review).

## 2. 벤치마킹 및 차별화 전략

### 차별화 전략

#### - 메인페이지 챗봇 기능 추가

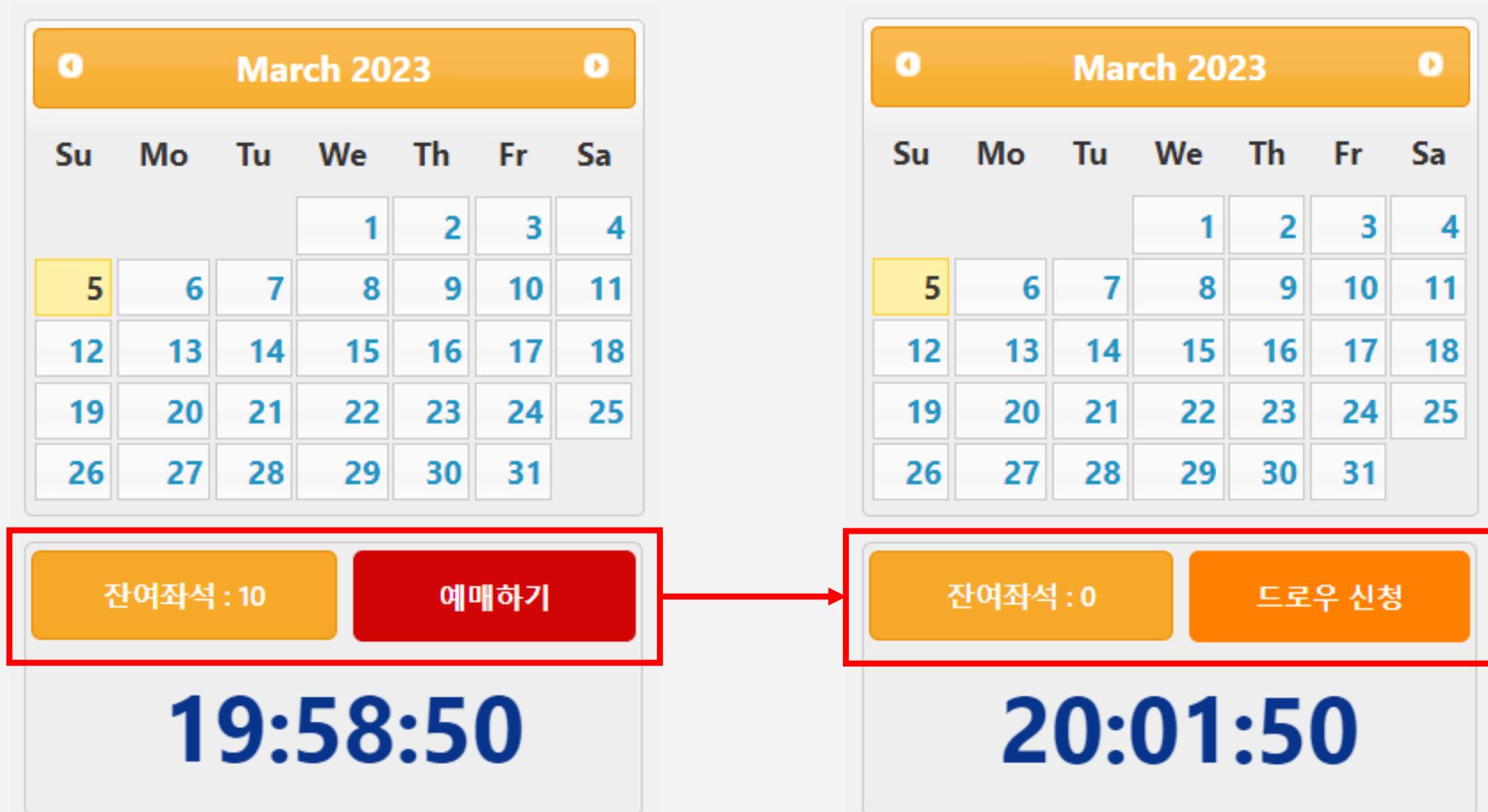
: 카카오 챗봇 api를 활용해 문의사항이 있을 시 해당 기능 사용



## 2. 벤치마킹 및 차별화 전략

### - DRAW 기능 사용

: 예매하지 못한 이용자들은 “취켓팅(예매 취소된 표를 구하기 위해 자정 ~ 새벽에 하는 티켓팅)”을 통해 어렵게 표를 구함  
이러한 번거로움을 줄이기 위해 만석일 경우, 드로우 신청을 받고  
예매 취소로 공석 발생 시 무작위 추첨을 통해 신청자에게  
표를 가질 수 있는 기회를 제공



만석(잔여좌석이 0석)일 경우,  
드로우 신청 버튼 활성화  
→ 공석 발생 시 무작위 추첨  
→ 상영일 하루 전 자동으로 무작위 추첨  
→ [마이페이지] - [드로우내역]에서 당첨 확인

## 2. 벤치마킹 및 차별화 전략

### - 사이트 내 서버 시간 제공

: 티켓 예매 서비스를 제공하는 기존 사이트들은 대부분의 경우 서버 시간을 제공하지 않아 이용자들이 서버 시간 안내 사이트를 따로 확인하며 티켓팅을 진행해야 하는 번거로움이 존재 이를 해결하기 위하여 사이트 내 예매 페이지에서 서버 시간을 제공

The screenshot shows a movie listing for '그래비티' (Gravity) on the T-CATCH website. The movie poster features a hand reaching out against a dark background with Earth visible. The listing includes the following details:

- 관람일 : 2023-03-07 20:00:00
- 장소 : CGV 용산
- 관람등급 : 12세 이상
- 관람시간 : 90분
- 가격 : 0원(전석 초대)

To the right of the movie listing is a calendar for March 2023. The date 2023-03-07 is highlighted in yellow. A red box surrounds the time '20:28:10' displayed prominently below the calendar. A red arrow points from this red box down to a text box at the bottom right.

사이트 내  
서버 시간 제공

### 3. 요구사항 명세서

#### 이용자

서비스	내용
이용자	- 아이디 적합성 및 중복확인 - 비밀번호 적합성 및 입력 자리수 지정 - 전화번호, 이메일 중복확인 - 전화번호 인증
	- 로그인 전: 회원가입, 로그인, 검색, 상세보기 등 사용 가능 - 로그인 후: 예매 및 드로우 신청, 마이페이지 등 사용 가능
	- 카테고리 별로 검색
	- 개인 관심사 맞춤형 랭킹 제공 - 평점 높은 순, 예매자 많은 순으로 집계
	- 상품의 날짜, 위치, 상세정보, 후기(별점, 코멘트), 잔여 좌석수, 서버시간 등 제공 - 예매한 사람들의 세대별, 성별 분석
	- 서버 시간 제공하고 예매 오픈 시간에 예매 버튼 활성화 - 좌석정보 확인 - 매진 시 드로우 옵션 선택 가능
	- 개인정보 수정 - 내가 예매한 내역, 신청한 드로우 내역 및 결과 보기 - 내가 작성한 후기 모아보기 - 1:1 문의내역 확인 (글 공개여부 결정) - 알람

### 3. 요구사항 명세서

#### 관리자

서비스		내용
관리자	작품등록	<ul style="list-style-type: none"><li>- DB로 직접 등록하는 것이 아니라 View를 통해 판매 작품 등록 및 수정, 삭제</li></ul>
	회원관리	<ul style="list-style-type: none"><li>- View를 통해 전체 회원 관리</li></ul>
	고객센터	<ul style="list-style-type: none"><li>- 공지사항 작성 및 수정, 삭제</li><li>- 1:1 문의에 답변 달기</li></ul>

#### 권한 설정

서비스		내용
권한 설정	관리자	<ul style="list-style-type: none"><li>- 공지사항 및 1:1 문의에 글 작성</li><li>- 회원관리</li><li>- 작품등록</li></ul>
	이용자	<ul style="list-style-type: none"><li>- 마이페이지</li><li>- 1:1 문의</li></ul>
	그 외	<ul style="list-style-type: none"><li>- 게시판 열람</li></ul>

# 4. 역할분담 및 작업일정

---

## 역할분담

**이명진** : 뮤지컬 인스턴스 작성 및 전체 테이블 인스턴스 정리, SQL 작성, 서버시간 기능, 잔여좌석 기능, 랭킹 기능, 카카오맵 api 기능, 성별 통계, 마이 페이지 - 후기 작성 및 수정/삭제 기능

**김고운** : 관리자 페이지 구현, 공지사항 및 QnA 게시판 기능 구현, 시사회 인스턴스 작성, 페이징 처리

**신윤경** : 전반적인 View 화면 설계 및 전체 CSS 담당, 연극 인스턴스 작성, 로그인/회원가입

**유근형** : 콘서트 인스턴스 작성, 로그인/회원가입 기능, 메인 페이지에 현재 상영작/개봉예정작에 장르별 작품 출력 기능, 장르별 페이지 기능, 검색 페이지 기능, 마이 페이지 - 예매내역 기능

**조영민** : 세션스토리지 기능, 전반적인 기능 구현 및 오류 수정, 드로우 기능 구현, 프로젝트 클래스 정리 및 Merge

**황은선** : 테이블 별 DAO/VO 클래스 CURD 작성, SQL 작성, View 화면 설계(예매 페이지, 마이 페이지 일부), 세대별 통계, 예매하기 버튼 기능, 마이 페이지 – 내정보 기능

# 4. 역할분담 및 작업일정

## 작업일정 - 1주차

일정	15 (수)	16 (목)	17 (금)	18 (토)	19 (일)	20 (월)	21 (화)
View 화면 설계							
CSS							
DAO/VO 클래스 (CRUD)							
메인 페이지 기능 구현							
로그인/회원가입							
장르별 페이지 기능 구현							
상세 페이지 기능 구현							
검색 페이지 기능 구현							
예매/드로우 기능 구현							
마이페이지 – 내정보 기능 구현							
마이페이지 – 예매내역 기능 구현							
마이페이지 – 드로우내역 기능 구현							
관리자 페이지 – 티켓 등록 기능 구현							
관리자 페이지 – 회원 관리 기능 구현							
게시판 페이지 – 공지사항 기능 구현							
게시판 페이지 – 1:1 문의 기능 구현							

1주차  
중간  
점검  
및  
수정

# 4. 역할분담 및 작업일정

## 작업일정 - 2주차

일정	22 (수)	23 (목)	24 (금)	25 (토)	26 (일)	27 (월)	28 (화)
View 화면 설계							
CSS							
DAO/VO 클래스 (CRUD)							
메인 페이지 기능 구현							
로그인/회원가입							
장르별 페이지 기능 구현							
상세 페이지 기능 구현							
검색 페이지 기능 구현							
예매/드로우 기능 구현							
마이페이지 – 내정보 기능 구현							
마이페이지 – 예매내역 기능 구현							
마이페이지 – 드로우내역 기능 구현							
관리자 페이지 – 티켓 등록 기능 구현							
관리자 페이지 – 회원 관리 기능 구현							
게시판 페이지 – 공지사항 기능 구현							
게시판 페이지 – 1:1 문의 기능 구현							

2주차  
중간  
점검  
및  
수정

# 4. 역할분담 및 작업일정

## 작업일정 - 3주차

일정	1 (수)	2 (목)	3 (금)	4 (토)	5 (일)	6 (월)	7 (화)
View 화면 설계							
CSS							
DAO/VO 클래스 (CRUD)							
메인 페이지 기능 구현							
로그인/회원가입							
장르별 페이지 기능 구현							
상세 페이지 기능 구현							
검색 페이지 기능 구현							
예매/드로우 기능 구현							
마이페이지 – 내정보 기능 구현							
마이페이지 – 예매내역 기능 구현							
마이페이지 – 드로우내역 기능 구현							
관리자 페이지 – 티켓 등록 기능 구현							
관리자 페이지 – 회원 관리 기능 구현							
게시판 페이지 – 공지사항 기능 구현							
게시판 페이지 – 1:1 문의 기능 구현							

Merge  
후  
기능  
구현  
검토  
및  
보완

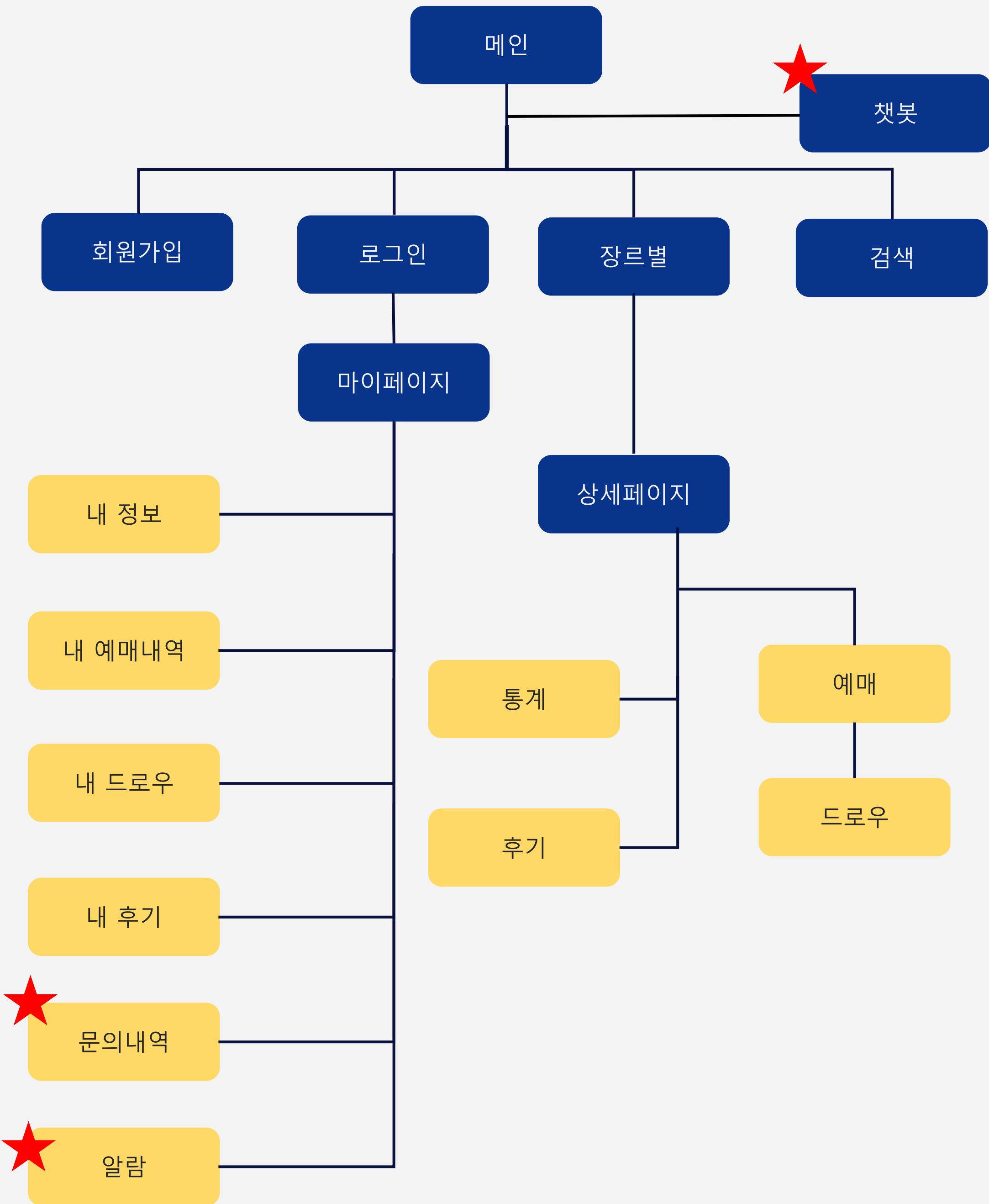
최종  
검토  
및  
발표

# 5. 프로젝트 프로파일러

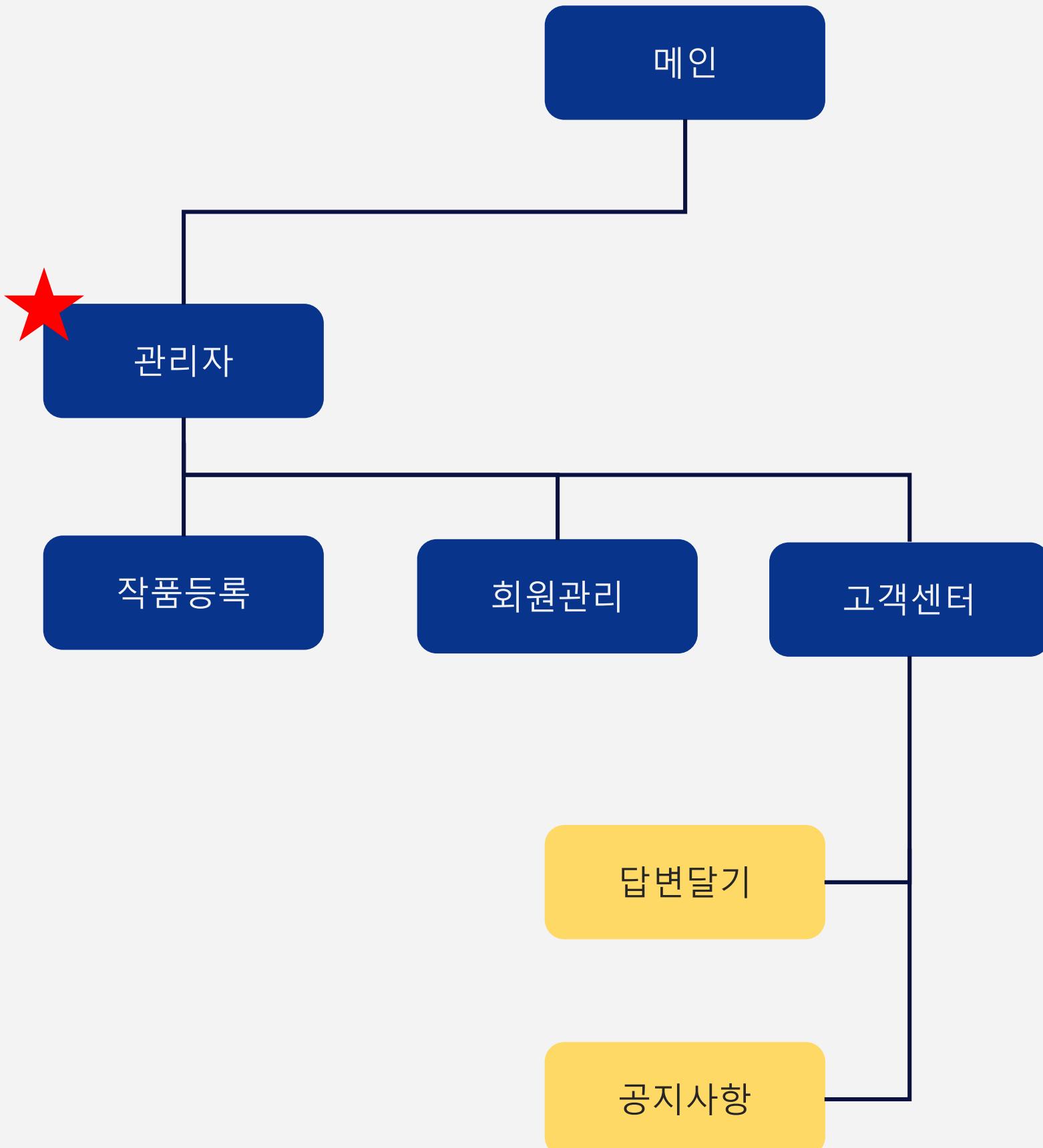
※ 모든 부분(클래스, 메소드, 변수, SQL 등)에 주석 표시 ※

구분	프로파일러
클래스	<ul style="list-style-type: none"><li>- 첫 글자는 대문자로 시작</li><li>- 카멜 표기법에 따라 영단어 공백은 대문자로 구분 ex) BookDAO</li></ul>
메소드	<ul style="list-style-type: none"><li>- 첫 글자는 소문자로 시작, 카멜 표기법에 따라 그 이후 오는 첫 단어는 대문자로 작성 ex) listBook</li></ul>
변수	<ul style="list-style-type: none"><li>- 소문자로 시작 ex) int custid</li><li>- 카멜 표기법과 스네이크 표기법(공백을 '_'로 구분) 적절히 혼용</li></ul>
들여쓰기	<ul style="list-style-type: none"><li>- 시스템 규칙에 따름</li><li>- 중괄호 다음 줄 들여쓰기</li></ul>
DB연결/DB계정	<p>&lt;DB연결&gt; MyBatis, JPA</p> <p>&lt;DB계정&gt;</p> <ul style="list-style-type: none"><li>- 사용자명 : c##final / 비밀번호 : final123</li><li>- 테이블명, 컬럼명 소문자로 작성</li></ul>
Dependency	Spring Web, Spring Security, Lombok, Thymeleaf, Oracle, Mybatis, JPA, Spring boot DevTools,
기타	<ul style="list-style-type: none"><li>- 하루 한 일 Github Readme에 작성</li></ul>

# 6. 사용자 흐름도



# 6. 사용자 흐름도



# 7. 사용자 화면 정의서

< 메인 페이지 >

The screenshot shows the T-CATCH website homepage. At the top, there is a navigation bar with categories: 흥 (Hyeong), 시사회 (Society), 뮤지컬 (Musical), 연극 (Theater), and 콘서트 (Concert). On the right side of the header are icons for 프로그램 (Program), 회원가입 (Join), and 마이페이지 (My Page). A search bar is located at the top right. The main banner features a scene from the movie Monsters, Inc. with Sulley and other monsters. Below the banner, there is a section titled '장르별 랭킹' (Genre Ranking) showing three musicals: 1. 네스노트 (Nessnote), 2. 엘리자벳 (Elizabeth), and 3. 드라큘라 (Dracula). Further down, there is a section titled '현재상영작' (Currently Playing) showing three more musicals: 1. 마틸다 (Matilda), 2. 엘리자벳 (Elizabeth), and 3. 킹기부츠 (Kinky Boots).

**T-CATCH**

흥 시사회 뮤지컬 연극 콘서트

장르별 랭킹

1 네스노트 2 엘리자벳 3 드라큘라

현재상영작

1 마틸다 2 엘리자벳 3 킹기부츠

# 7. 사용자 화면 정의서

< 메인 페이지 - 상단 >

The screenshot shows the top navigation bar of the T-CATCH website. On the left is the logo 'T-CATCH' with a cartoon character icon. To the right are three user account icons: '로그인' (Login), '회원가입' (Sign Up), and 'マイ페이지' (My Page). Below the header is a search bar with the placeholder '검색어 입력...' (Enter search term...) and a magnifying glass icon.

Home 시사회 뮤지컬 연극 콘서트

현재상영작  
개봉예정작  
과거상영작

STANDBY SCARE

怪物公司 (Monsters, Inc.)

< 메인 페이지 – 장르별 랭킹 (후기 평점 높은순) >

The screenshot displays the '장르별랭킹' (Genre Ranking) section of the website. It features three posters for musicals, each numbered 1, 2, and 3. The genres are indicated by small circles above the numbers: 1 (시사회), 2 (연극), and 3 (뮤지컬).

1. 데즈노트 (The Death Note)
2. 엘리자베스 (Elizabeth - Das Musical)
3. 드라큘라 (DRACULA)

시사회 뮤지컬 연극 콘서트

# 7. 사용자 화면 정의서

## < 메인 페이지 – 현재상영작, 개봉예정작 >

홈 시사회 연극 뮤지컬 콘서트 검색어 입력...

**현재상영작**

시사회 연극 뮤지컬 콘서트

**개봉예정작**

시사회 연극 뮤지컬 콘서트

# 7. 사용자 화면 정의서

< 회원가입 화면 >

## SignUp

H

아이디  
\* 6자~12자의 소문자 영문, 숫자로만 입력

중복확인

P

비밀번호  
\* 최소 6자 이상 소대문자 영문, 숫자, 특수문자로 입력

C

비밀번호 확인  
(비밀번호와 일치하는지 확인)

N

이름  
(필수 입력)

E

이메일주소  
@ gmail.com

D

생년월일  
\* 예시 : 1999/09/09

T

전화번호  
(필수 입력)

중복확인

I

전화번호 인증  
(인증번호를 입력하세요)

인증번호 보내기

인증번호 확인

A

주소  
(선택 입력)

우편번호 찾기

S

성별  
(선택 입력)

M

F

O

관심사  
(선택 입력)

S

M

Y

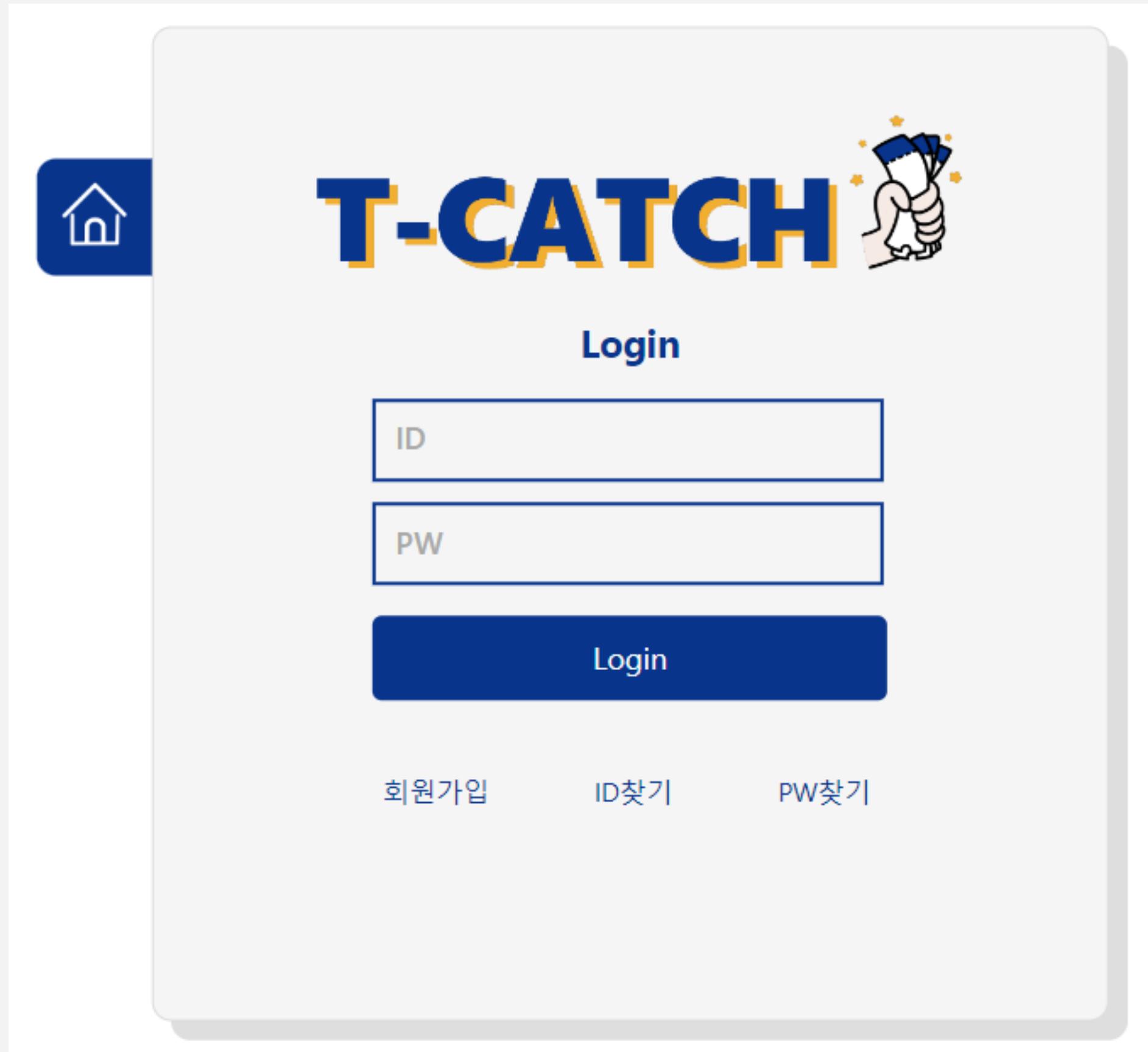
C

취소

가입하기

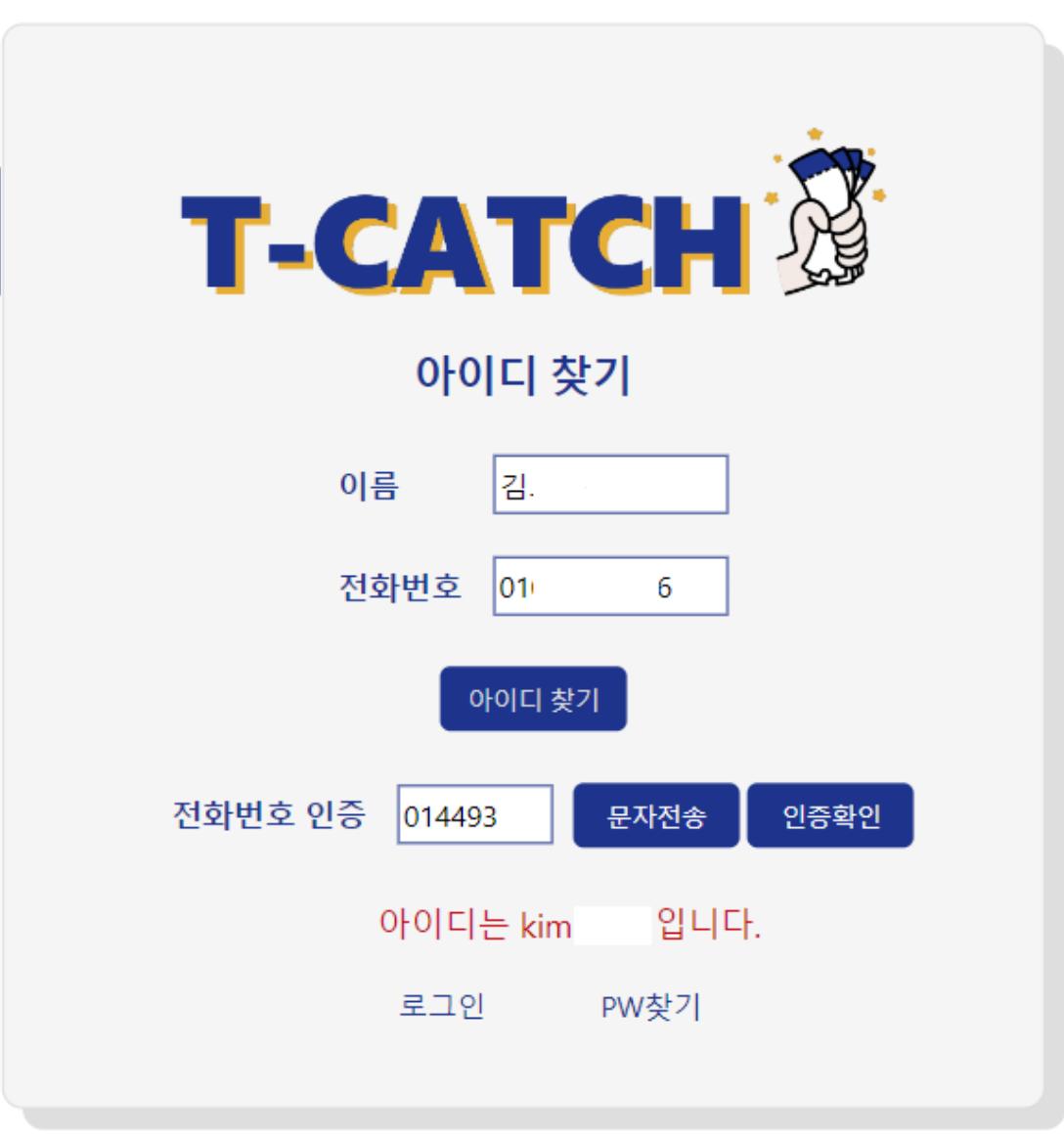
# 7. 사용자 화면 정의서

< 로그인 화면 >



# 7. 사용자 화면 정의서

< 아이디/비밀번호 찾기 >



The screenshot shows the T-CATCH user interface for recovering an ID. It features a logo with a blue house icon and the text "T-CATCH". Below it is a search bar labeled "아이디 찾기" (ID Recovery). The search fields are "이름" (Name) with "김." entered and "전화번호" (Phone Number) with "010" and "6" entered. A blue button labeled "아이디 찾기" (Find ID) is below the fields. At the bottom, there are three buttons: "전화번호 인증" (Phone Verification) with "014493", "문자전송" (Text Message), and "인증확인" (Verification Confirmation).

아이디 찾기

이름 김.

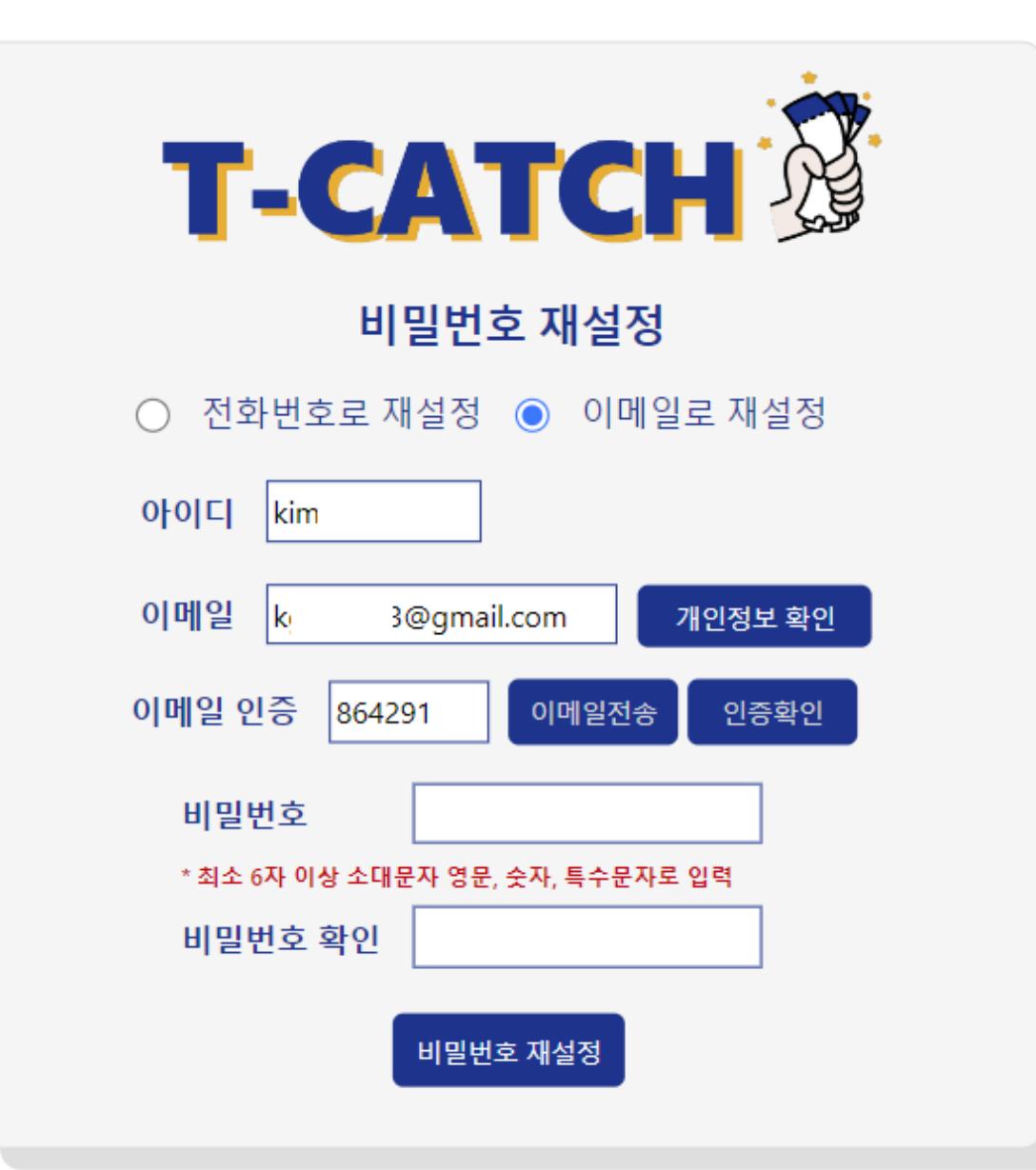
전화번호 010 6

아이디 찾기

전화번호 인증 014493 문자전송 인증확인

아이디는 kim 입니다.

로그인 PW찾기



The screenshot shows the T-CATCH user interface for password reset. It features a logo with a blue house icon and the text "T-CATCH". Below it is a search bar labeled "비밀번호 재설정" (Password Reset). There are two radio buttons: "전화번호로 재설정" (Reset via Phone Number) and "이메일로 재설정" (Reset via Email), with the latter being selected. The "아이디" (ID) field contains "kim". The "이메일" (Email) field contains "k...@gmail.com". A blue button labeled "개인정보 확인" (Check Personal Information) is next to the email field. At the bottom, there are three buttons: "이메일 인증" (Email Verification) with "864291", "이메일전송" (Email Send), and "인증확인" (Verification Confirmation). Below these, there are input fields for "비밀번호" (Password) and "비밀번호 확인" (Password Confirmation), both with placeholder text. A note at the bottom states "\* 최소 6자 이상 소대문자 영문, 숫자, 특수문자로 입력" (\* Minimum 6 characters, lowercase letters, numbers, and special characters required). A blue button labeled "비밀번호 재설정" (Reset Password) is at the bottom right.

비밀번호 재설정

○ 전화번호로 재설정 ○ 이메일로 재설정

아이디 kim

이메일 k...@gmail.com 개인정보 확인

이메일 인증 864291 이메일전송 인증확인

비밀번호

\* 최소 6자 이상 소대문자 영문, 숫자, 특수문자로 입력

비밀번호 확인

비밀번호 재설정

# 7. 사용자 화면 정의서

< 장르별 페이지 – 상단 메뉴바 [메뉴-서브메뉴 클릭] >

The screenshot shows the T-CATCH movie recommendation website's genre page. At the top, there is a navigation bar with links for Home, 시사회 (Society), 연극 (Theater), 뮤지컬 (Musical), 콘서트 (Concert), and a search bar. On the right side of the navigation bar are icons for logging out and viewing my page. Below the navigation bar, the page title is "현재상영작" (Currently Playing Movies). There are three movie cards displayed:

- 에브리씽 에브리웨어 올 앤 원스**  
개봉예정작  
과거상영작  
다니엘 환(감독), 양자경, 스테파니 수, 키 흐이 환  
2022/11/25 15:10  
139  
CGV 명동
- 그래비티**  
알폰소 쿠아론(감독), 산드라 블록, 조지 클루니  
2022/11/24 20:00  
90  
CGV 용산
- 헤어질 결심**  
박찬욱(감독), 탕웨이, 박해일, 이정현  
2022/11/26 19:00  
138  
CGV 강변

At the bottom of the page, there is a footer with the text "© CodingDoit corp."

# 7. 사용자 화면 정의서

< 검색 페이지 – 검색어는 제목과 캐스트에서 키워드 선택 >

The screenshot shows the T-CATCH website interface. At the top, there is a navigation bar with categories: 홈 (Home), 시사회 (Press Conference), 뮤지컬 (Musical), 연극 (Theater), and 콘서트 (Concert). On the right side of the header is a search bar with the placeholder "검색어 입력..." and a magnifying glass icon. Below the header, a banner displays the text "'박찬욱'에 대한 검색 결과입니다". The main content area features a movie poster for "헤어질 결심" (Haircut) with actors Park Chan-wook and Tang Wei. To the right of the poster, the movie's title "헤어질 결심" is displayed in large blue text, followed by the cast "박찬욱(감독), 탕웨이, 박해일, 이정현", release date "2022/11/26 19:00", runtime "138", and venue "CGV 강변". The bottom of the page has a dark blue footer bar with the copyright notice "© CodingDoit corp."

# 7. 사용자 화면 정의서

## < 작품 상세페이지 - 예매 >

The screenshot shows a movie ticket purchase page on the T-CATCH website. At the top, there's a logo for 'T-CATCH' with a cartoon character, and navigation links for 'Home', '시사회' (Society), '연극' (Theater), '뮤지컬' (Musical), and '콘서트' (Concert). A search bar and user icons for '로그아웃' (Logout) and '마이페이지' (My Page) are also at the top right.

The main content area displays a movie poster for '에브리씽 에브리웨어 올 앤 원스'. Below the poster, event details are listed: 관람일 : 2022/11/25 15:10, 장소 : CGV 명동, 관람등급 : 15, 관람시간 : 139, 가격 : 전석 초대. To the right of the poster is a calendar for November 2022, showing the date 24 highlighted in yellow. Below the calendar are buttons for '잔여좌석 : 10' (Remaining Seats: 10) and '예매하기' (Buy Now), and a large digital clock showing 12:27:49.

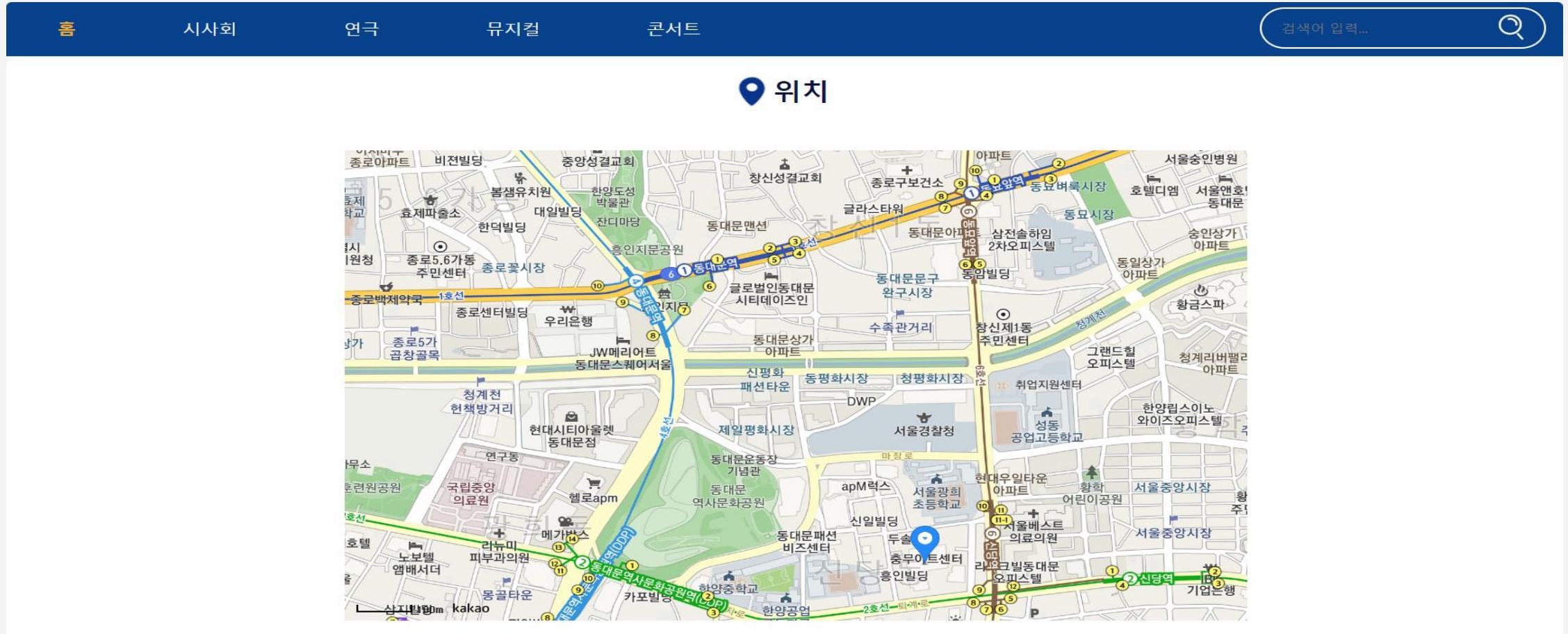
## < 작품 상세페이지 – 상세 정보 >

The screenshot shows a detailed page for the musical '킹키부츠'. The top navigation bar includes 'Home', '시사회', '연극', '뮤지컬', '콘서트', a search bar, and user icons. The '상세정보' (Detailed Information) tab is currently selected.

The main content features a large image of the cast performing on stage, with the text '2020 뮤지컬 <킹키부츠>' and 'Raise You Up MV'. Below the image is a 'YouTube' link labeled '다음에서 보기' (Watch on Naver). To the right of the image, a badge indicates '150만' (1.5 million). Below the image are sections for '[캐스트]' (Cast) and '[Content]'. A small note at the bottom left discusses the plot involving two male leads creating a 1.8m tall '팅키부츠'.

# 7. 사용자 화면 정의서

## < 작품 상세페이지 - 위치 >



## < 작품 상세페이지 – 통계&후기 >

### 후기

  
남자 :

  
여자 :

~10대  
20대  
30대  
40대  
50대 ~

<p><b>별점</b></p> <p>작성자</p> <p>후기 내용</p>	<p><b>관람일</b></p>
<p><b>별점</b></p> <p>작성자</p> <p>후기 내용</p>	<p><b>관람일</b></p>

# 7. 사용자 화면 정의서

## < 예매화면 >



# 7. 사용자 화면 정의서

## < 마이페이지 - 내정보 >

MyPage

- 내 정보
- 예매내역
- 드로우내역
- 내 후기

### 회원정보 수정

아이디 mj1234

비밀번호 .....  
\* 최소 6자 이상 영문, 숫자로만 입력

비밀번호 확인

이름 이명진

이메일주소 dlsdjdmqlc3@naver.com

생년월일 1996/03/11

전화번호 010-1234-1234 증복확인

성별 남성  여성

관심사 시사회  뮤지컬  연극  콘서트

취소 수정하기

# 7. 사용자 화면 정의서

## < 마이페이지 - 예매내역 >

The screenshot shows the user interface of the T-CATCH MyPage section for reservation history. At the top, there is a navigation bar with links for Home, Social, Music, Concert, and Concerts. On the right side of the header are icons for logging out and the user's profile, along with a search bar.

**Left Sidebar (MyPage):**

- 내 정보
- 예매내역** (highlighted)
- 드로우내역
- 내 후기

**Reservation History Content:**

**ART**

예매번호 : 6
관람일 : 2022/11/29 14:00
상영관 : 예스24스테이지 1관
관람인원 : 30명
관람좌석 : d6

**Buttons:**

- 후기작성 (Review Writing) button with a checkmark icon.
- 예매취소 (Cancel Reservation) button.

**Page Footer:**

© CodingDoit corp.

# 7. 사용자 화면 정의서

## < 드로우 예매 >

**T-CATCH**

고객센터 로그인 회원가입

검색어 입력...

홈 시사회 뮤지컬 연극 콘서트

**MyPage**

- 내 정보
- 예매내역
- 드로우내역
- 내 후기
- 내 문의내역

드로우번호 : 27  
제목 : 스톰 보이  
관람일 : 2023-03-10 20:00:00  
상영관 : 메가박스 아트나인  
관람좌석 : none

드로우번호 : 30  
제목 : 더 퍼스트 슬램덩크  
관람일 : 2023-03-08 10:00:00  
상영관 : 메가박스 코엑스  
관람좌석 : none

드로우번호 : 1  
제목 : 그래비티  
관람일 : 2023-03-07 20:00:00  
상영관 : CGV 용산  
관람좌석 : a1

**결제하기**

# 7. 사용자 화면 정의서

## < 마이페이지 - 내후기 >

The screenshot displays the 'MyPage' interface, specifically the '내후기' (Reviews) section. On the left, a sidebar menu includes 'MyPage', '내 정보', '예매내역', '드로우내역', and '내후기'. The main area shows two review entries in a card format.

**Review 1:**

- 별점: 5
- 작성자: mj1234
- 관람일: 2022/11/24 20:00
- 작품명: 그래비티
- 내용: 우주에 떠다니는 느낌이 들었어요!
- Buttons: 수정 (blue), 삭제 (red)

**Review 2:**

- 별점: 4
- 작성자: mj1234
- 관람일: 2022/11/25 19:00
- 작품명: 킹키부츠
- 내용: 몰입감 최고! 팝콘 2개 필수!
- Buttons: 수정 (blue), 삭제 (red)

# 7. 사용자 화면 정의서

## < 고객센터 >

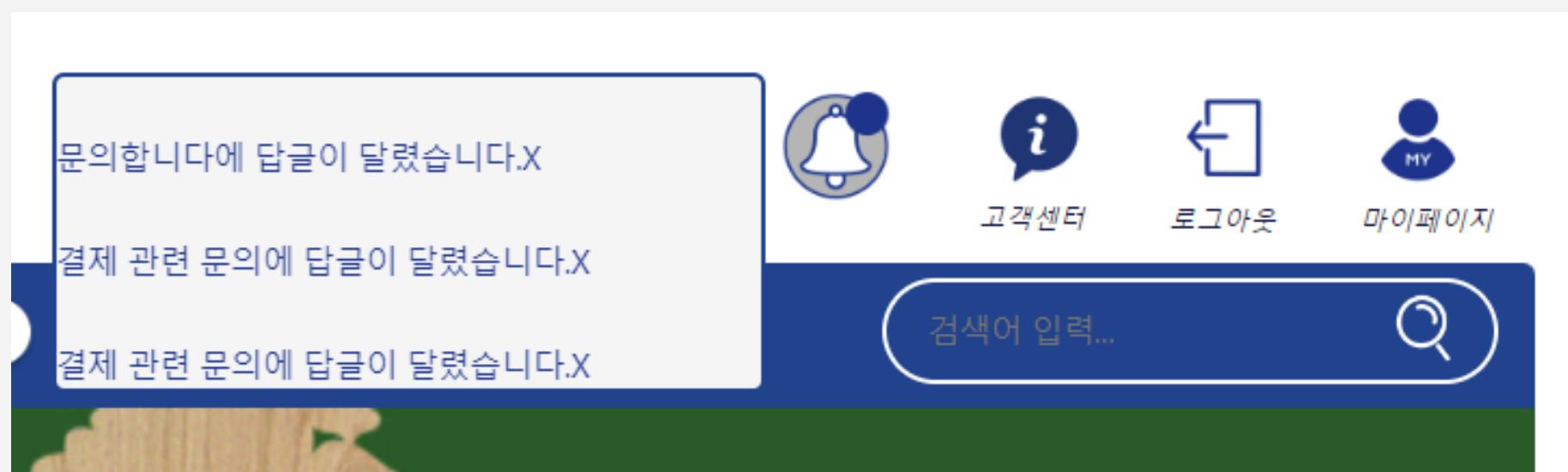
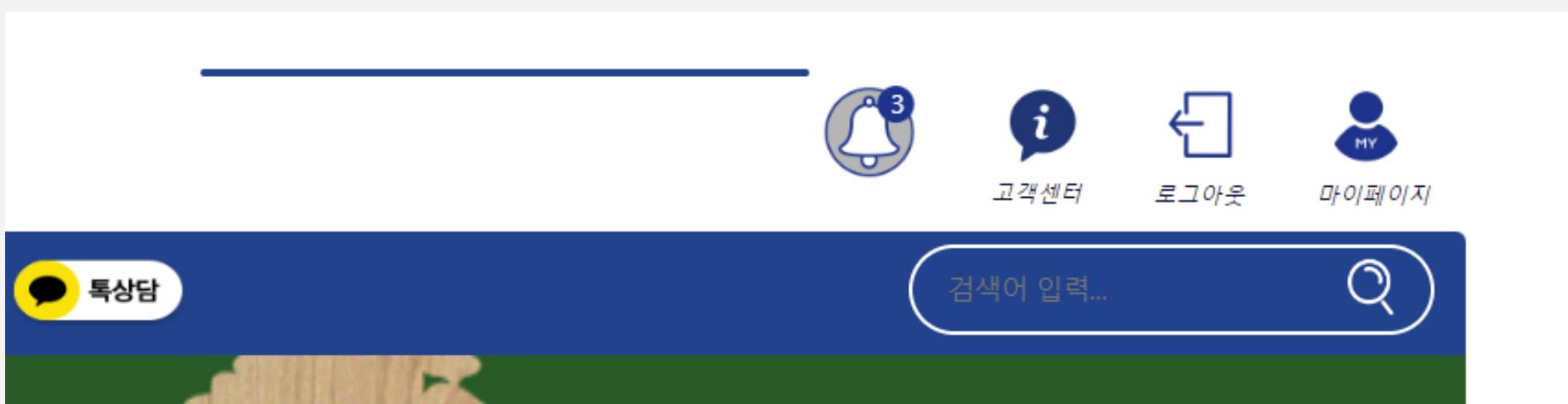
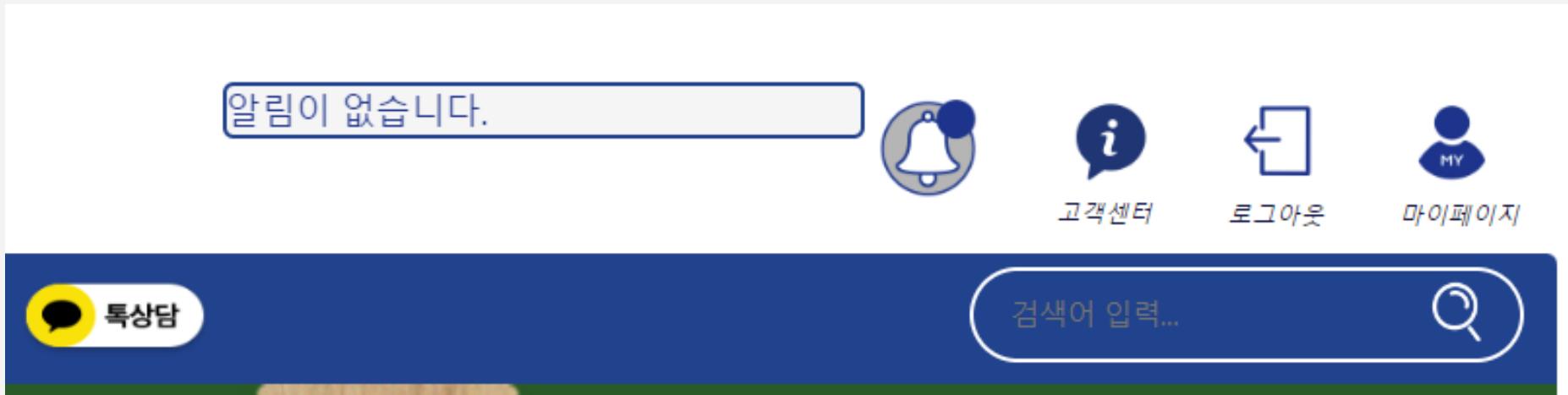
The screenshot shows the 'Customer Service' (고객센터) section of a website. The top navigation bar includes links for Home (홈), Social Events (시사회), Music (뮤지컬), Performances (연극), Concerts (콘서트), a Chat icon (톡상담), and a search bar. The main content area is titled '공지사항' (Announcements). On the left, a sidebar lists categories: '고객센터', '공지사항', 'Q&A', '카테고리', '전체', '예매/드로우', '결제/환불', and '기타'. The '카테고리' section is highlighted with a blue background. The '공지사항' section displays a single announcement table:

글번호	분류	제목	조회수	등록일
182	예매/드로우	d	7	2023/03/07 09:11

At the bottom of the page are search and filter controls: a dropdown menu labeled '제목', a search input field, and a button labeled '검색 초기화'.

# 7. 사용자 화면 정의서

## < 알람 >



# 7. 사용자 화면 정의서

## < 관리자 - 작품 등록 >

### 작품 등록하기

티켓ID

장르 **영화** ▾

장소  주소  주소 검색

위도  경도

장소유형 **a** ▾

이름  가격

날짜 연도-월-일  -- --:--

내용

비디오  제한연령

러닝타임

캐스트

이미지  선택된 파일 없음

등록

취소

# 7. 사용자 화면 정의서

## < 관리자 - 작품 관리 >

### 작품 리스트

ticketid	ticket_name	cateid	price	ticket_date	삭제
0	<a href="#">ticketid0</a>	시사회	0	2022-02-02 00:00:00	<a href="#">삭제하기</a>
1	<a href="#">에브리씽 에브리웨어 올 앤 원스</a>	시사회	0	2022-11-25 15:10:00	<a href="#">삭제하기</a>
2	<a href="#">그래비티</a>	시사회	0	2022-11-24 20:00:00	<a href="#">삭제하기</a>
3	<a href="#">헤어질 결심</a>	시사회	0	2022-11-26 19:00:00	<a href="#">삭제하기</a>
4	<a href="#">사랑할 땐 누구나 최악이 된다</a>	시사회	0	2022-10-23 16:00:00	<a href="#">삭제하기</a>
5	<a href="#">썸머 필름을 타고!</a>	시사회	0	2022-08-13 15:00:00	<a href="#">삭제하기</a>
6	<a href="#">애프터 양</a>	시사회	0	2022-11-25 17:00:00	<a href="#">삭제하기</a>
7	<a href="#">벌새</a>	시사회	0	2022-12-12 18:00:00	<a href="#">삭제하기</a>
8	<a href="#">고양이를 부탁해</a>	시사회	0	2022-12-24 16:00:00	<a href="#">삭제하기</a>
9	<a href="#">옥자</a>	시사회	0	2022-12-07 20:00:00	<a href="#">삭제하기</a>

검색

추가하기

처음 1 2 3 4 5 다음 끝

# 7. 사용자 화면 정의서

## < 관리자 - 회원 관리 >

### 고객 리스트

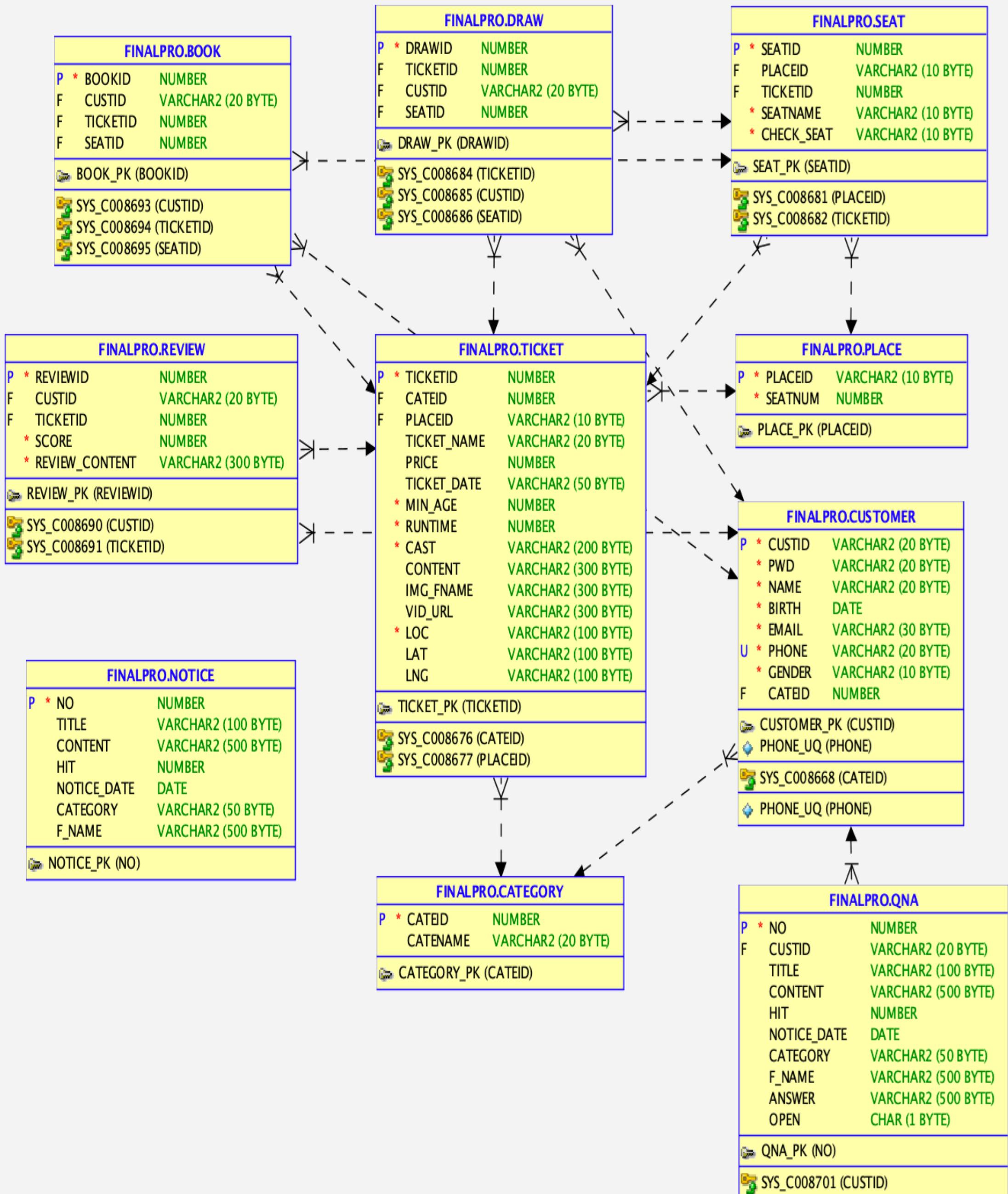
custid	name	birth	phone	gender	예매내역	Q&A내역	삭제
<a href="#">admin</a>	관리자	1995/02/03	010-2039-4356	여자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">asd123</a>	ym	19961225	01041418894	남자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">es1234</a>	황은선	1995/04/14	010-3456-3456	여자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">gh1234</a>	유근형	1994/09/18	010-4567-4567	남자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">kim3322</a>	김고운	1995/02/03	01020394355	여자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">kim3323</a>	김고운	1995/02/03	01020394356	여자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">mj1234</a>	이명진	1996/03/11	010-1234-1234	여자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">yk1234</a>	신윤경	1998/01/01	010-2345-2345	여자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>
<a href="#">ym1234</a>	조영민	1998/03/29	010-5678-5678	남자	<a href="#">내역보기</a>	<a href="#">내역보기</a>	<a href="#">삭제하기</a>

검색

처음 1 끝

# 8. ER-다이어그램, 테이블 명세서

## ER-다이어그램



# 8. ER-다이어그램, 테이블 명세서

## 테이블 명세서

### - CATEGORY

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	cateid	number		Y	N		pk
2	catename	varchar2	20	N	N		

### - CUSTOMER

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	custid	varchar2	20	Y	N		pk
2	cateid	number		N	N		fk
3	pwd	varchar2	20	N	N		
4	name	varchar2	20	N	N		
5	birth	varchar2	20	N	N		
6	phone	varchar2	20	Y	N		
7	email	varchar2	50	Y	N		
8	addr_postcode	varchar2	300	N	N		
9	addr_address	varchar2	100	N	Y		
10	addr_detail	varchar2	100	N	Y		
11	addr_extra	varchar2	100	N	Y		
12	gender	varchar2	10	N	N		
13	role	varchar2	10	N	N	default 'customer'	

# 8. ER-다이어그램, 테이블 명세서

## - PLACE

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	placeid	varchar2	10	Y	N	a,b,c	pk
2	seatnum	number		N	N		

## - TICKET

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	ticketid	number		Y	N		pk
2	cateid	number		N	N		fk
3	placeid	varchar2	10	N	N		fk
4	ticket_name	varchar2	50	N	N		
5	price	number		N	N		
6	ticket_date	date		N	N		
7	content	varchar2	3000	N	N		
8	img_fname	varchar2	300	N	N		
9	img_fname_main	varchar2	300	N	Y		
10	vid_url	varchar2	100	N	Y		
11	min_age	number		N	N		
12	runtime	number		N	N		
13	cast	varchar2	200	N	N		
14	loc	varchar2	100	N	N		
15	lat	number		N	Y		
16	lng	number		N	Y		

# 8. ER-다이어그램, 테이블 명세서

## - SEAT

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	seatid	number		Y	N	sequence	pk
2	placeid	Varchar2	10	N	N		fk
3	ticketid	number		N	N		fk
4	seatname	varchar2	10	N	N		
5	check_seat	char		N	N	default 'n'	

## - BOOK

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	bookid	number		Y	N	sequence	pk
2	custid	varchar2	20	N	N		fk
3	ticketid	number		N	N		fk
4	seatid	number		N	N		fk

## - DRAW

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	drawid	number		Y	N	sequence	pk
2	custid	varchar2	20	N	N		fk
3	ticketid	number		N	N		fk
4	seatid	number		N	N		fk

# 8. ER-다이어그램, 테이블 명세서

## - REVIEW

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	reviewid	number		Y	N	sequence	pk
2	custid	varchar2	20	N	N		fk
3	ticketid	number		N	N		fk
4	score	number		N	N		
5	review_content	varchar2	500	N	N		

## - NOTICE

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	notice_no	number		Y	N	sequence	pk
2	notice_category	varchar2	20	N	N		
3	notice_title	varchar2	100	N	N		
4	notice_content	varchar2	1000	N	N		
5	notice_hit	number		N	N	default 0	
6	notice_date	date		N	N	default sysdate	
7	notice_fname	varchar2	300	N	Y		

# 8. ER-다이어그램, 테이블 명세서

## - QNA

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	qna_no	number		Y	N	sequence	pk
2	custid	varchar2	100	N	N		fk
3	ticketid	number		N	N		fk
4	qna_category	varchar2	20	N	N		
5	qna_title	varchar2	100	N	N		
6	qna_content	varchar2	1000	N	N		
7	qna_hit	number		N	N	default 0	
8	qna_date	date		N	N	default sysdate	
9	qna_fname	varchar2	300	N	Y		
10	qna_open	char		N	N		
11	qna_answer	varchar2	1000	N	Y		

## - NOTIFICATION

NO	컬럼명	자료형	크기	유일키	NULL 허용	인덱스	키
1	notif_no	number		Y	N	sequence	pk
2	custid	varchar2	20	N	N		fk
3	qna_no	number		N	N		fk
4	notif_date	date		N	N	default sysdate	
5	checked	char		N	N	default 'n'	

# 9. 클래스 명세서, 서비스 명세서

## 클래스 명세서

		vo/entity	db	dao/service	controller
테이블	Category	CategoryVO/ Category	Category Mapper	Category DAO/Service	
	Customer	CustomerVO/ Customer	Customer Mapper	Customer DAO/Service	Customer Controller
	Place	PlaceVO/ Place	Place Mapper	Place DAO/Service	
	Ticket	TicketVO/ Ticket	Ticket Mapper	Ticket DAO/Service	Ticket Controller
	Seat	SeatVO/ Seat	Seat Mapper	Seat DAO/Service	Seat Controller
	Book	BookVO/ Book	Book Mapper	Book DAO/Service	Book Controller
	Draw	DrawVO/ Draw	Draw Mapper	Draw DAO/Service	Draw Controller
	Review	ReviewVO/ Review	Review Mapper	Review DAO/Service	Review Controller
	Notice	NoticeVO/ Notice	Notice Mapper	Notice DAO/Service	Notice Controller
	Qna	QnaVO/ Qna	Qna Mapper	Qna DAO/Service	Qna Controller
기능	Notification	NotificationVO	Notification Mapper		
	랭킹	RankingVO	Ranking Mapper		
	문자 전송			Send Message	BookSMS Controller
	메일 전송			Email Service	Email Controller
	페이지/관리자			Paging	Admin Controller

# 9. 클래스 명세서, 서비스 명세서

## 서비스 명세서

	기능	서블릿 클래스명	메소드명
메인 페이지	랭킹 출력	RankingTicket.java	reviewdao.ranking()
	현재 상영작/개봉 예정작 출력	SelectTicketMain.java	ticketdao.selectTicketByCategory()
로그인	로그인	LoginCustomer.java	customerdao.login()
아이디 찾기	아이디 찾기	FindCustomerId.java	dao.findCustid()
비밀번호 찾기	비밀번호 찾기	FindCustomerPwd.java	dao.findPwd()
회원가입	회원가입	SignupCustomer.java	customerdao.insertCustomer()
	아이디 중복 확인	ConfirmCustomerId.java	customerdao.confirmCustomerID()
	전화번호 중복 확인	ConfirmCustomerPhone.java	customerdao.confirmCustomerPhone()
장르별 페이지	장르별, 현재 상영작/개봉 상영작/지 난 상영작 출력	SelectTicketMain.java	ticketdao.selectTicketByCategory()
상세 페이지	작품 상제정보 출력	DetailAction.java	dao.findById
	오픈일에 예매버튼 활성화	GetOpendateAction.java	ticketDAO.openDate
	남은 좌석수	LeftSeatNumAction.java	dao.leftSeat
	서버 시간 제공	GetServerTime.java	dao.nowTime
	작품 후기 출력	TicketReviewAction.java	dao.findByTicketid
	작품 평점 출력	AvgScoreAction.java	dao.avgScore
	예매자 성별 통계	CountGenderAction.java	dao.countGender
	예매자 세대별 통계	CountGenerationAction.java	dao.countGeneration
	서버시간 제공	GetServerTimeAction.java	dao.nowTime
예매 페이지	예매 좌석 화면에 나타내기	ListSeat.java	dao.listSeatByTicketid()
	예매하기	RegistBook.java	bd.registBook(), sd.registSeat()
검색 페이지	검색 결과 출력	SearchTicket.java	ticketdao.searchTicket
마이 페이지 - 내 정보	회원정보 출력	GetCustomerAction.java	customerDAO.findById()
	회원정보 수정	UpdateCustomerAction.java	customerDAO.updateCustomer()
마이 페이지 - 예매 내역	예매내역 출력	BookByCustid.java	ticketdao.selectTicketByCustid()
	예매내역 삭제	DeleteBook.java	bookdao.deleteBook()
	후기 작성 및 등록	InsertReview.java	dao.getNextReviewid()
마이 페이지 - 내 후기	후기내역 출력	GetReviewAction.java	dao.findByCustid()
	후기내역 수정	UpdateReviewAction.java	dao.updateReview()
	후기내역 삭제	deleteReviewAction.java	dao.deleteReview()

# 10. SQL 명세서, 인스턴스 명세서

---

## SQL 명세서 (1)

### – 사용자 생성 SQL

```
create user c##final identified by final123;  
grant connect,resource,dba to c##final;
```

### – 테이블 생성 SQL (1)

```
create table category(  
    cateid number primary key,  
    catename varchar2(20) not null  
)
```

```
create table customer(  
    custid varchar2(20) primary key,  
    cateid number references category(cateid) not null,  
    pwd varchar2(20) not null,  
    name varchar2(20) not null,  
    birth varchar2(20) not null,  
    phone varchar2(20) not null,  
    email varchar2(50) not null,  
    addr_postcode varchar2(300) not null,  
    addr_address varchar2(100),  
    addr_detail varchar2(100),  
    addr_extra varchar2(100),  
    gender varchar2(10) not null,  
    role varchar2(10) default 'customer',  
    constraints phone_uq unique (phone),  
    constraints email_uq unique (email)  
)
```

```
create table place(  
    placeid varchar2(10) primary key,  
    seatnum number not null  
)
```

# 10. SQL 명세서, 인스턴스 명세서

## - 테이블 생성 SQL (2)

```
create table ticket(
    ticketid number primary key,
    cateid number references category(cateid) not null,
    placeid varchar2(10) references place(placeid) not null,
    ticket_name varchar2(50) not null,
    price number not null,
    ticket_date date not null,
    content varchar2(3000) not null,
    img_fname varchar2(300) not null,
    img_fname_main varchar2(300),
    vid_url varchar2(100),
    min_age number not null,
    runtime number not null,
    cast varchar2(200) not null,
    loc varchar2(100) not null,
    lat number,
    lng number
);
```

```
create table seat(
    seatid number primary key,
    placeid varchar2(10) references place(placeid) not null,
    ticketid number references ticket(ticketid) not null,
    seatname varchar2(10) not null,
    check_seat char default 'n'
);
```

```
create table book(
    bookid number primary key,
    custid varchar2(20) references customer(custid) not null,
    ticketid number references ticket(ticketid) not null,
    seatid number references seat(seatid) not null
);
```

```
create table draw(
    drawid number primary key,
    custid varchar2(20) references customer(custid) not null,
    ticketid number references ticket(ticketid) not null,
    seatid number references seat(seatid) not null
);
```

```
create table review(
    reviewid number primary key,
    custid varchar2(20) references customer(custid) not null,
    ticketid number references ticket(ticketid) not null,
    score number not null,
    review_content varchar2(500) not null
);
```

# 10. SQL 명세서, 인스턴스 명세서

## – 테이블 생성 SQL (3)

```
create table notice(
    notice_no number primary key,
    notice_category varchar2(20) not null,
    notice_title varchar2(100) not null,
    notice_content varchar2(1000) not null,
    notice_hit number default 0,
    notice_date date default sysdate,
    notice_fname varchar2(300)
);

create table qna(
    qna_no number primary key,
    custid varchar2(20) references customer(custid) not null,
    ticketid number references ticket(ticketid) not null,
    qna_category varchar2(20) not null,
    qna_title varchar2(100) not null,
    qna_content varchar2(1000) not null,
    qna_hit number default 0,
    qna_date date default sysdate,
    qna_fname varchar2(300),
    qna_open char not null,
    qna_answer varchar2(1000)
);

create table notification(
    notif_no number primary key,
    custid varchar2(20) references customer(custid) not null,
    qna_no number references qna(qna_no) not null,
    notif_date date default sysdate,
    checked char default 'n'
);
```

## – 시퀀스 생성 SQL

```
create sequence seq_seatid;
create sequence seq_bookid;
create sequence seq_drawid;
create sequence seq_reviewid;
create sequence seq_notice_no;
create sequence seq_qna_no;
create sequence seq_notif_no;
```

# 10. SQL 명세서, 인스턴스 명세서

## SQL 명세서 (2)

### – CustomerMapper

<!-- page에 따라 startRecord와 endRecord 구해서 출력하기 -->

<!-- search에 따라 출력하기 -->

<!-- order 기능 추가 -->

```
<select id="findCustomerPagingSearchOrderBy" resultType="customerVO" parameterType="java.util.HashMap">
    select * from
        (select rownum n, a.* from
            (select * from customer
                <if test="keyword != null and keyword != "">
                    where (custid like '%'||#{keyword}||'%' or name like '%'||#{keyword}||'%')
                </if>
                order by ${order}) a)
            where n between #{startRecord} and #{endRecord}
    </select>
```

<!-- admin에서 customer의 목록을 페이징 처리하기 위해 customer의 totalRecord 숫자 구하기 -->

```
<select id="getTotalRecord" resultType="java.lang.Integer">
    select count(*) from customer
    <if test="keyword != null and keyword != "">
        where (custid like '%'||#{keyword}||'%' or name like '%'||#{keyword}||'%')
    </if>
    </select>
```

<!-- findCustid 아이디 찾기 -->

```
<select id="findCustid" resultType="customerVO">
    select custid from customer where name=#{name} and phone=#{phone}
</select>
```

<!-- findPwd 전화번호로 개인정보 확인 -->

```
<select id="checkByPhone" resultType="customerVO">
    select * from customer where custid=#{custid} and phone=#{phone}
</select>
```

<!-- 전화번호로 비밀번호 재설정 -->

```
<update id="updatePwdbyPhone" parameterType="customerVO">
    update customer set pwd=#{pwd} where custid=#{custid} and phone=#{phone}
</update>
```

# 10. SQL 명세서, 인스턴스 명세서

## – TicketMapper

<!-- main.html에서 카테고리, 시간 별로 상영작 출력하는 mapper sql 코드 -->

```
<select id="findAllTicketByCategory" resultType="ticketVO">
    select * from ticket where cateid=#{cateid} and
    <if test="time==0">
        ticket_date < to_char(sysdate, 'yyyy/mm/dd')
    </if>
    <if test="time==1">
        ticket_date > to_char(sysdate, 'yyyy/mm/dd') and ticket_date <= to_char(sysdate+14, 'yyyy/mm/dd')
    </if>
    <if test="time==2">
        ticket_date > to_char(sysdate+14, 'yyyy/mm/dd')
    </if>
</select>
```

<!-- main.html에서 카테고리, 시간 별로 상영작 출력하는 mapper sql 코드 -->

```
<select id="findAllTicketByCategoryPaging" resultType="ticketVO">
    select * from
    (select rownum n, a.* from
    (select * from ticket where cateid=#{cateid} and
    <if test="time==0">
        ticket_date < to_char(sysdate, 'yyyy/mm/dd')
    </if>
    <if test="time==1">
        ticket_date > to_char(sysdate, 'yyyy/mm/dd') and ticket_date <= to_char(sysdate+7, 'yyyy/mm/dd')
    </if>
    <if test="time==2">
        ticket_date > to_char(sysdate+7, 'yyyy/mm/dd')
    </if>
    order by ticketid) a)
    where n between #{startRecord} and #{endRecord}
</select>
```

<!-- keyword' 를 매개변수로 ticket 리스트 검색 결과를 출력하기 -->

```
<select id="findSearchTicket" resultType="ticketVO">
    select * from ticket where ticket_name like '%'||#{keyword}|| '%' or cast like '%'||#{keyword}|| '%'
</select>
```

# 10. SQL 명세서, 인스턴스 명세서

## – SeatMapper

```
<!-- 잔여좌석 출력 -->
<select id="selectLeftSeat" resultType="int">
    select count(*) from seat where ticketid=#{ticketid} and check_seat='n'
</select>
```

```
<!-- 드로우 잔여좌석 출력 -->
<select id="selectDrawLeftSeat" resultType="seatVO">
    select * from seat where ticketid=#{ticketid} and check_seat='n'
</select>
```

```
<!-- 좌석예매 -->
<update id="registSeat" parameterType="int">
    update seat set check_seat='y' where seatid=#{seatid}
</update>
```

```
<!-- 좌석취소 -->
<update id="cancleSeat" parameterType="int">
    update seat set check_seat='n' where seatid=#{seatid}
</update>
```

## – BookMapper

```
<!-- 성별통계 -->
<select id="countGender" resultType="countGenderVO">
    select gender,count(*) cnt from customer c, book b where c.custid = b.custid and b.ticketid=#{ticketid}
    group by gender
</select>
```

```
<!-- 세대통계 -->
<select id="countGeneration" resultType="countGenerationVO">
    select generation, count(*) num
    from (select custid,trunc((to_char(sysdate,'yyyy')-substr(birth,0,4)+1)/10)*10 generation from customer) c, book b
    where c.custid=b.custid and b.ticketid=#{ticketid} group by generation
</select>
```

```
<!-- QNA insert화면에 티켓 목록 출력을 위한 sql -->
<select id="findTicketidByCuistid" parameterType="java.lang.String" resultType="java.lang.Integer">
    select ticketid from book where custid=#{custid}
</select>
```

```
<!-- 예약 확정 문자 보내기 위한 ajax를 위한 bookid 찾기 -->
<select id="findBookidByOthers" parameterType="bookVO" resultType="java.lang.Integer">
    select bookid from book where custid=#{custid} and ticketid=#{ticketid} and seatid=#{seatid}
</select>
```

# 10. SQL 명세서, 인스턴스 명세서

## – DrawMapper

```
<!-- 드로우 신청 -->
<insert id="drawInsert" parameterType="drawVO">
    insert into draw(drawid,custid,ticketid) values(seq_drawid.nextval,#{custid},#{ticketid})
</insert>
```

```
<!-- 드로우 중복 신청 방지 -->
<select id="selectDrawNoSame" resultType="drawVO">
    select drawid,ticketid,custid from draw where ticketid=#{ticketid} and custid=#{custid}
</select>
```

## – ReviewMapper

```
<!-- 리뷰 스코어로 정렬 -->
<select id="findReviewByTicketid" resultType="reviewVO">
    select * from review where ticketid=#{ticketid}
    <if test="re>0">
        order by score desc
    </if>
    <if test="0>re">
        order by score
    </if>
    <if test="re=0">
        order by score desc
    </if>
</select>
```

```
<!-- 리뷰내역이 있나 확인 -->
<select id="checkReview" resultType="java.lang.Integer" parameterType="reviewVO">
    select count(*) from review r, ticket t where t.ticketid=r.ticketid and r.ticketid=#{ticketid} and
    r.custid=#{custid}
</select>
```

```
<!-- 평균평점 구하기 -->
<select id="findAvgScore" resultType="int">
    select avg(score) from review where ticketid=#{ticketid}
</select>
```

# 10. SQL 명세서, 인스턴스 명세서

## – NoticeMapper

```
<!-- 조회수 증가 -->
<update id="updateHit" parameterType="java.lang.Integer">
    update notice set notice_hit=notice_hit+1 where notice_no=#{notice_no}
</update>

<!-- 페이지를 위한 전체 레코드 카운트 -->
<select id="getTotalRecord" resultType="java.lang.Integer" parameterType="java.util.HashMap">
    select count(notice_no) from notice
    <if test="category!=null">
        where notice_category=#{category}
        <if test="keyword!=null">
            and ${searchColumn} like '%' || #{keyword} || '%'
        </if>
    </if>
    <if test="category==null">
        <if test="keyword!=null">
            where ${searchColumn} like '%' || #{keyword} || '%'
        </if>
    </if>
</select>

<!-- 페이지를 위한 전체 레코드 카운트 -->
<!-- 공지사항 게시판 전체 레코드 출력 -->
<select id="findAll" resultType="noticeVO" parameterType="java.util.HashMap">
    select notice_no, notice_category, notice_title, notice_content, notice_hit,
    notice_date, notice_fname from (select rownum r, a.* from
    (select * from notice
        <if test="category!=null">
            where notice_category=#{category}
            <if test="keyword!=null">
                and ${searchColumn} like '%' || #{keyword} || '%'
            </if>
        </if>
        <if test="category==null">
            <if test="keyword!=null">
                where ${searchColumn} like '%' || #{keyword} || '%'
            </if>
        </if>
        order by NOTICE_DATE desc) a)
        where r between #{startNo} and #{endNo}
</select>
```

# 10. SQL 명세서, 인스턴스 명세서

## – QnaMapper

```
<!-- custid 고객 아이디에 따라 작성한 qna 목록 출력하기 -->
<select id="selectQnaByCustid" resultType="qnaVO" parameterType="java.util.HashMap">
    select * from
    (select rownum n, a.* from
    (select * from qna where custid=#{custid} order by qna_no) a)
    where n between #{startRecord} and #{endRecord}
</select>
```

```
<!-- myPage에서 custid에 따른 qna Record 수자를 구하기 -->
<select id="getTotalQnaRecord" resultType="java.lang.Integer">
    select count(*) from qna where custid=#{custid}
</select>
```

## – NotificationMapper

```
<!-- 체크 안 된 알림 갯수 세기 -->
<select id="countNChecked" resultType="java.lang.Integer">
    select count(notif_no) from notification where checked='n'
</select>
```

```
<!-- 'checked' 칼럼 업데이트 ('n'에서 'y'로) -->
<update id="updateCheckedToY">
    update notification set checked='y' where checked='n'
</update>
```

## – RankingMapper

```
<!-- ranking을 출력하는 코드 -->
<select id="findAllRankingOrderByScore" resultType="rankingVO">
    select t.ticketid, avg(score), (select img_fname from ticket a where a.ticketid=t.ticketid) as img_루문
    from review r, ticket t
    where t.ticketid=r.ticketid and cateid=#{cateid}
    group by t.ticketid having avg(score)>0
    order by avg(score) desc
</select>
```

# 10. SQL 명세서, 인스턴스 명세서

## 인스턴스 명세서

### - Category

```
insert into category(cateid,catename) values(1,'시사회');  
insert into category(cateid,catename) values(2,'뮤지컬');  
insert into category(cateid,catename) values(3,'연극');  
insert into category(cateid,catename) values(4,'콘서트');
```

### - Customer

\* Spring Security를 이용하기 때문에 DB에 직접 입력할 수 없음

```
insert into  
customer(custid,cateid,pwd,name,birth,phone,email,addr_postcode,addr_address,addr_detail,addr_extra,gender)  
values('gu1234',1,'Rhdms@11','김고운','1995/02/09','01011111111','rhdns@google.com','서울','여자');  
insert into  
customer(custid,cateid,pwd,name,birth,phone,email,addr_postcode,addr_address,addr_detail,addr_extra,gender)  
values('yk1234',2,'Dbsrud@11','신윤경','1995/04/09','01022222222','dbsrud@naver.com','인천','여자');  
insert into  
customer(custid,cateid,pwd,name,birth,phone,email,addr_postcode,addr_address,addr_detail,addr_extra,gender)  
values('mj1234',3,'Audwls@11','이명진','1999/01/09','01033333333','audwls@google.com','서울','여자');  
insert into  
customer(custid,cateid,pwd,name,birth,phone,email,addr_postcode,addr_address,addr_detail,addr_extra,gender)  
values('gh1234',4,'Rmsgud@11','유근형','1994/05/09','01044444444','rmsgud@naver.com','서울','남자');  
insert into  
customer(custid,cateid,pwd,name,birth,phone,email,addr_postcode,addr_address,addr_detail,addr_extra,gender)  
values('ym1234',1,'Dudals@11','조영민','1996/09/09','01055555555','dudals@google.com','서울','남자');  
insert into  
customer(custid,cateid,pwd,name,birth,phone,email,addr_postcode,addr_address,addr_detail,addr_extra,gender)  
values('es1234',1,'Dmstjs@11','황은선','1992/07/09','01066666666','haesam92@naver.com','서울','여자');
```

\* 관리자 계정의 경우, View를 통해 가입한 후 CMD창에서 Custid와 Role 변경

```
insert into  
customer(custid,cateid,pwd,name,birth,phone,email,addr_postcode,addr_address,addr_detail,addr_extra,gender)  
values('admin1',1,'Admin@11','관리자','1999/09/09','01012345678','codingdoit.tcatch@google.com','04001','서울  
마포구 월드컵북로 21','2층 풍성빌딩','(서교동)','여자');  
update customer set custid='admin' and role='admin' where custid='admin1';
```

# 10. SQL 명세서, 인스턴스 명세서

## – Place

```
insert into place(placeid,seatnum) values('a',30);
insert into place(placeid,seatnum) values('b',10);
insert into place(placeid,seatnum) values('c',15);
```

## – Ticket

\* Qna 페이지에서 문의 카테고리 선택시 필요

```
insert into
```

```
ticket(ticketid,cateid,placeid,ticket_name,price,ticket_date,content,img_fname,min_age,runtime,cast,loc)
values(0,1,'a','ticketid0',0,
to_date('2022/08/08 09:00:00','yyyy/mm/dd hh24:mi:ss'),'ticketid0','ticketid0',0,0,'ticketid0','ticketid0');
```

```
insert into
```

```
ticket(ticketid,cateid,placeid,ticket_name,price,ticket_date,content,img_fname,img_fname_main,vid_url,
min_age,runtime,cast,loc,lat,lng)
```

```
values(
```

```
1,1,'b',
```

```
'에브리씽 에브리웨어 올 앤 원스',0,to_date('2022/11/25 15:10:00','yyyy/mm/dd hh24:mi:ss'),
```

'미국에 이민와 힘겹게 세탁소를 운영하던 에블린은 세무당국의 조사에 시달리던 어느 날 남편의 이혼 요구와 빼딱하게 구는 딸로 인해 대혼란에 빠진다. 그 순간 에블린은 멀티버스 안에서 수천, 수만의 자신이 세상을 살아가고 있다는 사실을 알게 되고, 그 모든 능력을 빌려와 위기의 세상과 가족을 구해야하는 운명에 처한다.',

```
'movie_EveryEveryAll.jpg','everything_main_1.jpg','https://www.youtube.com/embed/vOXJySPfMJY',
```

```
15,139,'다니엘 콴(감독), 양자경, 스테파니 수, 키 호이 콴',
```

```
'CGV 명동',37.563450,126.983350
```

```
);
```

```
insert into
```

```
ticket(ticketid,cateid,placeid,ticket_name,price,ticket_date,content,img_fname,vid_url,
```

```
min_age,runtime,cast,loc,lat,lng)
```

```
values(
```

```
51,2,'b','물랑루즈',160000,to_date('2023/03/05 17:00:00','yyyy/mm/dd hh24:mi:ss'),
```

'물랑루즈 최고의 스타 사틴과 무명 작곡가 크리스티안은 우연이 빚어낸 운명적 만남으로 헤어 나올 수 없는 사랑에 빠진다. 하지만 물랑루즈의 단장인 지들러와 사틴을 탐하는 귀족 몬로스 공작에 의해 그 사랑은 위기를 맞는다. 크리스티안은 낭만적인 보헤미안 로트렉과 함께, 재정 위기에 빠진 물랑루즈를 구하고 사틴의 마음까지 사로잡을 공연 <보헤미안 랩소디>를 제작하기로 한다. 그런데 그 이야기는 사틴, 크리스티안, 몬로스 공작의 엇갈린 관계를 거울처럼 비추는데...!',

```
'moulinrouge.jpg','https://youtu.be/qMghyN4kqwo',
```

```
10,170,'크리스티안: 홍광호, 사틴: 아이비, 몬로스 공작: 손준호',
```

```
'블루스퀘어',37.541183,127.002749
```

```
);
```

# 10. SQL 명세서, 인스턴스 명세서

---

## – Seat

```
insert into seat(seatid,placeid,ticketid,check_seat,seatname) values(1,'a',26,'y','a1');  
insert into seat(seatid,placeid,ticketid,check_seat,seatname) values(2,'a',26,'y','a2');  
insert into seat(seatid,placeid,ticketid,check_seat,seatname) values(3,'a',26,'y','a3');
```

## – Book

```
insert into book(bookid,custid,ticketid,seatid) values(1,'mj1234',26,1);  
insert into book(bookid,custid,ticketid,seatid) values(2,'yk1234',26,2);  
insert into book(bookid,custid,ticketid,seatid) values(3,'es1234',26,3);
```

## – Draw

```
insert into draw(drawid,ticketid,custid,seatid) values(1,2,'es1234',31);
```

## – Review

```
insert into review(reviewid,custid,ticketid,score,review_content)  
values(1,'mj1234',1,4,'몰입감 최고! 팝콘 2개 필수!');  
insert into review(reviewid,custid,ticketid,score,review_content)  
values(2,'yk1234',1,1,'별로에요');  
insert into review(reviewid,custid,ticketid,score,review_content)  
values(3,'mj1234',25,3,'보통이에요');
```

# 11. 주요 코드

## Draw

### – DrawController (1)

```
47     // 잔여좌석이 0이거나 cateid=1일 때 드로우 버튼 활성화
48     @RequestMapping("/DrawButtonOpen")
49     @ResponseBody
50     public TicketVO openDrawButton(int ticketid) {
51         System.out.println("ticket번호:"+ticketid);
52         return DBManager.findByTicketid(ticketid);
53     }
54
55     //드로우 신청한 회원정보 가져오기
56     @GetMapping("/drawList")
57     @ResponseBody
58     public List<DrawVO> drawList(int ticketid){
59         List<DrawVO> draw = null;
60         draw = DBManager.drawTest(ticketid);
61         return draw;
62     }
63
64     // 결제완료하면 드로우 인스턴스 삭제
65     @RequestMapping("/drawDelete")
66     @ResponseBody
67     public int drawDelete(int drawid){
68         return DBManager.drawDelete(drawid);
69     }
70
71     public int[] leftSeat(int ticketid){
72         List<SeatVO> seat = DBManager.drawLeftSeat(ticketid);          //seatTable에 남은 좌석 정보 가져오기
73         int seatArr[] = new int[seat.size()];
74         for(int i = 0; i < seat.size(); i++){
75             seatArr[i] = seat.get(i).getSeatid();
76             System.out.println(seatArr[i]);
77         }
78         return seatArr;
79     }
80 }
```

# 11. 주요 코드

## – DrawController (2)

```
81 //드로우 가동
82 @GetMapping("/drawExec")
83 @ResponseBody
84 public String drawExec(int ticketid){
85     int count = DBManager.findLeftSeatByTicketid(ticketid);
86     List<DrawVO> draw = drawList(ticketid);
87     System.out.println("드로우 사이즈:"+draw.size());
88     String custidArr[] = new String[count];           //드로우에 당첨된 회원아이디 저장
89     Random r = new Random();
90
91     int seatArr[] = leftSeat(ticketid);
92
93
94     lucker = new String[count];                      //전역변수에 저장하여 드로우 된 회원들을 저장
95
96     LinkedList<DrawVO> list = new LinkedList<>();
97
98     for(DrawVO d : draw){
99         list.add(d);
100    }
101
102    for (int i = 0; i < count; i++){
103        int number = r.nextInt(list.size());          //랜덤함수를 드로우를 신청한 회원 수 만큼의 크기로 설정
104        custidArr[i] = list.get(number).getCustid();   //랜덤함수에 배정된 회원아이디를 배열에 저장
105        lucker[i] = custidArr[i];
106        list.remove(number);                         //당첨된 회원은 linkedList에서 제거하고 반복문 가동
107    }
108
109
110    DBManager.drawDeleteSeatId(ticketid);           //여러번 드로우 할 경우를 대비해 드로우 전 draw테이블에 당첨여부 초기화
111    for(int i = 0; i < count; i++){
112        DBManager.drawUpdate(lucker[i], seatArr[i]);  //당첨된 회원이름과 남은 좌석을 draw 테이블에 반영
113    }
114    return "success";
115 }
116 }
```

# 11. 주요 코드

## - ScheduledDraw

: 상영일 전날 오후 12시에 자동으로 드로우 추첨

```
28 //매월 오후 12시에 다음날이 상영일인 상품에 대해 드로우 진행
29 @Scheduled(cron = "0 0 12 * * ?")
30 public void draw(){
31     List<Ticket> tickets = ticketDAO.findAll();
32     int toYear = date.getYear()+1900;
33     int toMonth = date.getMonth()+1;
34     int toDay = date.getDate();
35     for(Ticket t : tickets){
36
37         String date = t.getTicket_date().substring(0,10);          //년 월 일 까지만 잘라준다.
38         String arr[] = date.split("-");                         // 년 월 일 기준으로 잘라 배열에 저장.
39
40         // 현재 년 월이 일치하고 상영일이 today의 다음 날일때 draw 실행.
41         if(toYear == Integer.parseInt(arr[0]) && toMonth == Integer.parseInt(arr[1]) && toDay == Integer.parseInt(arr[2])-1{
42             System.out.println("draw실행:"+ t.getTicketid());
43             drawController.drawExec(t.getTicketid());
44         }
45     }
46 }
47 }
```

# 11. 주요 코드

## 모듈화

### - SearchService (1)

: Notice와 Qna 페이지 내에서 검색하는 기능이 비슷해 모듈화

```
13     HashMap<String, Object> hashMap=new HashMap<String, Object>();
14
15     // 카테고리별로 보기
16     if(category!=null) {
17         switch (category) {
18             case "book":
19                 category = "예매/드로우";
20                 break;
21             case "transaction":
22                 category = "결제/환불";
23                 break;
24             case "etc":
25                 category = "기타";
26                 break;
27         }
28
29         // 카테고리가 all이면 세션에 저장된 카테고리 지우기
30         if(category.equals("all")) {
31             if(session.getAttribute(tableName+"Category")!=null) {
32                 session.removeAttribute(tableName+"Category");
33             }
34             category = null;
35         }else{ // all이 아니라 특정 카테고리를 선택했을 경우
36             //카테고리를 세션에 저장한다.
37             session.setAttribute(tableName+"Category",category);
38             //세션에 원래 있던 카테는 필요없음.
39         }
40         // 카테고리를 클릭 안했을 경우
41     }else{
42         // 그 전에 클릭했던 게 있다면 (=세션에 저장되어 있다면)
43         if(session.getAttribute(tableName+"Category")!=null) {
44             //세션에 저장되어 있는 카테를 가져온다.
45             category = (String) session.getAttribute(tableName+"Category");
46
47             //둘다 없다면 => 카테고리별로 보기 x => null => 처리 필요 없음.
48         }
49     }
```

# 11. 주요 코드

## - SearchService (2)

: 필요한 데이터들은 map에 담아 전달

```
51         hashMap.put("category", category);
52
53     //검색
54     if(keyword!=null) {
55         session.setAttribute(tableName+"Keyword", keyword);
56         session.setAttribute(tableName+"SearchColumn", searchColumn);
57         hashMap.put("searchColumn", searchColumn);
58         hashMap.put("keyword", keyword);
59     }else{
60         if(session.getAttribute(tableName+"Keyword")!=null){
61             keyword= (String) session.getAttribute(tableName+"Keyword");
62             searchColumn= (String) session.getAttribute(tableName+"SearchColumn");
63             hashMap.put("searchColumn", searchColumn);
64             hashMap.put("keyword", keyword);
65         }
66     }
67     return hashMap;
68 }
69 }
```

# 11. 주요 코드

## - Paging

: 페이지 기능 모듈화

```
61     public Paging(int totalRecord, int page){  
62         // totalRecord = 총 record 숫자  
63         // page = 현재 페이지  
64  
65         // 현재 페이지  
66         setPage(page);  
67         // page2  
68         setPage2( (int)Math.ceil ((double)page / showingPageSize));  
69  
70         // Record 총 갯수  
71         setTotalRecord(totalRecord);  
72  
73         // page의 총 갯수  
74         // 총 페이지 숫자 (총 record 숫자를 한 페이지 최대 갯수로 나눈 뒤 올림 해준다)  
75         setTotalPage((int)Math.ceil(((double)totalRecord/pageSize)));  
76  
77         // start, end 레코드 계산  
78         setStartRecord((pageSize*page) - (pageSize-1));  
79         setEndRecord(page*pageSize);  
80  
81         // start, end 페이지 계산  
82         setStartPage((showingPageSize*page2) - (showingPageSize-1));  
83         setEndPage(page2*showingPageSize);  
84  
85         // 만약 totalPage가 6보다 작은, 예를 들어 3이면 1 2 3 페이지만 나오도록.  
86         // 그러면 3페이지가 마지막인데 endPage가 계산되어 5페이지까지 나옴  
87         if(totalPage < endPage){  
88             setEndPage(totalPage);  
89         }  
90     }
```

# 11. 주요 코드

## - CategoryNavSearch

: 내비게이션바(카테고리별/현재·미래·과거상영작/검색) 기능 모듈화

```
12     $(cate_current).click(function(e){  
13         var target = this.id;  
14         var cateid_num = $("#" + target).attr("cateid");  
15         var time_num = $("#" + target).attr("time");  
16  
17         time = time_num;  
18         cateid = cateid_num;  
19         console.log(cateid);  
20  
21         sessionStorage.setItem("time", time);  
22         sessionStorage.setItem("cateid", cateid);  
23  
24         var url = "/category?time="+time+"&cateid="+cateid;  
25         location.replace(url);  
26     });
```

```
60     // 검색하면 결과가 나오도록  
61     var searchKeyword = function(f){  
62         $("#search_txt").keydown(function(e){ // 엔터 누르면 검색되도록 (아직 구현 X)  
63             if(e.which == 13){  
64                 var keyword = $("#search_txt").val();  
65                 sessionStorage.setItem("keyword", keyword);  
66                 var url = "search?keyword="+keyword;  
67                 location.replace(url);  
68             }  
69         });  
70  
71         $("#search_btn").click(function(){ // 버튼 누르면 검색됨  
72             var keyword = $("#search_txt").val();  
73             sessionStorage.setItem("keyword", keyword);  
74             var url = "search?keyword="+keyword;  
75             location.replace(url);  
76         });  
77     } // 검색 end
```

# 12. 프로젝트를 마치며

---

## 소감

**이명진** : 처음부터 순조롭게 계획을 세워 프로젝트를 진행해 큰 어려움 없이 완성할 수 있었습니다. 특히 결제모듈을 이용해 예매하는 기능을 구현할 때 구글링을 통해 방법을 찾아가며 코드를 완성한 부분이 인상 깊었습니다.

**김고운** : 협업을 하면서 웹사이트 하나를 완성하는 경험을 해볼 수 있어서 유익하고 즐거웠습니다.

**신윤경** : 2차 때 해결하지 못했던 부분을 최종 프로젝트 때 쉽게 해결할 수 있어 비교적 2차 때보다 순조롭게 진행했습니다. 비록 개발자 쪽으로는 취업하지 못했지만 틈틈히 프론트엔드 공부를 하면서 자기개발하겠습니당😊 1조 모두 고생하셨습니다!!

**조영민** : 2차 프로젝트에서 완성하지 못했던 핵심 기술을 완성할 수 있었고 그것을 통해 팀원들과 협업을 하는데 있어서 유익한 시간이 되었습니다.

**황은선** : 2차 때 부족했던 부분을 보완하여 프로젝트를 시작해 수월하게 진행할 수 있었습니다. 프로젝트를 할 때마다 저의 부족한 부분이 아쉽기도 했지만 좋은 팀원분들 덕분에 많은 것을 배우고 성장했습니다. 모두들 감사합니다😊 이번 프로젝트 땐 활용하지 못했지만 추후에 기회가 된다면 파이썬을 활용한 프로젝트도 진행해보고 싶습니다.

Thank  
you!