



강민희 고민석 김성현 김지욱 두원재 성지혜

# CONTENTS

01 프로젝트 소개

02 분석/설계

03 사이트 시연

04 핵심 기능 소개

05 소감

# 01. 프로젝트 소개

---

- 조원소개 및 역할담당
- 사이트소개
- 개발환경

# 1 프로젝트 소개

## 조원소개 및 역할담당

- Admin 매출
- Admin 후기
- Client & Admin CSS

강민희

- CEO 극장관리
- Admin 계정관리
- Admin 블랙리스트 관리

고민석

- Client 예매 & 결제
- Client 메인JS & 리뷰등록 & 찜하기
- 회원정보 수정

김성현

- CEO 예매관리
- CEO 문의사항 & 환불
- Git 형상관리

김지욱

- 로그인 & 회원가입
- CEO CSS

두원재

- Client 카카오맵 & 스케줄
- Client 문의사항

성지혜

# Why?

소규모 극단은 **결제**하기가 어려워서  
**개인 SNS로 활동을** 많이 한다

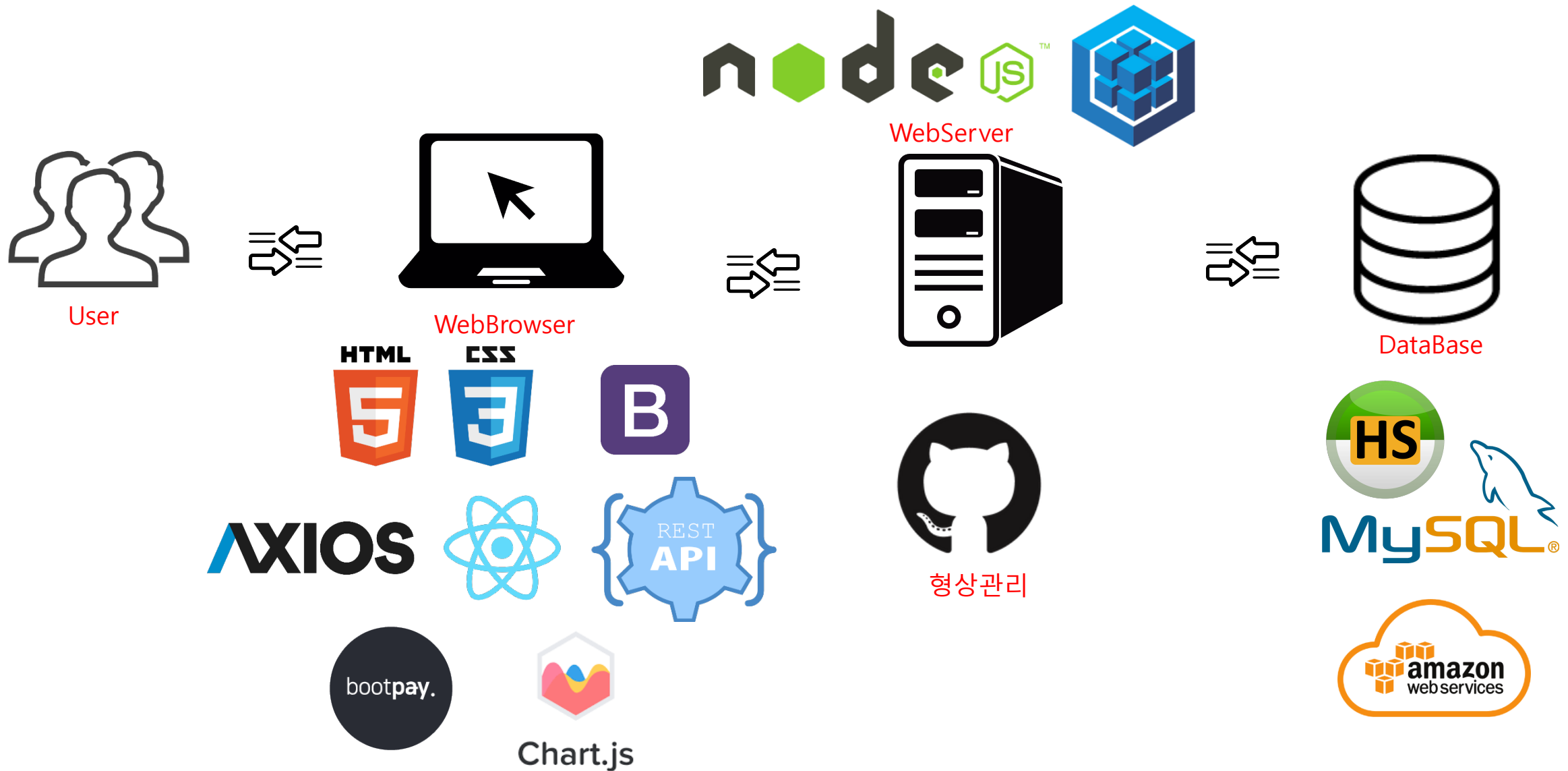
연극이나 소규모 극단에서는  
**매출관리 시스템**을 사용하기  
부담이 된다.

문화공연 산업은 시장이 작지만  
**잠재적인 수요**를 가짐

소비자, 사업자  
둘 다 서로  
**Win Win** 하기 위해 제작

1

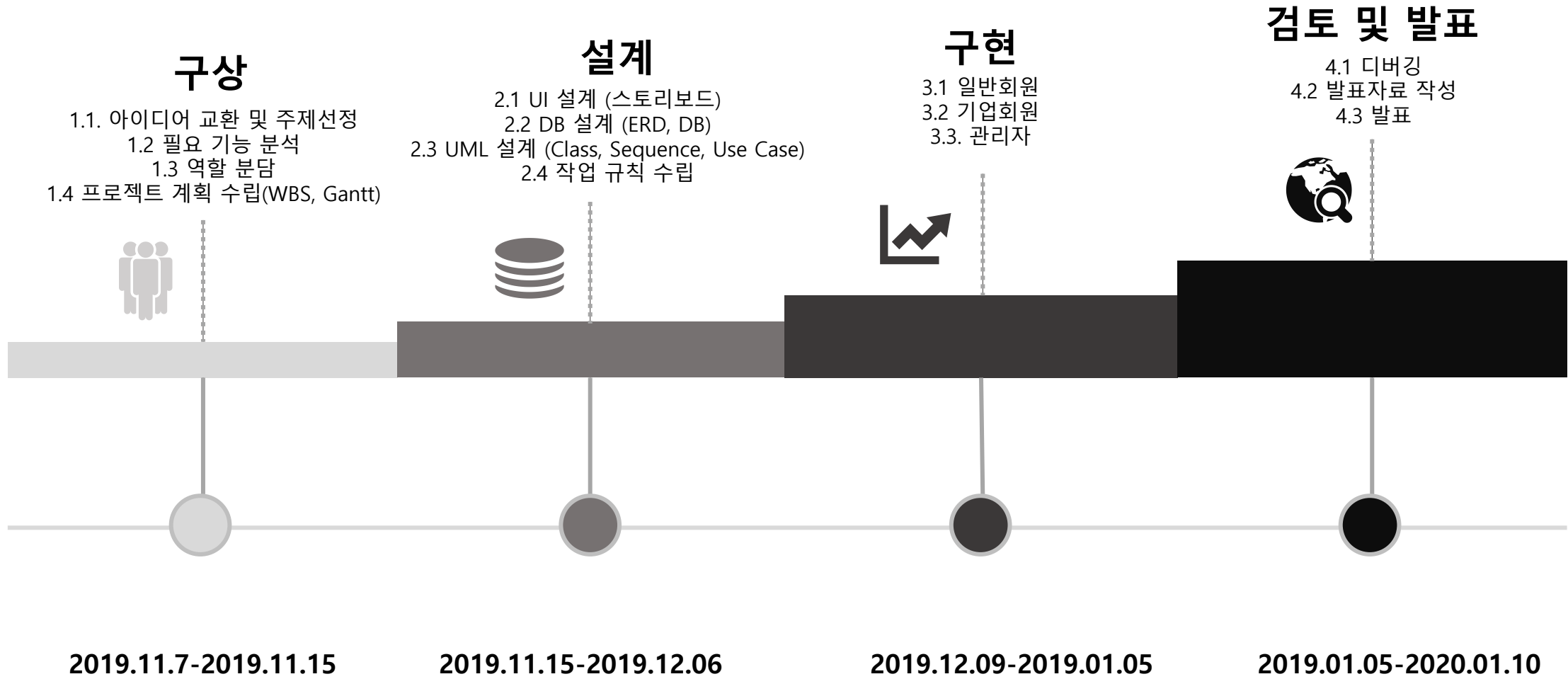
# 프로젝트 소개 개발환경



## 02. 분석/설계

---



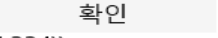
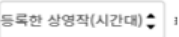



- 개발계획
- 스토리보드
- 프로젝트구조
- 시퀀스 다이어그램
- ERD





## 규칙 명세서

페이지 번호 규칙		
사장님 사이트(Ceo)	Ceo_00_00	<ul style="list-style-type: none"><li>- 첫 문자는 사이트에 따라 결정</li><li>- (상단)메뉴가 바뀌면 2번째 문자 변경</li><li>- (좌측)사이드바 변경 시 숫자 증가</li><li>- 해당 페이지 내에서 UI 변경 시, .00로 표현한다.</li><li>- 예) 회원사이트 메인 페이지_메인.썸네일의 경우 아래와 같이 작성 Cl_Ma_01.01</li><li>- 세부 페이지 변경사항 없을 시 .00은 생략가능 예) Cl_Ma_01</li></ul>
회원 사이트(Client)	Cl_00_00	
관리자 사이트(Admin)	Ad_00_00	
메인 페이지(Main)	00_Ma_00	
회원가입(Signup)	00_Su_00	
로그인(Login)	00_Lg_00	
마이페이지(Mi-Page)	00_Mi_00	
이벤트(Event)	00_Ev_00	
문의사항(Q&A)	00_Qa_00	
공연일정(Schedule)	00_Sc_00	
글 등록(Add)	00_Add_00	
극단 등록(Register)	00_Rg_00	
극단 관리(Management)	00_Mn_00	
매출 분석(Analyst)	00_An_00	
후기관리(Review)	00_Re_00	
계정관리(Account Manage)	00_Am_00	
페이지 가이드 라인		
설명 번호 위치	좌 상단	
설명 번호 모양(빨강)	<div><div>1</div><ul style="list-style-type: none"><li>- 높이 : 0.44cm</li><li>- 너비 : 0.45cm</li><li>- 글자 크기 : 10cm</li><li>- 텍스트 상자 왼쪽 여백 : 0cm 오른쪽 여백 0cm</li></ul></div>	
설명 구획 선택	- 선 너비 : 1 pt	
설명 구획 방향	<div><div></div><ul style="list-style-type: none"><li>- 높이 0.36cm</li><li>- 너비 1.25cm</li></ul></div>	
설명 번호 순서	왼쪽 -> 오른쪽 -> 아래	

UI 수정 규칙 / 추후에 규칙 수정	
메인 틀 및 로고 등 추가 사항	비율이나 사이즈를 수정하지 않는다. 해당 화면에서의 필요성에 의해서 추가/삭제 한다.
전체 페이지 공통 표시 영역	메인 페이지에서만 보여준다.
이미지 표시 (로고 포함)	<p>흰 바탕에 검은색 선으로 X를 표시한다.</p> 
UI 사용 이미지	<p>체크 박스 -&gt; <input checked="" type="checkbox"/></p> <p>옵션 버튼 -&gt; </p> <p>버튼 -&gt;  (색상: RGB(224,224,224))</p> <p>옵션 버튼 -&gt; </p> <p>배너 -&gt; </p> <p>수량 선택 삭제 or 닫기 버튼 -&gt; </p> <p>현재 선택되어 있는 탭 표시 -&gt; </p> <p>라디오 버튼 -&gt; <input type="radio"/></p>

파일 위치	
Component 위치	Adminpage/client/src/components/Login.js
Page(View) 위치	Adminpage/client/src/pages/LoginRegister.js
Sever위치	Adminpage/server.js
파일 명명 규칙	
클래스 명	★ 클래스 파일 이름은 첫 음절 마다 대문자(파스칼 표기법) AdminCeoSales.js

### 클라이언트 page & server

```
▼ CLIENT
  > client
  > database
  > models
  > routes
  > upload
  ◆ .gitignore
  {} database.json
  {} package-lock.json
  {} package.json
  JS server.js
  yarn.lock
```

### 사업자 page & server

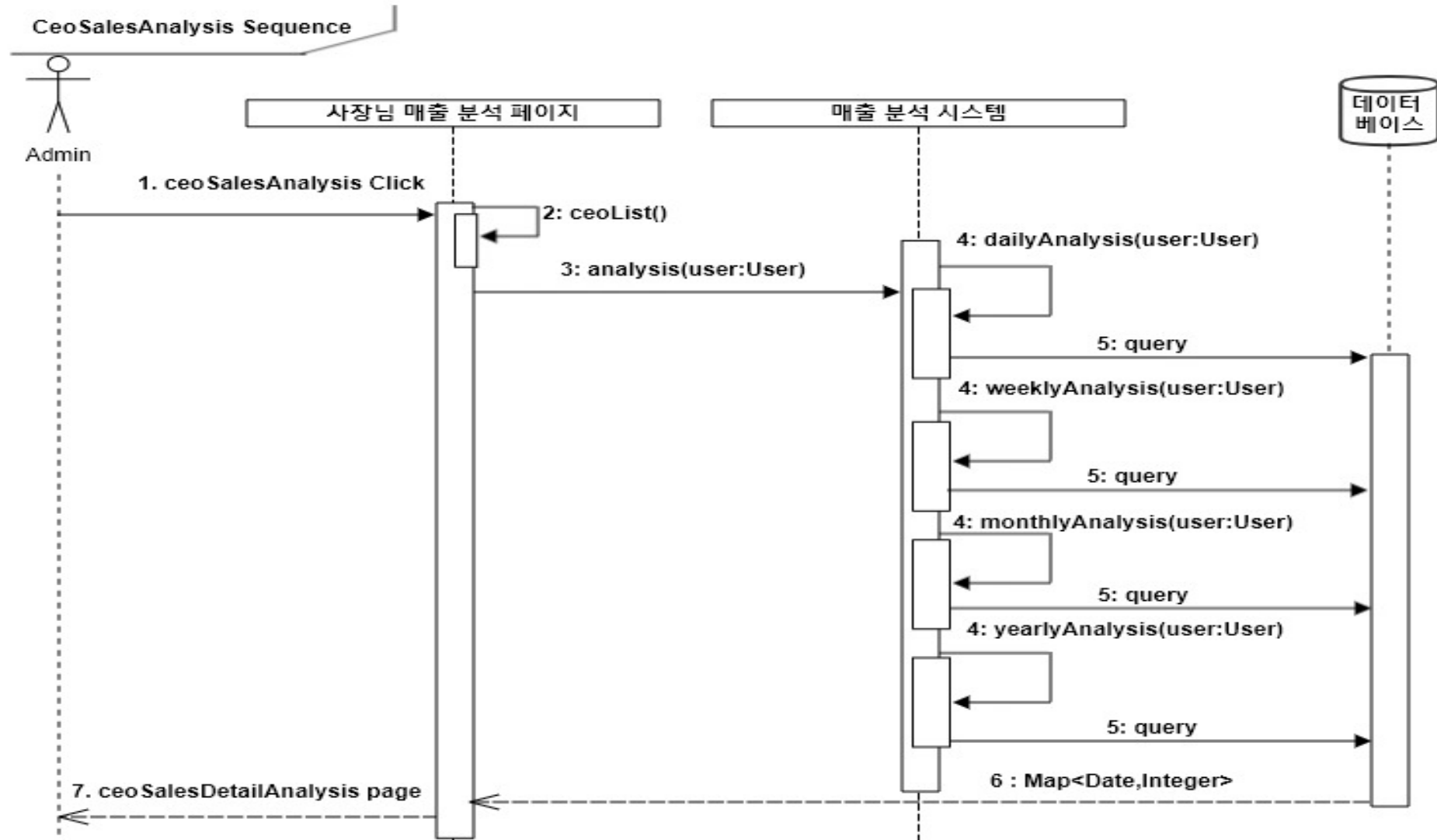
```
▼ CEO
  > client
  > database
  > models
  > routes
  ◆ .gitignore
  {} package-lock.json
  {} package.json
  JS server.js
  yarn.lock
```

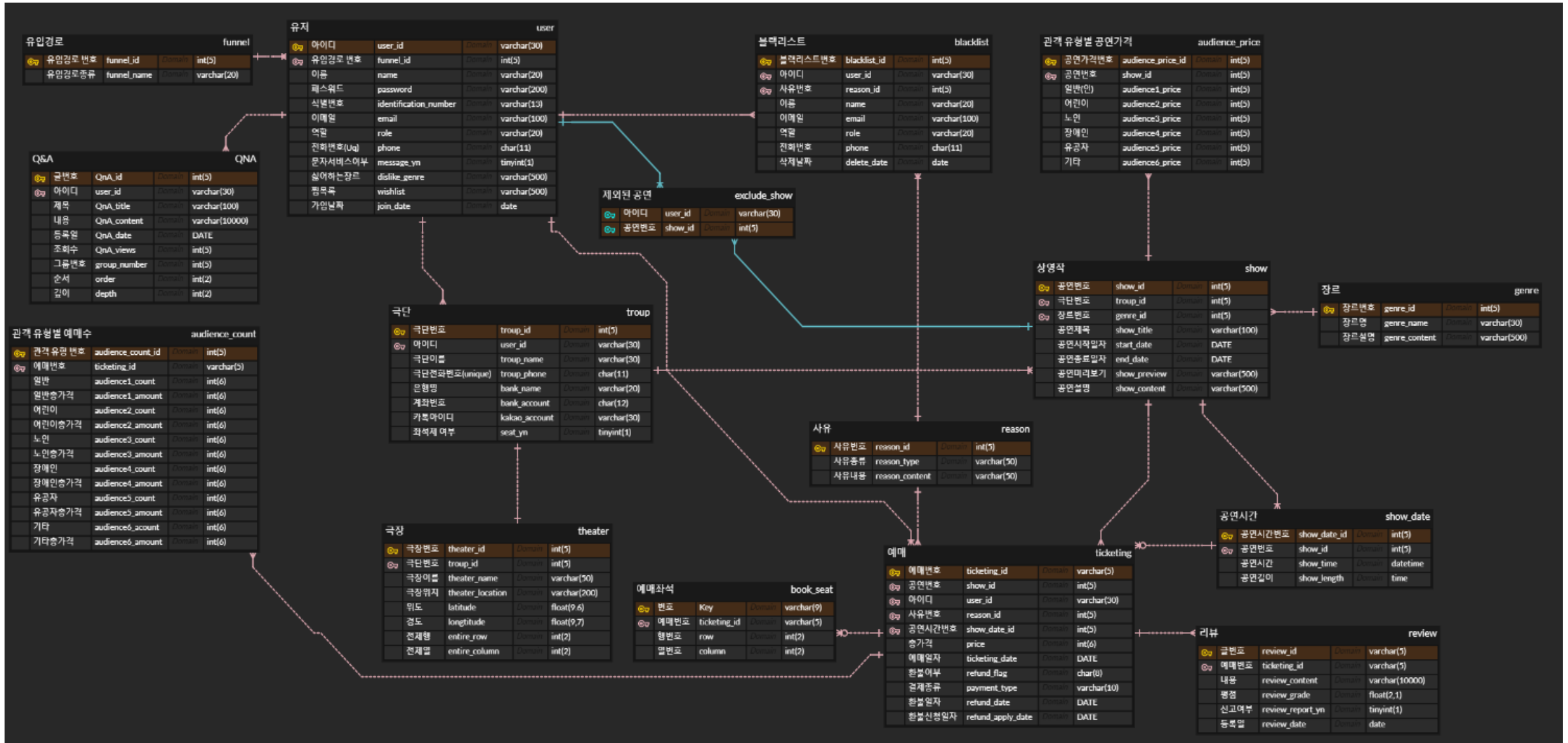
### 관리자 page & server

```
▼ ADMIN
  > client
  > database
  > models
  > routes
  ◆ .gitignore
  {} package-lock.json
  {} package.json
  ⓘ README.md
  JS server.js
  yarn.lock
```

### Database

```
showdb
  ▼ Tables
    ├── audience_count
    ├── audience_price
    ├── blacklist
    ├── book_seat
    ├── exclude_show
    ├── funnel
    ├── genre
    ├── QNA
    ├── reason
    ├── review
    ├── show
    ├── show_date
    ├── theater
    ├── ticketing
    ├── troupe
    └── user
```





## 03. 사이트 시연

---



## 04. 핵심 기능 소개

---

- Client
- CEO
- Admin

view

## 회원가입

아이디

a1

아이디를 올바르게 입력해주세요

비밀번호

.

비밀번호가 너무 쉽습니다.

.....|

비밀번호가 약합니다.

.....|

비밀번호가 괜찮습니다.

.....|

비밀번호가 매우 좋습니다.

## JS Register.js

```
validateName = name => {
  const nameRegExp = /[a-z0-9] | [\ ] {} < > ? | ` ~ ! @ # $ % ^ & * _ + = , . : ; " ' \ \ / g;

  if (name.match(nameRegExp) || name.length < 2) {
    this.setState({
      name_valid: false,
      name
    });
  }
};
```

```
validatePassword = (password) => {
  const { passwordScore } = this.state;
  const { score } = zxcvbn(password);
  this.setState({ passwordScore: score });
};
```

```
.then(user => {
  if (!user) {
    bcrypt.hash(req.body.password, 10, (err, hash) => {
      userData.password = hash,
      User.create(userData)
        .then(user => {
          res.json({ status: user.user_id + "님 회원가입 축하드립니다." })
        })
    })
  }
});
```

view

로그인

아이디

ejungsuk39

비밀번호

....

로그인

로그인



안녕하세요! 이종석 님

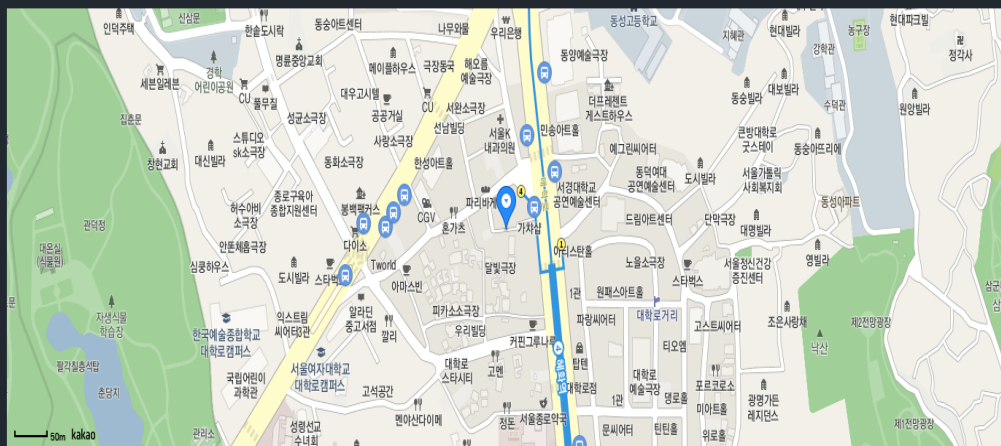
JS Users.js

```
User.findOne({
  where: {
    user_id: req.body.user_id
  }
})
```

```
componentDidMount() {
  this.timer = setInterval(this.progress, 20);
  const token = localStorage.getItem('usertoken')
  if(token !== null) {
    const decoded = jwt_decode(token)
    this.setState({
      user_id: decoded.user_id,
      name: decoded.name
    });
  }
}
```



view



에메

♡찜하기

📍터이상 보지 않기

공연정보

극장위치

리뷰

극단의 다른 작품

components

JS Map.js

```

callApi = async () => {
  let show_id = this.props.match.params.show_id;
  const response = await fetch('/api/theater/'+show_id);
  const body = await response.json();
  return body;
}

```

```

componentDidUpdate() {
  const {address} = this.state;
  const script = document.createElement("script");
  script.async = true;
  script.src =
    "https://dapi.kakao.com/v2/maps/sdk.js?appkey=33ccad9cee7a819a54a9b315939115a5";
  document.head.appendChild(script);
  script.onload = () => {
    var container = document.getElementById("map");
    var options = {
      center: new kakao.maps.LatLng(address[0]['longitude'], address[0]['latitude']),
      level: 3
    };
    var map = new kakao.maps.Map(container, options);
    var coords = new kakao.maps.LatLng(address[0]['longitude'], address[0]['latitude']);
    var marker = new kakao.maps.Marker({
      map: map,
      position: coords,
      clickable: false
    });
    marker.setMap(map);
    map.setCenter(coords);
  }
}

render() {
  const {address} = this.state;
  return (
    <React.Fragment>
      <div id="map" className="kakao"></div>
    </React.Fragment>
  )
}

export default withRouter(Map);

```

```

render() {
  return (
    <div>
      {this.state.show ? this.state.show.map(c => {
        <TableRow>
          <TableCell width="1400px">
            <Map/>
          </TableCell>
        </TableRow>
      }) : "no data"}
    </div>
  )
}

export default TheaterLocation;

```

view

연극  
어때?  
오늘은 영화 말고.

[등록](#) [관리](#) [분석](#) [문의게시판](#)[로그아웃](#)[+ 상영작등록](#)[+ 극단등록](#)

## 상영작 등록

공연 제목 \*

공연일자 시작 \*

연도. 월. 일.

공연일자 종료 \*

연도. 월. 일.

공연 시간 \*

런닝타임

공연 장르

[공연사진첨부](#)

공연 설명 \*

일반가격 \*

어린이가격 \*

노인가격 \*

장애인가격 \*

유공자가격 \*

기타 \*

[등록](#)[확인](#)[취소](#)

```
components
JS ShowAdd.js

handleFormSubmit = (e) => {
  e.preventDefault()
  this.axios();
  this.addCustomer();
  this.setState({
    troupe_id: '',
    genre_id: '',
    show_id: '',
    show_title: '',
    start_date: '',
    end_date: '',
    show_preview: '',
    show_content: '',
    show_time: '',
    show_length: '',
    audience1_price: '',
    audience2_price: '',
    audience3_price: '',
    audience4_price: '',
    audience5_price: '',
    audience6_price: '',
    file: null,
    fileName: '',
    open: false
  })
}
```

## JS server.js

```
//공연시간테이블 등록
app.post('/api/showManagementAddShowDate',parser,(req,res)=>{
  let sql ='INSERT INTO `show_date` VALUES ((select max(show_date_id) from `show_date` alias_for_subquery where show_date_id = show_date_id) +1 , ?,?,?)';

//관객유형별공연가격테이블 등록
app.post('/api/showManagementAddAudiencePrice',parser,(req,res)=>{
  let sql ='INSERT INTO `audience_price` VALUES ((select max(audience_price_id) from `audience_price` alias_for_subquery where audience_price_id

//상영작테이블 등록
app.post('/api/showManagementAddShow',upload.single('image'),parser,(req,res)=>{
  let sql ='INSERT INTO `show` VALUES ((select max(show_id) from `show` alias_for_subquery where show_id = show_id) +1 , ?, ?, ?, ?, ?, ?, ?)';
```

view

극단 등록

극단 이름 \*

전화번호 \*

은행명

계좌 번호 \*

카카오톡 ID \*

극장이름 \*

극장위치 \*

위도 \*

경도 \*

우편번호검색

|예) 판교역로 235, 분당 주공, 삼평동 681

**tip**

아래와 같은 조합으로 검색을 하시면 더욱 정확한 결과가 검색됩니다.

도로명 + 건물번호

예) 판교역로 235, 제주 첨단로 242

지역명(동/리) + 번지

예) 삼평동 681, 제주 영평동 2181

지역명(동/리) + 건물명(아파트명)

예) 분당 주공, 연수동 주공3차

사서함명 + 번호

예) 분당우체국사서함 1~100

```
components handleAddress = (data) => {  
  JS TroupAdd.js  let fullAddress = data.address;  
                  let extraAddress = '';  
                  let postcode=data.zonecode;  
  
                  if (data.addressType === 'R') {  
                    if (data.bname !== '') {  
                      extraAddress += data.bname;  
                    }  
                    if (data.buildingName !== '') {  
                      extraAddress += (extraAddress !== '' ? ` , ${data.buildingName}` : data.buildingName);  
                    }  
                    fullAddress += (extraAddress !== '' ? ` (${extraAddress})` : '');  
                  }  
                  this.setState({theater_location:data.address});  
  
                  console.log(postcode+" "+fullAddress);  
                  this.togglePopup();  
  
                }  
                togglePopup() {  
                  this.setState({  
                    showPopup: !this.state.showPopup  
                  });  
                }  
              }
```

## view

## 예매 변경

변경유형      시간대 변경      예매 번호\*      변경좌석(행)\*      변경좌석(열)\*

시간대 변경      18:00~20:00      예매 번호\*      변경좌석(행)\*      변경좌석(열)\*

변경      취소

## JS TicketManagement.js

```
formAxios = () =>{
  var params = new URLSearchParams();
  params.append('show_time', this.state.show_time)
  params.append('row', this.state.row)
  params.append('column', this.state.column)
  params.append('ticket_id', this.state.ticket_id)
  formAxios.post('/ticketing/changeSeat', params)
  .then((Response) => {
    console.log(Response);
  }).catch((ex) =>{
    console.log(ex)
  })
  console.log(params)
}
```

```
handleFormSubmit2(e) {
  e.preventDefault();
  this.formAxios();
  this.setState({
    show_time: '',
    seatNo: '',
    row: '',
    column: '',
    ticket_id: ''
  })
}
```

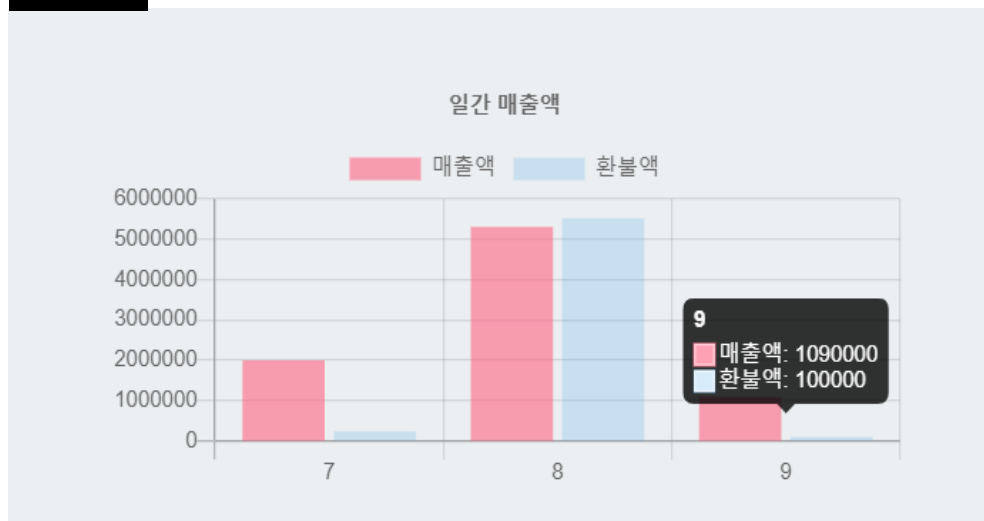
```
<TextField
  required
  id="ticket_id"
  name="ticket_id"
  label="예매 번호"
  value={this.state.ticket_id}
  onChange={this.handleChange}
  fullWidth/>
</Grid>
<Grid item xs={12} sm={2}>
<TextField
  required
  id="row"
  name="row"
  label="변경 좌석 (행)"
  value={this.state.row}
  onChange={this.handleChange}
  fullWidth/>
</Grid>
<Grid item xs={12} sm={2}>
<TextField
  required
  id="column"
  name="column"
  label="변경 좌석 (열)"
  value={this.state.column}
  onChange={this.handleChange}
  fullWidth/>
</Grid>
<Button className={classes.button} type="submit"
  variant="contained" color="primary">변경</Button>
<Button className={classes.button} type="reset"
  variant="contained" color="primary">취소</Button>
</Grid>
```

## 4 핵심 기능 소개 CEO 예매관리(예매변경)

JS server.js

```
// 예매 변경 적용
app.post('/ticketing/changeSeat', parser, (req, res) => {
  let sql = "update book_seat bs, ticketing t set bs.`key`=concat(?,concat(?,?)), " +
    "bs.`row` = ?, bs.`column` = ? WHERE bs.ticketing_id = t.ticketing_id and t.ticketing_id=?"
  let id1 = req.body.ticket_id;
  let row1 = req.body.row;
  let column1 = req.body.column;
  let row2 = req.body.row;
  let column2 = req.body.column;
  let id2 = req.body.ticket_id;
  let params = [id1, row1, column1, row2, column2, id2];
  console.log(params);
  connection.query(sql, params,
    (err, rows, fields) => {
      res.send(rows);
    }
  )
});
```

## view



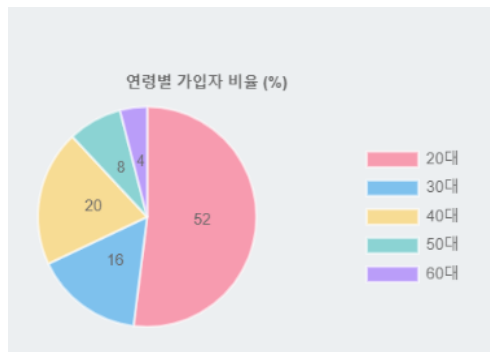
## JS AdminCeoSalesChartBarChart8.js

```
const {data, pair,data2} = this.props;

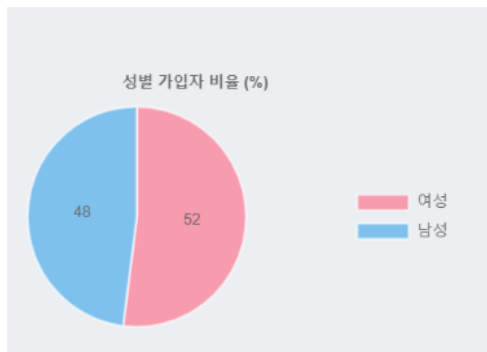
const config = {
  type:"bar",
  data: {
    labels: [dd-2,dd-1,dd],
    datasets: [
      {
        label: "매출액",
        data: data.map(d => d.value),
        fill: false,
        backgroundColor: 'rgba(255, 99, 132, 0.6)',
        borderColor: 'rgba(255, 99, 132, 0.6)',
        lineTension: 0,
        pointRadius: 0
      }
    ]
  }
}
```

```
options:{
  responsive: true,
  title: {
    display: true,
    text: `${pair} 매출액`
  },
  tooltips: {
    mode: "index",
    intersect: false
  },
  hover: {
    mode: "nearest",
    intersect: true
  },
  plugins:{
    datalabels: {
      formatter : function(value,ctx){
        return null;
      }
    }
  },
  scales:{
    xAxes:[{
      display: true,
    }],
    yAxes: [{
      display: true,
      ticks: {
        scaleBeginAtZero : true,
        beginAtZero: true
      }
    }]
  }
}
```

## view



## view



## JS AdminClientSalesChart6.js

```
const config = {
  type:"pie",
  data: {
    labels: data.map(d => d.date + '대'),
    datasets: [
      {
        label: "연령별 가입자수",

```



## Sql: 주간분석

```
-- 최근 3주간 결제 금액 쿼리문
-- DATE_SUB / DATE_FORMAT / DAYOFWEEK
-- INTERVAR
-- %U : 주간 (총 0~51주)
SELECT DATE_FORMAT(DATE_SUB(T.ticketing_date, INTERVAL (DAYOFWEEK(T.ticketing_date)-1) DAY), "%Y/%m/%d") as start,
       DATE_FORMAT(DATE_SUB(T.ticketing_date, INTERVAL (DAYOFWEEK(T.ticketing_date)-7) DAY), "%Y/%m/%d") as end,
       DATE_FORMAT(T.ticketing_date, "%Y-%m-%U") AS `date`,
       date_format(now(), "%Y-%m-%U") as now,
       sum(`price`) as sum
FROM ticketing T, user U, `show` S, troupe TR
where T.show_id=S.show_id and S.troupe_id = TR.troupe_id and TR.user_id = U.user_id
and U.phone = ?
and (DATE_FORMAT(T.ticketing_date, "%U") = 0 or
     DATE_FORMAT(T.ticketing_date, "%U") = 51 or
     DATE_FORMAT(T.ticketing_date, "%U") = 1)
GROUP BY date
order by date asc;
```

## Sql: 연령별 가입자수

```
--연령별 가입자수
select round(count(user_id)/(select count(user_id) from user where length(identification_number)=13)*100) as 인원,
substr(date_format(now(), "%Y") - (if (substr(identification_number,7,1) = 1 or 2 ,1900, 2000)
+substr(identification_number,1,2)) +1 ,1,1)*10 as 연령
from user where length(identification_number) =13
group by 연령
order by 연령
```

## Sql: 성별 가입자수

```
--성별 가입자수
select round(count(user_id)/(select count(user_id) from user where length(identification_number)=13)*100) as count ,
if(substr(identification_number,7,1) = 4, 2 ,if(substr(identification_number,7,1) =3 ,1,substr(identification_number,7,1)))
as gender
from user where length(identification_number)=13 group by gender
```

## view

QR코드로 결제

카톡 메시지로 결제

✕

pay

결제요청 메시지 전송을 위해  
아래 정보를 입력해주세요.

휴대폰 번호

예) 01012345678

생년월일

예) 840301

결제요청

pay

카카오페이

결제하기 &gt;

결제를 완료해주세요.

- 일시: 2020.01.09 07:17  
- 구매처: KAKAOPAY TEST  
- 상품명: 극적인하룻밤  
- 금액: 47,000원

수신설정 : 페이 설정 &gt; 알림받기 설정

모바일에서 확인하세요.

오전 7:17

## components

## JS Ticketing.js

```
handleFormSubmit = (e) => {
  e.preventDefault()
  this.state.price = this.state.basicPrice + this.state.childPrice+ this.state.seniorPrice
  +this.state.disabledPrice+this.state.meritPrice+this.state.etcPrice;

  this.ticket();           //결제 api호출
  this.addTicketing()      //결제 정보 저장
  this.addAudience()      //해당 결제의 관객 유형별 예매수 및 유형별 총가격 저장

  this.setState({          //정보 초기화
    basic: 0,
    child: 0,
    senior: 0,
    disabled: 0,
    merit: 0,
    etc: 0,
    price: 0,
    open: false
  })
}
```

```
ticket() {
  BootPay.request({
    price: this.state.price, //실제 결제되는 가격
    application_id: "59a4d323396fa607cbe75de4", // WEB SDK용 Application ID
    name: this.state.title[0].show_title, //결제창에서 보여질 이름
    pg: 'kakao',
    method: 'easy', //결제수단, 입력하지 않으면 결제수단 선택부터 화면이 시작합니다.
    show_agree_window: 0, // 부트페이 정보 동의 창 보이기 여부
  })
}
```

## 05. 소감

---

**강민희** 노드 js랑 리엑트를 1달만에 배우고, 서비스를 구현하는 것이 불가능하다고 생각했는데, 팀원들과 같이 공부하고, 반친구들의 적극적인 도움으로 완성할 수 있었습니다. 감사합니다.

**김성현** 리엑트의 라이프 사이클에 대한 공부가 부족한 탓에 원하는대로의 화면구현이 제대로 되지 않은것같고 환불기능이 완전하지 못한것도 아쉬웠고 하지만 단기간에 전혀 모르는 기술을 이용해 프로젝트를 구현해봄으로서 자신감을 얻은 것 같다.

**두원재** 이번 프로젝트 동안 조원들에게 너무 감사하고 새로운 기술들을 사용해서 의미 있었습니다. 앞으로 더 다양한 프로젝트들을 경험해보고 싶습니다.

**고민석** 이번에 노드js와 리엑트를 전부 프로젝트 시작하면서 처음 배웠는데 배우는 기간이 너무 짧아서 어렵고 막히는 부분이 많아서 처음에 생각했던 기능을 다 구현하지 못한게 아쉬웠고 그래도 어느정도 생각한대로 완성되어서 뿌듯합니다.

**김지욱** 처음에 기술 조율 할 때 의견을 많이 내고 해서 팀 프로젝트에 부담이 조금 됐지만, 다들 같이 열심히 공부하고 프로젝트를 만들어 내서 정말 감사합니다. 새로운 것에 도전하는 것에 두려움을 이겨내는 법을 배워서 좋았습니다.

**성지혜** 새로운 기술에 도전해본다는 사실이 이렇게 힘든 줄 몰랐습니다. 기능을 완성해내기엔 공부 할 시간이 많이 부족하고 저의 실력이 많이 부족 했던 것 같습니다. 앞으로 공부 방향을 잘 잡아서 지금보다 훨씬 더 발전 할 수 있도록 노력하겠습니다.