

1조



# 항공편 예약 시스템



표준태  
허재혁  
문나정

서예진  
이윤정

# 목차

## 흐름도

<항공편 예약 시스템>  
웹 프로젝트 처리 과정

## 코드 리뷰

분담 역할 구현 코드 설명

## 화면 명세서

항공편 예약 시스템 UI/UX  
구성 표현

## 회의록

프로젝트 기간 실행한  
회의록 정리

## 데이터베이스

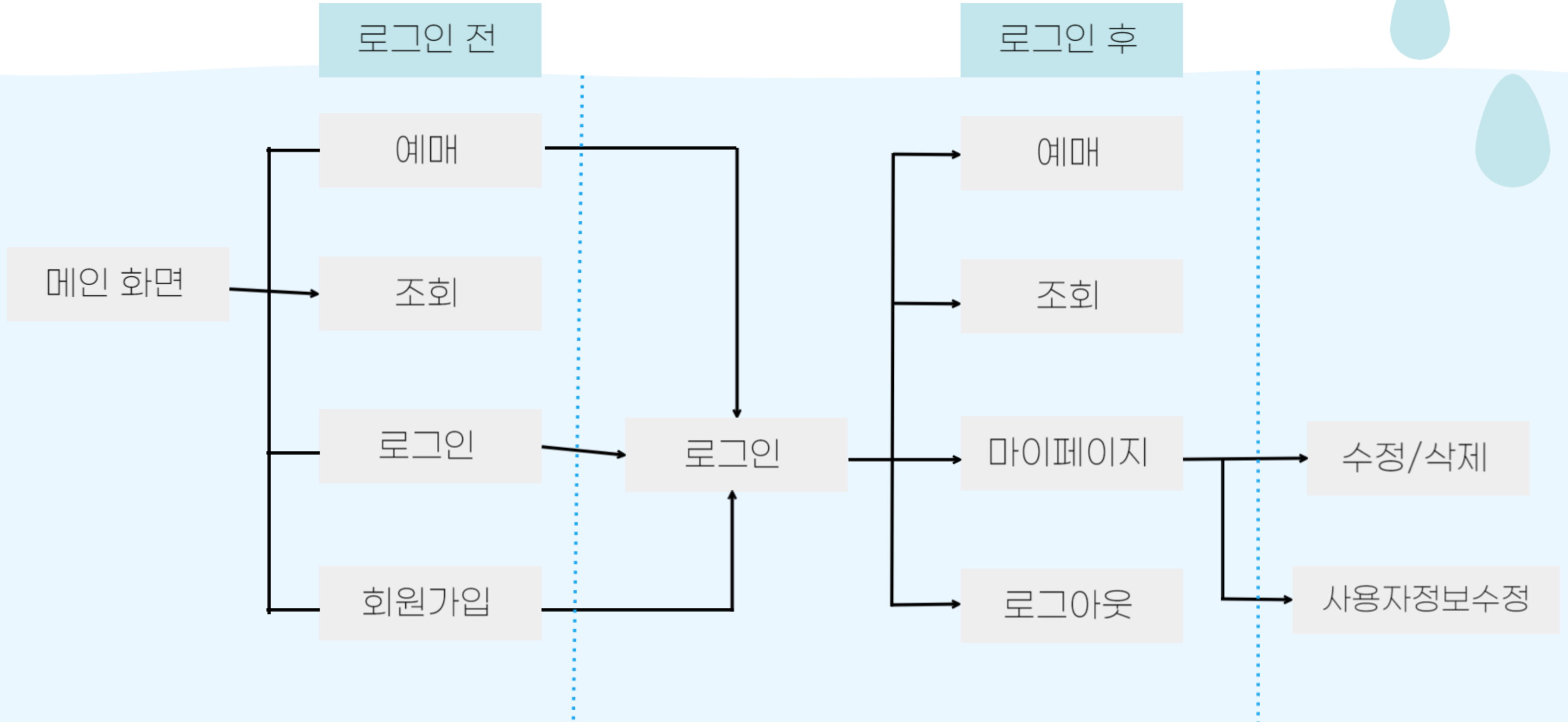
회원, 예약, 좌석, 항공편의 ERD

## 개발 명세서

각 맡은 업무별 개발 타임라인  
제시

# 흐름도

<항공편 예약 시스템> 웹 프로젝트 처리 과정



# 화면 명세서

로그인 전

로그인/회원가입

예매

조회

예매

# 화면 명세서

로그인 후

로그아웃/마이페이지

예매

조회

예매

# 화면 명세서

회원가입

회원가입

아이디

비밀번호

이름

전화번호

확인

# 화면 명세서

로그인

로그인

아이디

비밀번호

확인

# 화면 명세서

항공편 수정/삭제

마이페이지

항공편 수정/삭제

내 정보수정

항공편 수정/삭제



# 화면 명세서

내 정보수정

마이페이지

항공편 조회/삭제

내 정보수정

아이디

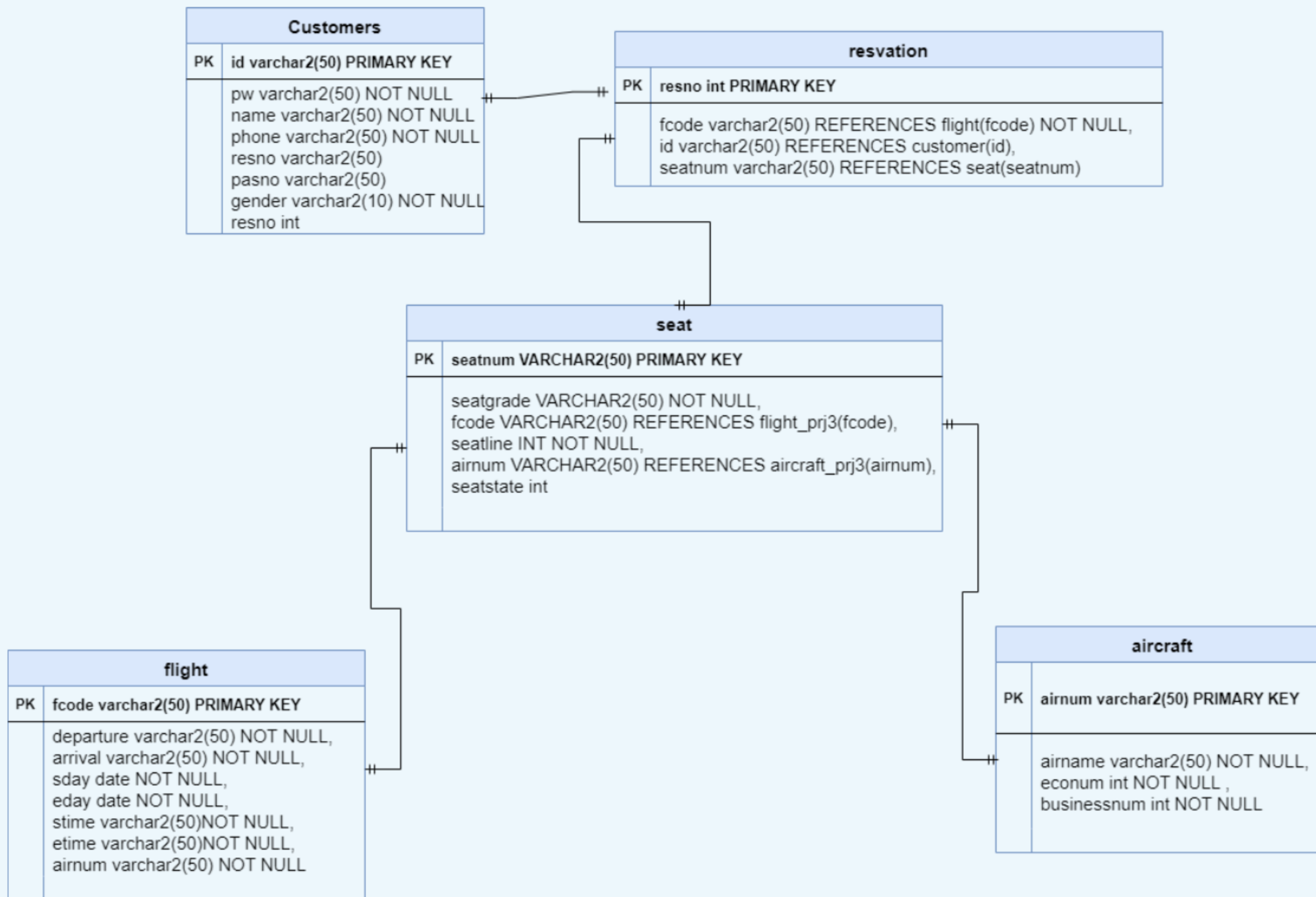
비밀번호

이름

전화번호

확인

# 데이터베이스



# 개발 명세서

회원가입\_이윤정

	요청 url	서블릿	요청방식	모델	뷰	
메인화면	index.do	HomeServlet	GET	X	main.jsp	메인화면
			POST	AirlineDAO AirlineService		
회원가입	l_join	L_joinServlet	GET	X	L_joinus.jsp	
			POST	joinServicejoinDA O		실패: 회원가입 화면 성공: 로그인 화면
로그인	h_login	H_indexServlet	GET	X	h_login.jsp	로그인

# 개발 명세서

로그인\_허재혁

	요청 url	서블릿	요청방식	모델	뷰		
메인화면	index.do	HomeController	GET	X	index.jsp	메인화면	
로그인	h_login	LoginController	GET	X	h_login.jsp	로그인시 로그인 후 페이지로 이동	실패: 로그인페이지 성공: 로그인 후 메인페이지
			POST	LoginService LoginDAO		입력된 ID의 비밀번호가 일치하면 메인 페이지로 이동	
로그아웃	index.do	H_indexServlet	GET	X	index.jsp	로그아웃시 로그인 전 페이지로 이동	

# 개발 명세서

항공권 조회 및 예약\_문나정

	요청 url	서블릿	요청방식	모델	뷰	
메인화면 (예약 및 조회)	index.do	HomeServlet	GET ----- POST	X ----- AirlineDAO	main.jsp -----	
항공편 조회	flight.do	flightServlet	POST	AirlineDAO	flight.jsp ----- flightBsn.jsp	이코노미 선택 ----- 비즈니스 선택
좌석선택	seat.do	SeatServlet	POST	AirlineDAO	SeatEconomy.jsp ----- SeatBusiness.jsp	이코노미석 예매 ----- 비즈니스석 선택
예약정보 입력	Booking.do	BookingServlet	POST	AirlineDAO	booking.jsp ----- check.jsp	
예약 완료	bookingcheck.do	BookingCompServlet	POST	AirlineDAO	bookingComp.jsp	

# 개발 명세서

항공권, 회원 정보 수정\_표준태

	요청 url	서블릿	요청방식	모델	뷰	
회원 수정	UUA.do	P_UserUpdateAjaX	GET	X	P_UserUpdateView2.jsp	
	UUA2.do	P_UserUpdateAjaX2	GET	P_DAO p_Service		
			POST	P_DAO p_Service		회원 수정 화면

# 개발 명세서

항공권 조회/삭제\_서예진

	요청 url	서블릿	요청방식	모델	뷰	
항공편 조회	scheck	S_flightServlet	GET	X	S_main.jsp	예약 조회 화면
					S_check2.jsp	예약 조회 완료 화면
항공편 취소	sdelete	S_DeleteServlet	POST	flightDAO S_flightService	h_index	삭제된 후 메인 페이지로 이동
메인화면	index.do	HomeServlet	GET	X	main.jsp	메인화면
			POST	AirlineDAO AirlineService		



# 코드 리뷰

## 좌석 선택\_문나정

```
259<%
260 ArrayList<Seat> BSNList = (ArrayList<Seat>)session.getAttribute("BSN");
261 ArrayList<Seat> line1 = new ArrayList<>();
262 ArrayList<Seat> line2 = new ArrayList<>();
263 for(int i=0; i<BSNList.size(); i++){
264     if(BSNList.get(i).getSeatline()==1){
265         line1.add(BSNList.get(i));
266     }
267     else if(BSNList.get(i).getSeatline()==2){
268         line2.add(BSNList.get(i));
269     }
270 }
271 %>
272
273<div class="plane">
274     <div class="cockpit">
275         <h1>비즈니스석</h1>
276     </div>
277     <div class="exit exit--front fuselage">
278
279     </div>
280     <ol class="cabin fuselage">
281         <li class="row row--1">
282             <ol class="seats" type="A">
283                 <% for(int i=0; i<line1.size(); i++){ %>
284                 <% if(line1.get(i).getSeatstate()==0){ %>
285                     <li class="seat">
286                         <input type="checkbox" id="<%=line1.get(i).getSeatnum() %>" name="seat" onclick="checkOnlyOne(this)" />
287                         <label for="<%=line1.get(i).getSeatnum() %>"><%=line1.get(i).getSeatnum() %></label>
288                     </li>
289                 <%}else{ %>
290                     <li class="seat">
291                         <input type="checkbox" id="<%=line1.get(i).getSeatnum() %>" disabled name="seat" />
292                         <label for="<%=line1.get(i).getSeatnum() %>"><b>X</b></label>
293                     </li>
294                 <%} %>
295             </ol>
296         </li>
```

Server에서 내가 선택한 항공권의 비즈니스석  
좌석 정보만 받아옴

라인별 좌석 정보를 저장해 놓고  
그 정보를 통해 좌석 예매 여부에 따라  
좌석을 생성



# 코드 리뷰

로그인\_허재혁

```
32     if (id == null) {
33         %>
34         <div class="a_box">
35             <a href="<%=request.getContextPath() %>/h_login">로그인</a>
36         </div>
37         <%
38
39         %>
40         <div class="a_box">
41             <a href="<%=request.getContextPath() %>/l_join">회원가입</a>
42         </div>
43         <%
44     } else {
45         %>
46         <div class="idview">
47             <p>
48                 환영합니다 [<%=id%>]님 ㄴㄴㅇㅇ?
49             </p>
50         </div>
51     </div>
52     <div class="a_box">
53         <a href="<%=request.getContextPath() %>/h_logout">로그아웃</a>
54     </div>
55     <div class="a_box">
56         <a href="<%=request.getContextPath() %>/smain">마이페이지</a>
57     </div>
58
59     <%
60     }
61     %>
```

ID값이 null인지 비교하여

로그인이 되어 있는 상태면  
로그아웃과 마이페이지 출력  
로그인이 되어 있지 않으면  
로그인/회원가입 출력

# 코드 리뷰

회원가입\_이윤정

```
public boolean checkId(String id) {
    Connection con = dbcon();
    String sql = "select id from customer_prj3 where id = ?";
    PreparedStatement pst = null;
    ResultSet rs = null;

    try {
        pst = con.prepareStatement(sql);
        pst.setString(1, id);
        rs = pst.executeQuery();

        if (rs.next()) {
            return true;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(rs, pst, con);
    }

    return false;
}
```

ID 값이 존재하는지 boolean으로 체크하여

DB에 동일한 ID가 있다면 true,  
동일한 ID가 없다면 false로 출력함

# 코드 리뷰

항공권 삭제\_서예진

```
public void deleteFlightinfo(String resno) {
    Connection con = db();
    String sql = "DELETE FROM resvation_prj3 r " +
        "WHERE r.RESNO = ? " +
        "AND EXISTS (SELECT 1 FROM flight_prj3 f " +
        "                WHERE r.FCODE = f.FCODE " +
        "                AND EXISTS (SELECT 1 FROM aircraft_prj3 a " +
        "                            WHERE f.airnum = a.airnum))";

    PreparedStatement pst = null;
    try {
        pst = con.prepareStatement(sql);
        pst.setString(1, resno);
        pst.executeUpdate();

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    close(con, pst);
}
```

resno 받아와서 예약번호와 관련된 비행정보를  
데이터 베이스에서 삭제하는 메서드

EXISTS 연산자를 사용해서 관련된 비행 정보와  
항공기 정보가 있는 경우에만 삭제 진행

# 코드 리뷰

## 개인정보 수정\_표준태

```
.ajax({
  type : "get",
  dataType : "json",
  url : "/prj3_1/UUA2.do",
  success : function(data) {
    console.log(data);
    output = '';
    output += '아이디 <br> <input id="id" type="text" value="' + data.id + '" readonly="readonly" ><br>';
    output += '비밀번호 <br> <input id="pw" type="text" value="' + data.pw + '" ><br>';
    output += '이름 <br> <input id="name" type="text" value="' + data.name + '" readonly="readonly" ><br>';
    output += '전화번호 <br> <input id="phone" type="text" value="' + data.phone + '" ><br>';
    output += '성별 <br> <input id="gender" type="text" value="' + data.gender + '" readonly="readonly" ><br>';
    output += '<button class="butt" onclick="update(this)">수정하기</button>';

    document.getElementById('disp').innerHTML = output;
  },
  error : function() {
    alert("요청실패");
  }
});
} // select
function update(obj) {

  let div = obj.parentElement;
  let id = div.querySelector("#id").value;
  let pw = div.querySelector("#pw").value;
  let name = div.querySelector("#name").value;
  let phone = div.querySelector("#phone").value;
  let gender = div.querySelector("#gender").value;

  $.ajax({
    type : 'post',
    dataType : 'text',
    data : "id=" + id + "&pw=" + pw + "&name=" + name + "&phone="
      + phone + "&gender=" + gender,
    url : "/prj3_1/UUA2.do",
    success : function(data) {
      alert(data);
    },
  });
}
```

ajax를 통해서 회원정보를 받아오고 수정하는 메소드

수정되는 것과 안 되는 것을 구분하여  
다른 형태로 출력

# 9/15/프로젝트 1일차 회의록

## 역할 분담

준태 : 회원정보 수정(MVC -> ajax로 구현)

재혁 : 로그인 (유효성 검사)

나정: jsp, DB 수정, 메인 페이지

윤정: 회원가입 (유효성 검사)

예진 : 예약 정보 조회, 삭제

## 주소 매핑 짜기

index(메인 페이지)로 이동

## sequence 기능 넣기

sequence로 예약번호 차례대로 넣기

## ajax 페이지 구성안

버튼 클릭시 페이지 이동 가능하게 수정

## 현재 진행 상황

ERD 구성, UI/UX 화면 설계,  
개발명세서 작성

1

2

1

항공편 수정

1

2

2

회원정보 수정



# 9/18/프로젝트 2일차 오전 회의록

## 오전 진행 상황 공유

준태 : DB 수정, DAO ajax로 표현

재혁 : 로그인/로그아웃 수정,

로그인 된 ID 받아와서 화면 출력

나정: ajax로 항공편 페이지 구현 중

윤정: 회원가입 유효성 검사 후 DB에 값 넣기

안 되는 부분 수정

예진 : 조회, 예약번호 받아오기

## 진행 목표

★ 수요일 오전까지 백단 완성

# 9/18/프로젝트 2일차 오후 회의록

## 오후 진행 상황

준태 : ajax로 예약 내역 조회 가능, 회원정보 수정 중

재혁 : 로그인/로그아웃 완료, 회원가입 ID 중복 검사 백업

나정 : 메인 페이지 form 태그 쓰기, 항공편 조회 중

윤정 : 회원가입 라디오 버튼 유효성 검사 수정, ID 중복 여부 체크

예진 : 예약 조회 띄우기 완료, table 받아와서 sequence로 가능하게 수정

## 진행 목표

재혁, 윤정, 예진 분담 역할 끝내고 하나로 엮기

CSS 작업 돌입

## 각자 겪은 어려움

윤정 : DB를 통한 ID 중복 검사 하기

준태 : MVC로 jsp 구성은 괜찮지만 ajax 구조 파악이 어려움

나정 : ajax 구조 파악이 어려워서 조회 목록 띄우기에 어려움이 있음

# 9/19/프로젝트 3일차 오전 회의록

## 문제 상황 발생

ajax로 구현한 항공편 조회, 수정까지 거의 완료  
항공편 수정의 의미 -> '항공편 예매'와 같기에 항공편 수정 무의미

## 현재 상황 공유

준태: MVC 구현 회원정보 수정 -> ajax로 변환  
재혁: 완료된 페이지(로그인/로그아웃, 회원가입) 역기  
나정: server로 값 전송이 원활하지 않음  
윤정: 회원가입(null, DB 체크 유효성) 완료  
예진: 예약 조회, 삭제 완료 후 DB값 다시 넣기

## 진행 목표

항공편 예매는 편도로 가능하게 출력  
회원정보 수정 ajax 변환  
DB table 수정



# 9/19/프로젝트 3일차 오후 회의록

## 현재 상황 공유

윤정, 재혁, 예진, 준태 - 주어진 역할 완료 후 여유, 실행 가능  
나정: 메인 페이지 항공편 조회 완료

## 각자 겪은 어려움

준태: ajax로 항공편 예약 수정을 구현했으나 기능의 유용성이 없어  
처음 설계를 잘해야겠다고 생각함  
나정: req로 서버에 보낼 때 문자열 깨지는 걸 공백을 지우니 해결됨  
재혁: 기능 추가  
윤정: git 올리는 방법, 페이지 이동 방법 파악  
예진: join되는 테이블의 해당 데이터를 삭제

## 진행 목표

CSS는 목요일 오전(09.21)까지 작업  
filter 기능 추가  
금요일 오전까지 발표 자료, 기능 구현, CSS 작업까지 마무리