



BANGKOK

모든 여행의 시작, BANGKOK

2024.09.30

| 목 차 |

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항
개발 내용과 범위
소요자원

03 Front-end

메뉴의 흐름
화면 구현

04 Back-end

Data Flow Diagram
DB

05 프로젝트 시연 및 핵심코드

프로젝트 시연
모듈 핵심코드
모듈 오류 수정
추후 개발 내용
프로젝트 후기

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

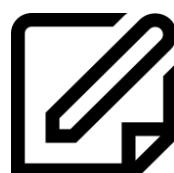


항공권 예약

호텔 예약

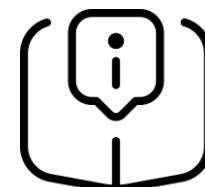


BANGKOK



여행 계획

여행 정보





BANGKOK

목적

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

| 여행의 관련된 모든 준비를 할 수 있는 종합 서비스 플랫폼 구축

| 사용자 친화적인 간편한 시스템 개발

| 여행과 관련된 상품 판매의 확장성 보유

| 판매를 통한 판매 수수료로 인한 수익 창출

02

개발 전반

요구 사항

개발 내용과 범위

소요 자원

01 프로젝트 개요
주제와 목적02 개발 전반
요구 사항
개발 내용과 범위
소요 자원03 Front-end
메뉴의 흐름
화면 구현04 Back-end
DFD
DB05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

요구 기능	요구 사항 키워드
회원관리	회원 등급 리스트, 등급 등록/삭제, 회원 리스트, 상세보기, 회원가입, 회원 정보수정, 비밀번호 변경 등..
항공권 관리	항공권 검색, 항공권 리스트, 항공사 관리, 기종 등록 및 관리, 노선 및 운임 관리, 스케줄 관리 등..
항공권 예약	항공권 예약, 좌석 등록, 결제, 주문 관리 리스트, 주문 수정, 주문 환불 등..
호텔 관리	호텔 리스트, 호텔 상세보기, 객실 리스트, 호텔 등록, 호텔 수정, 호텔 삭제 등..
호텔 예약	호텔 예약, 예약 내역, 예약 상세보기, 예약 취소, 예약 리스트, 고객 상태 변경 등..
호텔 리뷰	후기 리스트, 후기 작성, 후기 상세페이지, 후기 변경, 삭제
여행 정보 게시판	여행 정보 리스트, 정보 상세보기, 정보 등록, 정보 수정, 정보 삭제
여행 정보 게시판 댓글	댓글 리스트, 댓글 등록, 댓글 수정, 댓글 삭제
여행 계획 짜기	여행 계획 리스트, 계획 상세보기, 계획 등록, 계획 수정, 계획 삭제, 방문지 리스트, 방문지 등록 등..
이벤트 게시판	이벤트 리스트, 이벤트 상세보기, 이벤트 등록, 이벤트 수정, 이벤트 삭제

개발 내용과 범위

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위
소요 자원03 Front-end
메뉴의 흐름
화면 구현04 Back-end
DFD
DB05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

역할	이름	담당 범위
팀장	박지현	호텔 관리
PM	원필재	항공권 관리
부PM	나태준	호텔예약 / 이벤트 / 리뷰
PL	윤해연	항공권 예약
부PL	함수진	회원 관리
DBA	지유빈	여행 계획 짜기 / 댓글 가능
서기	고민희	여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

Front - 부PL 함수진

Default_decorator.jsp

- 홈페이지 기본 틀 잡기
- 각각의 모듈 siteMesh 적용
- 전체 디자인

Main.jsp

- 메인 페이지 디자인
- 메인 페이지 각 모듈 연결

Back - PL 윤해연, DBA 지유빈

DB.class

- 데이터 베이스 연결

AOP

- 실행 로그 출력 – Around Advice (실행 전 후)

Interceptor Util

- 매핑하는 모든 url 서버에서 Filter 처리
- 권한 처리

개발 내용과 범위 : 각 모듈 개발 범위

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

회원 관리

관리자

회원 리스트, 회원 상태/ 등급 수정

일반/비회원

로그인&아웃, 회원가입, 정보 수정, 보기, 탈퇴

이메일 찾기, 비밀번호 찾기

사업자

사업장 정보 수정

API

구글/카카오 로그인&로그아웃, 주소 검색

항공권 관리

관리자

국가, 공항, 항공사 등록 · 수정 · 삭제 관리

일반/비회원

항공권 검색, 공항 리스트 보기, 탑승 유형 선택

사업자

항공사 기종 리스트 · 등록 · 수정 · 삭제 관리

항공사 노선 리스트 · 등록 · 수정 · 삭제 관리

노선에 대한 운임 등록 · 수정 관리

운항 스케줄 리스트 · 등록 · 수정 · 삭제 관리

이벤트

이벤트

이벤트 리스트

이벤트 등록

이벤트 상세보기

이벤트 삭제

이벤트 수정

항공권 예약

주문/관리

항공 주문 리스트

항공권 예약 (탑승자, 수하물 등록)

항공 주문 수정 (여권번호 수정, 예약 취소 처리)

좌석 리스트

좌석 예약 등록

사업자

전체 주문 내역 리스트, 수정



BANGKOK

개발 내용과 범위 : 각 모듈 개발 범위

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

호텔 관리

상품 관리

호텔 리스트

호텔 상세보기 (객실 리스트)

호텔 등록 (객실 등록, 편의시설 등록)

호텔 수정 (객실 수정, 편의시설 수정)

호텔 삭제

여행 플랜 게시판

여행 계획

계획 리스트

계획 상세보기

계획 등록

계획 수정

계획 삭제

방문지 관리

방문지 관리

방문지 리스트

방문 일정 상세보기

방문지 추가

방문 일정 수정

방문지 삭제

호텔 예약

호텔 예약

예약 리스트

예약 결제

예약 상세보기

예약 삭제

관리자 리스트

예약자 상태 변경

리뷰

리뷰

리뷰 리스트

리뷰 상세보기

리뷰 등록

리뷰 삭제

여행 정보

여행 정보 게시판

리스트

상세보기

등록

수정

삭제

댓글

댓글

댓글 리스트

댓글 작성

댓글 수정

댓글 삭제

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

Hardware

자원분류	자원 이름	비고
CPU	Intel Core i5-3550U @ 3.30GHz	최저 사양
RAM	DDR3 8GB 이상	최저 사양
Storage	SSD 256GB 이상	최저 사양

Software

자원분류	자원 이름	비고
OS	Window 10, 11	
DB 설계	SQL Developer	
IDE	Spring Tool Suite 3	
DBMS	Oracle 11g	
WAS	Tomcat 9	
개발 프레임 워크	JDK 11, Spring framework	
DB Modeling	ER-Win 7	
Team Work	Visual SVN	
사용 라이브러리	jQuery, Bootstrap4, cos, JSTL, PageObject, fileUtil, SiteMesh, myBatis, Lombok, AspectJ	

03 Front-end

메뉴의 흐름

화면 구현



BANGKOK

메뉴의 흐름 상세도 – 회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

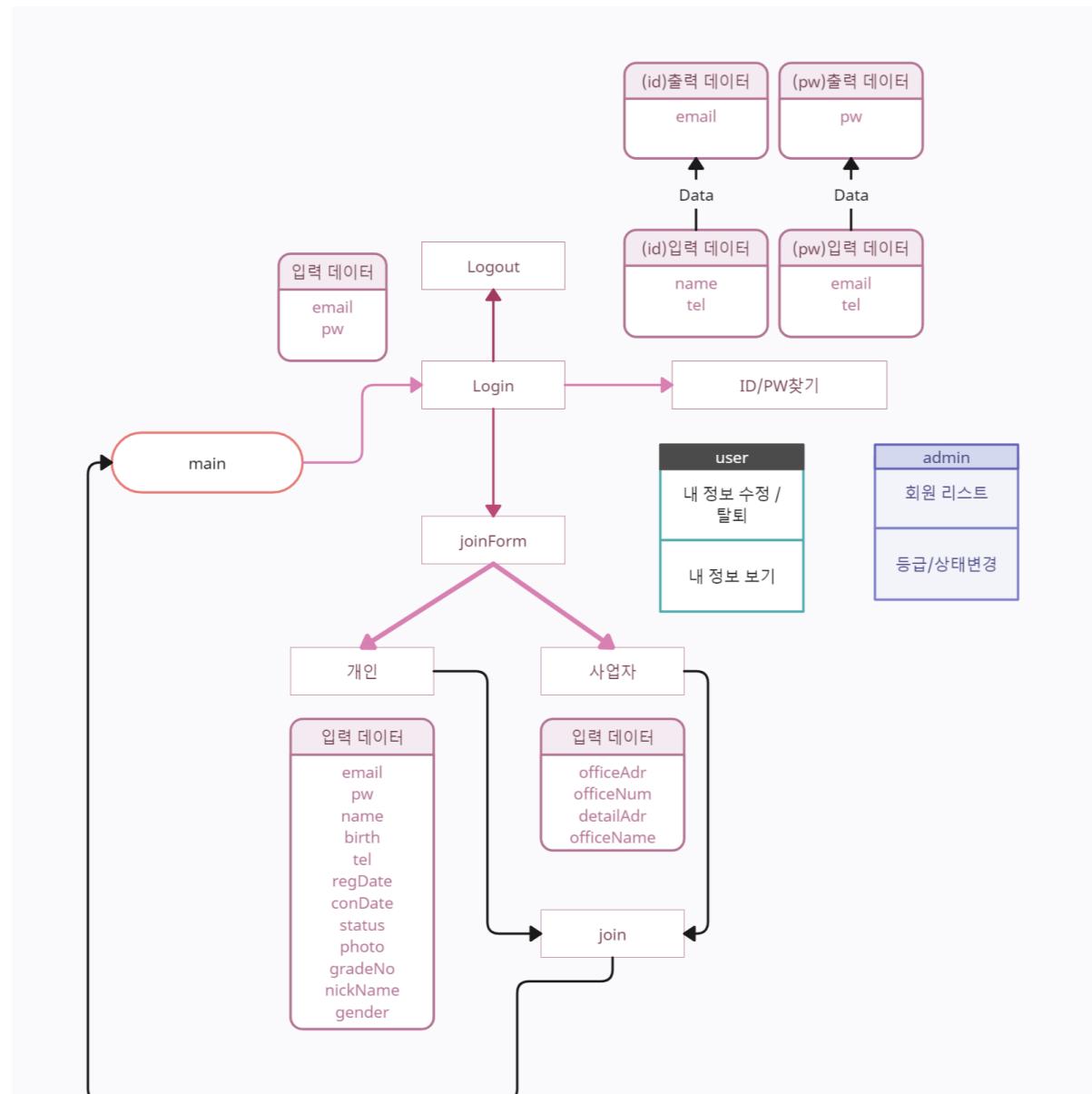
시연

핵심코드

오류수정

추후 개발 내용

후기



01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

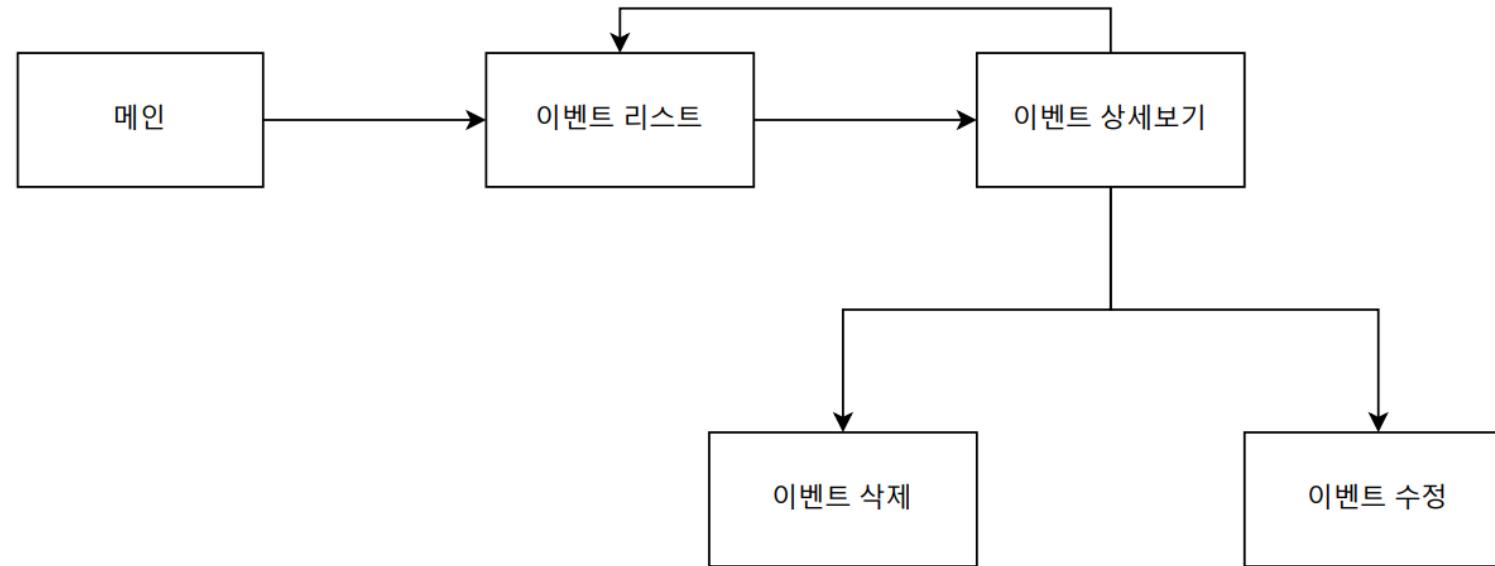
핵심코드

오류수정

추후 개발 내용

후기

이벤트



입력 데이터
이미지, 글 이미지, 이벤트 시작일, 이벤트 종료일

출력 데이터
-이벤트 번호, 제목, 글 이미지, 이미지, 이벤트 시작일, 이벤트 종료일

메뉴의 흐름 상세도 – 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

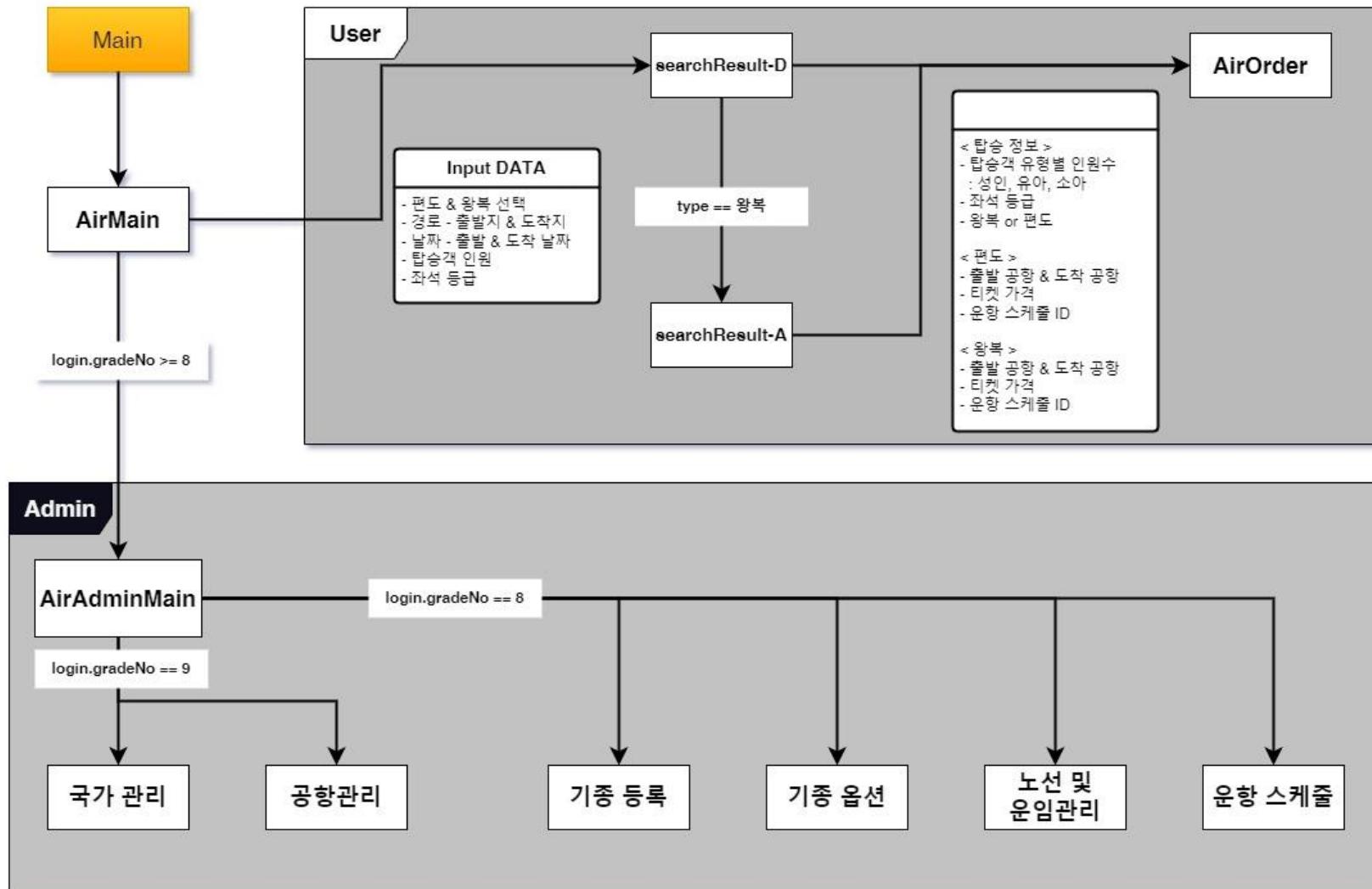
핵심코드

오류수정

추후 개발 내용

후기

< 항공 상품 관리 - 메뉴 흐름도 >



메뉴의 흐름 상세도 – 항공권 예약

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

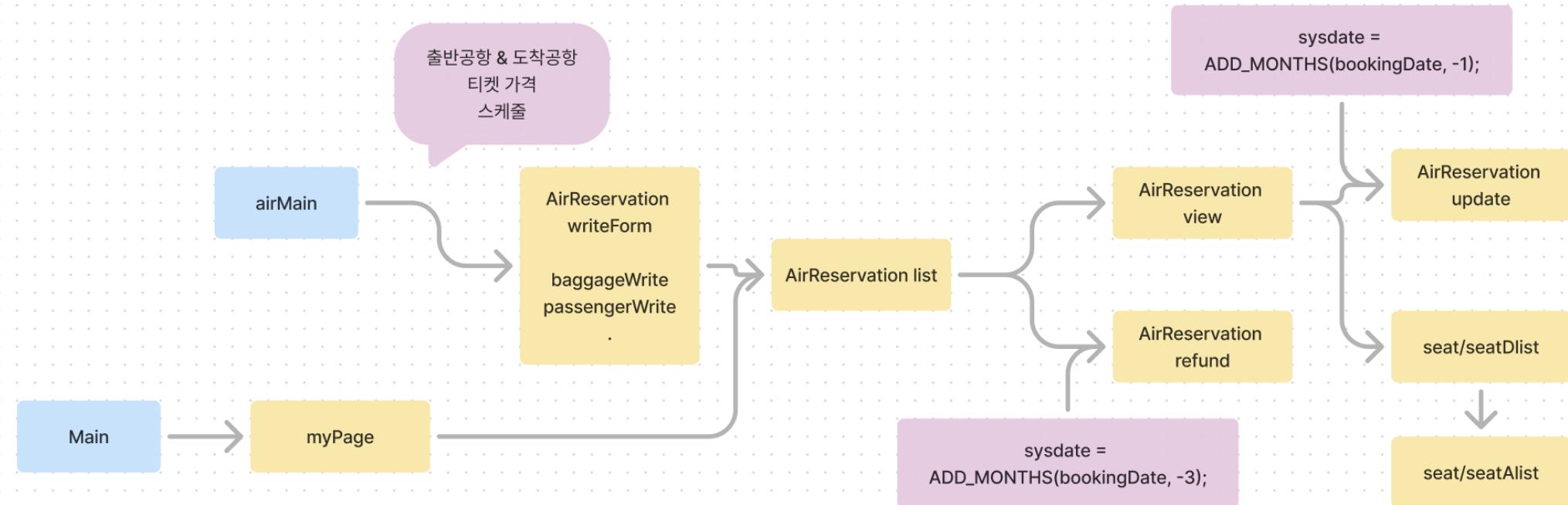
시연

핵심코드

오류수정

추후 개발 내용

후기





BANGKOK

메뉴의 흐름 상세도 – 호텔 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

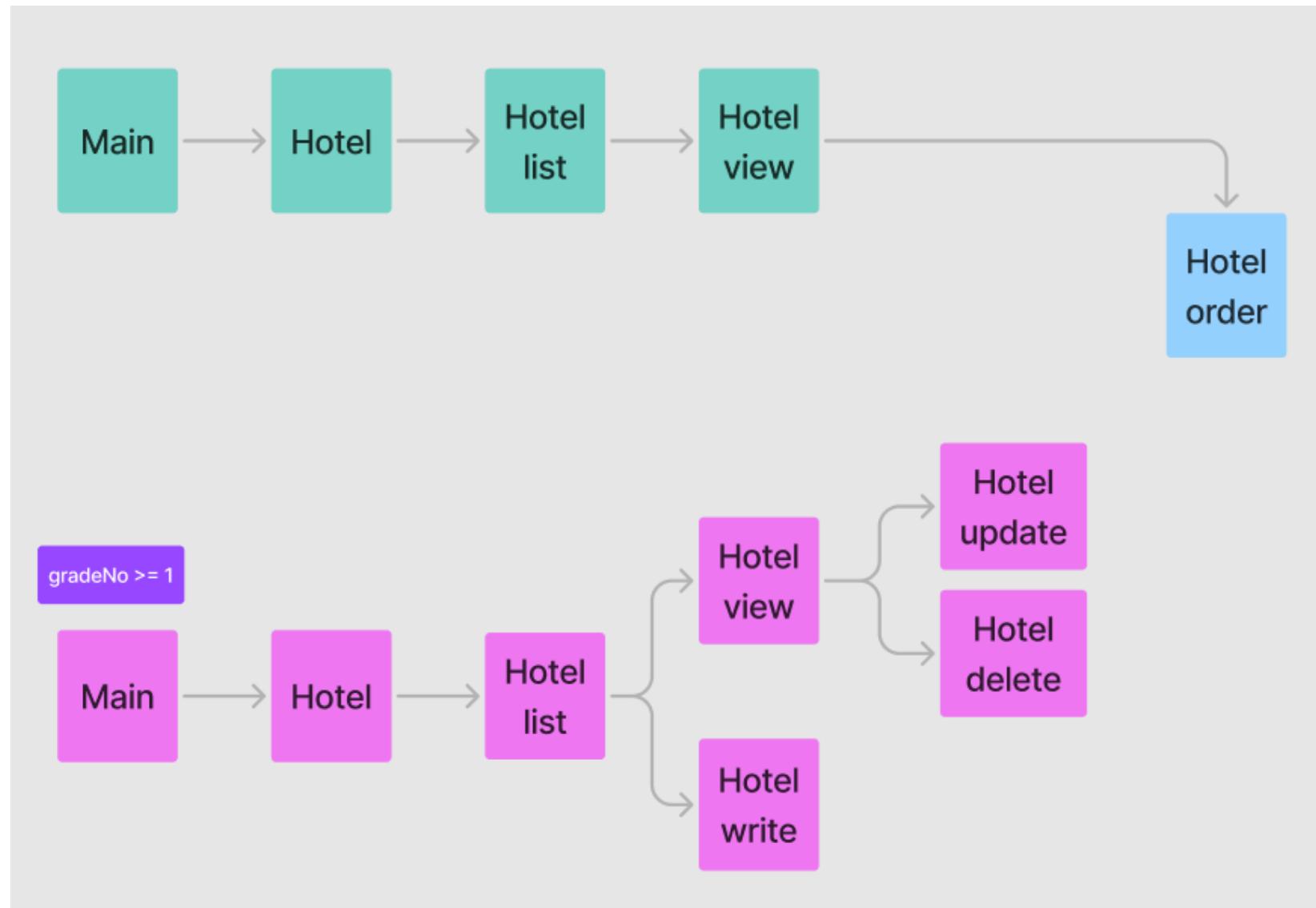
시연

핵심코드

오류수정

추후 개발 내용

후기



01 프로젝트 개요

주제와 목표

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

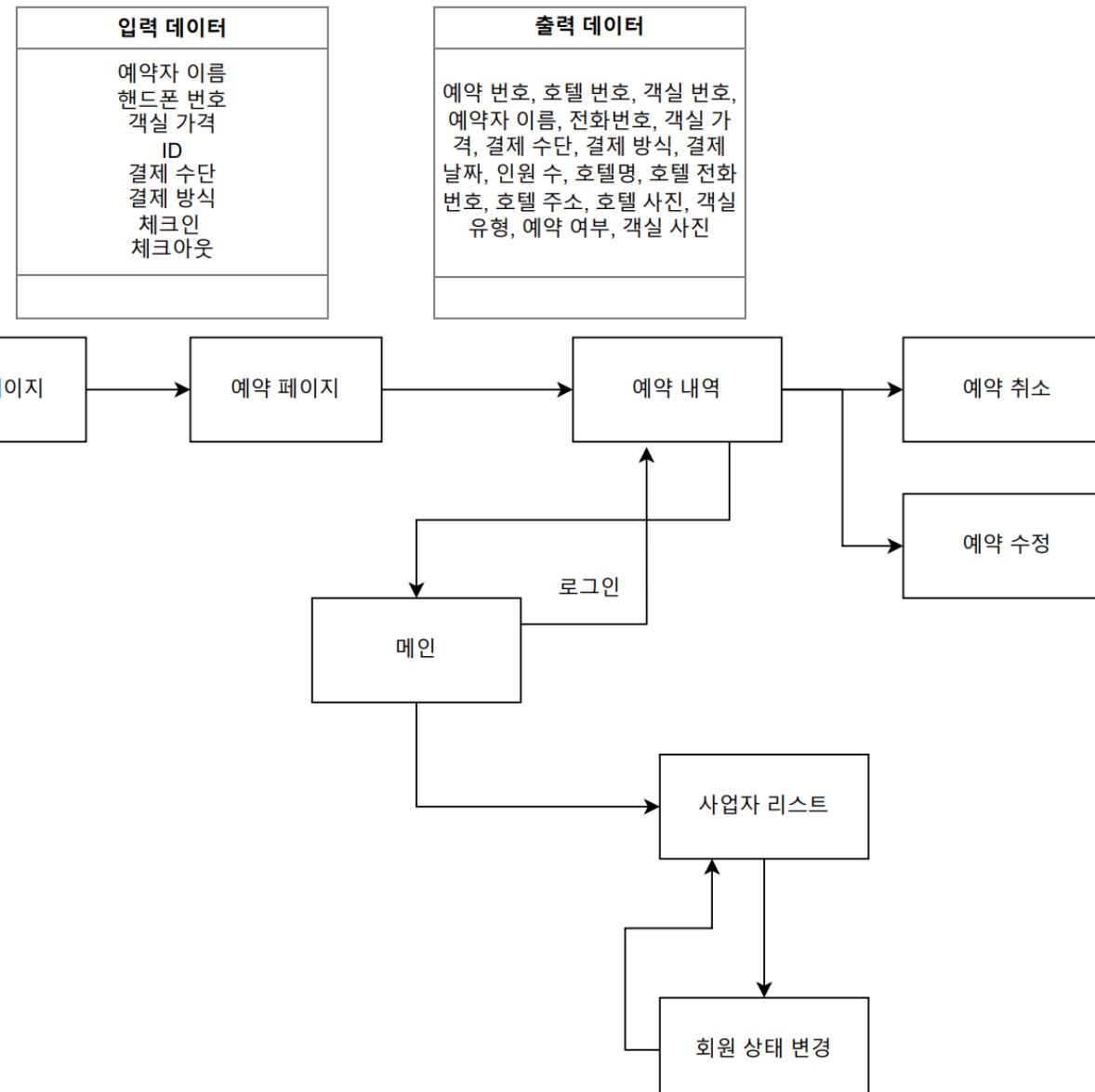
핵심코드

오류수정

추후 개발 내용

후기

예약



01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

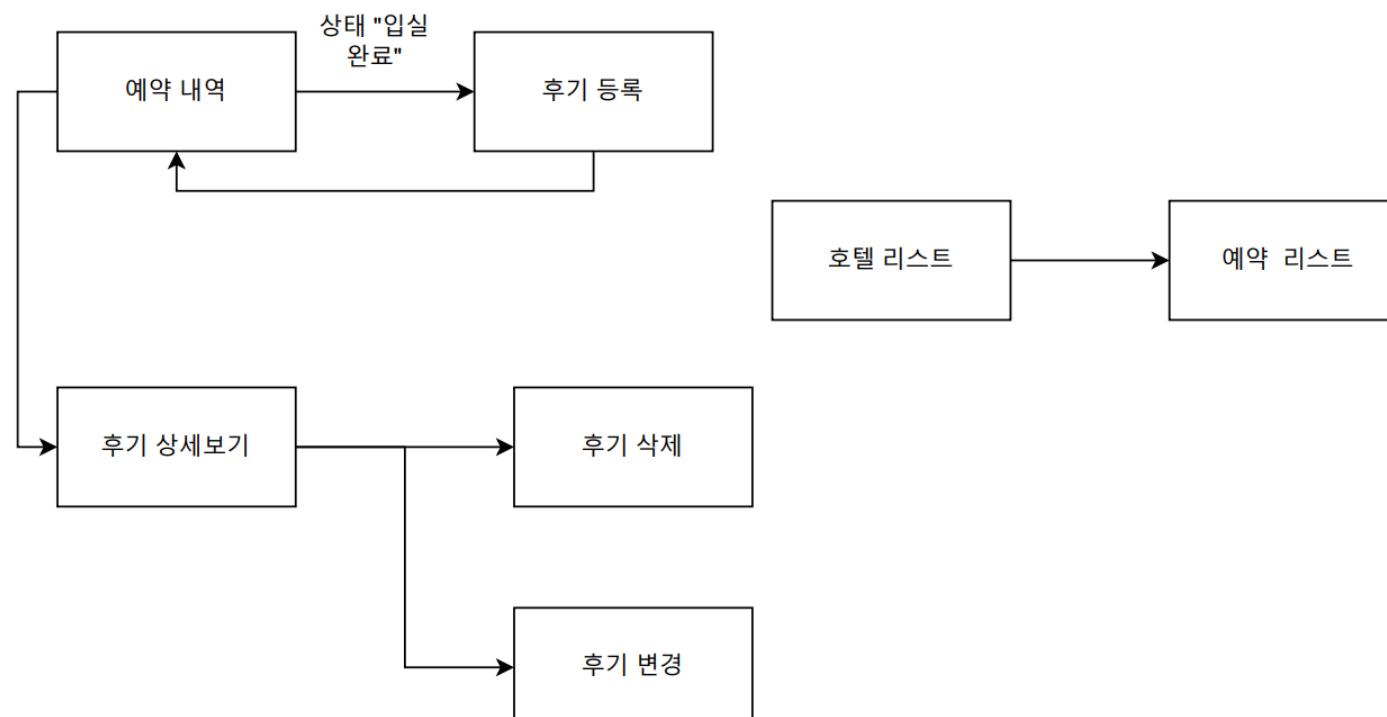
핵심코드

오류수정

추후 개발 내용

후기

후기



01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

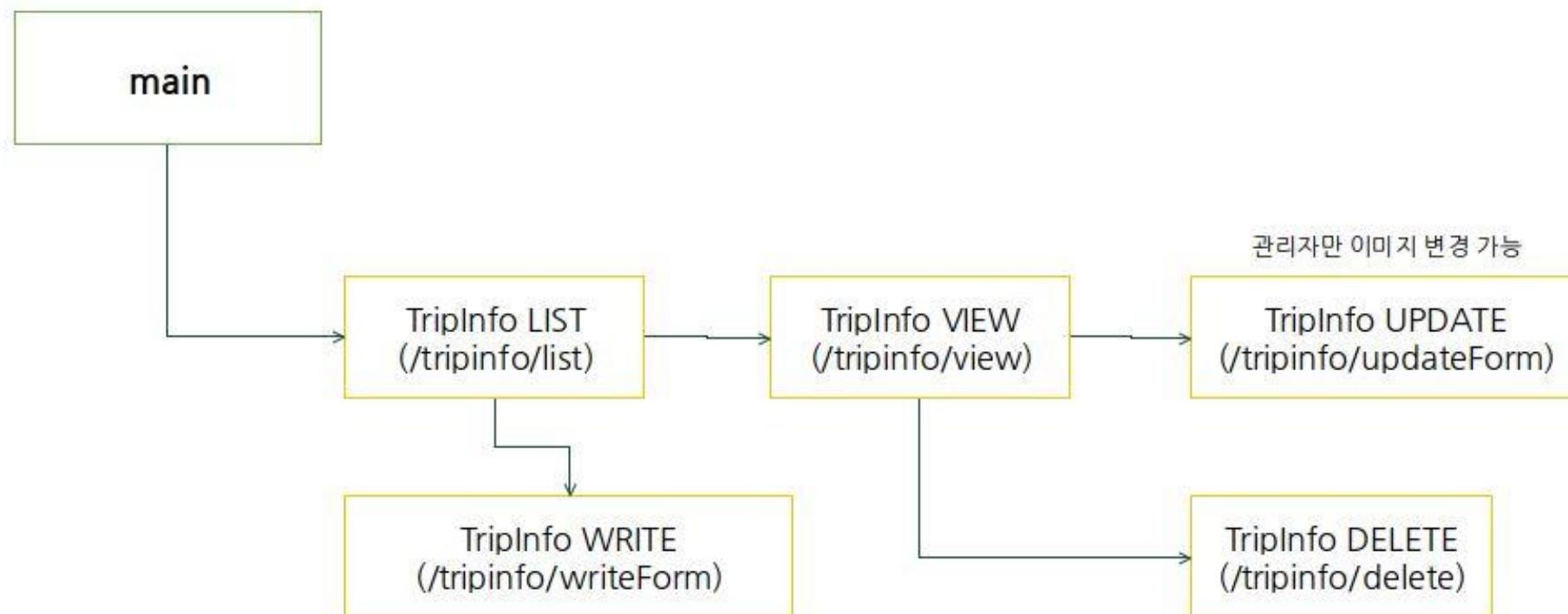
핵심코드

오류수정

추후 개발 내용

후기

화면 흐름도



01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

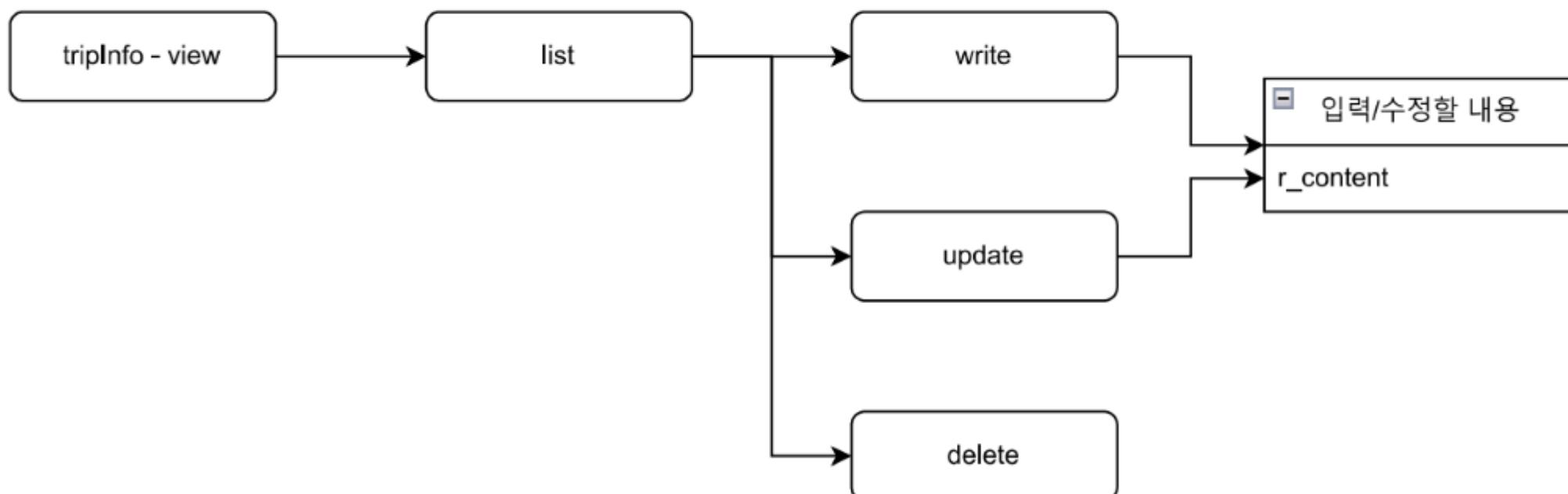
핵심코드

오류수정

추후 개발 내용

후기

reply





BANGKOK

메뉴의 흐름 상세도 – 여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

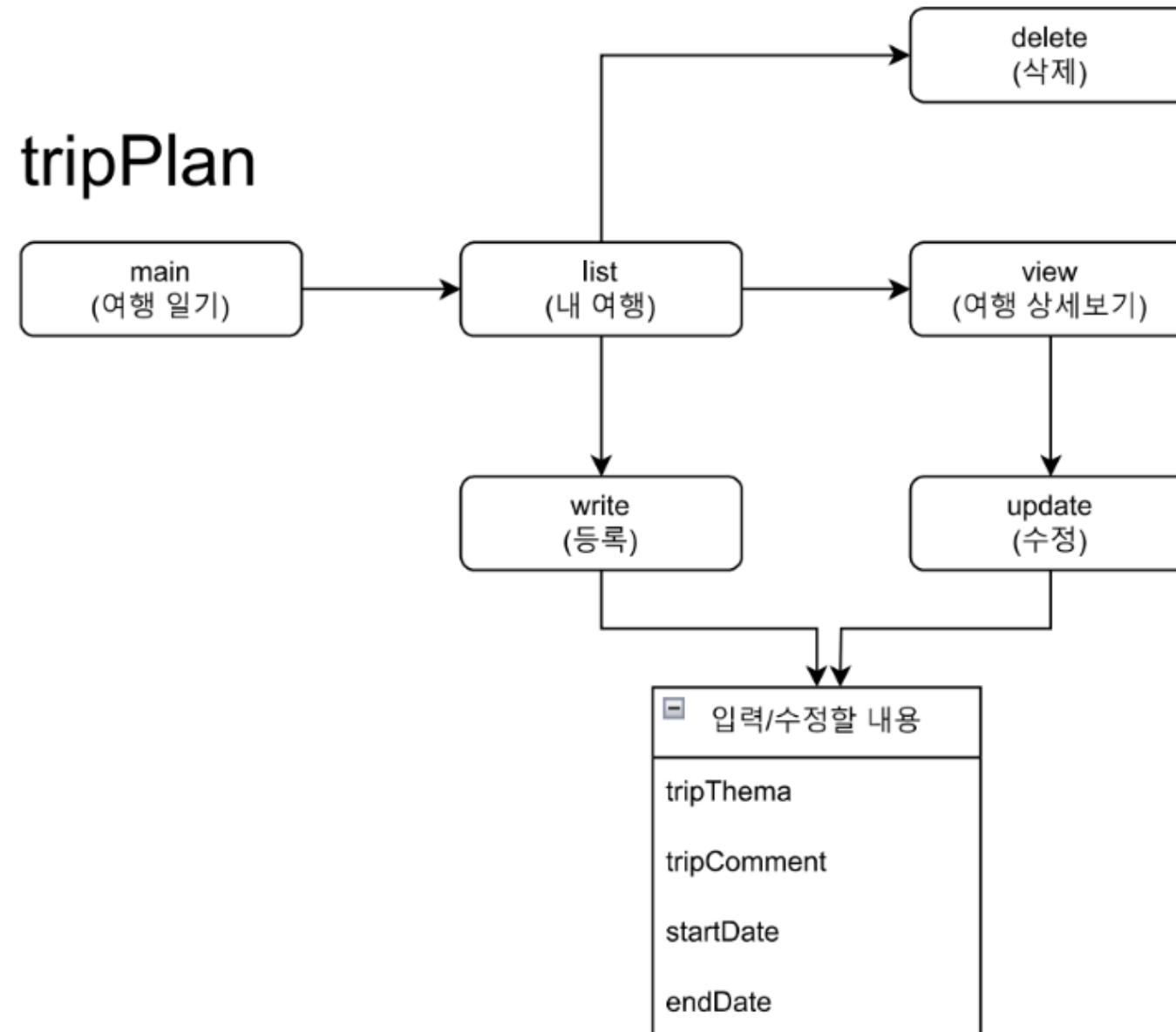
시연

핵심코드

오류수정

추후 개발 내용

후기



01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

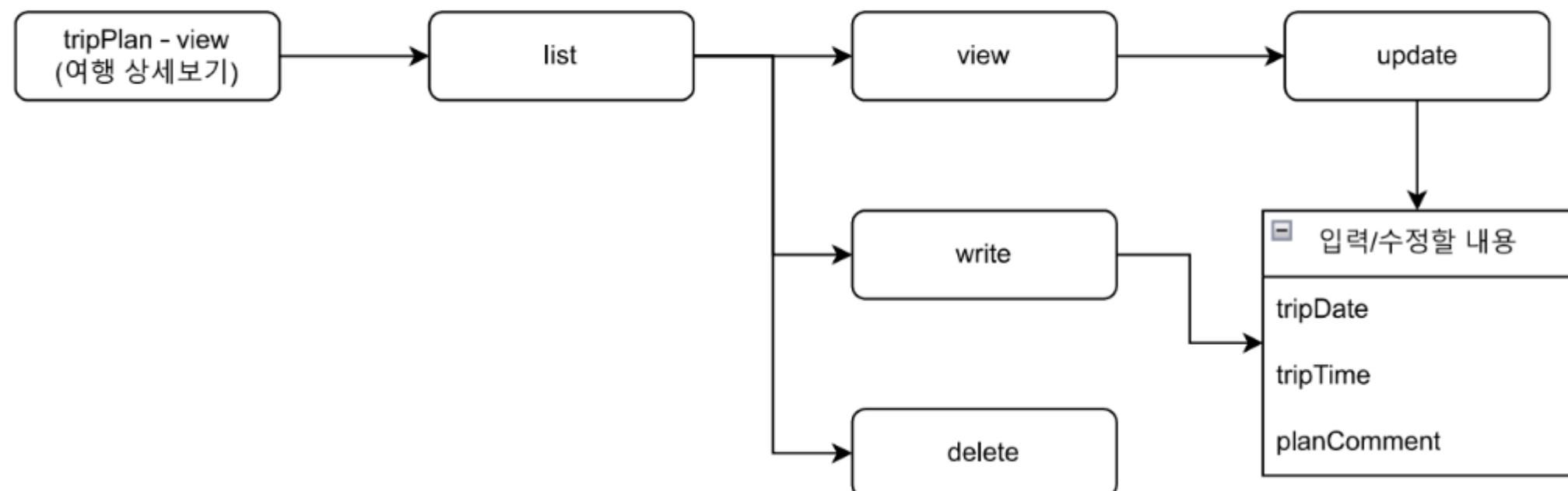
핵심코드

오류수정

추후 개발 내용

후기

planDetail



03 Front-end : 화면 구현

회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK login_first

1 / 9

BANGKOK

여행일기
호텔
여행
즐길거리

A2

A3 Login

A4 사진

A5 이메일로 진행하기

A6 카카오 A7 구글

회원관리는 관리자(9)
예약내역은 모든 회원
보여지게

로그인 A1

[출력]
A1 : 로그인 버튼
A2 : 로그인 모달창
A3 : 로그인 텍스트
A4 : 로그인 관련 사진
A5 : 이메일 전용 회원가입/id,pw찾기
A6 : 카카오 api 로그인 버튼
A7 : 구글 api 로그인 버튼

[이동]
A5 : /member/loginModal.do
A6 : /member/kakaoLogin.do
A7 : /member/googleLogin.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK login

2 / 9

The diagram illustrates the user interface flow for the '회원관리' (Member Management) feature. It starts with the main menu on the left, which includes categories like 여행일기, 호텔, 여행, 즐길거리, 이벤트, 예약내역, 회원관리, 마이페이지, and 사업자페이지. A large orange button labeled '로그인' (Login) is also present. A green arrow points from the '회원관리' menu item to the 'Login' screen on the right. The 'Login' screen has fields for 'Login' (B1), 'email@domain.com' (B2), 'pw' (B3), and a '로그인' (Login) button (B4). Below these are links for '회원가입' (B5) and 'ID/PW찾기' (B6).

[입력]

- B2 : 이메일 입력칸
- B3 : 비밀번호 입력칸

[출력]

- B1 : 로그인 텍스트

[이동]

- B4 : /member/login.do
- B5 : /member/joinForm.do
- B6 : /member/findForm.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK joinForm

3 / 9

BANGKOK

여행일기

호텔

여행

즐길거리

이벤트

예약내역

회원관리

마이페이지

사업자페이지

로그인

회원가입

C1 개인 C2 사업자

이름 : C3 이름 입력

닉네임 : 중복확인 C4

이메일 : 중복확인 C4

비밀번호 :

비밀번호 확인 :

첨부파일 등 입력폼 ...
사업자 전용 추가 폼 등

C5 가입 C6 취소

reset

[입력]

C3 : 이름, 닉네임, 이메일, 비밀번호,
비밀번호 확인, 프로필 사진
- 사업자 -
사업장 주소, 회사명, 사업자 번호

[출력]

C1 : 회원가입 텍스트
C2: 개인 / 사업자 라디오 버튼

[이동]

C4 : /member/checkNick.do
/member/checkEmail.do
C5 : /member/join.do
C6 : /main/main.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK findEmail

4 / 9

[출력]

- D1 : 프로젝트 로고
- D2 : 이메일 찾기 텍스트
- D6 : 찾기 결과창
- D7 : 이메일 / 비밀번호 라디오버튼

[입력]

- D3 : 이름 / 연락처 입력

[이동]

- D4 : /member/findEmail.do
- D5 : /main/main.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK findPw

5 / 9

[출력]
D10 : 비밀번호 찾기 결과 팝업

[입력]
D8 : 이메일 / 이름 입력

[이동]
D9 : /member/findPw.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK memberList

6 / 9

BANGKOK

여행일기

호텔

여행

즐길거리

이벤트

예약내역

회원관리

마이페이지

사업자페이지

로그아웃

E1 회원 관리

E2 이름	성별	생년월일	이메일	연락처	상태	등급이름
홍길동	남자	1981-12-31	test@naver.com	010-1234-5678	정상	일반회원
박순자	여자	1956-05-31	test2@naver.com	010-9876-5432	정상	사업자

E5 변경

E6 취소

[출력]
 D1 : 회원관리 텍스트
 D2 : 회원 기본 정보
 D3 : 상태 이름
 D4 : 등급 이름

[이동]
 D5 : /member/memberUpdate.do
 D6 : /member/Memberlist.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK myPage

7 / 9



01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK updateForm

8 / 9

BANGKOK

여행일기

호텔

여행

즐길거리

이벤트

예약내역

회원관리

마이페이지

사업자페이지

로그아웃

G1 내 정보 수정

G2 닉네임 :

닉네임을 입력해주세요

연락처 :

'-' 포함 입력해주세요

비밀번호 :

비밀번호를 입력해주세요

비밀번호 확인 :

비밀번호 재입력해주세요

프로필 사진 :

첨부파일

G3

변경

취소

G4

[입력]

G2 : 회원 정보 입력란

[출력]

G1 : 내 정보 수정 텍스트

[이동]

G3 : /member/update.do

G4 : /member/myPage.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

BANGKOK officeUpdateForm

9 / 9

BANGKOK

여행일기

호텔

여행

즐길거리

이벤트

예약내역

회원관리

마이페이지

사업자페이지

로그아웃

사업자 정보 수정

G5 사업장 주소 :

닉네임을 입력해주세요

사업자 번호 :

'-' 포함 입력해주세요

비밀번호:

비밀번호를 입력해주세요

비밀번호 확인 :

비밀번호 재입력해주세요

G6

변경

취소

G7

[입력]

G5 : 사업장 정보 입력란

[이동]

G6 : /member/officeUpdate.do

G7 : /member/officePage.do

03 Front-end : 화면 구현

이벤트 게시판

화면구현 : 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

이벤트

D4

인기 호텔 연휴 예약은 지금부터!

**추석 미리 예약
최대 5만원 할인**

D1



qqqqa@ 마감임박

2024-09-02 ~ 2024-09-19

D2

D3

추석 10월 연휴 국내여행은!
**숙박세일 페스타
최대 3만원 할인**

L3



test

마감임박

2024-09-01 ~ 2024-09-02

인기 호텔 연휴 예약은 지금부터!

**추석 미리 예약
최대 5만원 할인**



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!

**추석 미리 예약
최대 5만원 할인**



상쾌한 북한 여행입니다 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!
**추석 미리 예약
최대 5만원 할인**

L3



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!

**추석 미리 예약
최대 5만원 할인**



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!

**추석 미리 예약
최대 5만원 할인**



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!
**추석 미리 예약
최대 5만원 할인**

L3



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!

**추석 미리 예약
최대 5만원 할인**



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!

**추석 미리 예약
최대 5만원 할인**



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

인기 호텔 연휴 예약은 지금부터!
**추석 미리 예약
최대 5만원 할인**

L3



상쾌한 북한 여행 마감임박

2024-07-29 ~ 2024-10-30

이벤트 등록

L2

출력

D1:이벤트 이미지

D2:이벤트 제목

D3:이벤트 시작일 ~ 종료일

D4:이벤트 글 이미지

링크

L1 :/event/list.do?page=1~10

L2 :/event/writeForm.do

L3 :/event/view.do?eventNo=" + eventNo + "&\${pageObject.pageQuery}



BANGKOK

화면구현 : 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

이벤트 등록

제목	제목 입력 : 3자 이상 100자 이내	A1
이미지	파일 선택 선택된 파일 없음	A2
글 이미지	파일 선택 선택된 파일 없음	A3
이벤트 시작일	A4	
이벤트 종료일	L2	A5
<input type="button" value="등록"/> <input type="button" value="다시입력"/> <input type="button" value="취소"/>		L1 L3

입력

A1: 이벤트 제목

A2:이벤트 이미지

A3:이벤트 글 이미지

A4:이벤트 시작일

A5:이벤트 종료일

링크

L1 :/event/write.do

L2 :button type=reset

L3 :button onclick=history.back();

화면구현 : 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기



출력

D1:이벤트 제목

D2:이벤트 이미지

링크

L1 :/event/updateForm.do

L2 :/event/delete.do

L3 :/event/list.do?page=""&perPageNum=""



BANGKOK

화면구현 : 이벤트 게시판

01 프로젝트 개요

주제와 목적

출력

D1: 번호 세팅 및 입력x

02 개발 전반

요구 사항

출력 + 입력

개발 내용과 범위

DA1: 제목 세팅 및 입력

소요 자원

DA2: 시작일 세팅 및 입력

03 Front-end

메뉴의 흐름

DA3: 종료일 세팅 및 입력

화면 구현

입력

A1: 이미지 등록

04 Back-end

DFD

A2: 이미지 글 등록

DB

05 프로젝트 시연

시연

링크

L1 :/event/update.do

핵심코드

L2 :button type=reset

오류수정

L3 :button onclick=history.back();

추후 개발 내용

이벤트 수정

번호	22	D1
제목	qqqqaa	DA1
이미지	<input type="file"/> 선택된 파일 없음	A1 A2
글 이미지	<input type="file"/> 선택된 파일 없음	
이벤트 시작일	2024-09-02	DA2
이벤트 종료일	2024-09-19	DA3
<input type="button" value="등록"/> L2 <input type="button" value="다시입력"/> L1 <input type="button" value="취소"/> L3		

화면구현 : 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기



링크

L1 : 이벤트 삭제 /event/delete.do

03 Front-end : 화면 구현

항공권 상품 관리

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

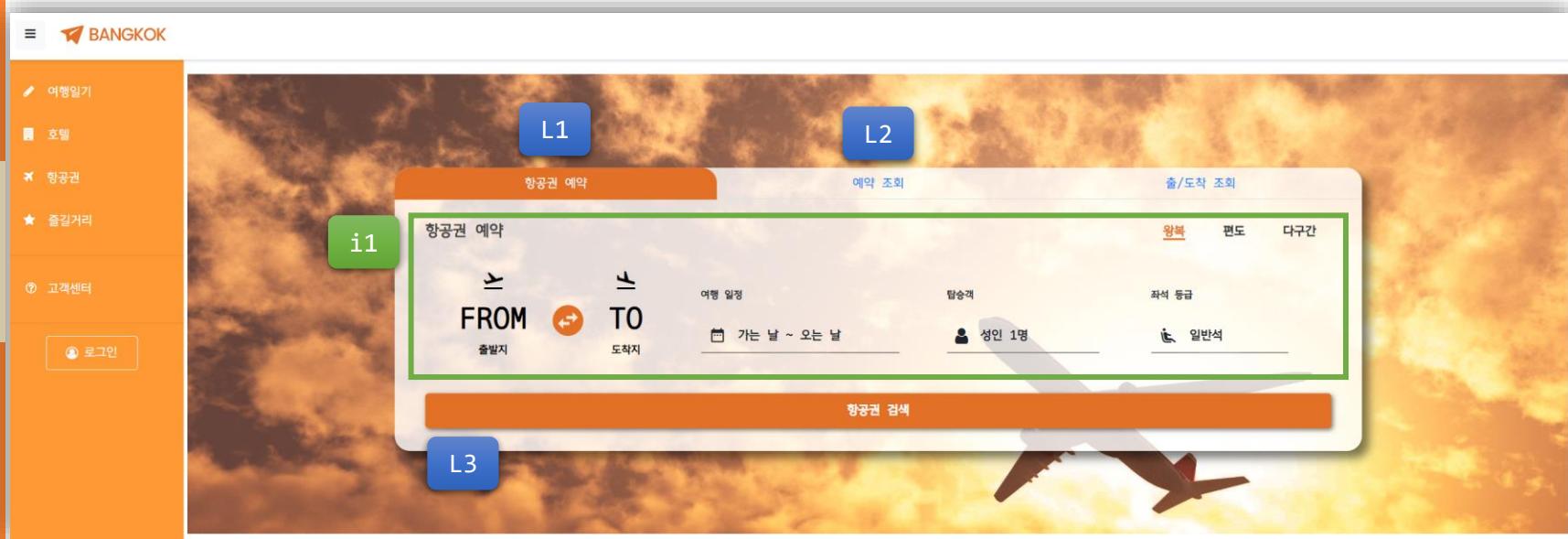
시연

핵심코드

오류수정

추후 개발 내용

후기



입력 데이터 <i>

i1 – 출발지 선택, 도착지 선택
여행 일정, 편도&왕복 선택
탑승객 유형
좌석 등급

출력 데이터 <p>

링크 <L>

L1 – air/main.do
L2 – air/orderList.do
L3 – air/searchResult.do

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

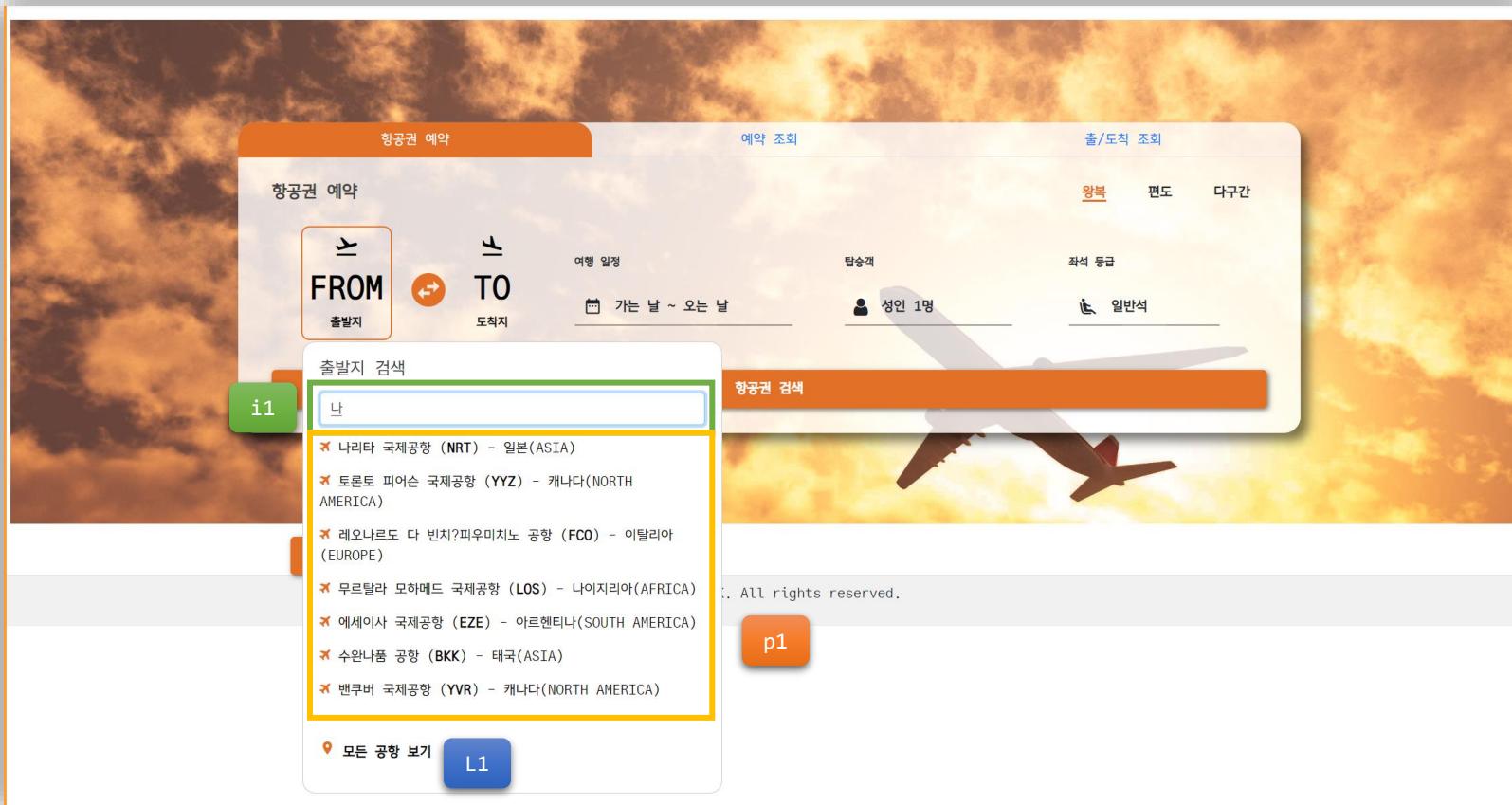
시연

핵심코드

오류수정

추후 개발 내용

후기



입력 데이터 <i>

i1 – 공항, 국가, 도시 검색

출력 데이터 <p>

p1 – 공항 한글, 공항 코드, 국가, 국가코드

링크 <L>

L1 – air/airportList.do

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

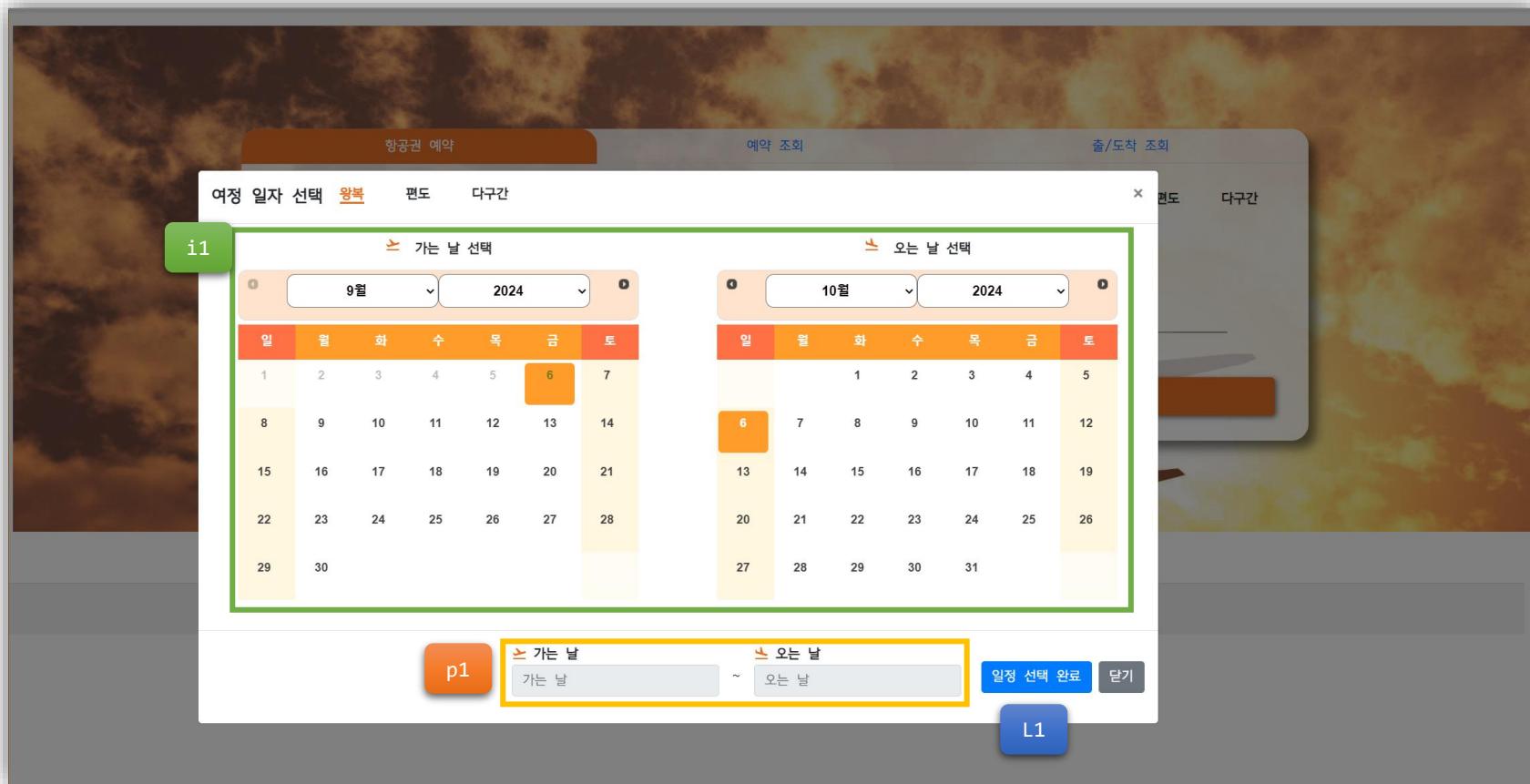
시연

핵심코드

오류수정

추후 개발 내용

후기



입력 데이터 <i>

i1 – 출발일, 복귀일 선택

출력 데이터 <p>

p1 – 선택한 날짜로 업데이트

링크 <L>

L1 – main 페이지에 데이터 전달

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

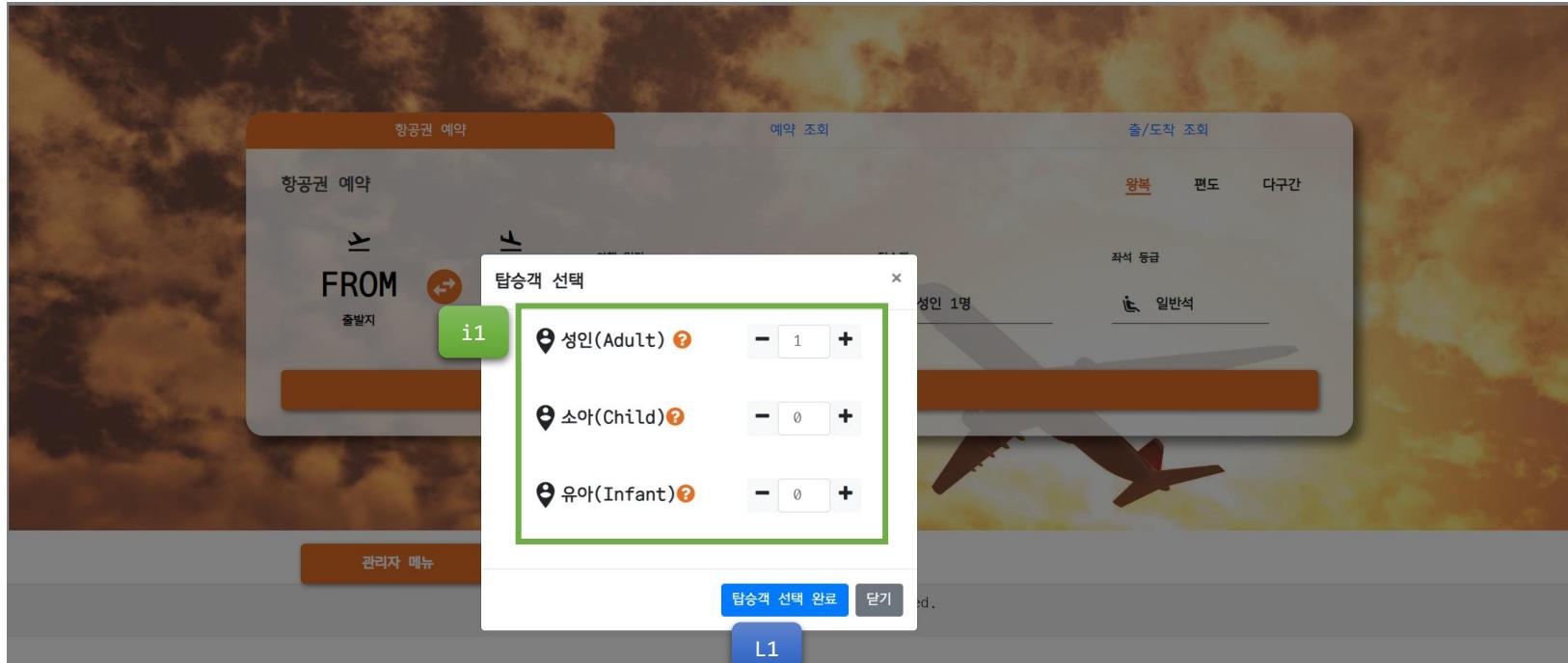
시연

핵심코드

오류수정

추후 개발 내용

후기



입력 데이터 <i>

I1 – 연령별 탑승객 유형 인원 수 입력

출력 데이터 <p>

P1 – 없음.

링크 <L>

L1 – main 페이지에 데이터 전달



화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

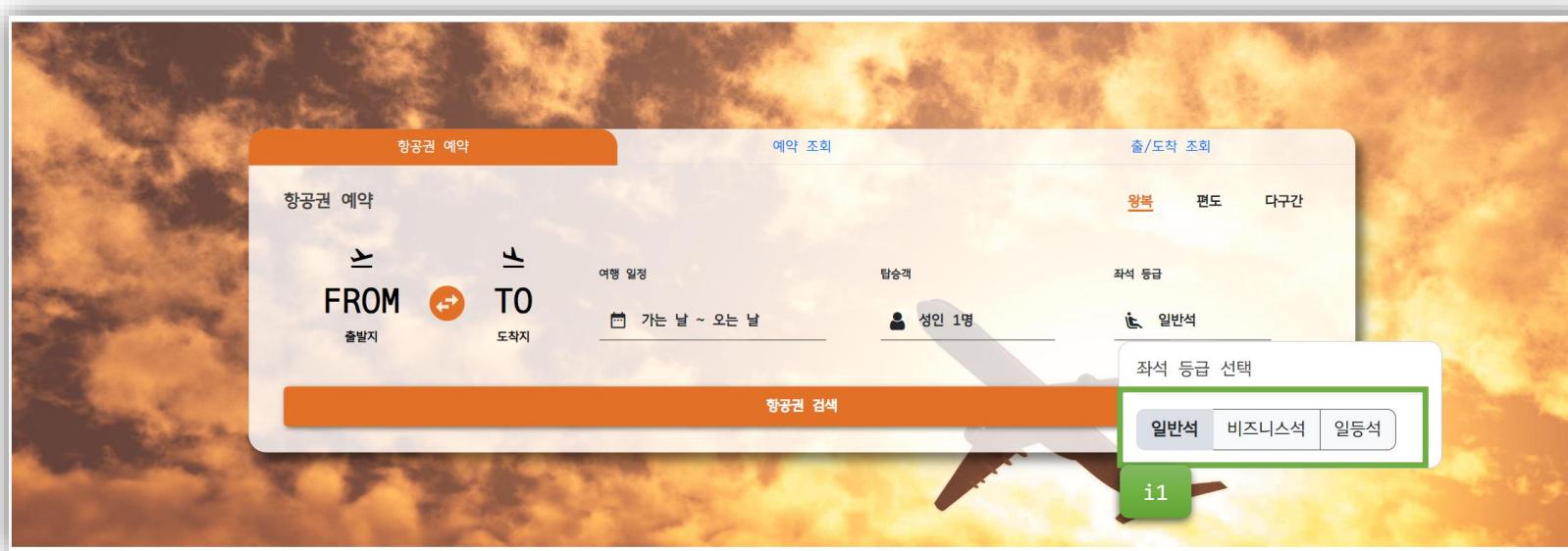
시연

핵심코드

오류수정

추후 개발 내용

후기



입력 데이터 <i>

|1 – 검색할 좌석 등급 선택

출력 데이터 <p>

P1 – 없음.

링크 <L>

L1 – 없음.

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

The screenshot shows a flight search interface with the following details:

- ICN** (인천 국제공항) - **NRT** (나리타 국제공항)
- 2024-10-01**
- 성인 : 1**
- 일반석**
- 항공편 검색**
- L1**

The results table displays five flight options:

항공사	편	출발	도착	소요시간	기준	선택
KOREAN AIR	KE5741	06:45	09:20	2시간 35분	192000 원	항공편 선택
KOREAN AIR	KE703	10:10	12:40	2시간 30분	192000 원	항공편 선택
KOREAN AIR	KE711	13:00	15:25	2시간 25분	192000 원	항공편 선택
KOREAN AIR	KE5743	14:45	17:10	2시간 25분	192000 원	항공편 선택
KOREAN AIR	KE713	17:25	19:45	2시간 20분	192000 원	항공편 선택

i1 (Top bar)

p1 (Flight KE5741 row)

L2 (Flight KE703 row)

L3 (Flight KE713 row)

p2 (Bottom summary bar: 가는 편 : 대한항공 (KE5741) 2024-10-01 (06:45 ~ 09:20) 일반석 총액 : 192000 원)

L4 (Bottom right button: 항공편 예약)

입력 데이터 <i>

I1 – 출발지 선택, 도착지 선택
여행 일정, 편도&왕복 선택
탑승객 유형
좌석 등급

출력 데이터 <p>

P1 – 검색된 항공편 리스트

항공사 이미지, 항공사, 편명
운항 스케줄 (출발 ~ 도착), 소요시간
출발지 – 도착지
검색한 좌석 등급
항공편 금액

링크 <L>

L1 – 항공편 재 검색

L2 – 항공편 선택 : 아래 필드로 데이터 전달

L3 – 편도인 경우 : /air/order.do
왕복인 경우 : /air/selectFlight-A.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

항공 관리

요약 국가 및 공항 관리 항공사 기종 관리 노선 및 운임 관리 운항 스케줄 관리

국가 및 공항 관리 add L1

국가 옆에 [공항 리스트] 버튼을 누르면 등록된 공항을 볼 수 있습니다.

대륙	국가	국가 코드	공항 리스트	수정	삭제
아시아 (Asia)	대한민국 South Korea	KR	[공항 리스트]	L2	L3
	일본 Japan	JP	[공항 리스트]		
	중국 China	CN	[공항 리스트]		
	인도 India	IN	[공항 리스트]		
	사우디아라비아 Saudi Arabia	SA	[공항 리스트]		
	태국 Thailand	TH	[공항 리스트]		
	베트남 Vietnam	VN	[공항 리스트]		
	싱가포르 Singapore	SG	[공항 리스트]		
	필리핀 Philippines	PH	[공항 리스트]		
북미 (North America)	미국 United States	US	[공항 리스트]		
	캐나다 Canada	CA	[공항 리스트]		
	멕시코 Mexico	MX	[공항 리스트]		

입력 데이터 <i>

I1 - 없음.

출력 데이터 <p>

P1 - 대륙 리스트

P2 - 대륙별 국가 리스트
: 국가, 영문, 코드

링크 <L>

L1 - add: 국가 추가

L2 - 수정: 국가 이름 수정 모달

L3 - 삭제: 국가 삭제

L4 - 공항리스트
: 해당 국가 공항 리스트 collapse

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

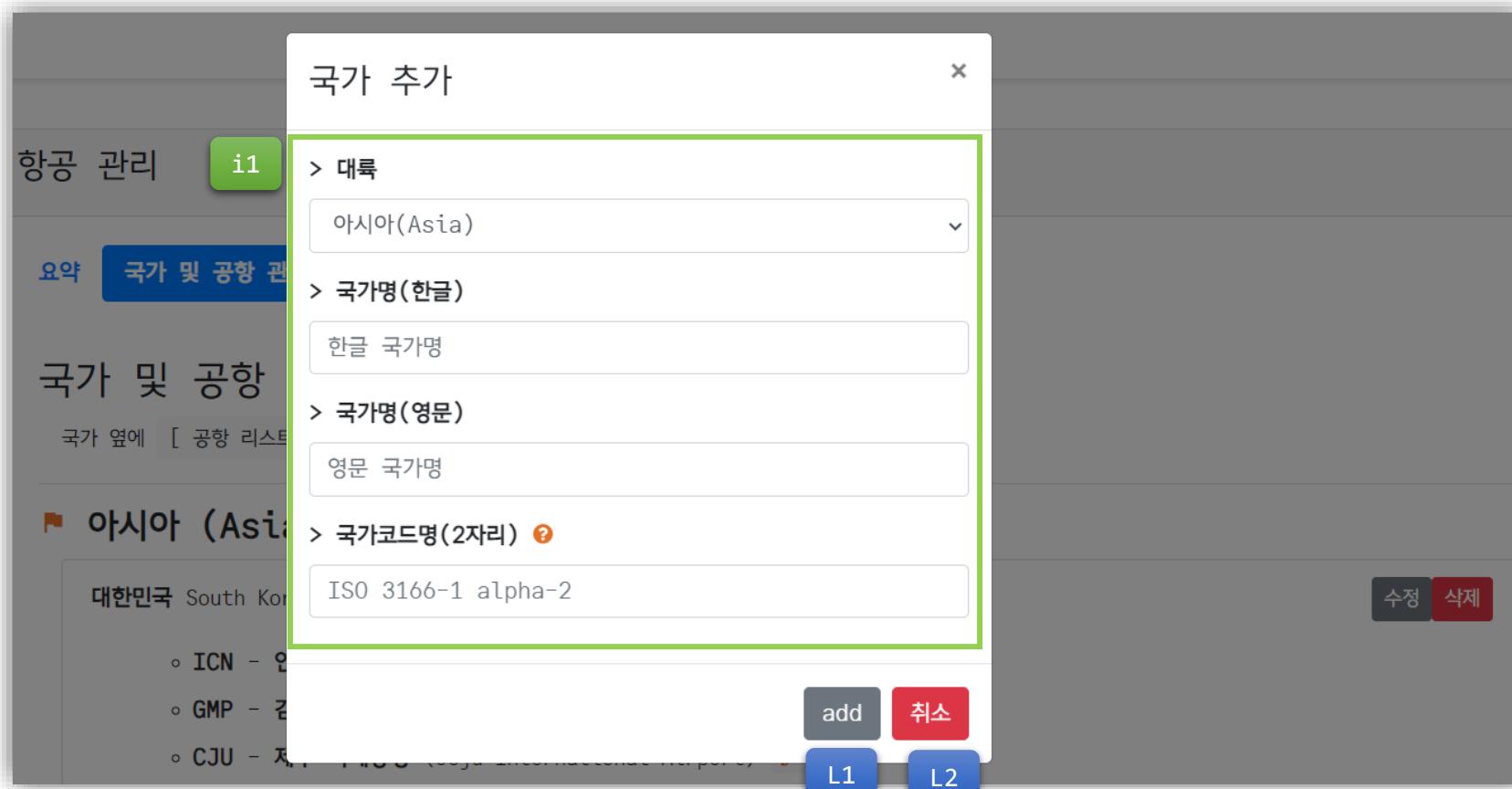
시연

핵심코드

오류수정

추후 개발 내용

후기



입력 데이터 <i>

I1 – 대륙, 국가 한글, 국가 영문, 코드

출력 데이터 <p>

P1 – 없음.

링크 <L>

L1 – add : 국가 추가 -> AdminNoc.do
L2 – 취소 : modal.hide();



화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

항공 관리

요약 국가 및 공항 관리 항공사 기종 관리 노선 및 운임 관리 운항 스케줄 관리

국가 및 공항 관리 [add](#)

국가 옆에 [공항 리스트] 버튼을 누르면 등록된 공항을 볼 수 있습니다.

▶ 아시아 (Asia)

국가	국가 코드	공항 리스트	수정	삭제
대한민국 South Korea	KR	<ul style="list-style-type: none"> ICN - 인천 국제공항 (Incheon International Airport) 수정 삭제 GMP - 김포 국제공항 (Gimpo International Airport) 수정 삭제 CJU - 제주 국제공항 (Jeju International Airport) 수정 삭제 	수정	삭제
일본 Japan	JP	[공항 리스트]	수정	삭제
중국 China	CN	[공항 리스트]	수정	삭제
인도 India	IN	[공항 리스트]	수정	삭제

+ 공항 추가 [L1](#)

입력 데이터 <i>

I1 – 없음.

출력 데이터 <p>

P1 – 국가별 공항 리스트

링크 <L>

L1 – 공항 추가 : 공항 추가 모달
 L2 – 수정 : 공항 이름 수정 모달
 L3 – 삭제 : 공항 삭제

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

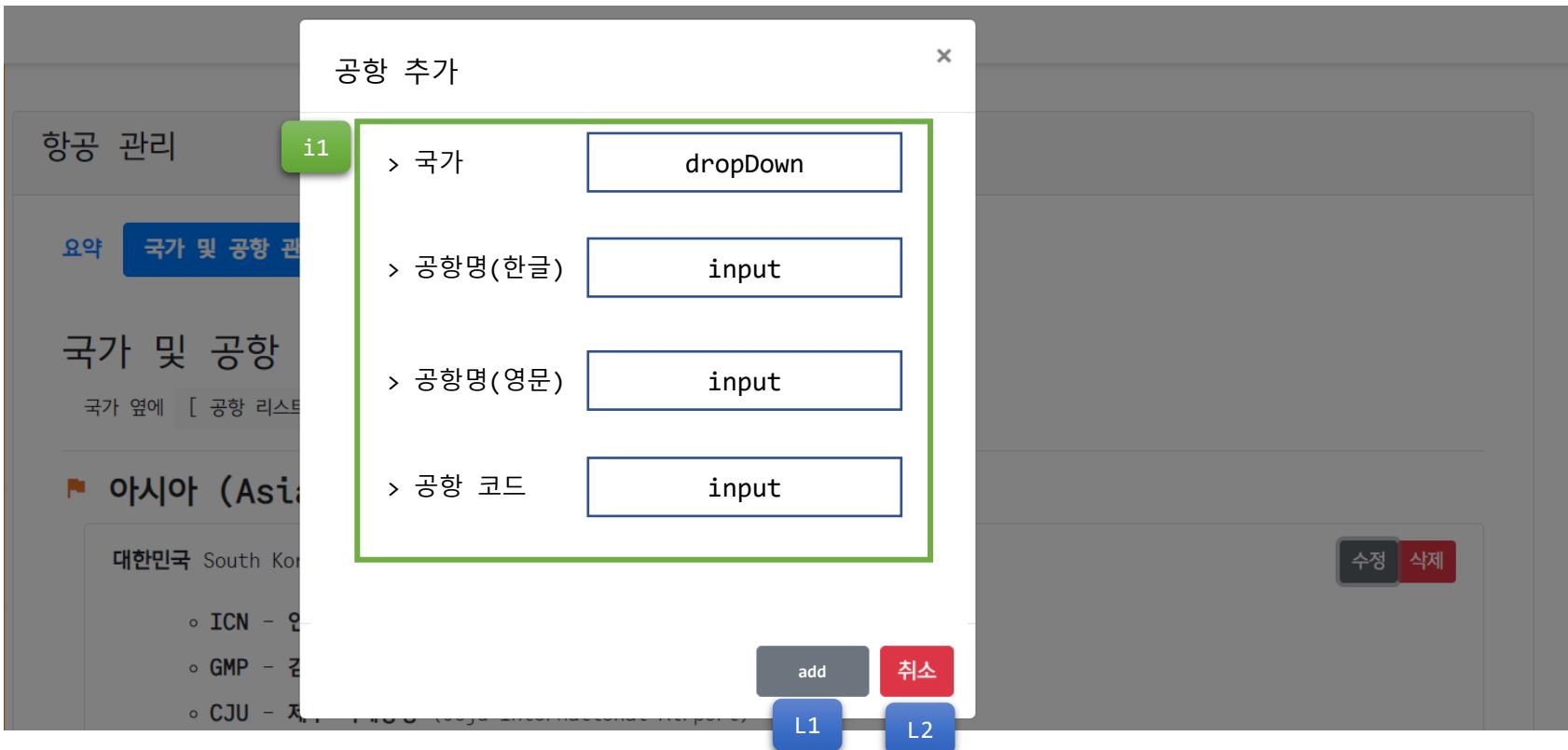
시연

핵심코드

오류수정

추후 개발 내용

후기



입력 데이터 <i>

I1 – 대륙, 국가 한글, 국가 영문, 코드

출력 데이터 <p>

P1 – 없음

링크 <L>

L1 – add : 국가 추가 ->

AdminNoc.do

L2 – 취소 : modal.hide();

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

항공 관리

요약 국가 및 공항 관리 항공사 기종 관리 노선 및 운임 관리 운항 스케줄 관리

L1

항공사 기종 관리

새로운 항공편을 등록하려면 [항공편 등록] 버튼을 누르세요.
새로운 항공편의 노선과 스케줄을 등록하려면 [노선 및 운임 관리] 또는 [운항 스케줄 관리]에서 등록할 수 있습니다.

* [수정] : 해당 항공편을 선택하면 자세한 내용을 확인할 수 있으며, 기종, 제조사, 좌석, 출발지, 도착지, 운항 스케줄을 변경할 수 있습니다.
* [삭제] : 해당 항공편과 관련된 모든 데이터가 삭제 됩니다.

순번	편명	기종	제조사	총 좌석수	일반석	프리미엄석	비즈니스석	일등석	운항여부	출발지	도착지
1	KE714	B737	BOEING	300	216	48	24	12	○	NRT	ICN
2	KE713	B737	BOEING	300	216	48	24	12	○	ICN	NRT
3	KE712	A320	AIRBUS	300	216	48	24	12	○	NRT	ICN
4	KE711	A320	AIRBUS	300	216	48	24	12	○	ICN	NRT
5	KE706	A320	AIRBUS	300	216	48	24	12	○	NRT	ICN
6	KE705	A320	AIRBUS	300	216	48	24	12	○	ICN	NRT
7	KE704	B787	BOEING	300	216	48	24	12	○	NRT	ICN
8	KE703	B787	BOEING	300	216	48	24	12	○	ICN	NRT
9	KE5744	B777	BOEING	300	216	48	24	12	○	NRT	ICN
10	KE5743	B777	BOEING	300	216	48	24	12	○	ICN	NRT

L2

p1

« < > »

입력 데이터 <i>

I1 – 없음.

출력 데이터 <p>

P1 – 항공편 데이터
편명, 기종, 제조사, 총 좌석수,
등급별 좌석 수, 운항 여부,
출발지, 도착지

링크 <L>

L1 – 항공편 등록 폼 오픈 버튼
L2 – 항공편 상세보기 collapse

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위
소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

항공 관리

요약 국가 및 공항 관리 항공사 기종 관리 노선 및 운임 관리 운항 스케줄 관리

항공사 기종 관리

[항공편 등록]

새로운 항공편을 등록하려면 [항공편 등록] 버튼을 누르세요.
 새로운 항공편의 노선과 스케줄을 등록하려면 [노선 및 운임 관리] 또는 [운항 스케줄 관리]에서 등록할 수 있습니다.

- * [수정] : 해당 항공편을 선택하면 자세한 내용을 확인할 수 있으며, 기종, 제조사, 좌석, 출발지, 도착지, 운항 스케줄을 변경할 수 있습니다.
- * [삭제] : 해당 항공편과 관련된 모든 데이터가 삭제 됩니다.

순번	편명	기종	제조사	총 좌석수	일반석	프리미엄석	비즈니스석	일등석	운항여부	출발지	도착지																																		
1	p1 KE714	B737	BOEING	300	216	48	24	12	○	NRT	ICN																																		
KOREAN AIR ✈ 편명 : KE714 L1																																													
* 항공사 기본 정보 <table border="1"> <tr> <th>제조사</th><th>모델</th><th>소속</th><th>소속 번호</th><th>국가(한글)</th><th>국가(영문)</th><th>국가코드</th></tr> <tr> <td>BOEING</td><td>B737</td><td>대한항공 (Korean Air)</td><td>1</td><td>대한민국</td><td>South Korea</td><td>KR</td></tr> </table> * 경로 정보 <table border="1"> <tr> <th>경로 번호</th><th>유형</th><th>출발지</th><th>출발 공항</th><th>도착지</th><th>도착 공항</th><th>출발 시간</th><th>도착 시간</th><th>소요 시간</th><th>거리</th></tr> <tr> <td>2</td><td>국제선</td><td>NRT</td><td>나리타 국제공항</td><td>ICN</td><td>인천 국제공항</td><td>20:55</td><td>23:25</td><td>2시간 30분</td><td>1257.98 km</td></tr> </table>												제조사	모델	소속	소속 번호	국가(한글)	국가(영문)	국가코드	BOEING	B737	대한항공 (Korean Air)	1	대한민국	South Korea	KR	경로 번호	유형	출발지	출발 공항	도착지	도착 공항	출발 시간	도착 시간	소요 시간	거리	2	국제선	NRT	나리타 국제공항	ICN	인천 국제공항	20:55	23:25	2시간 30분	1257.98 km
제조사	모델	소속	소속 번호	국가(한글)	국가(영문)	국가코드																																							
BOEING	B737	대한항공 (Korean Air)	1	대한민국	South Korea	KR																																							
경로 번호	유형	출발지	출발 공항	도착지	도착 공항	출발 시간	도착 시간	소요 시간	거리																																				
2	국제선	NRT	나리타 국제공항	ICN	인천 국제공항	20:55	23:25	2시간 30분	1257.98 km																																				
2	KE713	B737	BOEING	300	216	48	24	12	○	ICN	NRT																																		
3	KE712	A320	AIRBUS	300	216	48	24	12	○	NRT	ICN																																		

입력 데이터 <i>

I1 – 없음.

출력 데이터 <p>

P1 – 항공편 상세 데이터
 편명, 기종, 제조사, 소속 항공사, 국가 등급별 좌석 수, 경로 정보
 운항 스케줄

링크 <L>

L1 – 항공편 수정 모달 오픈 버튼
 L2 – 항공편 삭제 버튼

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

순번	편명	기종	제조사	총 좌석수	일반석	프리미엄석	비즈니스석	일등석	운항여부	출발지	도착지
1	KE714	B737	BOEING	300	216	48	24	12	<input checked="" type="radio"/>	NRT	ICN
2	KE713	B737	BOEING	300	216	48	24	12	<input checked="" type="radio"/>	ICN	NRT
3	KE712	A320	AIRBUS	300	216	48	24	12	<input checked="" type="radio"/>	NRT	ICN

입력 데이터 <i>

I1 – 제조사 선택, 모델 선택, 편명, 총 좌석 수, 등급별 좌석 수

출력 데이터 <p>

P1 – 제조사 리스트, 비행기 모델

링크 <L>

L1 – 항공편 등록 폼 추가 버튼
 L2 – 항공편 등록 폼 삭제 버튼
 L3 – 항공편 일괄 등록 버튼

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

항공 관리

항공사 기종 관리 노선 및 운임 관리 운항 스케줄 관리

노선 및 운임 관리

새로운 노선을 등록하려면 [노선 등록] 버튼을 누르세요.

아래 노선 탭에서 노선을 선택하면 운임에 대한 상세정보를 볼 수 있습니다.

입력 데이터 <i>

I1 – 운임 등록 및 수정

출력 데이터 <p>

P1 – 운행 타입, 출발지, 도착지, 거리
 P2 – 기본 운임, 세금, 할증료, 수수료
 일반석, 비즈니스, 일등석 금액

링크 <L>

L1 – 노선 수정
 L2 – 노선 삭제
 L3 – 운임 등록 및 수정

ID	Type	Departure Country	Departure Airport	Arrival Country	Arrival Airport	Distance
1	국제선	대한민국	ICN (인천 국제공항)	일본	NRT (나리타 국제공항)	1,257.98 km
2	국제선	일본	NRT (나리타 국제공항)	대한민국	ICN (인천 국제공항)	1,257.98 km
21	국제선	미국	JFK (존 F. 케네디 국제공항)	대한민국	ICN (인천 국제공항)	11,102.9 km
22	국제선	대한민국	ICN (인천 국제공항)	미국	JFK (존 F. 케네디 국제공항)	11,102.9 km
43	국제선	독일	FRA (프랑크푸르트 공항)	중국	HKG (홍콩 국제공항)	6,000 km
44	국제선	중국	HKG (홍콩 국제공항)	독일	FRA (프랑크푸르트 공항)	6,000 km

[번호 : 1 구간 : ICN ~ NRT]
 * 기본 운임을 입력하면 자동계산 됩니다.

기본 운임	130000 원
세금	28,000 원
유류 할증료	50,000 원
발권 수수료	1,000 원
총액 (일반석)	
209,000 원	
총액 (비즈니스)	
418,000 원	
총액 (일등석)	
836,000 원	



BANGKOK

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

노선 및 운임 관리

새로운 노선을 등록하려면 [노선 등록] 버튼을 누르세요.
아래 노선 탭에서 노선을 선택하면 운임에 대한 상세정보를 볼 수 있습니다.

[노선]

노선 등록

번호	출발지	출발공항	도착지	도착공항	거리	타입	왕복
	ICN	인천 국제공항	FRA	프랑크푸르트 공항	3333	국제선	<input checked="" type="checkbox"/>

i1

i2

i3

L1

등록

입력 데이터 <i>

i1 – 출발지
i2 – 도착지
i3 – 거리

출력 데이터 <p>

P1 – 없음

링크 <L>

L1 – 노선 등록

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

항공 관리

항공사 기종 관리 노선 및 운임 관리 운항 스케줄 관리

운항 스케줄 관리

해당 노선을 선택하면 상세화면으로 이동해 스케줄을 새로 [등록]하거나 [변경] 할 수 있습니다.

모두 검색



p1	번호	타입	출발 국가	출발지	출발 공항	도착 국가	도착지	도착 공항	거리
	1	국제선	대한민국	ICN	인천 국제공항	일본	NRT	L1	나리타 국제공항 1,257.98 km
	2	국제선	일본	NRT	나리타 국제공항	대한민국	ICN	인천 국제공항	1,257.98 km
	21	국제선	미국	JFK	존 F. 케네디 국제공항	대한민국	ICN	인천 국제공항	11,102.9 km
	22	국제선	대한민국	ICN	인천 국제공항	미국	JFK	존 F. 케네디 국제공항	11,102.9 km
	43	국제선	독일	FRA	프랑크푸르트 공항	중국	HKG	홍콩 국제공항	6,000 km
	44	국제선	중국	HKG	홍콩 국제공항	독일	FRA	프랑크푸르트 공항	6,000 km

◀ ◀ 1 ▶ ▶

입력 데이터 <i>

I1 - 없음

출력 데이터 <p>

P1 - 노선 리스트

링크 <L>

L1 - 노선 클릭시 상세보기 이동
</airScheduleDetail.do>

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

p1

항공 스케줄 상세보기 선택한 노선에 대한 스케줄 리스트입니다.

[노선 정보]								
노선번호	타입	출발 국가	출발지	출발 공항	도착 국가	도착지	도착 공항	거리
2	국제선	일본	NRT	나리타 국제공항	대한민국	ICN	인천 국제공항	1,257.98 km

p2

항공 스케줄 리스트

+add

L1

노선 번호	출발지	출발 시간	도착지	도착 시간	소요시간	운항 편명	상태	편집	
966	NRT	2024-10-31 20:55	ICN	2024-10-31 23:25	2시간 30분	KE714	운항		
923	NRT	2024-10-31 18:20	ICN	2024-10-31 20:55	2시간 35분	KE5744	운항		
880	NRT	2024-10-31 17:20	ICN	2024-10-31 20:00	2시간 40분	KE712	운항		
837	NRT	2024-10-31 14:00	ICN	2024-10-31 16:30	2시간 30분	KE704	운항		
794	NRT	2024-10-31 10:20	ICN	2024-10-31 13:00	2시간 40분	KE5742	운항		
751	NRT	2024-10-31 09:20	ICN	2024-10-31 11:45	2시간 25분	KE706	운항		
965	NRT	2024-10-30 20:55	ICN	2024-10-30 23:25	2시간 30분	KE714	운항		
922	NRT	2024-10-30 18:20	ICN	2024-10-30 20:55	2시간 35분	KE5744	운항		
879	NRT	2024-10-30 17:20	ICN	2024-10-30 20:00	2시간 40분	KE712	운항		
836	NRT	2024-10-30 14:00	ICN	2024-10-30 16:30	2시간 30분	KE704	운항		

<< < 1 2 3 4 5 6 7 8 9 10 > >>

입력 데이터 <i>

I1 - 없음

출력 데이터 <p>

P1 - 노선 정보
P2 - 스케줄 리스트

링크 < L >

L1 - 노선 등록 폼 이동
</airScheduleWirteForm.do>

화면구현 : 항공권 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

p1

구조화된 항공편 정보 표시

선택한 항공편에 대한 세부 정보

[노선 정보]								
노선번호	타입	출발 국가	출발지	출발 공항	도착 국가	도착지	도착 공항	거리
2	국제선	일본	NRT	나리타 국제공항	대한민국	ICN	인천 국제공항	1,257.98 km

운항 스케줄 등록

+ 스케줄 추가

i1

반복 등록 연도-월-일 부터 연도-월-일 까지 일괄 등록

노선	운항편명	출발일	출발시간	도착일	도착시간	소요시간	L1
NRT ~ ICN	KE310 (A330)	연도-월-일	-- --:--	연도-월-일	-- --:--		
NRT ~ ICN	KE310 (A330)	연도-월-일	-- --:--	연도-월-일	-- --:--		x
NRT ~ ICN	KE310 (A330)	연도-월-일	-- --:--	연도-월-일	-- --:--		x
NRT ~ ICN	KE310 (A330)	연도-월-일	-- --:--	연도-월-일	-- --:--		x
NRT ~ ICN	KE310 (A330)	연도-월-일	-- --:--	연도-월-일	-- --:--		x

입력 데이터 <i>

I1 – 운항 편 선택, 출발일, 시간, 도착일, 시간
(반복 시작일 – 종료일)

출력 데이터 <p>

P1 – 노선 정보

링크 <L>

L1 – 노선 등록
</airScheduleWirte.do>

03 Front-end : 화면 구현

항공권 예약 관리

화면구현 : 항공권 예약

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

AirOrder.writeForm

인천 (ICN) → 후쿠오카 (FUK)

왕복 | 성인 1인 | 좌석 등급: 일반석 | 대한항공

D1

여정 정보

D2

출발: 인천, 2024.09.10 (화) 18:35
 도착: 후쿠오카, 2024.09.10 (화) 20:00
 비행 시간: 01시간 25분
 항공편: 대한항공 (KE) 0781편

요금 정보

D3

항공 운임: 87,300원
 유류 할증료: 33,600원
 제세 공과금: 43,300원
 발권 대행 수수료: 10,000원
총 금액: 174,200원

예약자 정보 입력

I1

이름:

예: 윤해연

이메일:

example@domain.com

휴대폰:

010-1234-5678

탑승자 정보 입력

I2

탑승자 추가

수하물 추가

I3

수하물 추가

D [출력]

D1 : 출발지 - 도착지
 편도 or 왕복
 인원수
 좌석 등급
 항공사

D2 : 여정 정보
 출발지 출발시간
 도착지 도착시간
 비행시간
 항공편

D3 : 요금 정보

I [입력]

/airReservation/list.do

L1

예약하기

화면구현 : 항공권 예약

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

AirOrder.list.do

예약 목록

MRT님의 예약번호는 1111-1111입니다.

D1 : 예약 번호
출발지
도착지
항공사
출발일
도착일

D2 : 출발시간
도착시간
운항날짜
좌석 등급
좌석 번호

L1 : 환불

예약 수정

예약 번호: 1111-1111
출발: 인천(ICN) 도착: 로스앤젤레스(LAX)
항공사: 아시아나항공
출발일: 2021-01-14(목) 도착일: 2021-04-10(토)

예약 번호: 1111-1111
출발: 인천(ICN) 도착: 로스앤젤레스(LAX)
항공사: Korean Air (대한항공)
출발 시간: 2024년 9월 1일 15:00 (현지 시간)
도착 시간: 2024년 9월 1일 09:00 (현지 시간)
운항 날짜: 2024년 9월 1일
좌석 등급: 이코노미 클래스
좌석 번호: 23A

MRT님의 예약번호는 1111-1111입니다.
총 결제 금액: 0원
예약 번호: 1111-1111
출발: 인천(ICN) 도착: 로스앤젤레스(LAX)
항공사: 아시아나항공
출발일: 2021-01-14(목) 도착일: 2021-04-10(토)

D [출력]

D1 : 예약 번호
출발지
도착지
항공사
출발일
도착일

D2: 출발시간
도착시간
운항날짜
좌석 등급
좌석 번호

L [링크]

L1 : airReservation/delete.do

화면구현 : 항공권 예약

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

항공 예약 상세 airReservation/view.do

예약 정보

이메일:	test1@naver.com	D1
출발:	ICN	
도착:	NRT	
총 가격:	574750	
예약 날짜:	2024-09-25	
결제 방법:		
비행기 이름:	KE5741	
좌석 등급:	1	

탑승객 목록

탑승객 번호	이름	성	생일	국적	여권 번호	성별	탑승객 유형	여권번호 수정
1	호병	윤	2024-09-01	KR	M12343341	M	성인	L1 changePassport
1	내연	유	2024-09-02	US	L231315432	F	소아	changePassport

뒤로

L2

좌석 예약하러 가기

D [출력]

D1: 이메일
출발지
도착지
총 가격
예약날짜
결제방법
비행기 이름
좌석 등급

D2
탑승자 번호
이름
성
생일
국적
여권번호
성별
탑승객유형

L [링크]

L1
[:airReservation/update.do](#)

L2
[:seat/seatDlist.do](#)

화면구현 : 항공권 예약

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

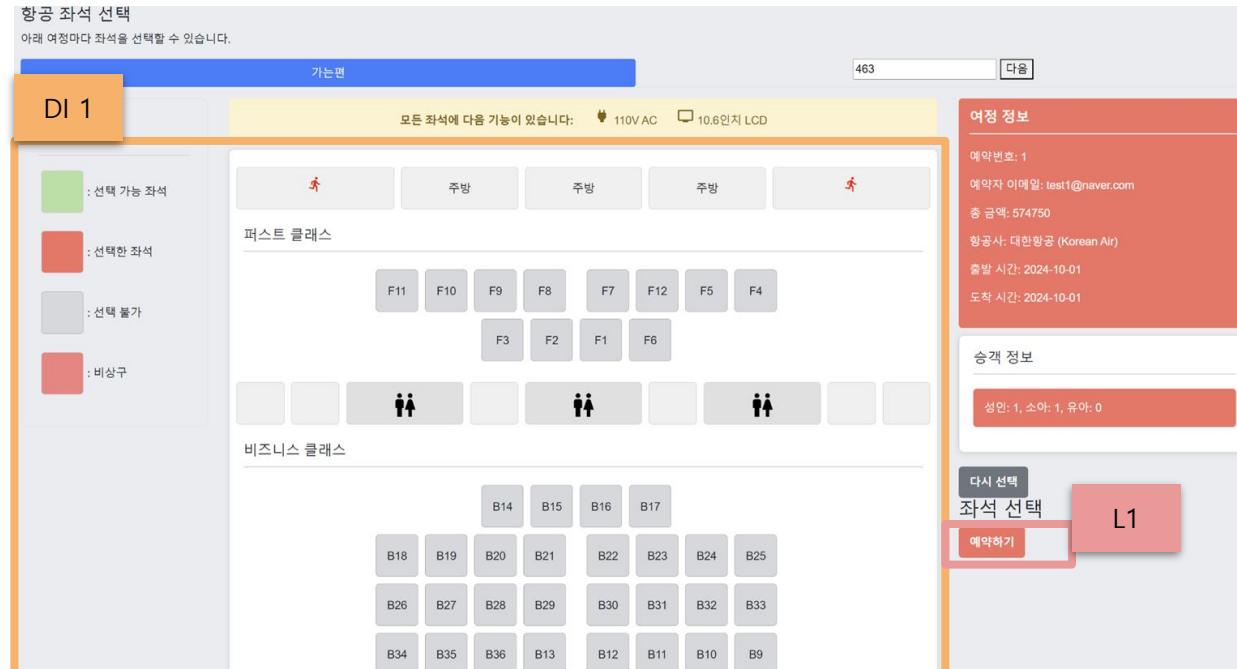
시연

핵심코드

오류수정

추후 개발 내용

후기



D & I [출력 & 입력]

DI : 좌석 리스트
좌석 선택 (입력)

L [링크]

L1 :airReservation/seatReservation.do



BANGKOK

화면구현 : 항공권 예약

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

AirOrderAdmin/list.do

예약 번호	이름	출발지	도착지	출발 날짜	좌석 등급	상태	환불	상태 변경
123456	홍길동	서울	뉴욕	2024-09-30	이코노미	예약됨	<button>환불</button>	<div><button>예약됨</button>▼ <button>변경 적용</button></div>
654321	김철수	서울	도쿄	2024-10-01	비즈니스	예약됨	<button>환불</button>	<div><button>예약됨</button>▼ <button>변경 적용</button></div>

D [출력]

D1 : 예약 번호
항공사
이름
출발지 - 도착지
출발 날짜
좌석 등급
상태
환불
상태변경

D2 : 예약 번호
항공사
이름
출발지 - 도착지
출발 날짜
좌석 등급
상태

I [입력]

I1: 상태 변경
예약 완료
예약 취소
환불 신청
환불 진행

L [링크]

L1 : airorder/refund.do

03 Front-end : 화면 구현

호텔 상품 관리

화면구현 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

의정부/포천/동두천

2024-09-08

2024-09-09

성인 2, 아동 0

호텔 등록

L1



L2

출력 D

- D1 – 호텔 정보
호텔 사진, 호텔명, 호텔 리뷰, 호텔등급, 객실 최소가, 체크인 시간



신라호텔

★ 4.5 (492)

5성급

₩100,000원~

숙박 16:00~



롯데호텔

★ 4.3 (81)

5성급

₩45,000원~

숙박 20:00~

이동 L

- L1 – writeForm.do
- L2 - searchModal



화면구현 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

D1


서울신라호텔

종대입구역 도보 300m

★ 4.3 (224)

[호텔 정보 수정](#) [호텔 삭제](#)

L1 L2

깨끗하고 서비스가 품질을 확보. 직원들이 친절하고 위치도 좋음.
2023.03.04

부가 아울렛을 편안한 숙박 경험을 할 수 있음.
2023.03.01

객실선택

체크인
성인 1, 어린이 0
성인 2, 어린이 0
객실 조회

D2

Room image
더블룸 ₩627,000 [객실 수정](#) [객실 삭제](#)

Room image
트윈룸 ₩627,000 [객실 수정](#) [객실 삭제](#)

출력 D

- D1 – 호텔 정보
호텔 사진, 호텔명, 호텔 리뷰, 공지사항, 소개, 전화번호, 주소, 편의시설
- D2 – 객실 정보
객실 유형, 인원수, 가격

이동 L

- L1 - updateForm.do
- L2 – delete.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

호텔 등록

호텔명			
호텔 등급	★★★★★		
호텔 소개			
전화번호			
공지사항			
주소			
편의시설	<input type="checkbox"/> 무료 와이파이 <input type="checkbox"/> 주차장 <input type="checkbox"/> 헬스장 <input type="checkbox"/> 수영장 <input type="checkbox"/> 레스토랑		
호텔 사진	<input type="button" value="파일 선택"/> 선택된 파일 없음		
객실 추가			
수정 L1			

출력 D

- D1 - 편의시설 리스트

입력 I

- 호텔 정보 입력
호텔명, 등급, 소개, 전화 번호, 공지, 주소

이동 L

- L1 – list.jsp



BANGKOK

화면구현 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

객실 등록

객실 유형

인원수

가격(1박)

객실 사진

파일 선택 선택된 파일 없음

객실 등록 L1

객실 추가

입력 |

- 객실 정보 입력
객실 유형, 인원수,
1박 가격, 객실 사진

이동 L

- L1 – write.jsp

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기



호텔명
신라호텔

호텔 등급
★★★★★

호텔 소개
신라호텔

전화번호
02-1234-1234

공지사항
신라호텔

주소
신라호텔

편의시설
 무료 와이파이
 수영장 주차장
 레스토랑 헬스장

호ôtel 사진
파일 선택 스크린샷 2024-08-26 162009.png

수정

출력 D

- D1 – 호텔 정보
호텔명, 호텔등급, 호텔 소개, 전화번호, 공지사항, 주소, 편의시설, 사진

입력 I

- I1 – 호텔 정보
호텔명, 호텔등급, 호텔 소개, 전화번호, 공지사항, 주소, 사진

이동 L

- L1 – view.jsp

03 Front-end : 화면 구현

호텔 예약 관리



화면구현 : 호텔 예약 관리

01 프로젝트 개요

주제와 목적

출력

02 개발 전반

요구 사항

개발 내용과 범위
소요 자원

D1: 예약 상태

D2: 객실 이미지, 객실 이름

D3: 체크인~체크아웃 날짜

03 Front-end

메뉴의 흐름

화면 구현

링크

L1 : 예약 내역 /hotelPay/view.do

L2 : 후기 등록 /reviews/writeForm.do

04 Back-end

DFD

DB

L3 : 국내 숙소 및 해외숙소 /hotelPay/list.do?type="~"

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

The screenshot shows a booking summary page with the following elements:

- Header:** 예약내역 (Booking Summary)
- Information:** 현재 숙소 외 예약내역은 여기어때 앱에서 확인 가능해요.
- Buttons:**
 - L3: 국내숙소 (Domestic Hotel) - highlighted in a box
 - 해외숙소 (Overseas Hotel)
- Section D1:** 예약확정 (Confirmed Booking)
 - 지금 바로 입실 가능 (Check-in available now)
 - 예약 상세 > (More details)
- Section D2:** ★오늘의 특가★ 호텔 크레센도 서울
 - [단독! 14시 체크아웃] 프티 퀸 · 1박
 - Image of a hotel room
- Section D3:** 체크인 9.13 (화) 15:00
- Section L1:** 숙소 상세 (Hotel Details)
- Section L2:** 주소 복사 (Address Copy)
- Section L3:** 이용완료 및 예약취소 (Completion and Cancellation)
- Text at the bottom:** 해당되는 예약 내역이 없습니다.



화면구현 : 호텔 예약 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

← 예약 확인 및 결제

예약자 정보

영문 이름 영문 성

Gildong Hong

띄어쓰기 없이 입력해주세요.

이메일

email@example.com

입력한 메일 주소로 예약 확인서를 보내드려요.
메일 주소가 정확한지 확인해주세요.

핸드폰 번호

+82

010-1234-5678

DA1

DA2

DA3

결제 수단



이 결제 수단을 다음에도 사용

모든 혜택 보기▼

취소 정책

* 2024-09-29 23:59 전 예약 취소 시 무료 취소 가능합니다.

2024-09-29 23:59 ~ 2024-09-30 23:59 기간에 예약 취소 시 35,970원이 취소 수수료로 부과됩니다.

2024-09-30 23:59 이후 예약 취소 시 179,850원의 취소 수수료가 부과됩니다.

(숙소 현지시간 기준)

예약한 호텔이나 숙소에 체크인하지 않을 경우 노쇼(No-Show)로 간주되며, 예약 요금의 100%가 취소 요금으로 부과됩니다.(호텔 정책)

● 24년 9월 29일 23:59 까지 무료취소
(숙소 현지 시간 기준)

[최저가보장] 렘 롯폰기
remm Roppongi



객실 Semi Double Room

일정 10.01 (화) 14:00 ~
10.02 (수) 12:00 (1박)
정확한 체크인/체크아웃 시간은 숙소에 문의 해주세요.

객실 수 객실 1개

인원 성인 2명
예약 인원과 실제 투숙 인원이 다를 경우, 세그 인이 불가하거나 추가 요금이 발생할 수 있습니다.

옵션 객실만

결제 정보

총 객실 가격(1개 x 1박) 179,850원

총 객실 가격 상세 보기 ▾

총 결제 금액 179,850원

약관 전체동의 ▾

L1 179,850원 결제하기

출력

D1:호텔 이름

D2:호텔 사진

D3:객실 이름

D4:체크인~체크아웃 날짜

D5:객실 수

D6:객실 옵션

출력 + 입력

DA1:이름 세팅 및 입력

DA2:이메일 세팅 및 입력

DA3:핸드폰 세팅 및 입력

링크

L1 : 예약 내역 /hotelPay/list.do

01 프로젝트 개요

주제와 목적

출력

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

D1:예약 번호

D15:객실 사진

D2:예약자 이름

D3:전화 번호

D4:결제 날짜

D5:상품 가격

D6:결제 수단

D7:결제 방식

D8:호텔명

D9:호텔 전화번호

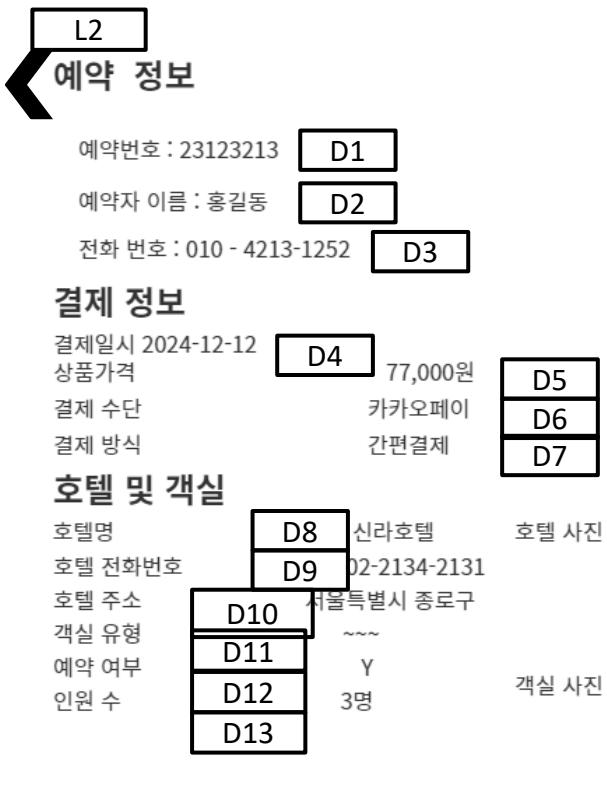
D10:호텔 주소

D11:객실 유형

D12:예약 여부

D13:인원 수

D14:호텔 사진



예약 취소

L1

링크

L1 : 예약 취소 /hotelPay/delete.do

L2 : 예약 내역 /hotelPay/list.do



화면구현 : 호텔 예약 관리

01 프로젝트 개요

주제와 목적

출력

02 개발 전반

요구 사항

D1: 예약 번호

개발 내용과 범위

D2: 이름

소요 자원

D3: 예약자 전화

03 Front-end

메뉴의 흐름

D4: 가격

화면 구현

D5: 예약 상태

04 Back-end

DFD

D6: 결제 수단

DB

D7: 결제 방식

05 프로젝트 시연

시연

D8: 가격

핵심코드

D9: 호텔 이름

오류수정

D10: 객실 이름

추후 개발 내용

D11: 인원 수

후기

링크

L1 : 예약 상태 /hotelPay/update.do

사업자 리스트										
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
예약 번호	이름	예약자 전화	가격	예약 상태	결제수단	결제 방식	주문금액 선불/충비포함	호텔 이름	객실 이름	인원 수
1	우소정	010-2232-1312	우소정	준비 변경	계좌이체	54000	확인	미사용	0	
2	김이젠	010-1232-1312	김이젠	배송중 변경	신용카드	72000	확인	미사용	0	

< 1 2 3 4 >

L2

L2 : 페이지 이동 /hotelPay/entr?page=1~10

L3 : 검색 이동 /hotelPay/entr.do

03 Front-end : 화면 구현

호텔 후기



화면구현 : 호텔 후기

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기



바른후기 남기기

출력

방문정보	야놀자 예약 / 숙박
숙소명	건대 호텔스타
객실정보	디럭스

D1

D2

이곳에서의 경험은 어떠셨나요?



탭해서 평가하기

생생한 평가를 해주세요.

서비스&친절도



A1

숙소&객실 청결도



시설&편의성



비품&용품 만족도



A2

이 곳에서 머물렀던 기억을 자세히 말해줄 수 있나요? (5자 이상 작성해주세요.)

A3



A4

등록하기

예상 적립 포인트 : 1,000P

D1: 숙소명

D2: 객실 유형

입력

A1: 클릭해서 별점 입력

A2: 제목 입력

A3: 내용 입력

A4: 이미지 등록

링크

L1 : 예약 내역 /reviews/write.do

L1

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

출력

D1:후기 제목

D2:후기 내용

D3:이미지

D4:별점

링크

L1 : 후기 수정 /reviews/updateForm.do

L1 : 후기 삭제 /reviews/delete.do

나의 후기

제목

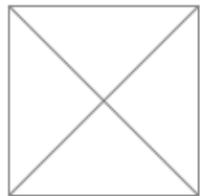
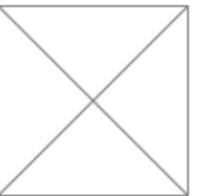
호텔 너무 이뻐요

D1

내용

제 마음에 쏙 들어요

D2



D3



D4

L1

L2

수정하기

삭제하기



01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

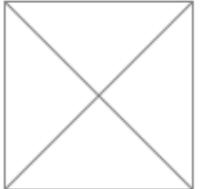
나의 후기

제목

호텔 너무 이뻐요

내용

제 마음에 쏙 들어요



수정하기

삭제하기

정말 삭제하시겠습니까?

L1

Cancel

OK

링크

L1 : 후기 삭제 /reviews/delete.do



화면구현 : 호텔 후기

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

출력 + 입력

DA1: 제목 세팅 및 입력

DA2: 내용 세팅 및 입력

DA3: 별점 세팅 및 입력

나의 후기 변경

제목

정말 이뻐요

DA1

내용

내용을 입력해주세요

DA2

입력

A1 : 이미지 등록

이미지 등록하기

A1

링크



DA4

L1 : 수정하기 :/reviews/update.do

L2 : 돌아가기 :onclick = history.back();

수정하기

돌아가기

L1

L2

화면구현 : 호텔 후기

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

L1 < D1 후기 (809) i D9 친절도 4.7 청결도 4.7 편의성 4.5 위치 4.7

숙소 평점 4.6 D2 숙소 후기사진 전체보기 (126) >


객실 전체 (809) 사진후기만 보기 추천순 ▾

D4	D8	D3 2024.03.07 ...
★★★★★ 심세* 슈페리어 더블		D7
직원분들 너무 친절하시고 설명도 잘해주셔서 감사했습니다 객실상태도 깨끗이 청소되어있었고 쾌적하게 잘 쉬다갑니다 다음 일정에 강남에 호텔잡을 일 생기면 재방문할께요		

D6 2024.05.16 ...

봉천동민사*
슈페리어 더블
아늘자 라이브특가로 구매하여 잘 다녀왔습니다!
시설도 너무 깔끔하고
조식이며, 룸컨디션도 너무 좋았어요 🔥
수영장, 헬스장 모두 깔끔하였는데...

더보기 ▾



출력

D1:총 별점
D2:리뷰 사진
D3:후기 등록 날짜
D4:별점
D5:회원 아이디 및 숙소유형
D6:내용
D7:좋아요
D8:제목
D9:친절도, 청결도, 편의성, 용품&비품

링크

L1 :되돌아가기 :onclick history.back();

03 Front-end : 화면 구현

여행 정보 게시판

01 프로젝트 개요

주제와 목적

1 / 5

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

trip info list

• 입력

•

•

•

•

• 출력

• search, rownum

• write button

• title, content, writeDate, writer

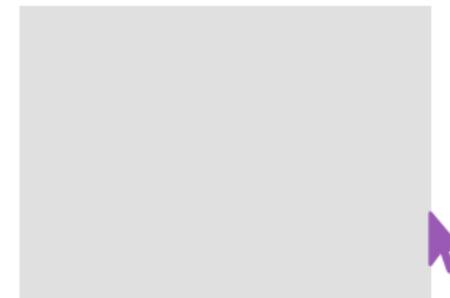
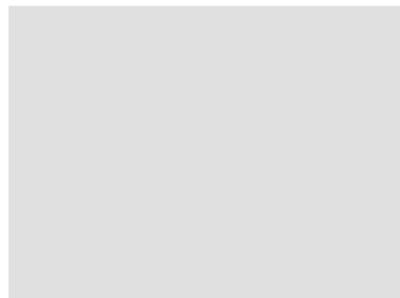
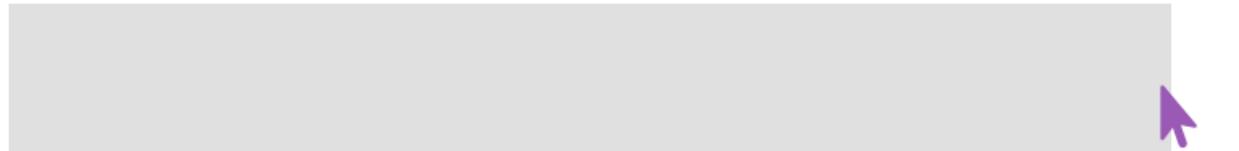
• header image

• image (FileName)

•

• 링크

• /tripinfo/writeForm.do

Hoxy 디즈니 좋아하세요? 2024-08-29
test(title)
(writer)

(writeDate)

등록

01 프로젝트 개요

주제와 목적

trip info view

2 / 5

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기



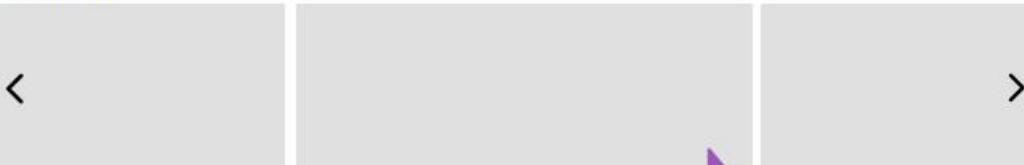
오락

명소

1. Hoxy 디즈니 좋아하세요?

test

2024-08-29



입력

•

•

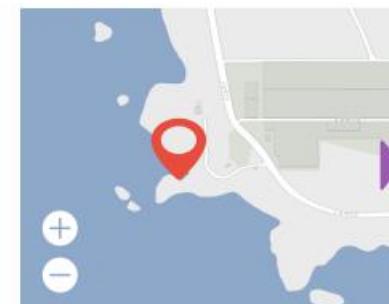
출력

- cate_code1, cate_name
- attracPlace
- image (FileName)
- title, content, writeDate, writer
- map (attracPlace)
- reply (tripPlan - reply)
-

•

링크

- /tripinfo/view.do
- /tripinfo/updateForm.do
- /tripinfo/delete.do
- /tripinfo/list.do



(content)

..

(reply)

수정

삭제

리스트



BANGKOK

화면구현 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

trip info write

3 / 5

여행정보 등록

카테고리 선택
제목
사진 업로드 선택된 파일 없음
장소 등록
내용
 등록 재입력 취소

< 1 2 3 4 >

- **입력**
- title, content
-
- **출력**
- cate_code1, cate_name
- image (FileName)
- title, content
- map (attracPlace)
- button
-
- **링크**
- /tripinfo/writeForm.do
- /tripinfo/write.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

trip info update

4 / 5

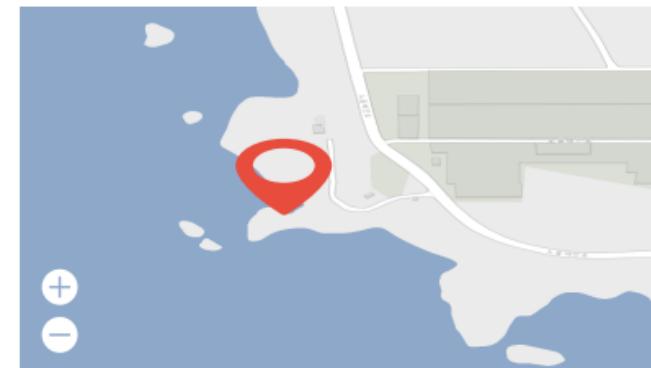
여행정보 수정

카테고리 선택

제목

사진 변경

장소 등록



내용

비밀번호 확인

수정

재입력

취소

- **입력**
- title, content
- checked pw
-
- **출력**
- cate_code1, cate_name
- image (FileName)
- title, content
- map (attractPlace)
- button
-
-
- **이미지 변경**
- gradeNo == 9
-
- **링크**
- /tripinfo/updateForm.do
- /tripinfo/update.do
- /tripinfo/view.do

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

trip info delete

5 / 5

trip info delete

오락 명소

1. Hoxy 디즈니 좋아하세요?
test
2024-08-29

< >

정말 삭제하시겠습니까?

CANCEL OK

지도 이미지 (content)

수정 삭제 리스트

1 2 3 4 >

- 입력
-
-
- 출력
- cate_code1, cate_name
- delete modal
- map (attracPlace)
- button
-
- 링크
- /tripinfo/delete.do
- /tripinfo/list.do

03 Front-end : 화면 구현

여행 정보 게시판 댓글



BANGKOK

화면구현 : 여행 정보 게시판 댓글

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

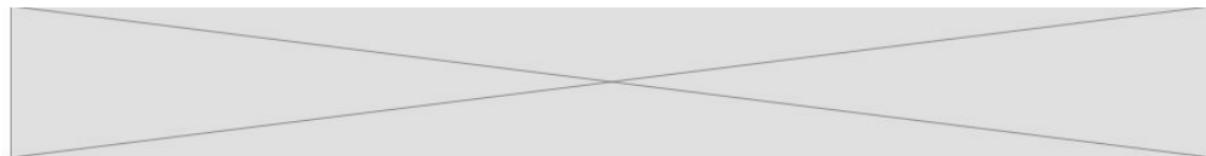
핵심코드

오류수정

추후 개발 내용

후기

- 댓글 list (/tripInfo/view.do)



출력

- 사용자 사진(member.photo)
- 사용자 닉네임(member.nickName)
- 사용자 아이디(reply.userId)
- 댓글 내용(reply.r_content)

링크

- /tripInfo/view.do(reply.updateForm)
- /tripInfo/view.do(reply.delete)

입력

화면구현 : 여행 정보 게시판 댓글

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 댓글 write (/tripInfo/view.do)



출력

1. 사용자 사진(member.photo)

링크

1. /tripInfo/view.do(reply.list)
2. /tripInfo/view.do(reply.write)

입력

1. 댓글 내용 (reply.r_content)

화면구현 : 여행 정보 게시판 댓글

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 댓글 update (/tripInfo/view.do)

The screenshot shows a reply update form on a tripInfo/view.do page. The form has the following fields:

- 댓글 개수 @개**: A dropdown menu set to "최신순".
- 사진**: A placeholder for a photo.
- 댓글 추가**: A text input field.
- 사진**: A placeholder for a photo.
- 2 user nickname <@userID>**: The nickname of the user who made the comment.
- 3 댓글 내용**: A text input field for the comment content.
- 4**: A small number indicating a step or action.
- ... (three dots)**: A more options icon.
- cancle**: A cancel button.
- submit**: A submit button.
- user nickname <@userID>**: The nickname of the user who made the comment.
- 댓글 내용**: A text input field for the comment content.
- ...**: A more options icon.

출력

- 사용자 사진(member.photo)
- 사용자 닉네임(member.nickName)
- 사용자 아이디(reply.userId)
- 댓글 내용(reply.r_content)

링크

- /tripInfo/view.do(reply.list)
- /tripInfo/view.do(reply.update)

입력

- 댓글 내용 (reply.r_content)

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

출력

- 댓글 delete (/tripInfo/view.do)



1. /tripInfo/view.do(reply.delete)
2. /tripInfo/view.do(reply.list)

링크

03 Front-end : 화면 구현

여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

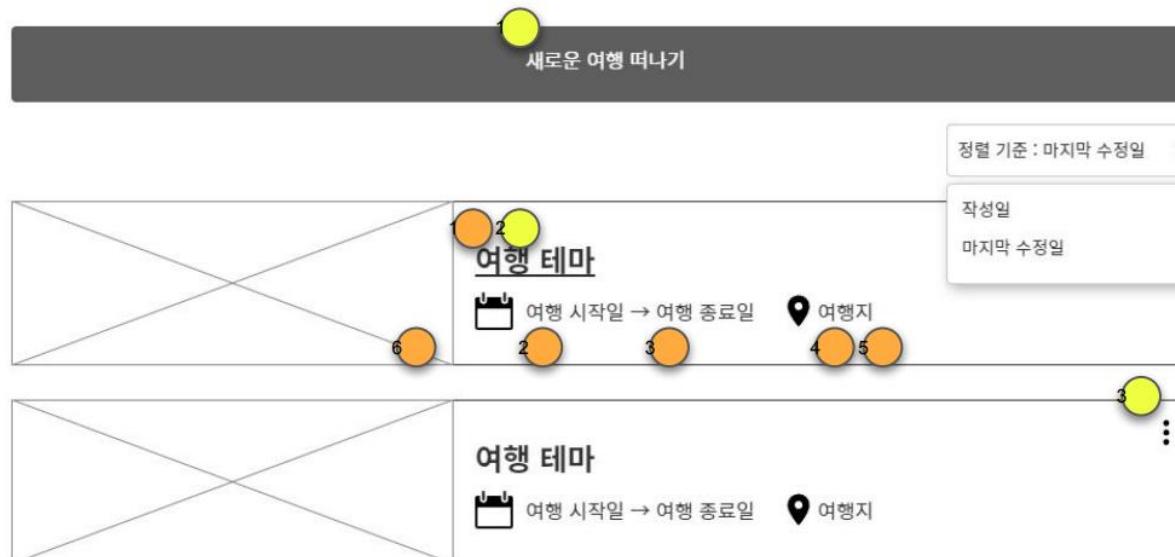
오류수정

추후 개발 내용

후기

- 여행 계획 list (/plan/list.do)

내 여행



출력

- 여행 테마(tripPlan.title)
- 여행 시작일(tripPlan.tripStartDate)
- 여행 종료일(tripPlan.tripEndDate)
- 여행 국가 (tripPlan.tripCountry)
- 여행지 (tripPlan.tripPlace)
- 여행지 사진 (tripInfo.image)

링크

- /plan/writeForm.do
- /plan/view.do
- /plan/delete.do

입력

화면구현 : 여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

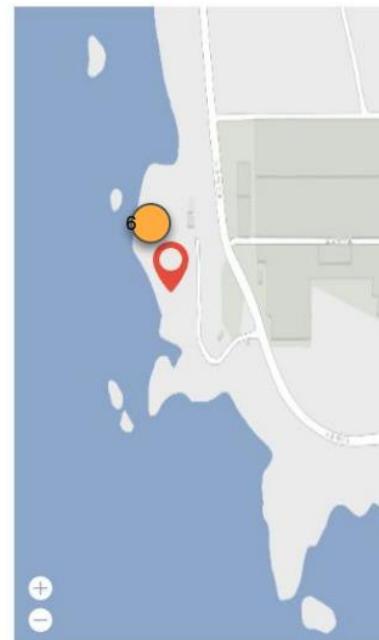
핵심코드

오류수정

추후 개발 내용

후기

- 여행 계획 view (/plan/view.do)



출력

- 여행 테마(tripPlan.title)
- 여행 시작일(tripPlan.tripStartDate)
- 여행 종료일(tripPlan.tripEndDate)
- 여행지 (tripPlan.tripPlace)
- 여행 당일 날짜(tripDate)
- 방문 장소(planDetail.attracPlace)

링크

- /plan/updateForm.do
- /plan/delete.do

입력

화면구현 : 여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 여행 계획 write (/plan/writeForm.do)

출력

링크

1. /plan/list.do

입력

- 여행 시작일(tripPlan.tripStartDate)
- 여행 종료일(tripPlan.tripEndDate)
- 여행 테마(tripPlan.tripThema)
- 여행 국가(tripPlan.tripPlace)
- 여행 일정(tripPlan.tripCountry)

화면구현 : 여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 여행 계획 update (/plan/updateForm.do)



출력

- 여행 시작일(tripPlan.startDate)
- 여행 종료일(tripPlan.endDate)
- 여행 테마(tripPlan.tripThema)
- 여행 국가(tripPlan.tripCountry)
- 여행지(tripPlan.tripPlace)

링크

- /plan/view.do

입력

- 여행 시작일(tripPlan.startDate)
- 여행 종료일(tripPlan.endDate)
- 여행 테마(tripPlan.tripThema)
- 여행 국가(tripPlan.tripCountry)
- 여행지(tripPlan.tripPlace)
- 세부사항(tripPlan.comment)

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 여행 계획 delete (/plan/delete.do)



출력

링크

1. /plan/list.do

입력

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위
소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

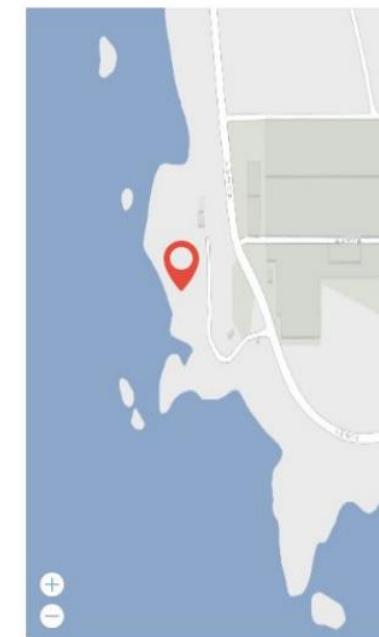
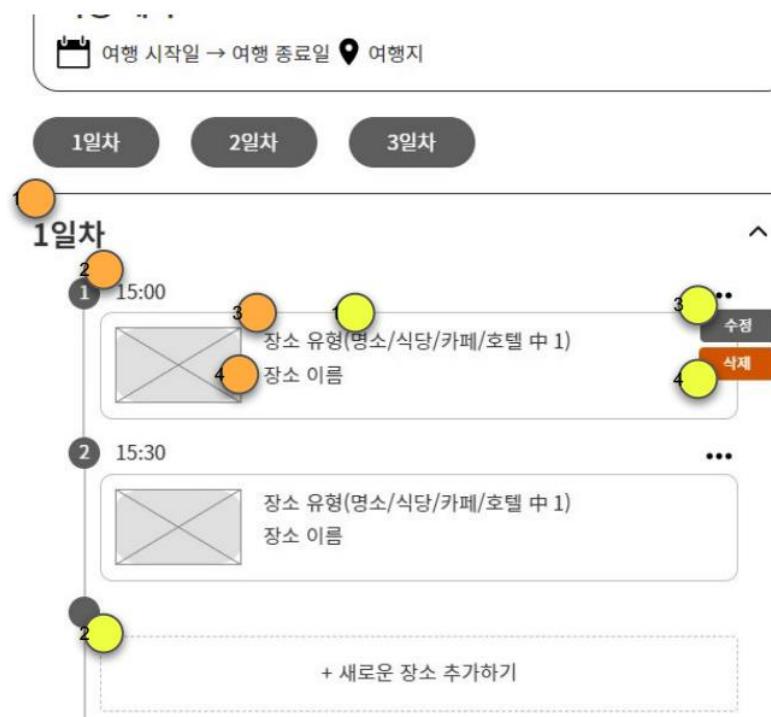
핵심코드

오류수정

추후 개발 내용

후기

- 방문지 관리 list (/plan/view.do)



출력

- 방문 날짜(planDetail.tripDate)
- 시간 (planDetail.tripTime)
- 방문지 카테고리(category.category_name)
- 방문 장소 (tripInfo.attracPlace)

링크

- /plan/view.do(planDetail.view)
- /plan/view.do(planDetail.writeForm)
- /plan/view.do(planDetail.updateForm)
- /plan/view.do(planDetail.delete)

입력

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

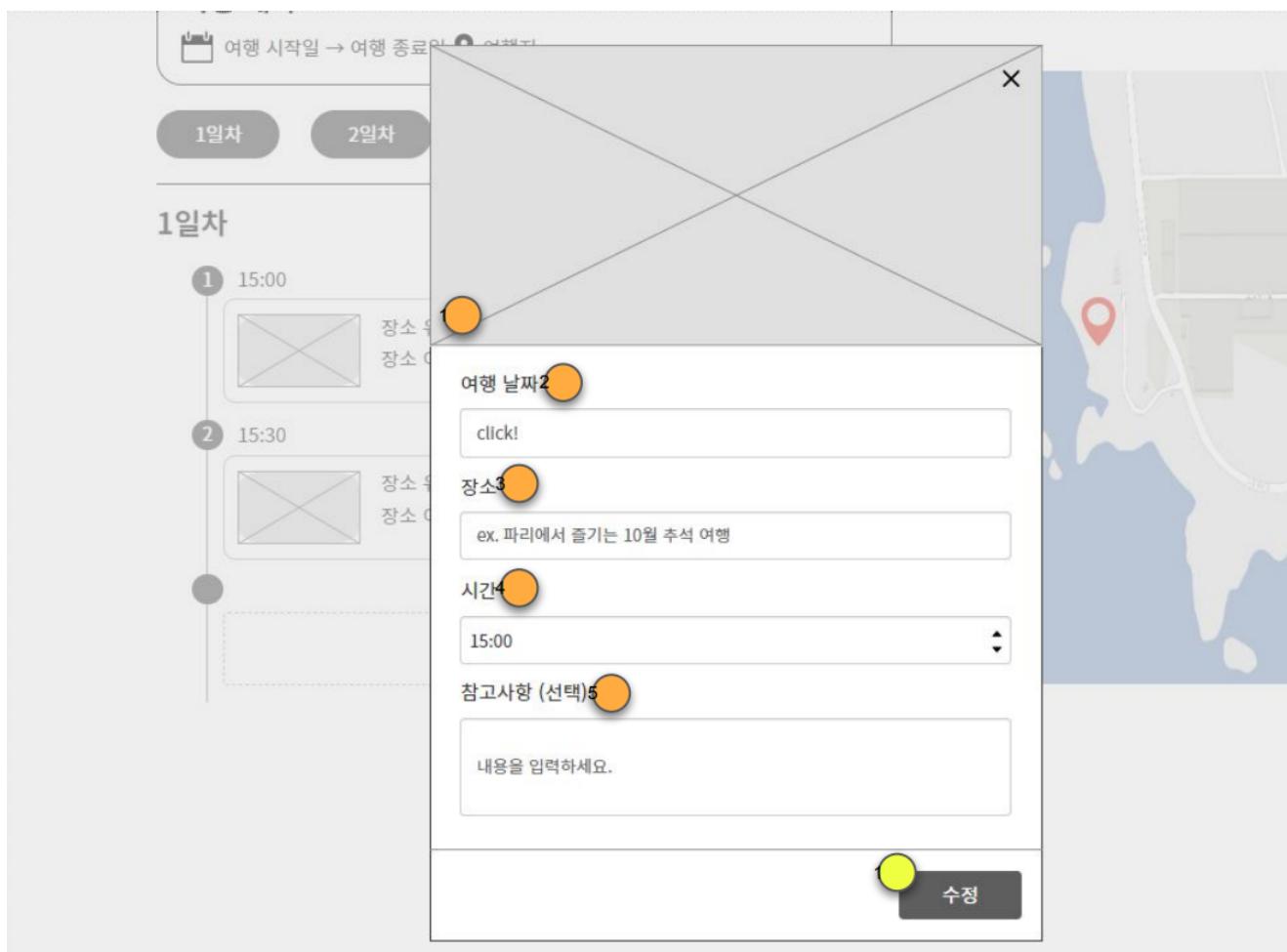
핵심코드

오류수정

추후 개발 내용

후기

● 방문지 관리 view (/plan/view.do)



출력

1. 여행지 사진(tripInfo.image)
2. 방문 날짜(planDetail.tripDate)
3. 방문 장소(planDetail.attracPlace)
4. 시간(planDetail.tripTime)
5. 참고사항(planDetail.tripContent)

링크

1. /plan/view.do(planDetail.updateForm)

입력

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 방문지 관리 write (/plan/view.do)

The screenshot shows a modal window titled "장소 추가하기" (Add Location) on top of a map of Bangkok. The modal has the following fields:

- 여행 날짜 1: A date input field labeled "click!".
- 장소 2: A text input field with placeholder text "ex. 파리에서 즐기는 10월 추석 여행".
- 시간 3: A dropdown menu showing "15:00".
- 참고사항 (선택) 4: A text input field with placeholder text "내용을 입력하세요."

At the bottom of the modal are two buttons: "추가" (Add) and "취소" (Cancel).

- 출력**
- 방문 날짜 (planDetail.tripDate)
 - 방문 장소 (planDetail.attracPlace)
 - 시간 (planDetail.tripTime)
 - 참고사항 (planDetail.tripContent)

링크

- /plan/view.do(planDetail.write)
- /plan/view.do(planDetail.list)

입력

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

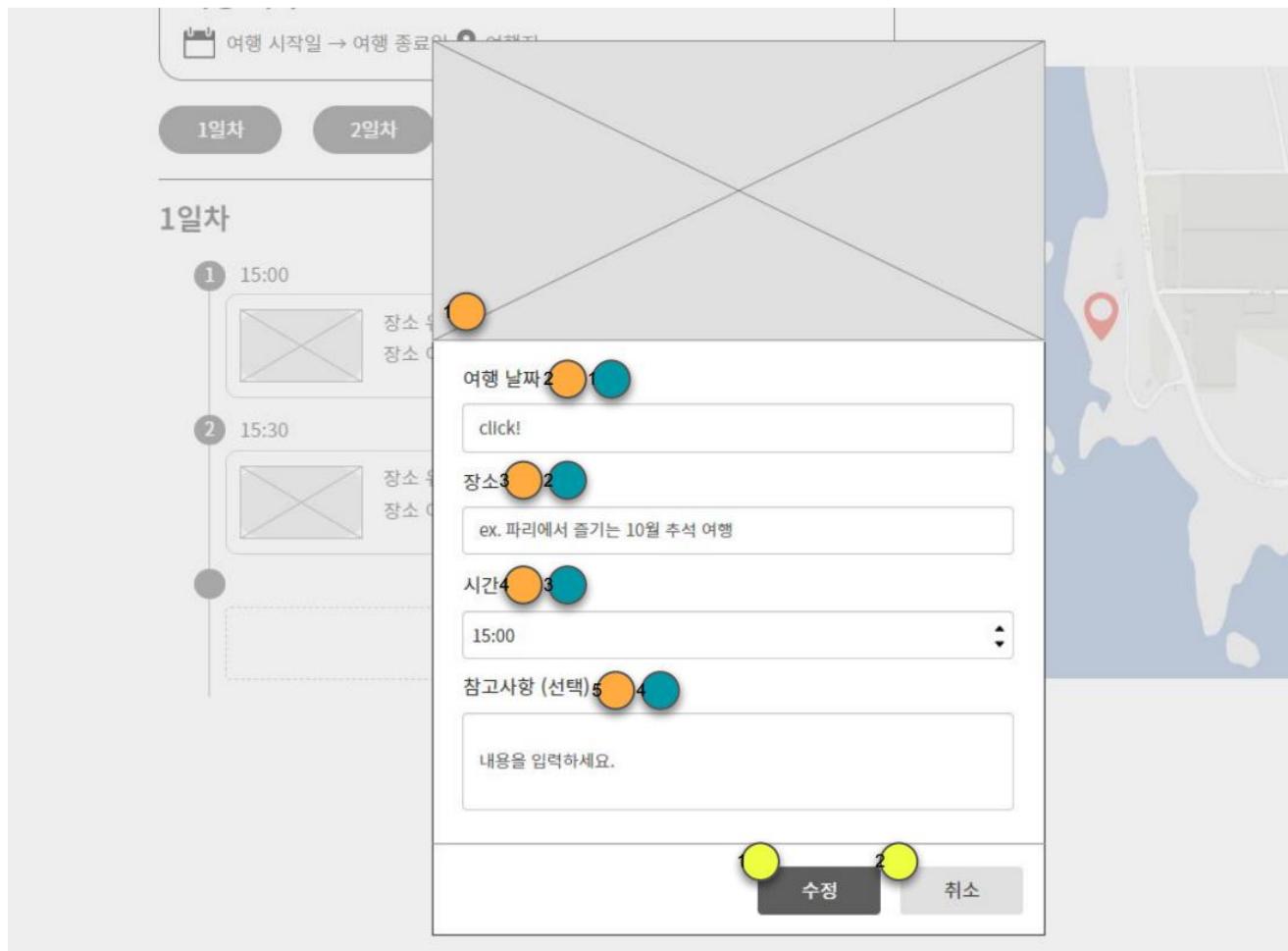
핵심코드

오류수정

추후 개발 내용

후기

- 방문지 관리 update (/plan/planUpdateForm.do)



출력

- 여행지 사진(tripInfo.image)
- 방문 장소(planDetail.attracPlace)
- 시간(planDetail.tripTime)
- 참고사항(planDetail.tripContent)

링크

- /plan/view.do(planDetail.update)
- /plan/view.do(planDetail.list)

입력

- 방문 날짜(planDetail.tripDate)
- 방문 장소(planDetail.attracPlace)
- 시간(planDetail.tripTime)
- 참고사항(planDetail.tripContent)

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

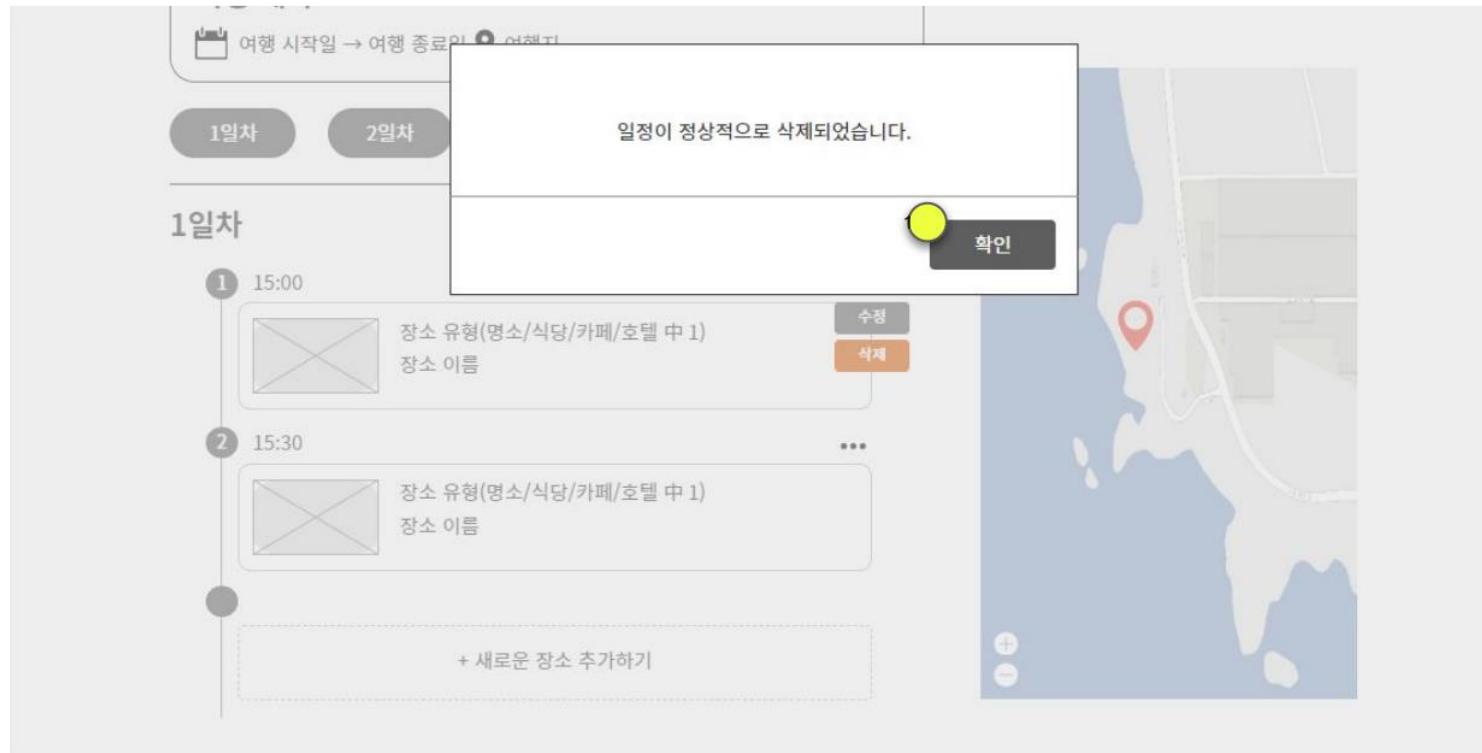
핵심코드

오류수정

추후 개발 내용

후기

- 방문지 관리 delete (/plan/planUpdateForm.do)



출력

링크

1. /plan/view.do(planDetail.list)

입력

04 Back-end

Data Flow Diagram

DB

Data Flow Diagram : 회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

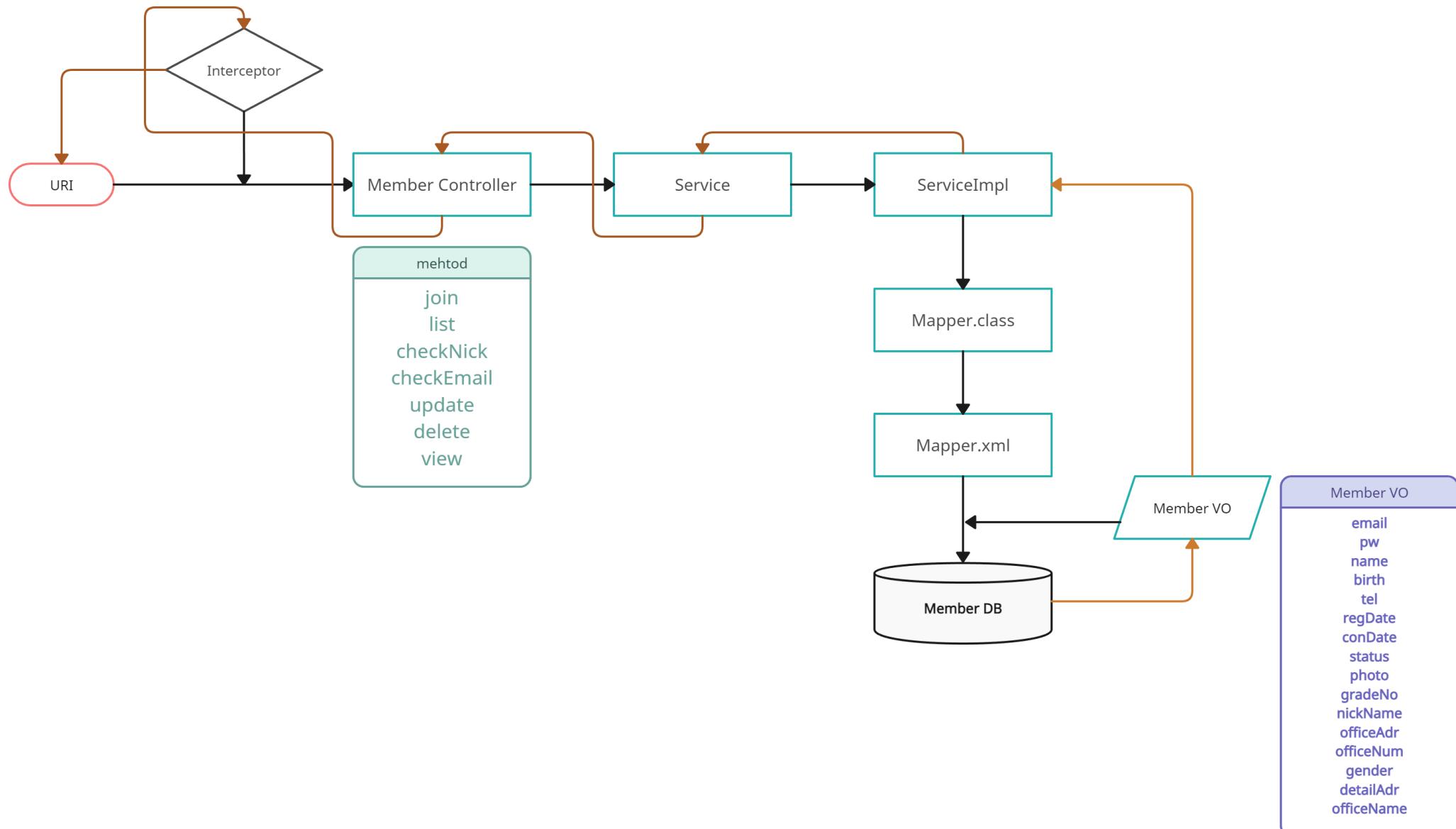
시연

핵심코드

오류수정

추후 개발 내용

후기



Data Flow Diagram : 이벤트

01 프로젝트 개요
주제와 목적

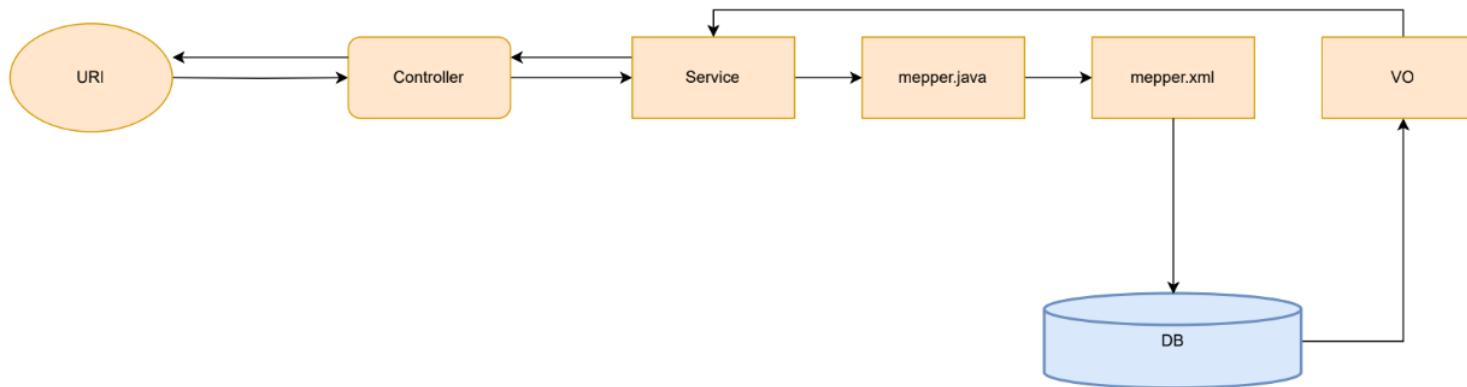
02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

이벤트



VO And Method

이벤트

eventVO

eventNo
image
title
category
sub_image
startDay
endDay
writeDate

실행되는 메소드

GET

event.view()
event.list()
event.updateForm()
event.writeForm()

POST

event.delete()
event.write()
event.update()
event.categoryWrite()
event.categoryUpdate()
event.categoryDelete()



BANGKOK

Data Flow Diagram : 항공권 관리

01 프로젝트 개요
주제와 목적

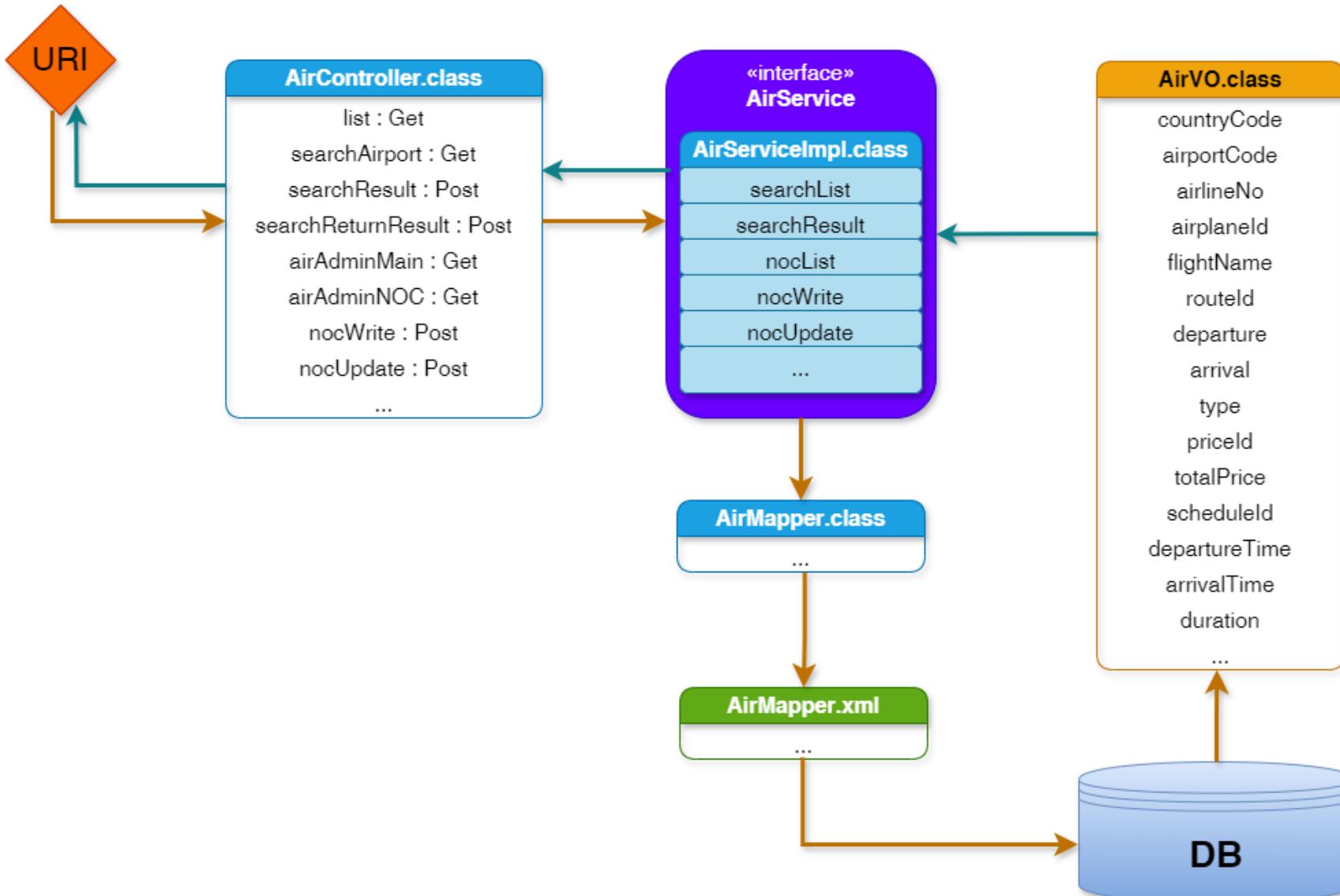
02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

< Air - Data Flow Diagram >



Data Flow Diagram : 항공권 예약

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

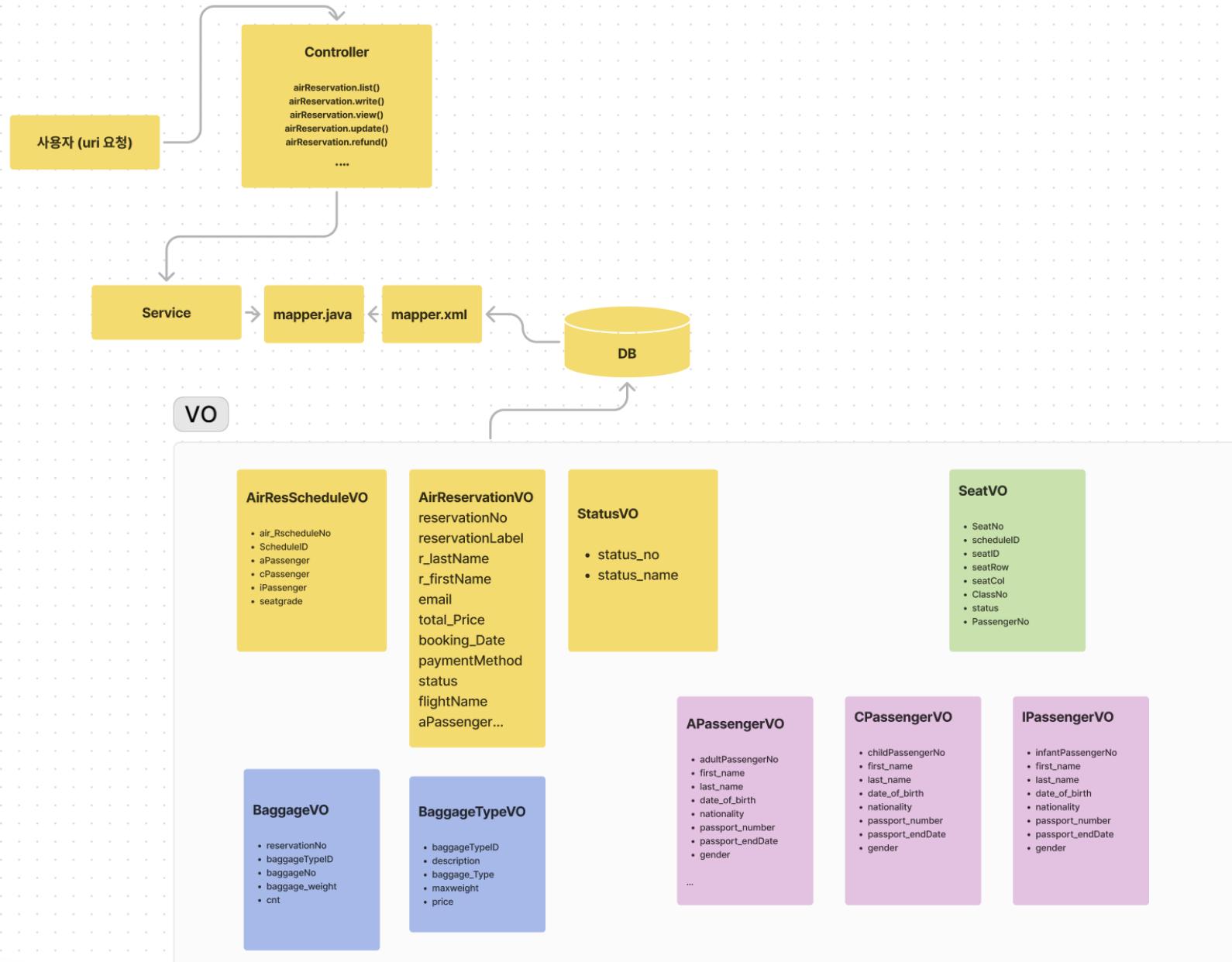
시연

핵심코드

오류수정

추후 개발 내용

후기





Data Flow Diagram : 호텔 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

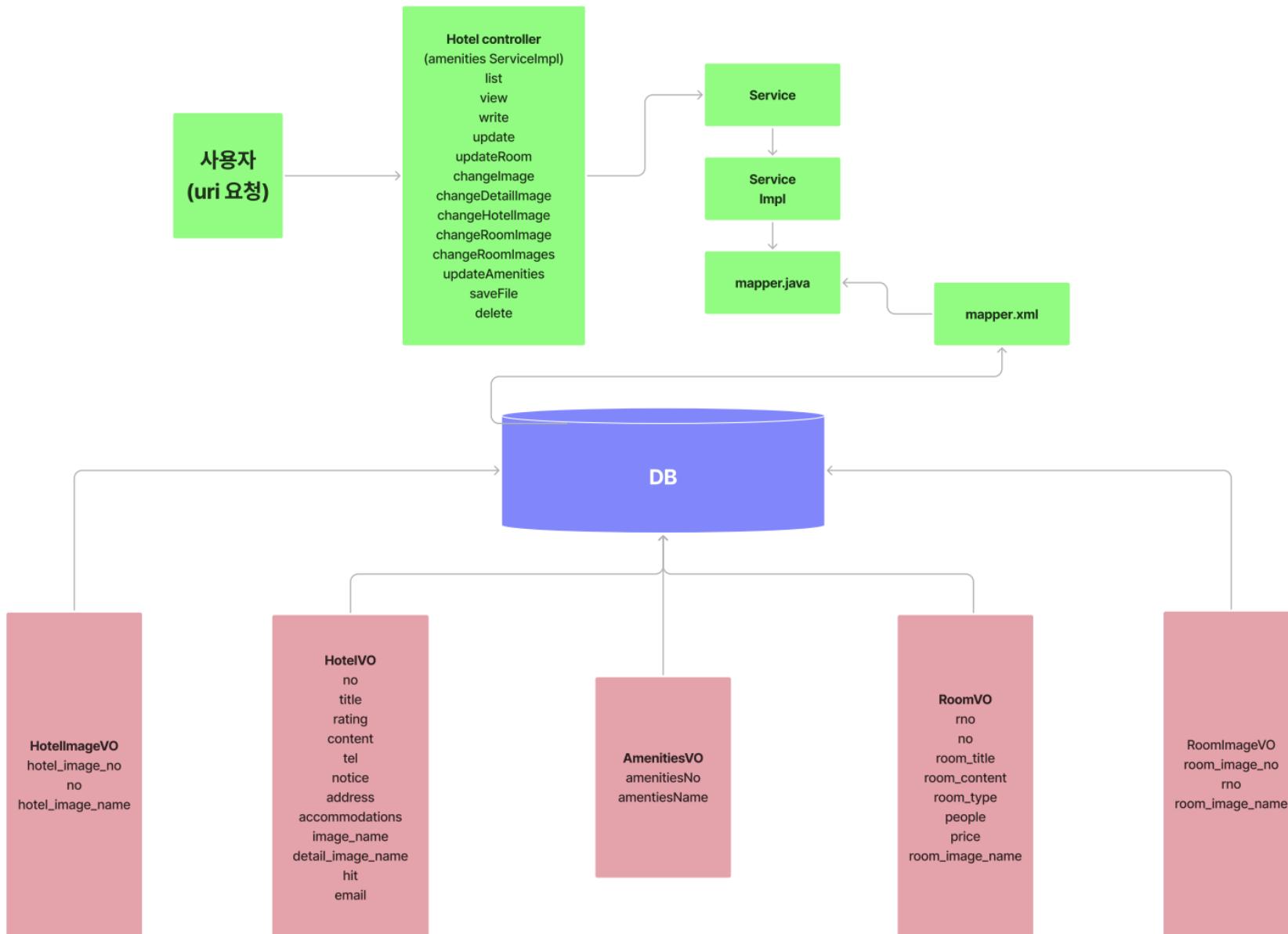
시연

핵심코드

오류수정

추후 개발 내용

후기



Data Flow Diagram : 호텔 예약

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

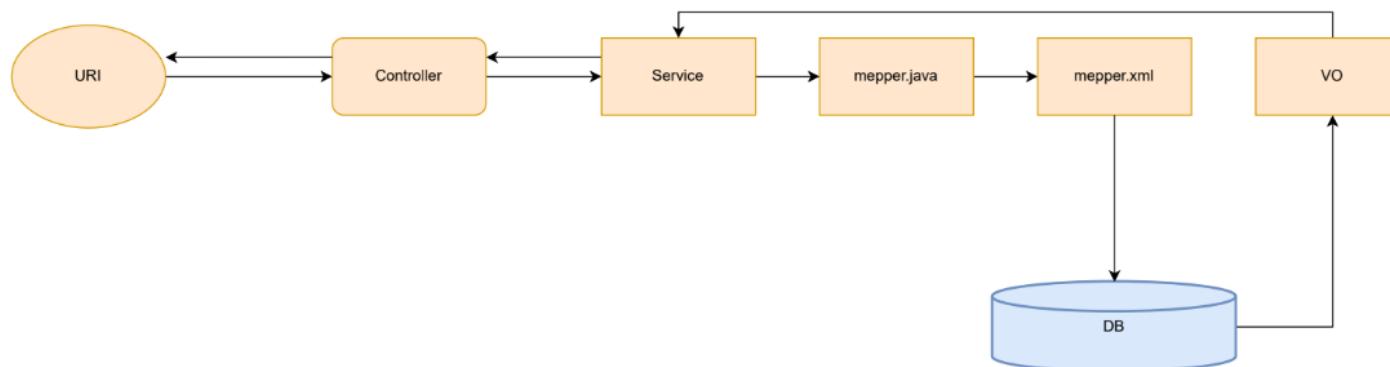
핵심코드

오류수정

추후 개발 내용

후기

호텔 예약



VO And Method

호텔 예약

hotel_reservationsVO

reservations_no
roomNumber
id
guestName
phoneNumber
paymentStatus
roomPrice
paymentMethod
payment_Type
paymentDate
start_date
end_date
people

실행되는 메소드

GET

- hotel_reservations.list()
- hotel_reservations.view()
- hotel_reservations.adminList()
- hotel_reservations.writeForm()

POST

- hotel_reservations.write()
- hotel_reservations.changeStatus()
- hotel_reservations.delete()

Data Flow Diagram : 호텔 후기

01 프로젝트 개요
주제와 목적

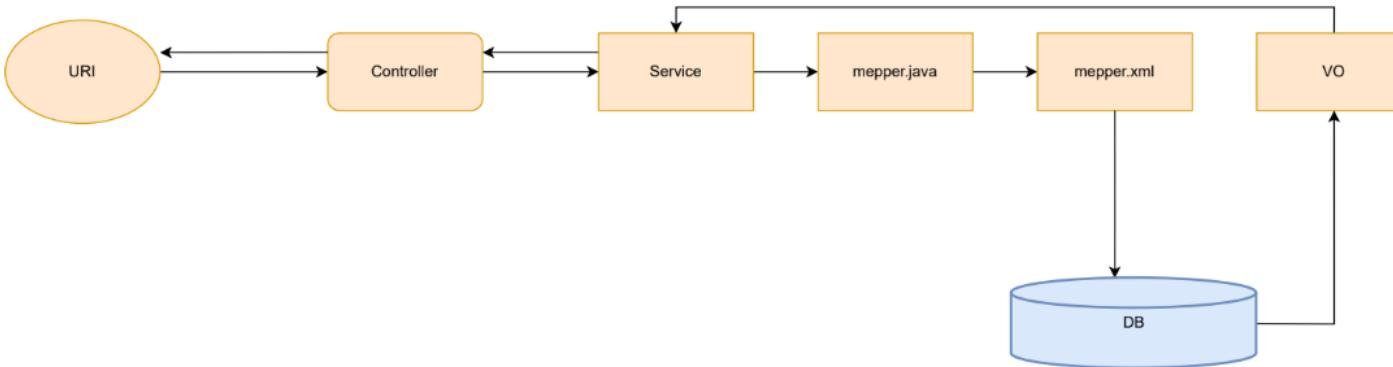
02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

리뷰



VO And Method

리뷰

reviewVO

reviewID
propertyName
roomDetails
service_Rating
cleanliness_Rating
convenience_Rating
Amenities_Rating
star
reviewContent
reviewDate
roomNumber
id
payNo
likeCount
reviewTitle

실행되는 메소드

GET
reviews.view()
reviews.list()
review.delete()

POST
reviews.count()
reviews.write()

Data Flow Diagram : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

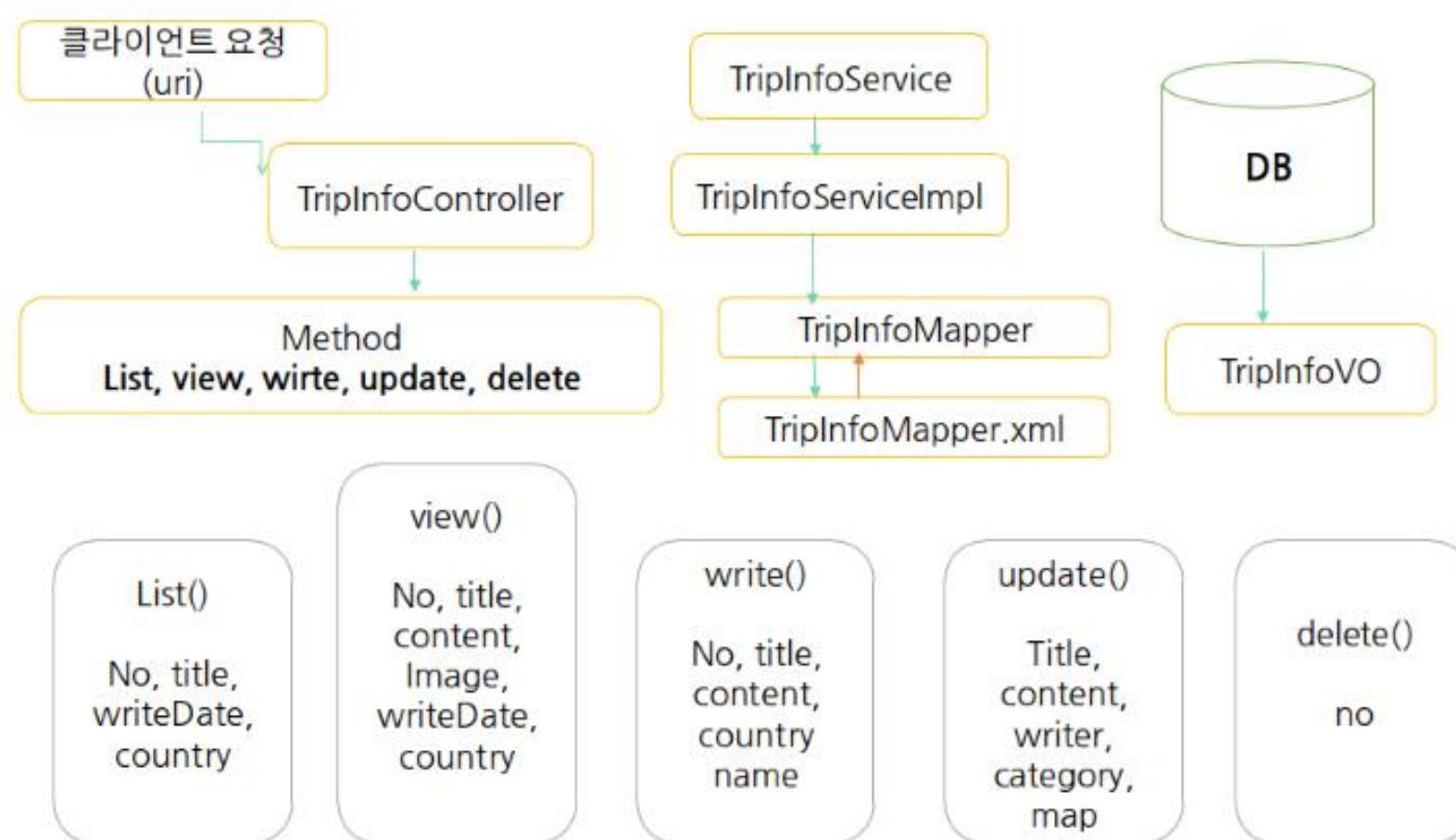
시연

핵심코드

오류수정

추후 개발 내용

후기



Data Flow Diagram : 여행 정보 게시판 댓글

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

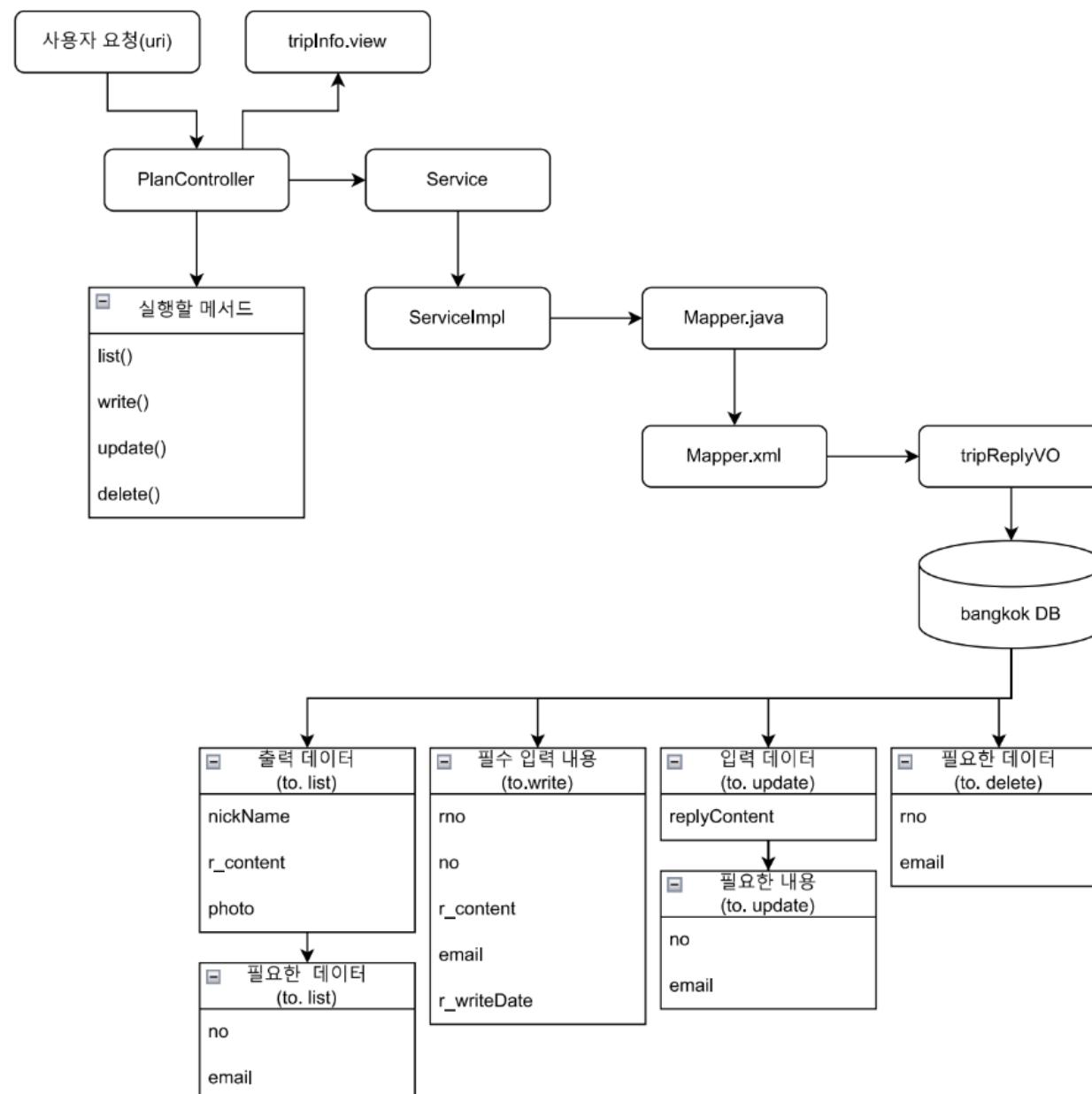
시연

핵심코드

오류수정

추후 개발 내용

후기



Data Flow Diagram : 여행 계획 짜기 게시판

01 프로젝트 개요
주제와 목적

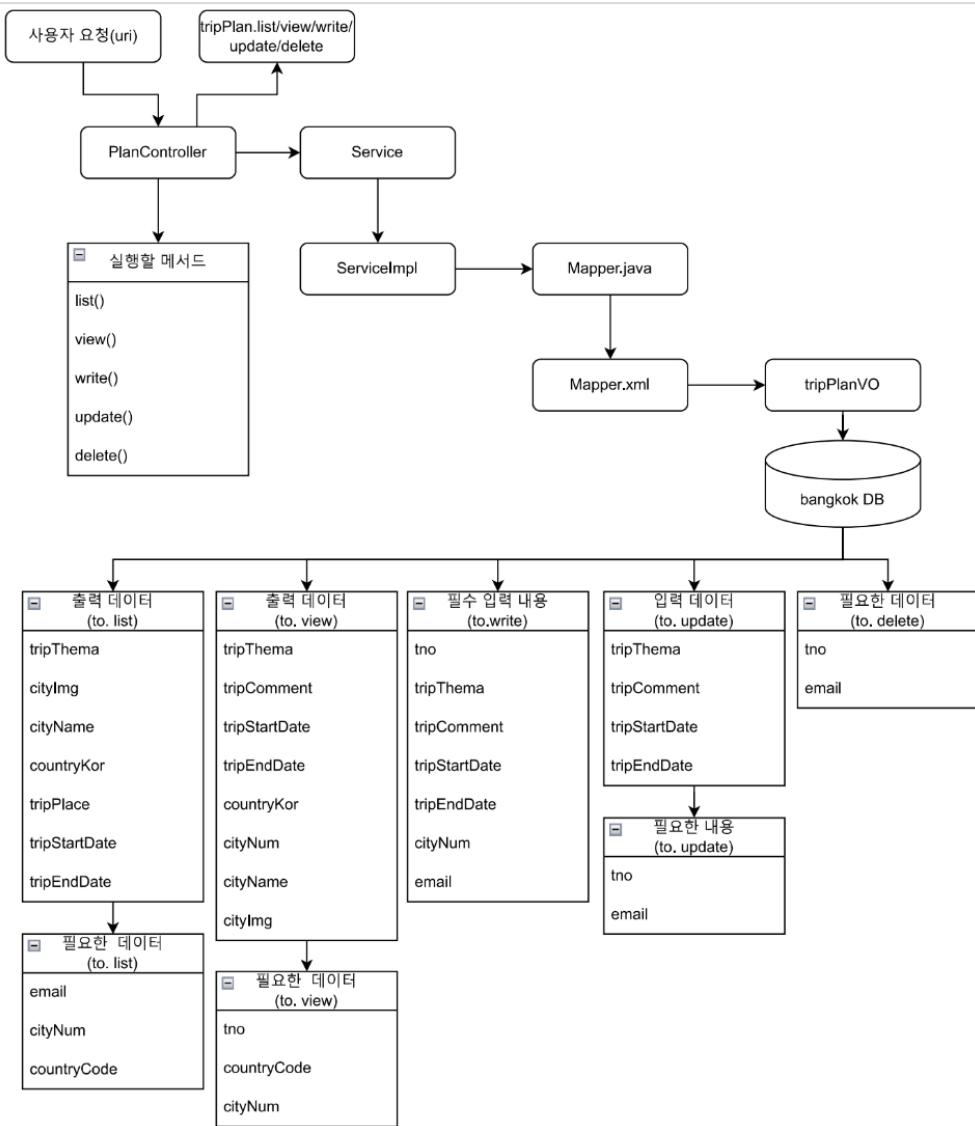
02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

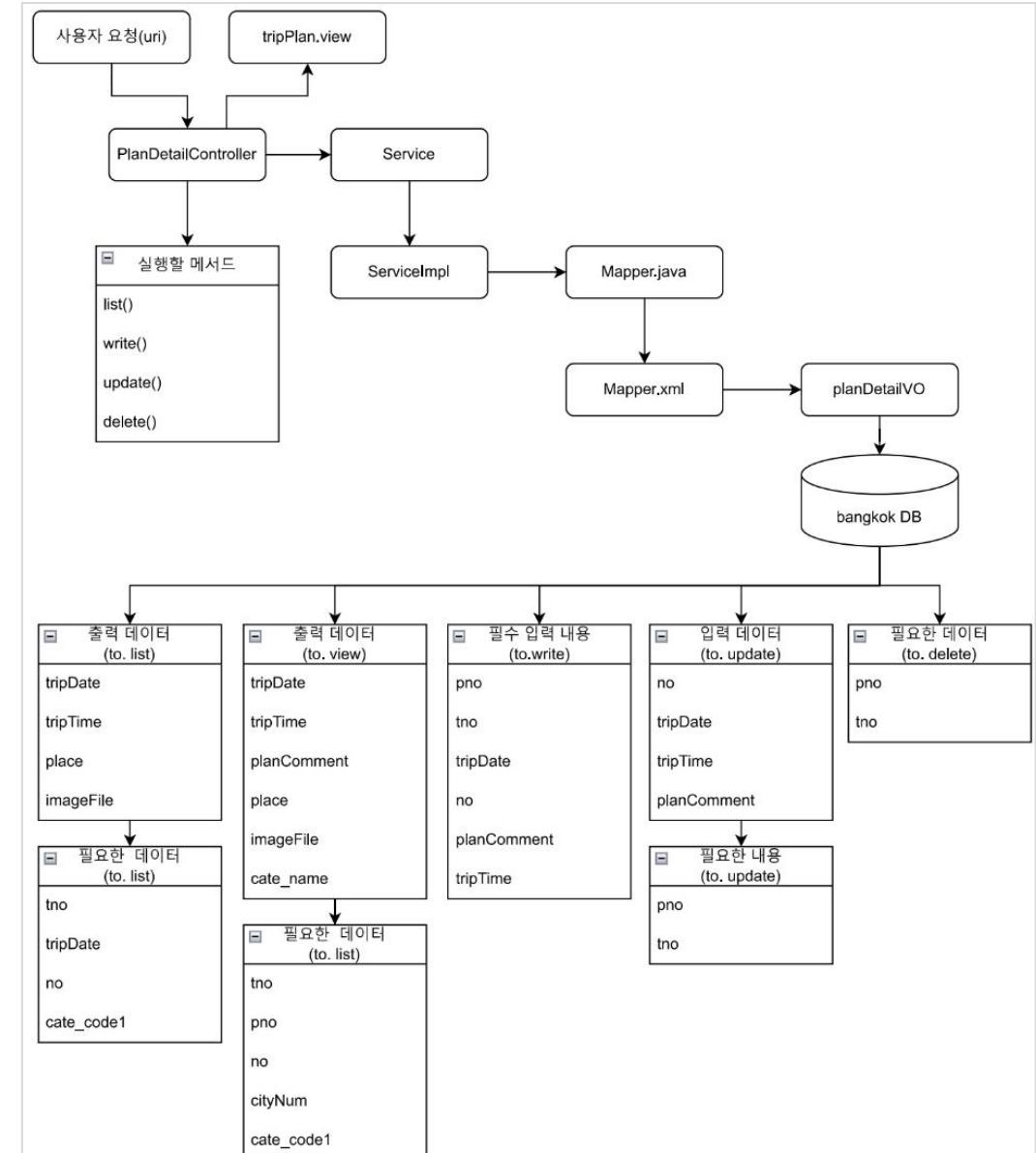
05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

[tripPlan]



출력 데이터 (to. list)	출력 데이터 (to. view)	필수 입력 내용 (to.write)	입력 데이터 (to.update)	필요한 데이터 (to.delete)
tripThema cityImg cityName countryKor tripPlace tripStartDate tripEndDate	tripThema tripComment tripStartDate tripEndDate countryKor cityNum cityName cityImg	tno tripThema tripComment tripStartDate tripEndDate cityNum cityName email	tno tripThema tripComment tripStartDate tripEndDate cityNum cityName email	tno email
				필수한 내용 (to. update)
				tno email
				필요한 데이터 (to. view)
				tno countryCode cityNum

[tripPlanDetail]



출력 데이터 (to. list)	출력 데이터 (to. view)	필수 입력 내용 (to.write)	입력 데이터 (to.update)	필요한 데이터 (to.delete)
tripDate tripTime place imageFile	tripDate tripTime place imageFile	pno tno place imageFile	pno tno place imageFile	pno tno
				필수한 데이터 (to. list)
				tno place imageFile
				cate_name
				tno tripDate no
				cate_code1
				tno pno no cityNum cate_code1

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

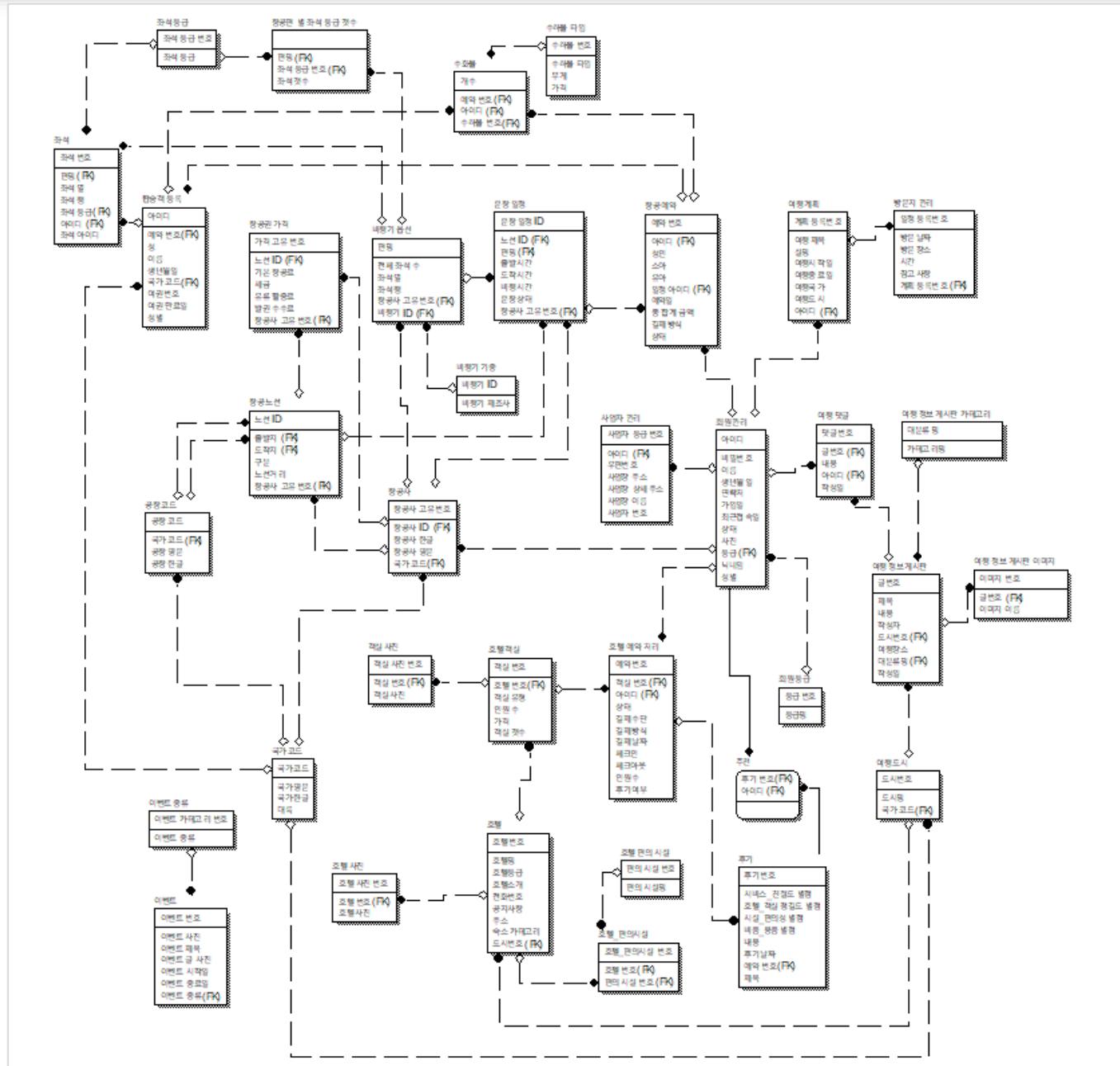
시연

핵심코드

오류수정

추후 개발 내용

후기





Data Base : Physical

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

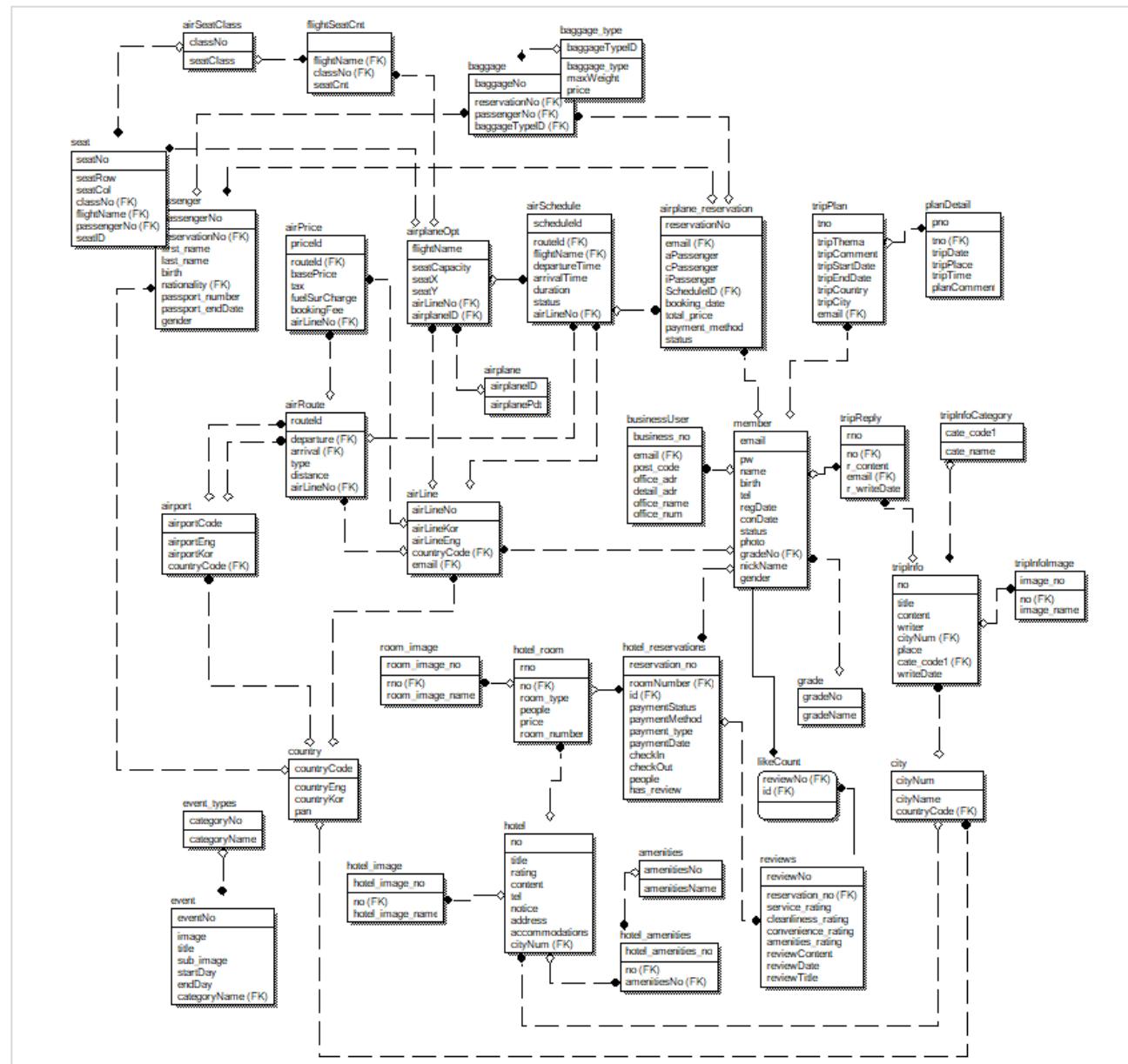
시연

핵심코드

오류수정

추후 개발 내용

후기



05 프로젝트 시연

회원관리

화면을 이동 합니다.

05 핵심코드 및 오류

회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

구글로그인 API

```

GoogleIdToken idToken = verifier.verify(idTokenString);
if (idToken != null) {
    GoogleIdToken.Payload payload = idToken.getPayload();
    MemberVO vo = service.findByEmail(email != null ? email : payload.getEmail());

    if (vo == null) {
        GoogleMemberVO googleMember = new GoogleMemberVO();
        googleMember.setEmail(email != null ? email : payload.getEmail());
        googleMember.setName((String) payload.get("name"));
        googleMember.setPhoto((String) payload.get("picture")); // 프로필 사진 URL도 포함
        service.saveGoogleMember(googleMember);
        vo = service.findByEmail(googleMember.getEmail()); // 새로 저장한 회원 정보 가져오기
    }

    session.setAttribute("user", vo); // "vo" 대신 "user"로 세션에 저장

    // 메시지 설정
    response.put("status", "success");
    response.put("msg", vo.getName() + "님이 구글 계정으로 로그인 되었습니다."); // 메시지를 response에 추가
} else {
    response.put("status", "error");
    response.put("msg", "유효하지 않은 ID 토큰입니다.");
}

```

토큰 / 이메일을 가져온다.

[DB에 동일한 이메일이 없으면 회원가입 / 있으면 로그인]

구글이 제공하는 토큰과 구글에 어플리케이션 만들때 체크한 동의 목록을 가져와서 MEMBER DB에 insert되어 자동 회원가입과 동시에 로그인 처리를 했다.
 ** 회원 토큰은 DB에 저장하지 않아도 된다. / 저장하면 보안에 더 위험하기 때문 /

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

구글로그인 API

Fetch API를 사용한 비동기 HTTP 요청

RestController에 구글 로그인 메서드를
가져와서 json 형태로 반환하는 형식
구글 토큰을 이용해서

회원가입 및 로그인이 되면
거기에 맞는 모달 메시지가 띄워진다.

```
function handleCredentialResponse(response) {
  const id_token = response.credential;
  if (id_token) {
    console.log('ID Token:', id_token);

    fetch('/auth/googleLogin.do', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ token: id_token })
    })
    .then(response => response.json())
    .then(data => {
      console.log(data); // 응답 데이터 확인

      // 응답 메시지를 모달에 삽입
      const modalMessage = document.getElementById('modalMessage');
      modalMessage.innerHTML = data.msg; // 서버 응답 메시지 삽입

      // 모달 띄우기
      $('#msgModal').modal('show');

      // "확인" 버튼 클릭 시 새로고침
      document.getElementById('modalCloseButton').addEventListener('click', function() {
        location.reload(); // 페이지 새로고침
      });
    })
  }
}
```

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

비밀번호 찾기 (SMTP)

```

// 비밀번호 찾기 처리
@PostMapping("/findPw.do")
public String findPassword(@RequestParam("name") String name,
                           @RequestParam("email") String email,
                           @RequestParam("tel") String tel,
                           Model model) {
    MemberVO member = service.findPassword(name, email, tel);

    if (member != null) {
        String token = UUID.randomUUID().toString();
        service.createPasswordResetTokenForUser(member, token);

        String resetUrl = "http://localhost:80/member/findPwResult.do?token=" + token;
        emailService.sendSimpleMessage(email, "[BANGKOK] 비밀번호 재설정", "비밀번호를 재설정하려면 아래 링크를 클릭하세요:\n" + resetUrl);
        model.addAttribute("msg", "비밀번호 재설정 링크가 이메일로 전송되었습니다."); // 포워드로 메시지 전달
    } else {
        model.addAttribute("msg", "입력하신 정보와 일치하는 사용자가 없습니다."); // 포워드로 메시지 전달
    }

    return "member/findPwForm"; // 비밀번호 찾기 폼으로 포워드
}

```

비밀번호 찾기 – SMTP 사용

이름 / 이메일 / 전화번호 일치 확인 후
토큰을 랜덤생성해서 회원 이메일주소로
비밀번호 변경 링크로 보내는 형식

[BANGKOK] 비밀번호 재설정 ➤ 받은편지함 ×



susu_en@naver.com

나에게 ▾

비밀번호를 재설정하려면 아래 링크를 클릭하세요:

[랜덤토큰번호](http://localhost:80/member/findPwResult.do?token=aa76436f-9b82-4b52-b3b1-71a1f0d90037)

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

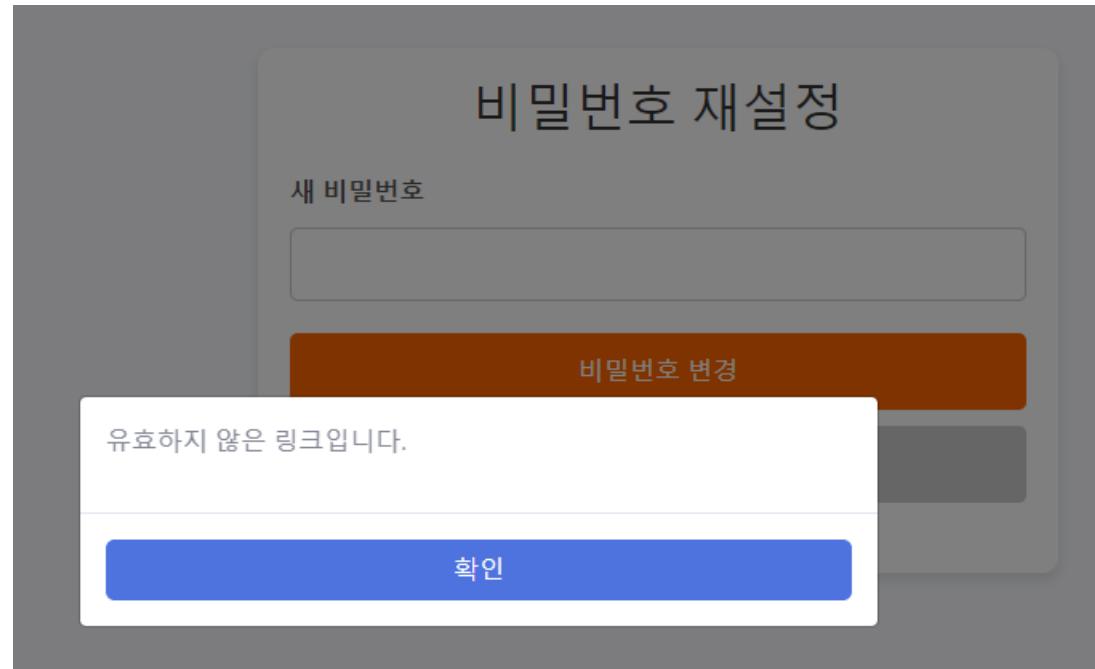
추후 개발 내용

후기

비밀번호 찾기 (SMTP)

비밀번호 찾기 – SMTP 사용

токен 유효 시간은 1시간
 1시간 이후에 누르게 된다면
 모달창으로 유효하지 않은 링크라고 뜨면서 메인
 화면으로 리다이렉트 된다.



```
<update id="createPasswordResetTokenForUser" parameterType="map">
<![CDATA[
UPDATE member
SET reset_token = #{reset_token}, reset_token_expiry = SYSDATE + INTERVAL '1' HOUR
WHERE email = #{email}
]]>
</update>
```

токен 시간 설정



오류 수정 : 회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

오류코드- 회원 사진 수정

MemberVO

```
private String photo; // 사진
```

오류 내용 :

타입이 vo는 String인데 컨트롤러에서는
MultipartFile로 되어 있어서
타입미스매치 에러가 났다.

```
INFO : com.bangkok.util.interceptor.AuthrityInterceptor - [ 권한 처리 Interceptor ] -----  
ERROR: com.bangkok.exception.CommonExceptionAdvice - Exception-----org.springframework.validation.  
Field error in object 'memberVO' on field 'photo': rejected value [org.springframework.web.multipart.support.Sta  
ERROR: com.bangkok.exception.CommonExceptionAdvice - {exception=org.springframework.validation.BindException: or
```

memberController

```
// 회원 정보 수정 처리  
@PostMapping(value = "/update.do", produces = "application/json;charset=UTF-8")  
@ResponseBody  
public ResponseEntity<Map<String, String>> update(  
    @RequestParam("email") String email,  
    MemberVO memberVO,  
    @RequestParam("photo") MultipartFile photo,  
    @RequestParam(value = "existingPhoto", required = false) String existingPhoto,  
    HttpServletRequest request) throws Exception {  
  
    log.info("memberController.update()-----");  
    log.info("전달된 이메일: " + email); // 전달된 이메일 로그 추가  
  
    // 이메일 설정  
    memberVO.setEmail(email); // 전달받은 이메일로 memberVO 설정  
  
    // 기존 사진이 있을 경우, 기본값으로 설정  
    if (existingPhoto != null && !existingPhoto.isEmpty()) {  
        memberVO.setPhoto(existingPhoto);  
    }  
  
    // 파일 업로드 처리  
    if (photo != null && !photo.isEmpty()) {  
        String photoPath = FileUtil.upload(path, photo, request);  
        memberVO.setPhoto(photoPath);  
        log.info("업로드된 사진: " + photo.getOriginalFilename());  
    }  
}
```

updateForm.jsp

```
<!-- 프로필 사진 -->  
<input type="file" id="fileInput" name="photo" accept="image/*">
```



BANGKOK

오류 수정 : 회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

오류코드- 회원 사진 수정

해결 내용 :

컨트롤러에서 요청 파라미터의 이름을 photo에서 photoFile로 변경하였더니,
파일 업로드가 정상적으로 처리되었습니다.

photoFile이 MultipartFile 타입으로 선언되어
파일 업로드 요청을 올바르게 매핑하고 바인딩할 수 있었기 때문입니다.

```
    @RequestParam("photoFile") MultipartFile photoFile,
```

```
<!-- 프로필 사진 -->
<input type="file" id="fileInput" name="photoFile" accept="image/*">
```

추후 개발 내용 : 회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

개발 모듈	추후 개발 내용	비고
회원 관리	네이버 로그인 api	
회원 관리	현재 : memberList(관리자 페이지) 직접 uri를 입력해서 들어가는 접근권한 처리는 했지만 이후에도 다른 페이지들도 마찬가지로 해보고 싶다.	

후기 : 회원 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

여행 사이트의 구조를 이해하게 되어 많은 도움이 되었습니다.

마지막 팀 프로젝트에서 회원 관리를 맡았지만, 팀원들이 개발한 다양한 기능들을 보며 전체적인 흐름을 공부할 수 있었습니다. 또한, 다양한 API를 활용하면서 문서와 구글링을 통해 직접 구현해보는 기회를 가지게 되어 더욱 값진 경험이 되었습니다.

저는 팀 프로젝트에서 가장 중요한 요소로 시간 관리와 소통을 꼽습니다.

회의 시간에는 서로의 아이디어와 일정을 공유하고 피드백을 주고받으며 큰 발전을 이뤘습니다. 하지만, 몇몇 상황에서는 소통의 부족으로 인해 작업이 지연되기도 했습니다.

이러한 경험을 통해 소통의 중요성을 다시 한 번 깨닫게 되었고,

앞으로는 이러한 문제를 효과적으로 해결할 수 있도록 노력하겠습니다.

이번 프로젝트에서 얻은 배움과 경험을 바탕으로 향후 프로젝트에서도 더욱 성장해 나가겠습니다.

05 프로젝트 시연

호텔 상품 관리

화면을 이동 합니다.

05 핵심코드 및 오류

호텔 상품 관리

핵심 코드 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

```
// 객실 추가 이미지 데이터를 세팅한다. - each(무한반복) - 핵심코드
$(".hotelRoomList").each(
    function() {
        let rno = $(this).find(".rno").val();
        $(this).find(".hotelRoom").load(
            "/ajax/getRoomImageList.do?rno=" + rno+"&no=${param.no}");
    });

```

객실 목록



Standards1

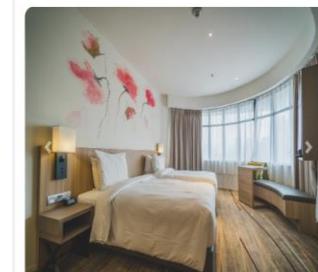
객실 정보: 두바이 객실 1 층 플랫

가격: 500,000 원

인원수: 2 명

남은 객실: 9 개

예약하기



Premieres2

객실 정보: 두바이 초클래스 객실 2

가격: 1,000,000 원

인원수: 4 명

남은 객실: 10 개

예약하기

핵심 코드 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

```
$(function() {
    // JSP에서 JSTL의 forEach를 이용해 amenities 데이터를 자바스크립트 배열로 변환
    var amenitiesPick = [];

    <c:forEach items="${amenitiesList}" var="amenity">
        amenitiesPick.push({
            amenitiesNo: '${amenity.amenitiesNo}',
        });
    </c:forEach>
    console.log(amenitiesPick);
    // amenitiesPick에 있는 선택된 편의시설을 체크 상태로 설정
    for (let i = 0; i < amenitiesPick.length; i++) {
        let amenitiesNo = amenitiesPick[i].amenitiesNo; // 선택된 편의시설 번호
        `$.amenitiesCheckBox[value='${amenitiesNo}']`.prop("checked", true); // 체크박스 선택
    }
}
```

amenitiesPick 배열을 사용하여 amenitiesNo 데이터를 클라이언트 측에서 사용할 수 있게 변환하고, 서버에서 JSP로 전달된 데이터를 클라이언트 측에서 효율적으로 다룰 수 있도록 처리합니다.

변환된 amenitiesPick 배열을 순회하면서 각 편의시설 번호에 해당하는 체크박스를 prop("checked", true)를 사용하여 해당 체크 박스를 선택 상태로 만듭니다.

기존에 호텔 등록시 같이 등록되었던 편의시설이 체크 된 상태로 출력 되는 것을 확인할 수 있습니다.

핵심 코드 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

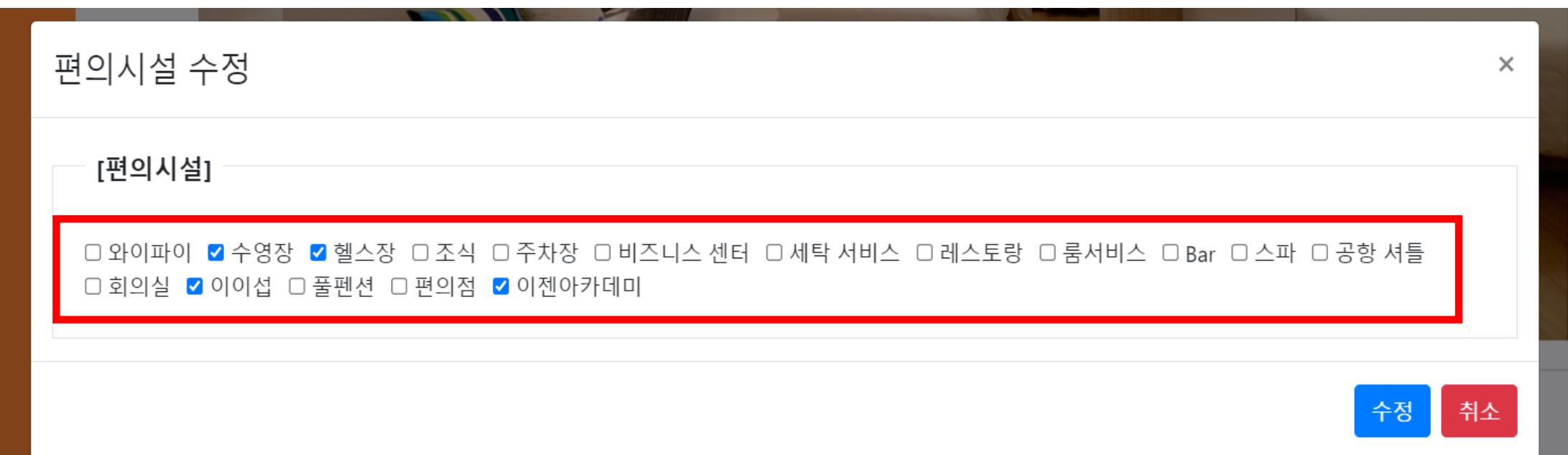
시연

핵심코드

오류수정

추후 개발 내용

후기





BANGKOK

오류 수정 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

에러코드 1-1

```
        class="rno">
<div class="card-body d-flex justify-content-between">
    <div id="hotelRoom"></div>
    <div class="info-container ml-3">
        <p>
            <strong>${hotelRoomVO.room_type}</strong> ${hotelRoom}
```

1번 객실

```
<!-- 숨겨서 넘겨야할 데이터 - 이미지 번호, 현재 파일이름(삭제)
<input type="hidden" name="no" value="5" class="no">
<input type="hidden" name="rno" value="37" class="rno">
<input name="deleteFileName" id="room_image_name" value="http
```

2번 객실

```
▶ <div class="modal-header">...</div> (flex)
▼ <form action="changeRoomImage.do" method="post" enctype="multip
    <!-- 숨겨서 넘겨야할 데이터 - 이미지 번호, 현재 파일이름(삭제)
    <input type="hidden" name="no" value="5" class="no">
    <input type="hidden" name="rno" value="" class="rno">
    <input name="deleteFileName" id="room_image_name" value type='
```

<c:foreach> 태그 내에 id속성으로 AJAX를 사용해 여러 개의 객실의 이미지를 불러오게 하였더니 첫번째의 객실에서만 반복을 하게 되어 2번 객실의 객실 번호를 가져오지 못하는 에러가 발생하였습니다.

01	프로젝트 개요 주제와 목적
02	개발 전반 요구 사항 개발 내용과 범위 소요 자원
03	Front-end 메뉴의 흐름 화면 구현
04	Back-end DFD DB
05	프로젝트 시연 시연 핵심코드 오류수정 추후 개발 내용 후기

에러코드 1-2

```
        class="rno">
<div class="card-body d-flex justify-content-between">
    <div class="hotelRoom"></div>
    <div class="info-container ml-3">
        <p>
```

1번 객실

```
<!-- 숨겨서 넘겨야할 데이터 - 이미지 번호, 현재 파일이름(삭제) -->
<input type="hidden" name="no" value="5" class="no">
<input type="hidden" name="rno" value="37" class="rno">
<input name="deleteFileName" id="room_image_name" value="http://localho
```

2번 객실

```
<!-- 숨겨서 넘겨야할 데이터 - 이미지 번호, 현재 파일이름(삭제) -->
<input type="hidden" name="no" value="5" class="no">
<input type="hidden" name="rno" value="38" class="rno">
<input name="deleteFileName" id="room_image_name" value="http://localho
```

속성을 id에서 class로 변경하고 확인해보니
수정하려는 객실의 알맞은 번호를 가져올 수
있는 것을 확인할 수 있었습니다.

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

에러코드 2

```
// 객실 이미지 변경
$(".changeRoomImageBtn").click(function() {
    let rno = $(this).closest(".room-carousel").data("rno");
    alert(rno);
    let room_image_name = $(this).closest(".room-carousel")
        .find("img").prop("src");

    $(".rno").val(rno);
    $("#room_image_name").val(room_image_name);
});
```

수정 전



다음은 AJAX에서 불러온 데이터를 변경하기 위해 쓰여진 코드입니다.

AJAX로 불러온 데이터에 이벤트 바인딩을 할 경우 .on() 메서드를 사용해야 하는데, .on() 메서드를 사용하지 않아서 rno의 값을 받아오지 못하는 경우를 찾을 수 있었습니다.

```
// 객실 이미지 변경
// Ajax를 사용했을 때는 on을 사용해서 Ajax를 호출하는 태그의 id를 $("oo")에 적고 ".click",
// "어떤것을 클릭했을때", function() 실행한다.
$(".hotelRoom").on("click", ".changeRoomImageBtn", function() {
    alert("test");
    let rno = $(this).closest(".room-carousel").find(".hotelRoomList").data("rno");
    alert(rno);
    let room_image_name = $(this).closest(".room-carousel").find("img").prop("src");

    $(".rno").val(rno);
    $("#room_image_name").val(room_image_name);
});
```

수정 후

추후 개발 내용 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

개발 모듈

추후 개발 내용

비고

호텔 객실의 이미지 추가, 삭제

현재 구현된 기능에서는 호텔, 객실의 이미지 추가 및 삭제가 구현되지 않았기에 사업자, 관리자의 호텔, 객실을 수정 할 때 이미지 추가, 삭제가 가능하지 않아서 아쉬움이 많이 남았기에 차후 개발 목록으로 선정하게 되었습니다.

관리자 페이지

사업자, 관리자로 로그인 했을시 호텔 상세보기 화면에서 수정 삭제 버튼들을 볼 수 있는데 이러한 버튼들과 버튼시 이동하는 수정 화면을 한 페이지에서 출력하고 관리 할 수 있지 않을까 하는 생각에 관리자 페이지를 차후 개발로 선정하게 되었습니다.



BANGKOK

후기 : 호텔 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

- 상품 관리 파트는 단순하지 않으며, 생각 이상으로 많은 기능을 요구하는 파트이다.
- 에러를 자주 볼수록 문제점은 점점 더 빨리 찾게 된다.
- Spring을 잘 다루려면, Servlet을 잘 다뤄야 한다.
- 마지막 프로젝트인 만큼 가장 어려웠던 프로젝트 였던 것 같다

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

05 프로젝트 시연

호텔 예약 주문 관리 / 호텔 후기 / 이벤트 게시판

화면을 이동 합니다.

05 핵심코드 및 오류

호텔 예약 주문 관리

핵심 코드 : 호텔 예약 주문 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

```

49      <!-- 호텔 예약 관리자 리스트 쿼리 -->
50<select id="business" parameterType="com.bangkok.util.page.PageObject" resultType="com.bangkok.hotel
51 select RESERVATION_NO, name, tel, price, PAYMENTSTATUS, paymentMethod, payment_Type, title, room_typ
52 from (
53     select rnum, RESERVATION_NO, name, tel, price, PAYMENTSTATUS, paymentMethod, payment_Type
54     from (
55         SELECT hr.RESERVATION_NO, m.name, m.tel, r.price, hr.PAYMENTSTATUS, hr.paymentMethod, hr.pay
56         FROM hotel_room r, Hotel h, member m, hotel_reservations hr
57         where(r.no = h.no) and (hr.ROOMNUMBER = r.rno) and (hr.id = m.email)
58             <include refid="search"/>
59             <include refid="searchPeriod"/>
60     )
61 )
62 where rnum between #{pageObject.startRow} and #{pageObject.endRow}
63 </select>
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
607
608
609
609
610
611
612
613
614
615
615
616
617
617
618
618
619
619
620
621
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
631
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
```



BANGKOK

오류 수정 : 호텔 예약 주문 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

오류 코드

```
</ui>
<div class="col-md" style="padding: 0;">
  <div style="font-size: 30px;">${vo.title}</div>
  <b style="font-size: 15px;">${vo.address}<i class="fa fa-clipboard clip" style="margin-left: 10px;</i></b>
  <input type="text" id="copyText" value="${vo.address}" />
</div>
</div>

<div class="row" style="margin-top: 20px;">
  <div class="col-md-1" style="padding: 0;">
    일정
  </div>
  <div class="col-md">
    <span><b><fmt:formatDate value="${vo.checkIn }" pattern="yyyy-MM-dd(EEE)HH:mm"/> ~ </b></span>
  </div>
</div>

<div class="row" style="margin-top: 20px;">
```



오류 내용

[title] 특성이 [com.bangkok.hotel_reservations.vo.Hotel_reservationsVO] 유형에 없습니다.
[BeanProperties.get\(BeanELResolver.java:253\)](#)

오류 해결 코드

```
</ui>
<div class="row" style="margin-top: 20px;">
  <div class="image-container col-md-4" style="text-align: center;">
    
  </div>
  <div class="col-md" style="padding: 0;">
    <div style="font-size: 30px;">${vo.title}</div>
    <b style="font-size: 15px;">${vo.address}<i class="fa fa-clipboard clip" style="margin-left: 10px;</i></b>
    <input type="text" id="copyText" value="${vo.address}" />
  </div>
</div>

<div class="row" style="margin-top: 20px;">
  <div class="col-md-1" style="padding: 0;">
    일정
  </div>
  <div class="col-md">
    <span><b><fmt:formatDate value="${vo.checkIn }" pattern="yyyy-MM-dd(EEE)HH:mm"/> ~ </b></span>
  </div>
</div>

<div class="row" style="margin-top: 20px;">
```



05 핵심코드 및 오류

호텔 리뷰

핵심 코드 : 호텔 리뷰

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

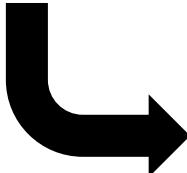
후기

```
// 리뷰 보기 버튼 클릭 시 AJAX 요청
$(document).on('click', '.reviewView', function() {
    let reservationNo = $(this).closest('.row').find('.room-type').data('no');

    $.ajax({
        url: '/reviews/view.do',
        type: 'GET',
        success: function(response) {
            $.ajax({
                url: '/reviews/view.do',
                type: 'GET',
                data: { reservations_no: reservationNo },
                success: function(response) {
                    // 별점 및 리뷰 내용 업데이트
                    setStars('star-service', response.service_Rating || 0);
                    setStars('star-cleanliness', response.cleanliness_Rating || 0);
                    setStars('star-facilities', response.convenience_Rating || 0);
                    setStars('star-equipment', response.amenities_Rating || 0);

                    $('#customReviewTitle').text(response.reviewTitle || '제목 없음');
                    $('#hotelName').text(response.title || '호텔명 없음');
                    $('#hotel_roomName').text(response.room_type || '객실명 없음');
                    $('#customReviewContent').text(response.reviewContent || '내용 없음');

                    $('#reviewModals').modal('show');
                }
            });
        }
    });
});
```





BANGKOK

오류 수정 : 호텔 리뷰

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

오류 코드

```
// --- 리뷰 보기 -----
@GetMapping("/view.do")
@ResponseBody
public ResponseEntity<?> view(@RequestParam("reservationNo") Long reservation_no) {
    log.info("view.do");

    // 서비스에서 데이터를 가져옴
    ReviewsVO reviewVO = service.view(reservation_no);

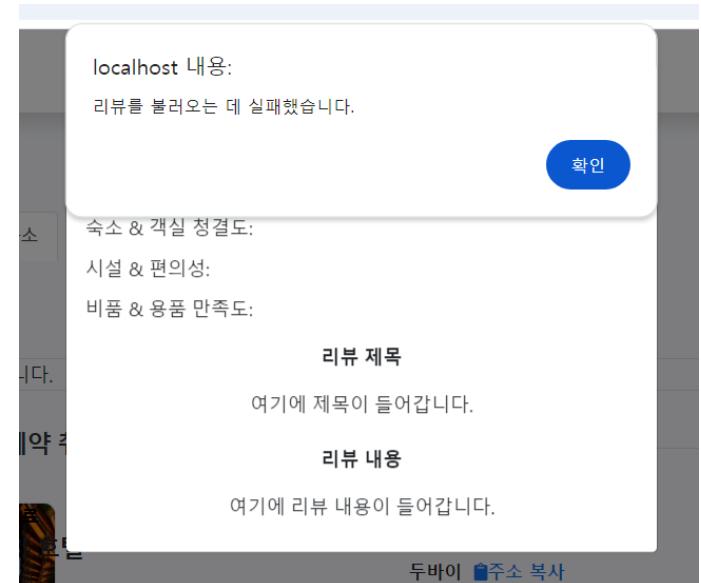
    // 데이터를 JSON 형식으로 반환
    return ResponseEntity.ok(reviewVO);
}
```

오류 해결 코드

```
// --- 리뷰 보기 -----
@GetMapping("/view.do")
@ResponseBody
public ResponseEntity<?> view(@RequestParam("reservation_no") Long reservation_no)
    log.info("view.do");

    // 서비스에서 데이터를 가져옴
    ReviewsVO reviewVO = service.view(reservation_no);
```

오류 내용



```
// 리뷰 보기 버튼 클릭 시 AJAX 요청
$(document).on('click', '.reviewView', function() {
    var reservationNo = $(this).closest('.row').find('.room-type').data('no');

    $.ajax({
        url: '/reviews/view.do',
        type: 'GET',
        data: { reservation_no: reservationNo },
        success: function(response) {
            console.log('Success:', response);
            console.log('Service Rating:', response.service_Rating);
            console.log('Cleanliness Rating:', response.cleanliness_Rating);
            console.log('convenience_Rating:', response.convenience_Rating);
            console.log('amenities_Rating:', response.amenities_Rating);
        }
    });
});
```

05 핵심코드 및 오류

이벤트 게시판



BANGKOK

핵심 코드 : 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

```
// --- 이벤트 등록 처리 -----
@PostMapping("/write.do")
public String write(EventVO vo, RedirectAttributes rttr, @RequestParam ArrayList<MultipartFile> files,
        HttpServletRequest request) throws Exception {
    Log.info("write.do");
    String path = "/upload/event";
    for(MultipartFile file : files) {
        Log.info("-----");
        Log.info("name : " +file.getOriginalFilename());
        Log.info("size : " +file.getSize());
        // file upload
        String add=(FileUtil.upload(path, file, request));
        if(vo.getImage() ==null)
            vo.setImage(add);
        vo.setSub_image(add);
    }
    service.write(vo);

    // 처리 결과에 대한 메시지 처리
    rttr.addFlashAttribute("msg", "이벤트가 정상 등록되었습니다.");
}

return "redirect:list.do";
}
```



BANGKOK

오류 수정 : 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

오류 코드

```
2 ②     @Override  
3  public List<EventVO> tripList() {  
4      // TODO Auto-generated method stub  
5      return mapper.tripList();  
6  }  
  
8 ③     @Override  
9  public List<EventVO> listEvent() {  
0      // TODO Auto-generated method stub  
1      return mapper.listEvent();  
2  }  
  
4 ④     @Override  
5  public EventVO checkId(String id) {  
6      // TODO Auto-generated method stub  
7      return checkId(id);  
8  }  
9 }  
0
```



오류 내용

```
-----Handler dispatch failed; nested exception is java.lang.StackOverflowError  
    tk.web.util.NestedServletException: Handler dispatch failed; nested exception is java.lang.StackOverflowError}
```



오류 해결 코드

```
// TODO Auto-generated method stub  
return mapper.checkIn(email);
```

추후 개발 내용 : 호텔 예약 관리 / 호텔 리뷰 / 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

개발 모듈

추후 개발 내용

비고

**호텔 예약 관리
호텔 리뷰
이벤트 게시판**

일단 이벤트에서 상세보기에서 딸랑 포스트만 보이는데 나중 개발 때는 안에 버튼을 넣어서 쿠폰을 받을 수 있게하거나 여러 이벤트 기능을 개발해보고 싶고, 호텔 예약에서는 더 개발하고 싶은게 없고 리뷰 리스트 부분에서 사용자 편의기능을 더 추가하고 싶다.

후기 : 호텔 예약 관리 / 호텔 리뷰 / 이벤트 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

1. 개발하면서 부족한 부분보다 아는 부분이 많아서 좋았고 좀 아쉬운 부분은 제 코드가 다른 개발자들이

봤을 때 전체적인 코드가 스파게티 코드 인 것 같아서 다음 프로젝트 때 어떻게 하면 다른 개발자나 제

03 Front-end

메뉴의 흐름

화면 구현

가 유지보수 했을 때 복잡하지 않고 단순화 할 수 있을까를 고민 해 봐야할 것 같습니다. 마지막으로 너

04 Back-end

DFD

DB

무 코드가 재 사용 되는 코드가 많았는데 그걸 저번 프로젝트 때 신경 쓰자고 했는데 이게 막상 여러 가

지 코드를 짜다 보니까 생각을 이번에도 못했는데 다음 프로젝트 때는 먼저 재 사용 되는 코드가 뭔지 생

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

각을 해보고 짜봐야할거같습니다.

05 프로젝트 시연

항공 상품 관리

화면을 이동 합니다.

05 핵심코드 및 오류

항공 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

스케줄 반복 등록

운항 스케줄 등록									
<input checked="" type="checkbox"/> 반복 등록 2024-10-01 <input type="button" value="선택"/> 부터 2024-10-31 <input type="button" value="선택"/> 까지									
노선	운행편명	출발일	출발시간	도착일	도착시간	소요시간	삭제		
HKG - FRA	KE8251 (A310)	2024-10-01 <input type="button" value="선택"/>	오전 09:00 <input type="button" value="선택"/>	2024-10-01 <input type="button" value="선택"/>	오후 03:30 <input type="button" value="선택"/>	6시간 30분			
HKG - FRA	KE8123 (A330)	2024-10-01 <input type="button" value="선택"/>	오후 01:00 <input type="button" value="선택"/>	2024-10-01 <input type="button" value="선택"/>	오후 07:00 <input type="button" value="선택"/>	6시간 0분			
HKG - FRA	KE853 (A340)	2024-10-01 <input type="button" value="선택"/>	오후 05:00 <input type="button" value="선택"/>	2024-10-01 <input type="button" value="선택"/>	오후 11:30 <input type="button" value="선택"/>	6시간 30분			

ScheduleVO

```
@Data  
public class ScheduleVO {  
  
    private Long airlineNo;  
    private Long routeId;  
    private String flightName;  
  
    private boolean repeat; // Red box highlights this line  
    @DateTimeFormat(pattern = "yyyy-MM-dd")  
    private Date startDate;  
    @DateTimeFormat(pattern = "yyyy-MM-dd")  
    private Date endDate;  
    private Long repeatDays;  
  
    private String departureDate;  
    private String departureTimeStr;  
    @DateTimeFormat(pattern = "yyyy-MM-dd HH:mm")  
    private Date departureTime;  
    private String arrivalDate;  
    private String arrivalTimeStr;  
    @DateTimeFormat(pattern = "yyyy-MM-dd HH:mm")  
    private Date arrivalTime;  
    private String durationStr;  
    private Long duration;  
  
    private List<ScheduleVO> list;
```

AirServiceImpl.class

```
// 스케줄 등록 처리
@Override
public Integer airScheduleWrite(Long airlineNo, ScheduleVO vos) {
    Integer result = 0;

    if (vos.isRepeat()) {
        List<ScheduleVO> list = vos.getList();
        Long repeatDays = vos.getRepeatDays();

        for(ScheduleVO vo : list) {

            vo.setAirlineNo(airlineNo);
            Date originalDepartureTime = vo.getDepartureTime();
            Date originalArrivalTime = vo.getArrivalTime();
            mapper.airScheduleWrite(vo);

            // 반복 등록
            for (long i = 1; i <= repeatDays; i++) {
                Date newDepartureTime = addDaysToDate(originalDepartureTime, i);
                Date newArrivalTime = addDaysToDate(originalArrivalTime, i);
                vo.setDepartureTime(newDepartureTime);
                vo.setArrivalTime(newArrivalTime);
                mapper.airScheduleWrite(vo);
            }
        }

        result = 1;
    } // end of for write

    else {
        List<ScheduleVO> list = vos.getList();

        for(ScheduleVO vo : list) {

            vos.setAirlineNo(airlineNo);
            mapper.airScheduleWrite(vo);
        } // end of for write
    }

    result = 1;
} // end of if repeat false
return result;
}
```

Repeat가 true 일 경우 처리

- 반복 종료일 – 시작일 : 일 수 계산
 - 기존에 입력한 일정 write
 - For문으로 일 수 만큼 출발, 도착 일 + 1일
 - 날짜에 1일 추가하는 메서드 작성

```
// Date 객체에 일(day)을 더하는 함수
private Date addDaysToDate(Date date, long days) {
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(date);
    calendar.add(Calendar.DATE, (int) days); // 일(day) 더하기
    return calendar.getTime(); // Date 객체로 반환
}
```

Repeat가 false 일 경우 처리

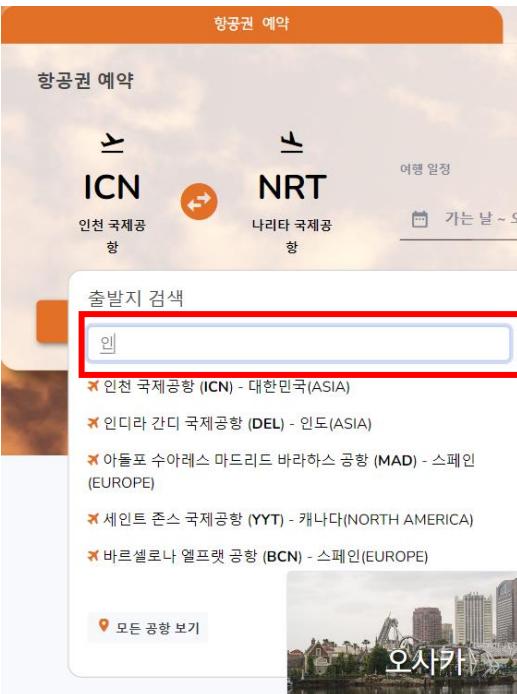
- 기존에 입력한 일정만 write

핵심 코드 : 항공 상품 관리

01 프로젝트 개요

주제와 목적

공항 검색



Keyup Event – ajax 요청

```
// AJAX 요청 보내기
$.ajax({
  url: '/air/searchAirport', // 서버의 공항 검색 API URL
  type: 'GET',
  data: { searchAirport: searchAirport }, // 요청에 검색어 전달
  dataType: 'json', // 반환되는 데이터 타입을 명시
  success: function(data, status, xhr) {

    console.log(data);
    // 검색 결과를 받아온 후 처리
    $('#airportListResults').empty(); // 기존 결과를 초기화
    if(data.list && data.list.length > 0) {
      // 검색된 공항 목록을 추가
      $.each(data.list, function(index, airport) {
        $('#airportListResults').append(
          '<p class="search-result" data-code="' + airport.airportCode + '" data-name="' +
          + airport.airportKor + '" style="cursor: pointer;">' +
          + '<i class="fa fa-plane" style="color:#E37027"></i>' +
          + airport.airportKor + '<b>' + airport.airportCode + '</b>' + -
          + airport.countryKor + '(' + airport.pan + ')</p>'
        );
      });
    } else {
      $('#airportListResults').append('<p>검색 결과가 없습니다.</p>');
    }
  },
  error: function(xhr, status, er) {
    // 오류 발생 시 처리
    $('#airportListResults').empty();
    $('#airportListResults').append('<p>오류가 발생했습니다.</p>');
  }
});
```

AirController.class

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

@ResponseBody를 사용하여 순수 데이터만 전달

검색 칼럼 : 공항명(한/영), 국가(한/영)
전달 되는 데이터 : 검색 결과 데이터

```
// 2. 공항 검색 리스트 - GET
// 넘겨주는 데이터는 기본이 XML로 보내고 JS에서 JSON으로 변환한다.
@GetMapping(value = "/searchAirport",
  produces = {
    MediaType.APPLICATION_XML_VALUE,
    MediaType.APPLICATION_JSON_UTF8_VALUE
  })
@ResponseBody
public ResponseEntity<Map<String, Object>> searchAirport(@RequestParam(value = "searchAirport") String searchAirport) {

  // DB에서 데이터를 가져와서 넘겨 준다.
  List<AirVO> list = service.searchList(searchAirport);
  Log.info("controller - service()");

  // list와 PageObject를 넘겨야 한다.
  Map<String, Object> map = new HashMap<String, Object>();
  map.put("list", list);

  // list와 pageObject의 데이터 확인
  Log.info("After - map : " + map);

  // ResponseEntity : 통신 상태를 확인하기 위해 ResponseEntity를 사용한다. 데이터 보내는 것 뿐만 아닌 상태까지 보낼 수 있다.
  // DB에서 조성 및 전체 공항명을 함께 검색
  return new ResponseEntity<Map<String, Object>>(map, HttpStatus.OK);
}
```



오류 수정 : 항공 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

항공사 기종 리스트를 불러올 때 오류 : MyBatis – Parameter 'XXX' not found

- 발생 원인

- Mapper XML 파일에서 SQL 매핑 시 파라미터(airlineNo)를 정확하게 인식을 하지 못하는 오류가 발생을 했습니다.
- DB 컬럼명과 VO 변수 이름, mapper로 전달되는 파라미터의 이름이 같으나 찾지 못하는 오류였습니다.

```
// 항공사 기종 리스트
public List<AirVO> flightList( PageObject pageObject, Long airlineNo);
```

```
Exception-----nested exception is org.apache.ibatis.binding.BindingException: Parameter 'airlineNo' not found.
Available parameters are [arg1, arg0, param1, param2]
ERROR: com.bangkok.exception.CommonExceptionAdvice - {exception=org.mybatis.spring.MyBatisSystemException: nested exception is
org.apache.ibatis.binding.BindingException: Parameter 'airlineNo' not found. Available parameters are [arg1, arg0, param1, param2]}
```

- 해결 방법

- @Param 어노테이션을 통해 Mapper XML 파일이 정확하게 찾을 수 있도록 하여 해결하였습니다.

```
// 항공사 기종 리스트
public List<AirVO> flightList(@Param("pageObject") PageObject pageObject, @Param("airlineNo") Long airlineNo);
```

INFO : jdbc.resultsettable -												
rnum	airplaneid	flightname	airplanepdt	seatcapacity	ecocnt	biscnt	fscnt	departure	arrival	airlineno	scheduleid	routeid
1	A330	KE310	AIRBUS	300	252	36	12	JFK	ICN	1	21	21
2	B777	KE3111	BOEING	300	252	36	12	ICN	JFK	1	236	22
3	A340	KE3124	AIRBUS	300	252	36	12	JFK	ICN	1	64	21
4	A310	KE359	AIRBUS	300	252	36	12	JFK	ICN	1	107	21
5	B747	KE364	BOEING	300	252	36	12	ICN	JFK	1	279	22
6	B757	KE371	BOEING	300	252	36	12	ICN	JFK	1	322	22
7	B727	KE372	BOEING	300	252	36	12	ICN	JFK	1	365	22
8	A380	KE377	AIRBUS	300	252	36	12	JFK	ICN	1	150	21
9	A220	KE379	AIRBUS	300	252	36	12	JFK	ICN	1	193	21
10	B707	KE380	BOEING	300	252	36	12	ICN	JFK	1	408	22



BANGKOK

추후 개발 내용 : 항공 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

개발 모듈	추후 개발 내용	비고
항공권 검색	최저/최고 금액 정렬, 시간대 검색, 항공사 검색	
관리자	- 기종 리스트 검색 기능, 노선 리스트 검색 기능, 스케줄 검색 기능 - 스케줄 일자별 보기 기능	

후기 : 항공 상품 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위
소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 항공 관련 모듈을 맡게 되어 첫 프로젝트 때 보다 더 많은 생각과 구상에 대한 생각을 많이하게 되었습니다.
- 항공 관리자에 대해 참고할 자료가 없어서 관리자 페이지 구상에 어려움이 있었지만 DB가 등록되는 과정의 기준으로 생각하다 보니 관리할 수 있는 페이지가 만들어졌습니다.
- 코딩을 시작하기 전에 미리 구상을 하고 시작하는 것이 시간 절약과 효율적인 코딩을 할 수 있다는 것을 배웠습니다.
- DB와 관련하여 사용하는 테이블이 많아질 때는 어떻게 효율적으로 조인을 하면 되는지에 대해 더 배우게 되었습니다.
- 코딩을 나만 잘하면 되는 것이 아닌 팀 동료의 진행 상황도 고려하여 진행해야 한다는 것을 알았습니다.
- 프로젝트를 진행 함에 있어 PM의 역할이 매우 중요하다는 것을 다시 한번 깨닫게 되었습니다.

05 프로젝트 시연

항공 주문 예약 관리

화면을 이동 합니다.

05 핵심코드 및 오류

항공 주문 예약 관리

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

수하물 등록

탑승객이 입력한 무게가 선택한 수하물의 최대 무게를 초과 할 때의 로직

모든 수하물 탑입을 초과했을 때의 로직

```
if (selectedBaggage) {
    let weight = parseFloat(this.value); // float로 변환
    if (weight > selectedBaggage.maxWeight) { // 탑승객이 입력한 무게가 선택한 수하물의 최대 무게를 초과하는 경우
        let newBaggage = baggageTypes.find(item => item.maxWeight >= weight);
        if (newBaggage) {
            select.value = '0'; // Reset to "선택하세요"
            descriptionDiv.textContent = '';
            additionalCostDiv.textContent = '규정된 무게를 초과했습니다. 적절한 수하물 종류를 선택해 주세요.';
            weightInput.value = '';
        } else { // 모든 수하물 탑입에 맞지 않으면
            select.value = '0'; // "선택하세요" option으로 바꾼다
            descriptionDiv.textContent = '';
            additionalCostDiv.textContent = '모든 수하물 종류의 무게를 초과했습니다. 적절한 수하물 종류를 선택해 주세요.';
        }
    } else {
        additionalCostDiv.textContent = ''; // 가격 입력창 비우기
    }
}
```

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

탑승객 등록

```
<h2>탑승자 정보 입력</h2>
<div id="passenger-container">
    <c:forEach var="i" begin="1" end="${param.aPassenger}">
        <div class="passenger-item">
            <h3>성인 ${i}</h3>
            <div class="form-group">
                <label for="Last-name-${i}">성:</label> <input type="text"
                    id="Last-name-${i}-a" placeholder="예: 홍"
                    name="apassengers[${i-1}].lastName">
            </div>
            <div class="form-group">
                <label for="first-name-${i}">이름:</label> <input type="text"
                    id="first-name-${i}-a" placeholder="예: 길동"
                    name="apassengers[${i-1}].firstName">
            </div>
        </div>
    </c:forEach>
</div>
```

입력한 사용자 수 만큼 동적으로 폼을 추가 시키기 위해 변수 i를 사용하고 end 포인트를 입력받은 탑승 객 수로 지정함

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

리스트 - 삭제품

```
// 환불 버튼 클릭 시 상세보기 이동 방지 및 모달에 예약 번호 전달
$(document).on('click', '.button.refund', function(event) {
    event.stopPropagation(); // 이벤트 전파 중지
    const card = $(this).closest('.reservation-card');
    const reservationNo = card.find('.reservationNo').val(); // 예약 번호 가져오기
    $('#modalReservationNo').val(reservationNo); // 모달에 예약 번호 설정
});
```

toggle이벤트 중첩 되는 사항이 있어 환불 버튼을 누르면 자연스레 상세보기로 넘어 가게하는 현상 방지 위해 이벤트 중지시킴

모달에 예약번호 전달

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

좌석 관리

```
// 탑승객과 번호를 결합한 고유 키 생성 (예: "adult_1", "child_1")
let passengerKey = `${selectedPassengerType}_${selectedPassengerNo}`

// 이미 좌석을 선택한 승객인지 확인
if (selectedSeats[passengerKey]) {
    alert("이미 이 승객을 위한 좌석이 선택되었습니다. 좌석을 취소한 후 다른 좌석을 선택하세요.");
    return;
}

// $(this).hasClass("open")) {
//     seatCnt++;
//     $(this).removeClass("open").addClass("choose");

//     let seatInput = `<input type="hidden" name="seats[\${seatCnt - 1}].seatID" value="\${originalSeatID}">
//     <input type="hidden" name="seats[\${seatCnt - 1}].seatNo" value="\${seatNum}">
//     <input type="hidden" name="seats[\${seatCnt - 1}].passengerNo" value="\${selectedPassengerNo}">
//     <input type="hidden" name="seats[\${seatCnt - 1}].passengerType" value="\${selectedPassengerType}">
//     <input type="hidden" name="reservations[\${seatCnt - 1}].seatNo" value="\${seatNum}">`;
```

탑승객 탑승(성인, 소아, 유아)과 탑승객 번호를 조합하기 위해 js에서 passengerKey 생성

open 클래스가 있는 항목(좌석)들을 선택하면 choose 클래스로 대체되고 js에서 처리한 요소들을 seatInput 변수를 만들어서 HTML에 append함

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

탑승객이 입력한 좌석 등급에 맞춰 입력한 좌석등급 외엔 예약 불가 하도록 block처리

```
<div class="row justify-content-center">
    <c:forEach var="seat" items="${seats}" varStatus="status">
        <c:if test="#{seat.classNo == 3}">
            <div
                class="seat-box ${seat.classNo == reservation.classNo && seat.status == 'available' ? 'open' : 'blocked'}"
                data-seatNum="#{seat.seatNo}">${seat.seatID}</div>
            <c:if test="#{(status.index + 1) % 4 == 0}">
                <div style="flex-basis: 20px;"></div>
            </c:if>
            <c:if test="#{(status.index + 1) % 8 == 0}">
                <div style="flex-basis: 100%; height: 0;"></div>
            </c:if>
        </c:if>
    </c:forEach>
</div>
```

줄바꿈 처리

01 프로젝트 개요
주제와 목적

02 개발 전반
요구 사항
개발 내용과 범위
소요 자원

03 Front-end
메뉴의 흐름
화면 구현

04 Back-end
DFD
DB

05 프로젝트 시연
시연
핵심코드
오류수정
추후 개발 내용
후기

좌석 관리

```
INFO : com.bangkok.airReservation.controller.SeatController - 전달된 좌석: SeatVO(seatNo=null, seatRow=null, seatID=null, seatCol=null, clas
sNo=null, scheduleID=null, status=null, passengerNo=null, seats=[SeatVO(seatNo=null, seatRow=null, seatID=, seatCol=null, classNo=null, sc
heduleID=null, status=null, passengerNo=null, seats=null)])  

INFO : com.bangkok.airReservation.controller.SeatController - 전달된 예약 정보: SeatReservationVO(seatReservationNo=null, reservationNo=n
ull, seatNo=null, seatReserve=null)  

INFO : com.bangkok.aop.LogAdvice - + ***** [ AOP 실행 전 로그 출력 ]***** +  

INFO : com.bangkok.aop.LogAdvice - + 실행 메서드 : Integer
```

```
com.bangkok.airReservation.service.SeatService.reservationSeat(List, List)  

INFO : com.bangkok.aop.LogAdvice - + 전달 데이터 : [[SeatVO(seatNo=null, seatRow=null, seatID=, seatCol=null, classNo=null, scheduleID=
null, status=null, passengerNo=null, seats=null)], null]
```

```
ERROR: com.bangkok.exception.CommonExceptionAdvice - Exception-----nested exception is org.apache.ibatis.binding.Bindi
ngException: Parameter 'seats' not found. Available parameters are [collection, list]  

ERROR: com.bangkok.exception.CommonExceptionAdvice - {exception=org.mybatis.spring.MyBatisSystemException: nested exception is org.a
pache.ibatis.binding.BindingException: Parameter 'seats' not found. Available parameters are [collection, list]}
```

jsp에서 전달받은 SeatVO객체가 자꾸 null로 들어오는 현상 발생
이유 : js에서 쓴 el객체 앞에 '\' 빼먹음
-> \' 붙여 제대로 전송되게 해결

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

좌석 관리

js에서 보내주는 seats[] 배열의 인덱스 값이 비어있다 하는 현상
 원인: \누락과, seatCnt가 1로 지정되어 있음
 -> \${seatCnt - 1} 로 인덱스 값을 지정

```
let seatInput = `<input type="hidden" name="seats[${seatCnt}].seatID" value="${originalSeatID}">
<input type="hidden" name="seats[${seatCnt}].seatNo" value="${seatNum}">
<input type="hidden" name="seats[${seatCnt}].passengerNo" value="${selectedPassengerNo}">`;
```

ERROR: com.bangkok.exception.CommonExceptionAdvice - Exception-----Invalid property 'seats[]' of bean class [com.bangkok.airReservation.vo.SeatVO]: Invalid index in property path 'seats[]'; nested exception is java.lang.NumberFormatException: For input string: "" ERROR: com.bangkok.exception.CommonExceptionAdvice - {exception=org.springframework.beans.InvalidPropertyException: Invalid property 'seats[]' of bean class [com.bangkok.airReservation.vo.SeatVO]: Invalid index in property path 'seats[]'; nested exception is java.lang.NumberFormatException: For input string: ""}



BANGKOK

오류 수정 : 항공 주문 예약 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

좌석 관리 - 오류코드

jsp에서 seat.classNo, reservation.classNo 태입이 달라 비교를 할 수 없다는 에러
-> Controller에서 받아온 reservation이 List형식이라 발생한 오류라 단일 객체 airReservationVO에 담아 고침

SEVERE: 경로 [/]의 컨텍스트 내의 서블릿 [appServlet]을(를) 위한 Servlet.service() 호출이, 근본 원인(root cause)과 함께, 예외 [행 [266]에서 [WEB-INF/views/airReservation/seatDList.jsp]을(를) 처리하는 중 예외 발생

```
263: <div class="row justify-content-center">
264:     <c:forEach var="seat" items="${seats}" varStatus="status">
265:         <c:if test="${seat.classNo != null && seat.classNo == 3}"> 266:
266:             <div class="seat-box ${seat.classNo != null && seat.classNo == reservation.classNo && seat.
267: status == 'available' ? 'open' : 'blocked'}"
267:                 data-seatNum="${seat.seatNo}">${seat.seatID}</div>
268:                 <c:if test="${(status.index + 1) % 4 == 0}">
269:                     <div style="flex-basis: 20px;"></div> Stacktrace:]을(를) 발생시켰습니다.
java.lang.NumberFormatException: For input string: "classNo"      at java.base/java.lang.NumberFormatException.forInputString
(NumberFormatException.java:65)
```

추후 개발 내용 : 항공 주문 예약 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

개발 모듈	추후 개발 내용	비고
항공 주문 관리	PL/SQL로 처리 하지 않고 관리자가 직접 좌석 여러 개 등록할 수 있는 기능 만들기	
항공 주문 관리	사용자 기능제한 (환불규정, 수정규정 등) 추가 해서 좀 더 상세하게 개발하기	
항공 주문 관리	API 다양하게 사용해서 질 높은 개발 해보기	

후기 : 항공 주문 예약 관리

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end 메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

1. 이번 프로젝트 개발을 진행하면서 테이블 구조가 많이 바뀌어서 정신이 없었는데 처음 설계할 때 많은 시간과 생각을 투자했음에도 처음 짰던 구조와 많이 바뀐 걸 보면 좀 더 정진 해야겠다는 생각이 들었다
2. 개발하면서 점점 생각의 굴레에 빠지며 이게 맞나? 라고 생각했고 실제로 나중에 다시 보니 코드를 왜 이렇게 짰을까 싶었던 점들이 많았다 아직 경험도 그렇고 지식도 많이 부족하단걸 깨닫았다

05 프로젝트 시연

여행 정보 게시판

화면을 이동 합니다.

05 핵심코드 및 오류

여행 정보 게시판

핵심 코드 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

```
<script type="text/javascript">
$(document).ready(function () {
    // 장소의 좌표 설정
    var placePoints = {
        1: {lat: 34.665790, lng: 135.432312}, // 유니버설 스튜디오 좌표
        2: {lat: 34.65461349487305, lng: 135.42892456054688}, // 가
        3: {lat: 34.0792, lng: -118.402}, // 베버리 힐즈 좌표
        4: {lat: 40.12150192260742, lng: -100.45039367675781}, // 몬태
        5: {lat: 37.5759, lng: 126.9768} // 경복궁 좌표
    };
    var placeName = {
        1: "Universal Studios Japan", // 장소 이름
        2: "가이유칸 수족관",
        3: "베버리 힐즈",
        4: "Mount Fortress",
        5: "경복궁"
    };

    // 지도 초기화 함수
    function initMap(location, elementId, title) {
        let map = new google.maps.Map(document.getElementById(elementId), {
            zoom: 14, // 줌 레벨
            center: location // 중심 좌표 설정
        });

        new google.maps.Marker({
            position: location, // 마커 위치
            map: map, // 마커가 표시될 지도
            title: title // 마커 제목
        });
    }
});
```



아시아 최초의 유니버설 스튜디오인 오사카 유니버설 스튜디오 재팬(USJ)은 디즈니랜드와 쌍벽을 이루는 세계적인 테마 파크입니다. 거장 스티븐 스필버그 감독이 고문을 맡아 할리우드의 유명 영화들을 테마로 제작한 어트랙션과 쇼들은 짜릿하고 흥미진진한 볼거리로 인기가 높은데요. 우메다역에서 11분, JR난바역에서 15분, 간사이국제공항에서 65분 소요되며, 유니버설 시티역에서 하차하시면 됩니다. 시즌 및 입장권 옵션 선택에 따라 입장료 상이 / 7,819 ~ 9,455JPY / 어린이(4~11세) : 5,091JPY ~ 6,182JPY / 65세 이상 : 7,000JPY ~ 8,546JPY

Map api 사용으로 클라이언트를 위한 위치 정보 출력

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

파일 업로드 시 유효성 검사 & 이미지 추가 관련 로직

```
// file upload 유효성 검사
if (mainFile.getSize() > 5 * 1024 * 1024) { // 5MB 이상
    rttr.addFlashAttribute("msg", "파일 크기는 5MB 이하이어야 합니다.");
    return "redirect:writeForm.do";
}
```

```
List<InfoImageVO> imageList = null; // 이미지 목록 초기화
// 추가 이미지 파일 처리
if(sub_image != null && sub_image.size() > 0) { // 추가 이미지가 있는 경우
    for(MultipartFile file : sub_image) {
        if(imageList == null) imageList = new ArrayList<>(); // 이미지 목록 초기화
        fileName = file.getOriginalFilename(); // 파일 이름 가져오기
        // 파일이 비어있지 않은 경우 처리
        if(mainFile != null && !mainFile.equals("")) {
            InfoImageVO imageVO = new InfoImageVO(); // 이미지 VO 생성
            imageVO.setFilename(FileUtil.upload(path, file, request)); // 추가 이미지 업로드
            imageList.add(imageVO); // 이미지 목록에 추가
        }
    }
} // DB에 정보와 이미지 목록 저장
service.write(vo, imageList);
```



BANGKOK

오류 수정 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

```

`ceptor - ----- [ 권한 처리 Interceptor ] -----
`ceptor - ----- [ 권한 처리 Interceptor ] -----
`ceptor - ----- [ 권한 처리 Interceptor ] -----
`e - Exception-----No primary or default constructor found for interface java.util.List
`e - {exception=java.lang.IllegalStateException: No primary or default constructor found for interface java.util.List}

```

03 Front-end

메뉴의 흐름

화면 구현

인터페이스 list에 대한 직접적인 객체 생성 오류

04 Back-end

DFD

DB

```

// 정보 작성은 처리하는 메서드
@PostMapping("/write.do")
public String write(InfoVO vo,
    @RequestParam("mainFile") MultipartFile mainFile, // 대표사진
    ArrayList<MultipartFile> sub_image, // 추가사진
    HttpServletRequest request,

```

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

```

// 페이지 번호를 받아올 수 있도록 Long 타입으로 선언.
@RequestParam(name = "perPageNum", defaultValue = "10") Long perPageNum,
RedirectAttributes rttr) throws Exception {

```

```
List<CityVO> cityList = cityService.cityList();
```

직접 파라미터로 받지 않고 서비스 계층을 통해 불러와야 함



BANGKOK

오류 수정 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

```
; codes [typeMismatch.infoVO.imageFile,typeMismatch.java.lang.String,typeMismatch]; arguments [org.s...
```

03 Front-end

메뉴의 흐름

화면 구현

등록 처리 시 multipartFile과 String 변수의 타입 불일치

04 Back-end

DFD

DB

MultipartFile로 받아 서버에 저장한 뒤, 해당 파일 경로를 InfoVO 객체
의 String 타입 imageFile 필드에 저장하는 방식으로 처리

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기



BANGKOK

오류 수정 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

도시 리스트를 가져올 수 없는 경우

여행정보 등록

카테고리 선택 오락 명소

도시 선택

제목

제목은 3자 이상 100자 이내로 입력해주세요.

Ajax 처리를 해서 정보들을 불러오기

```
// 카테고리 가져오기 AJAX 요청
$("#cate_code1").load("ajax/getcateList.do?cate_code1=" + $("#cate_code1").val());

// 도시 가져오기 AJAX 요청
$("#cityNum").load("ajax/cityList.do?cityNum=" + $("#cityNum").val());
```

```
----- 도시 가져오기 -----
@GetMapping("/city.do")
public String getCityList(Model model, @RequestParam Integer cate_code1) {
    Log.info("getCityList.do with cate_code1: " + cate_code1);
    // Fetch the city list based on the category code
    List<CityVO> cityList = cityService.cityList(); // cate_code1을 매개변수로 전달
    model.addAttribute("cityList", cityList);
    // Return the view name
    return "cityPlace/cityList";
}
```



BANGKOK

오류 수정 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

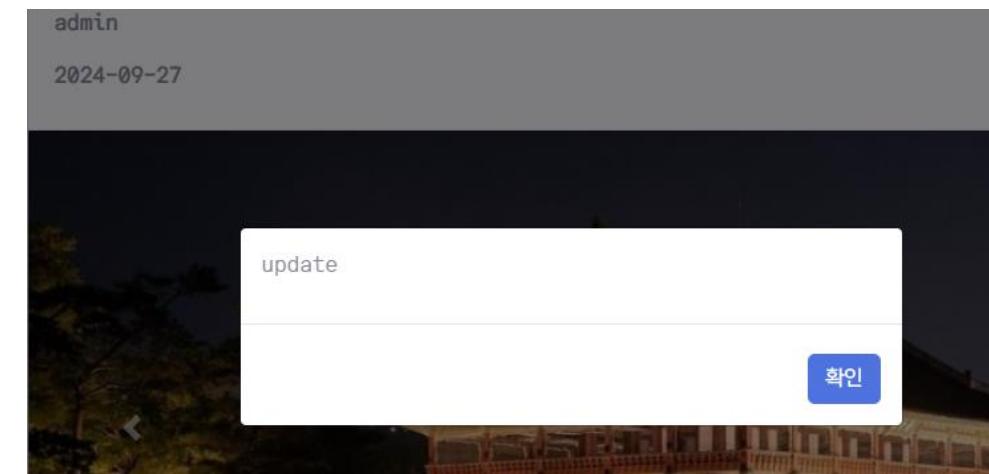
오류수정

추후 개발 내용

후기

Update 처리가 안되는 경우

```
### The error occurred while setting parameters
### SQL: UPDATE tripInfo      SET title = ?,           content = ?,           writer = ?,           place = ?
### Cause: java.sql.SQLSyntaxErrorException: ORA-01747: invalid user.table.column, table.column, or column specification
; bad SQL grammar []; nested exception is java.sql.SQLSyntaxErrorException: ORA-01747: invalid user.table.column, table.col
```



구문 오류라 ','를 제거하면 정상 처리

추후 개발 내용 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

개발 모듈

추후 개발 내용

비고

03 Front-end

메뉴의 흐름

화면 구현

여행 정보 게시판

지도 API를 DB에서 가져와 처리 (현재는 시간 상 하드코딩)

04 Back-end

DFD

DB

여행 정보 게시판

등록폼에서 스마트 에디터 API 적용

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

후기 : 여행 정보 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

- 여행정보 게시판이 이미지 게시판을 차용하는 방식이라 쉬울 줄 알았지만, 의외로 예상 밖의 변수 상황들이 많아 힘들었습니다.
- 그렇지만 어려운 부분에 대해서 질의하며, 코드의 흐름에 대해 다시 공부하고 이해하는 계기가 되었습니다.
- 특히나 API 사용과 데이터 전달, jquery 처리에서 미흡한 점이 보여 이 부분에 대해 될 때까지 연습하여 제 자신을 좀 더 정진시키도록 하겠습니다.
- 도와주신 모든 분들께 죄송하고 감사합니다.

05 프로젝트 시연

여행 정보 게시판 댓글 / 여행 계획 짜기 게시판

화면을 이동 합니다.

05 핵심코드 및 오류

여행 정보 게시판 댓글

핵심 코드 : 여행 정보 게시판 댓글

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

2. 한 개의 모달, n 개의 처리

```
// 수정 품 출력
$(".replyUpdateBtn").click(function(){
    $("#replyFormModal").find(".modal-title").text("댓글 수정");

    $("#replyFormModal").find(".rno")
        .attr("value", $(this).closest(".media-body").find(".card .rno").val());

    $("#replyFormModal").find(".form-control")
        .attr("value", $(this).closest(".media-body").find(".card-body").text());
});

$("#replyFormModal").find("form").attr("action", "/tripReplyAjax/update.do");

$("#replyFormModal").modal("show");
});

// 삭제 품 출력
$(".replyDeleteBtn").click(function(){
    $("#replyFormModal").find(".modal-title").text("댓글 삭제");

    $("#replyFormModal").find(".rno")
        .attr("value", $(this).closest(".media-body").find(".card .rno").val());
    $("#replyFormModal").find(".email")
        .attr("value", $(this).closest(".media-body").find(".userInfo .email").text());

    $("#replyFormModal").find(".card").remove();
    $("#replyFormModal").find(".modal-body").text("댓글을 삭제하시면 복원할 수 없습니다. 댓글을 삭제하시겠습니까?");

    $("#replyFormModal").find(".replyBtn").removeClass("btn-dark")
        .addClass("btn-danger").text("삭제");

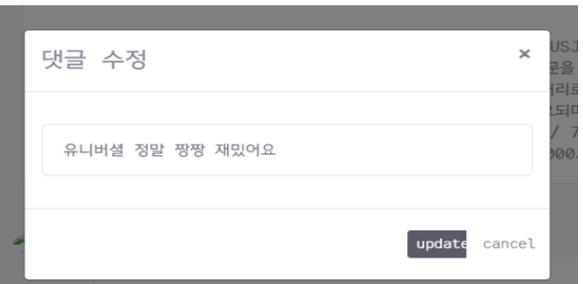
    $("#replyFormModal").find("form").attr("action", "/tripReplyAjax/delete.do");
});

$("#replyFormModal").modal("show");
});
```

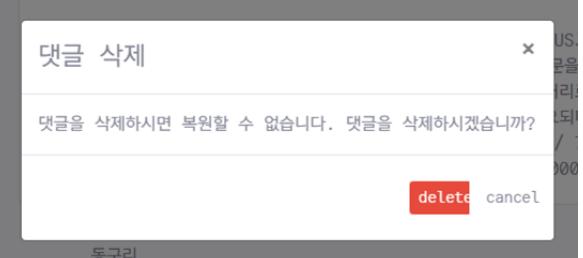


```
<div class="modal" id="replyFormModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <!-- Modal Header -->
            <div class="modal-header">
                <h4 class="modal-title">Modal Heading</h4>
                <button type="button" class="close" data-dismiss="modal">&times;</button>
            </div>
            <!-- Modal body -->
            <div class="modal-body">
                <form method="post">
                    <input type="hidden" name="no" class="no" value="${param.no }">
                    <input type="hidden" name="rno" class="rno">
                    <input type="hidden" name="email" class="email" value="${login.email }">
                    <div class="card form-group p-0">
                        <input class="card-body form-control border-0" name="r_content" style="height: 56px;">
                    </div>
                </form>
            </div>
            <!-- Modal footer -->
            <div class="modal-footer">
                <button type="button" class="btn btn-primary" data-dismiss="modal">Close</button>
            </div>
        </div>
    </div>

```



```
<div class="modal show" id="replyFormModal" style="display: block; padding-right: 17px;" aria-modal="true" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <!-- Modal Header -->
            <div class="modal-header"></div>
            <!-- Modal body -->
            <div class="modal-body">유니버설 정말 펑펑 재밌어요</div>
            <!-- Modal footer -->
            <div class="modal-footer"><button type="button" class="btn btn-primary" data-dismiss="modal">update</button> <button type="button" class="btn btn-secondary">cancel</button></div>
        </div>
    </div>
```



```
<div class="modal show" id="replyFormModal" style="display: block; padding-right: 17px;" aria-modal="true" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <!-- Modal Header -->
            <div class="modal-header"></div>
            <!-- Modal body -->
            <div class="modal-body">댓글을 삭제하시면 복원할 수 없습니다. 댓글을 삭제하시겠습니까?</div>
            <!-- Modal footer -->
            <div class="modal-footer"><button type="button" class="btn btn-primary" data-dismiss="modal">delete</button> <button type="button" class="btn btn-secondary">cancel</button></div>
        </div>
    </div>
```

05 핵심코드 및 오류

여행 계획 짜기 게시판



핵심 코드 : 여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 히를

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시여

해설코드

○ 21

초등 개발 내용

97

일차 / 날짜별 버튼

```
// 여행 등록 번호 가져오기
window.tno = $(".tripViewDiv").find(".tno").val();
// alert('tno : ' + tno);

// 시작일, 종료일 가져오기
window.startDate = $(".tripContent .tripStartDateBlock").text().replace(/[\s\t]+/g, '');
// alert('startDate : ' + startDate + ', 시작Date 타입: ' + typeof startDate);
window.endDate = $(".tripContent .tripEndDateBlock").text().replace(/[\s\t]+/g, '');
// alert('endDate : ' + endDate + ', endDate 타입: ' + typeof endDate);
```

```
// 날짜 배열 초기화
let dateArray = [];

// 날짜를 순회하여 배열에 추가
for (let date = startDateObj; date <= endDateObj; date.setDate(date.getDate() + 1)) {
    // 각 날짜를 'YYYY-MM-DD' 형식으로 포맷팅
    let formattedDate = date.toISOString().split('T')[0];
    dateArray.push(formattedDate);
}

// alert(dateArray); // 결과: 날짜 배열 출력
// 날짜 배열을 반복하며 버튼 생성
dateArray.forEach((date, index) => {
```

```
let planDateBtn = '';
planDateBtn += '<div class="btn-group planDateBtn">';
planDateBtn += '<button type="button" class="btn btn-outline-dark tripDateBtn mr-2 "';
planDateBtn += 'style="border-radius: 30px;">' + (index + 1) + '일자</button>';
planDateBtn += '</div>';

$(".planDateDiv").append(planDateBtn);
```

```
// 날짜별 버튼
let planListBtn = '';
planListBtn += '<div class="planListDiv">';
planListBtn += '<button type="button" class="btn planListBtn my-1"' +
    + 'data-tripdate="' + date + '" style="width: 100%;">';
planListBtn += '<h4 class="card-title text-left tripDate mb-0">';
planListBtn += date;
planListBtn += '<span class="float-right"><i class="fa fa-angle-down"></i></span>';
planListBtn += '</h4>';
planListBtn += '</button>;
```

여행 시작일과 여행 종료일 사이의 날짜들을 구하고 이를 배열에 저장

배열에 저장된 날짜를 하나씩 가져와, 일차별 / 날짜별 버튼을 생성

9월 미국 여행

2024-09-25 → 2024-09-30 미국, 로스엔젤리스

어떤 여행인가요? (선택사항)

1일차 2일차 3일차 4일차 5일차 6일차

2024-09-25 ▾

1

+ 장소 추가

2024-09-26 ▾

1

+ 장소 추가

2024-09-27 ▾

1

+ 장소 추가

오류 수정 : 여행 계획 짜기 게시판

01	프로젝트 개요
	주제와 목적
02	개발 전반
	요구 사항
	개발 내용과 범위
	소요 자원
03	Front-end 메뉴의 흐름
	화면 구현
04	Back-end
	DFD
	DB
05	프로젝트 시연
	시연
	핵심코드
	오류수정
	추후 개발 내용
	후기

모달 출력 오류

```

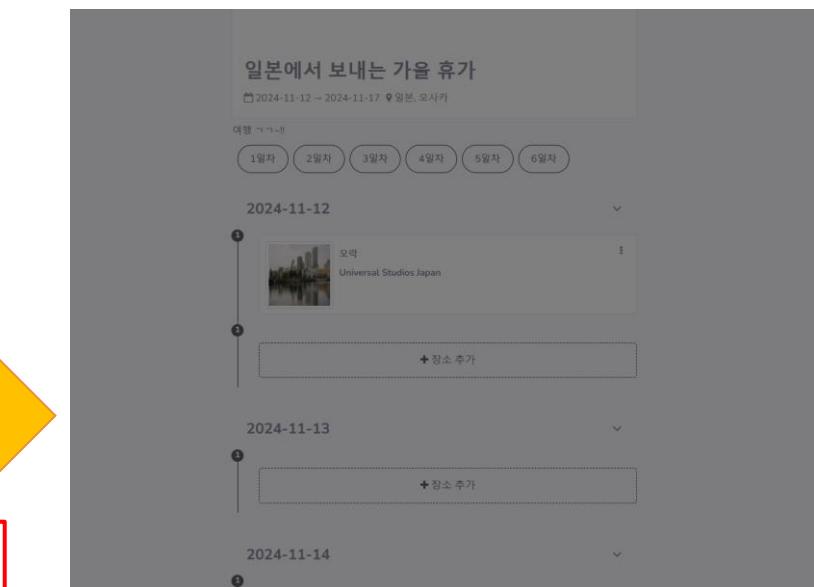
$(".tripPlanDiv").on("click", ".placeInfo", function(){
    console.log("일정 클릭");
    // 여행지 방문 번호(pno) 수집
    let pno = $(this).find(".pno").val();
    console.log("pno : " + pno);

    $(".planModalsDiv").load("/planAjax/getPlanView.do?tno=" + tno + "&pno=" + pno)
    // 여행 시간 출력 형태
    $("#planWriteModal").find(".tripTimeInput").css("display", "block");
    $("#planWriteModal").find(".tripTimeBlock").css("display", "none");

    $("#planWriteModal").find(".modal-footer .modalMultiBtn").hide();
    $("#planWriteModal").modal("show"); // 모달 표시
});

});
```

placeInfo 클래스를 클릭하면 로드는 되지만 모달이 출력되지 않음



```

<div class="modal" id="planWriteModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <!-- Modal Header -->
      <div class="modal-header"></div>
      <!-- Modal body -->
      <div class="modal-body">
        <!-- 여행지 사진 -->
        <div class="form-group placeImage"></div>
        <div class="form-group tripPeriodForm"></div>
        <div class="form-group tripPlaceForm" style="display: none;"></div>
        <div class="form-group tripTimeForm"></div>
        <div class="form-group"></div>
      </div>
      <!-- Modal footer -->
      <div class="modal-footer">
        <button class="btn btn-dark modalUpdateBtn">Update</button>
        <button class="btn btn-dark modalMultiBtn">Go</button>
        <button type="button" class="btn" data-dismiss="modal">cancel</button>
      </div>
    </div>
  </div>
</div>
```

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

모달 출력 오류 해결 방법

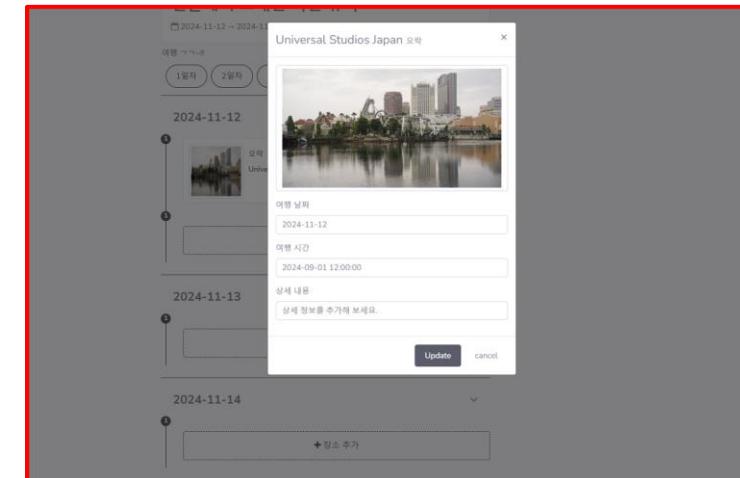
```
// 모달 출력 오류 코드 해결
$(".tripPlanDiv").on("click", ".placeInfo", function(){
    console.log("일정 클릭");
    // 여행지 방문 번호(pno) 수집
    let pno = $(this).find(".pno").val();
    console.log("pno : " + pno);

    $(".planModalsDiv").load("/planAjax/getPlanView.do?tno=" + tno + "&pno=" + pno,
        function(){
            // 여행 시간 출력 형태
            $("#planWriteModal").find(".tripTimeInput").css("display", "block");
            $("#planWriteModal").find(".tripTimeBlock").css("display", "none");

            $("#planWriteModal").find(".modal-footer .modalMultiBtn").hide();
            $("#planWriteModal").modal("show"); // 모달 표시
        });
});
```

로드하려는 url 뒤에, 로드하고 실행시킬 콜백함수
추가

```
<div class="planModalsDiv">
  <div class="modal show" id="planWriteModal" aria-modal="true" role="dialog" style="padding-right: 17px; display: block;">
    <div class="modal-dialog">
      <div class="modal-content">
        <!-- Modal Header -->
        <div class="modal-header"></div>
        <!-- Modal body -->
        <div method="post" action="/planAjax/planWrite.do">
          <input name="tno" class="tno" type="hidden">
          <input value="51" name="pno" class="pno" type="hidden">
        <!-- Modal footer -->
        <div class="modal-footer"></div>
      </div>
    </div>
  </div>
</div>
```





BANGKOK

추후 개발 내용 : 여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

개발 모듈

추후 개발 내용

비고

03 Front-end

메뉴의 흐름

화면 구현

여행 계획 짜기

구글 맵 API 로드

04 Back-end

DFD

DB

여행 계획 짜기

방문지 리스트에 있는 장소 마커 표시

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기



후기 : 여행 정보 게시판 댓글 / 여행 계획 짜기 게시판

01 프로젝트 개요

주제와 목적

02 개발 전반

요구 사항

개발 내용과 범위

소요 자원

1. 개인적인 개발 역량 기르기

2. 필요한 데이터

03 Front-end

메뉴의 흐름

화면 구현

04 Back-end

DFD

DB

05 프로젝트 시연

시연

핵심코드

오류수정

추후 개발 내용

후기

감사합니다.

Thank you for watching