

PORTFOLIO

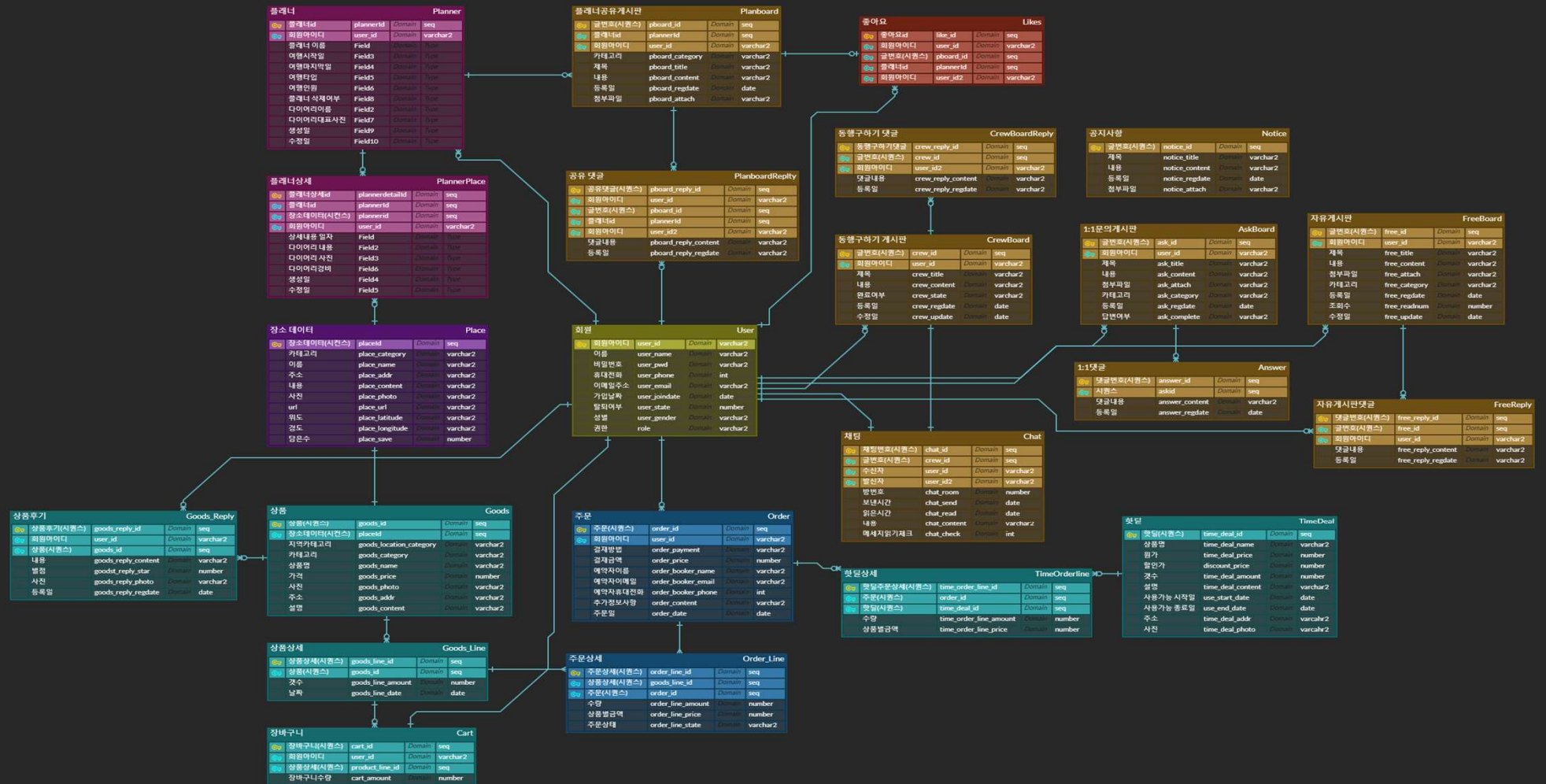
Spring boots & JPA Project

박민서

개요 & 개발환경

구분	내용
프로젝트명	제주도 여행 플래너 기능 제공 및 여행 상품판매 웹사이트
개발기간/참여인원	2022.05.07 ~ 2022.06.20(45일) / 5명
개발환경	사용언어 : JAVA Programming, JDBC, Oracle11g, JavaScript, jQuery, Spring boots & JPA Tool : Eclipse , SQL_Developer, My Batis
역할	DB 설계 및 게시판 CRUD 작업
프로젝트 개요	제주도 여행 계획을 간편하게 세울 수 있는 플래너 기능을 제공함으로써 기존 여행 상품 판매 사이트와 차별점을 둔다. 사용자가 플래너를 작성하여 저장한 장소와 숙소 데이터를 기반으로 각 사용자별 맞춤 여행 상품을 추천해 주며 상품의 구매율을 높이고 이용자의 만족도를 높인다.

DB설계



나의 역할부분

컨트롤러 & 서비스호출

1:1문의 상세보기

```
/**글 등록폼*/
@RequestMapping("/board/AskBoard")
public void write() {
    System.out.println("askboard test");
}
```

1:1 문의 등록

```
/**글 등록*/
@RequestMapping(value="/board/Askinsert", method = RequestMethod.POST)
public String insert(AskBoard askBoard, HttpSession session) {

    Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    askBoard.setUser(users);

    System.out.println("insert test");

    String uploadpath = session.getServletContext().getRealPath("/WEB-INF/") + "upload/ask/";
    askBoardService.addAskBoard(askBoard, uploadpath);

    return "redirect:/mypage/myask";
}
```

```
/**마이페이지에서 내가 쓴 글 목록 조회*/
@RequestMapping("/mypage/myask")
public String mylist(Model model, @RequestParam(defaultValue="1") int nowPage) {

    Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();

    DateTimeFormatter format = DateTimeFormatter.ofPattern("yyyy년 MM월 dd일");

    Pageable pageable = PageRequest.of( (nowPage-1), PAGE_COUNT, Direction.DESC, "askId");

    Page<AskBoard> myList = askBoardService.selectByUserId(users.getUserId(), pageable );

    List<AskBoard> list = myList.getContent();
    List<AskBoardDTO> asklist = new ArrayList<AskBoardDTO>();
    for(AskBoard x : list) {
        asklist.add(new AskBoardDTO (x.getAskId(), x.getUser().getUserId(),
            x.getAskCategory(), x.getAskTitle(), x.getAskContent(),
            x.getAskAttach(), x.getAskComplete(), x.getAskRegdate().format(format)) );
    }

    int temp = (nowPage-1)%BLOCK_COUNT;
    int startPage = nowPage - temp;

    model.addAttribute("asklist", asklist);
    model.addAttribute("users", users);
    model.addAttribute("blockCount", BLOCK_COUNT);
    model.addAttribute("startPage", startPage);
    model.addAttribute("nowPage", nowPage);

    return "mypage/myask";
}
```

마이페이지 - 내가
작성한 1:1문의 목록

```
/**상세보기 - board*/
@RequestMapping("/board/Ask_Detail/{askId}")
public String read(@PathVariable Long askId, Model model) {
    System.out.println("askboardId test");
```

```
    Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();
```

```
    AskBoard askBoard = askBoardService.getAskBoard(askId);
    List<AskReply> replylist = askBoard.getAskReplyList();
```

```
    System.out.println("rrrr");
    System.out.println(replylist.size());
```

```
    model.addAttribute("askboard", askBoard);
    model.addAttribute("replylist", replylist);
```

```
    return "/board/Ask_Detail";
}
```

1:1문의 삭제

```
/**삭제*/
@RequestMapping("/Askdelete")
public String delete(Long askId) {

    System.out.println("delete test1");

    askBoardService.deleteAskBoard(askId);

    System.out.println("delete test2");

    return "redirect:/mypage/myask";
}
```

컨트롤러 & 서비스호출

```
/**전체검색 - admin에서*/
@RequestMapping("/admin/AskList_Admin")
public void askListAdmin(Model model, @RequestParam(defaultValue= "1") int nowPage) {

    DateFormatter format = DateFormatter.ofPattern("yyyy년 MM월 dd일");

    Pageable pageable = PageRequest.of( (nowPage-1), PAGE_COUNT, Direction.DESC, "AskId");
    Page<AskBoard> pageList = askBoardService.getAllAskBoards(pageable);

    List<AskBoard> ask = pageList.getContent();
    List<AskBoardDTO> askList = new ArrayList<AskBoardDTO>();
    for(AskBoard a : ask) {
        askList.add(new AskBoardDTO (a.getAskId(), a.getUser().getId(), a.getAskCategory(), a.getAskTitle(),
            a.getAskContent(), a.getAskAttach(), a.getAskComplete(), a.getAskRegdate().format(format)) );
    }

    int temp = (nowPage-1)%BLOCK_COUNT;
    int startPage = nowPage - temp;

    System.out.println("adminList test");

    model.addAttribute("askList",askList);
    model.addAttribute("blockCount", BLOCK_COUNT);
    model.addAttribute("startPage", startPage);
    model.addAttribute("nowPage", nowPage);
}
```

관리자페이지 – 1:1 문의 전체목록

```
/**상세보기 답변달기 - admin*/
@RequestMapping("/reply/AskReply_Write/{askId}")
public ModelAndView read2(@PathVariable Long askId) {
    System.out.println("askboardId test");

    AskBoard askBoard = askBoardService.getAskBoard(askId);

    return new ModelAndView("reply/AskReply_Write", "askboard", askBoard);
}

/**관리자 상세보기*/
@RequestMapping("/admin/AskDetail_Admin/{askId}")
public String read2(@PathVariable Long askId, Model model) {
    System.out.println("askboardId test");

    AskBoard askBoard = askBoardService.getAskBoard(askId);
    List<AskReply> replylist = askBoard.getAskReplyList();

    System.out.println("rrrr");
    System.out.println(replylist.size());

    model.addAttribute("askboard",askBoard);
    model.addAttribute("replylist",replylist);

    return "/admin/AskDetail_Admin";
}
```

관리자페이지 – 1:1 문의 상세보기
및 답변등록

컨트롤러 & 서비스호출

```
@Controller
@RequiredArgsConstructor
public class AskReplyController {

    private final AskReplyService askReplyService;

    /**댓글 등록*/
    @RequestMapping("/reply/insert")
    public String insert(AskReply askReply , Long askId) {

        askReplyService.addAskReply(askId, askReply);

        return "redirect:/admin/AskDetail_Admin/" + askId;
    }

    /**댓글 삭제*/
    @RequestMapping("/reply/delete/{askReplyId}/{askId}")
    public String delete(@PathVariable Long askReplyId, @PathVariable Long askId) {

        askReplyService.deleteAskReply(askReplyId);

        System.out.println("reply delete");

        return "redirect:/admin/AskDetail_Admin/" + askId;
    }
}
```

관리자페이지 – 관리자가 1:1문의에
답변 등록

관리자페이지 – 관리자만이 1:1문의
답변 삭제할 수 있음

컨트롤러 & 서비스호출

```
/**관리자 삭제하기*/  
@RequestMapping("/admin/delete")  
public String delete2(Long askId) {  
  
    System.out.println("delete test11");  
  
    askBoardService.deleteAskBoard(askId);  
  
    System.out.println("delete test22");  
  
    return "redirect:/admin/AskList_Admin";  
}
```

관리자페이지 – 게시판에 어울리지 않는다면 삭제

```
/**답변 Y or N*/  
@RequestMapping("/admin/detail/complete")  
public ModelAndView complete(Long askId, String btnradio) {  
  
    askBoardService.complete(askId, btnradio);  
    return new ModelAndView("redirect:/admin/AskList_Admin");  
  
}
```

관리자페이지 – 1:1 문의 답변 등록 후 답변 완료 버튼누르기

컨트롤러 & 서비스호출

```
/**전체검색 & 카테고리 & 페이징*/
@RequestMapping("/board/PlanboardList")
public void planList(Model model, @RequestParam(defaultValue="1") int nowPage, String pboardCategory) {

    System.out.println("category?????" + pboardCategory);

    Page<PlanBoard> pageList = planBoardService.selectByCate(pboardCategory, nowPage, PAGE_COUNT);

    int temp = (nowPage-1)%BLOCK_COUNT;
    int startPage = nowPage - temp;

    model.addAttribute("pageList", pageList);
    model.addAttribute("blockCount", BLOCK_COUNT);
    model.addAttribute("startPage", startPage);
    model.addAttribute("nowPage", nowPage);
    model.addAttribute("pboardCategory", pboardCategory);
}
```

플래너게시판 전체조회,
카테고리, 페이징처리

```
/**등록 폼*/
@RequestMapping("/board/Planboard_Write")
public void write() {
    System.out.println("write");
}
```

플래너게시판 등록

```
/**등록하기*/
@RequestMapping(value= "/board/pinsert", method = RequestMethod.POST)
public String insert(PlanBoard planBoard, HttpSession session, Long plannerId) {
    Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    planBoard.setUser(users);

    planBoard.setUserPlan(plannerService.selectBy(plannerId));

    String uploadpath = session.getServletContext().getRealPath("/WEB-INF/") + "upload/planboard/";
    planBoardService.insertPlanBoard(planBoard, uploadpath);

    System.out.println("pboardinsert");

    return "redirect:/board/PlanboardList";
}
```

```
/**수정 폼*/
@RequestMapping("/board/pupdateForm")
public ModelAndView updateForm (Long pboardId) {

    PlanBoard planBoard = planBoardService.selectById(pboardId);

    System.out.println("update");

    return new ModelAndView("/board/Planboard_Update", "planBoard", planBoard);
}

/**수정하기*/
@RequestMapping(value = "/board/pupdate", method = RequestMethod.POST)
public String update(PlanBoard planBoard, HttpSession session, Long plannerId) {
    Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    planBoard.setUser(users);

    planBoard.setUserPlan(plannerService.selectBy(plannerId));

    System.out.println("pboardupdate00");
    String uploadpath = session.getServletContext().getRealPath("/WEB-INF/") + "upload/planboard/";
    PlanBoard dbBoard = planBoardService.updatePlanBoard(planBoard, uploadpath);

    System.out.println("pboardupdate");

    return "redirect:/board/Planboard_Detail/" + dbBoard.getPboardId();
}
```

플래너게시판 수정

```
/**삭제하기*/
@RequestMapping("/board/pdelete")
public String delete(Long pboardId) {

    System.out.println("삭제1");

    planBoardService.deletePlanBoard(pboardId);

    System.out.println("삭제2");

    return "redirect:/board/PlanboardList";
}
```

플래너게시판 삭제

컨트롤러 & 서비스호출

```
/**마이페이지에서 내가 쓴 글 목록 조회*/  
@RequestMapping("/mypage/myplanboard")  
public String myplan(Model model) {
```

```
    Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();  
    List<PlanBoard> myplan = planBoardService.selectByUserId(users.getUserId());
```

```
    model.addAttribute("myplan",myplan);  
    model.addAttribute("users",users);
```

```
    return "mypage/myplanboard";
```

```
}
```

마이페이지에서 사용자가 쓴 플래너게시판 목록

```
/**댓글 등록*/  
@RequestMapping("/reply/planinsert")  
@ResponseBody  
public void insert(String replyContent, String pboardId) {  
    System.out.println("insert 되냐?");  
  
    Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();  
    PlanReply reply = new PlanReply(null, null, users, replyContent, null);  
    planReplyService.insert(Long.valueOf(pboardId), reply);  
    System.out.println("insertreply");
```

```
}
```

```
/**댓글 삭제*/  
@RequestMapping("/reply/delete")  
@ResponseBody  
public void delete(String pboardReplyId) {  
    planReplyService.delete(Long.valueOf(pboardReplyId));  
    System.out.println("reply delete");  
}
```

플래너게시판 댓글 등록
및 삭제

```
/**댓글 조회*/  
@RequestMapping("/reply/Planboard_Detail")  
@ResponseBody  
public List<PlanReplyDTO> planReplyList(String pboardId) {  
    System.out.println("pboardId" + pboardId);  
  
    PlanBoard planboard = planBoardService.selectById(Long.valueOf(pboardId));  
    List<PlanReply> dblist = planboard.getPlanReply();  
    System.out.println(dblist.size());  
  
    List<PlanReplyDTO> list = new ArrayList<PlanReplyDTO>();  
    for(PlanReply x : dblist) {  
        list.add(new PlanReplyDTO(x.getPboardReplyId(), x.getPlanBoard().getPboardId(),  
            x.getUser().getUserId(), x.getPboardReplyContent()));  
    }  
  
    System.out.println("planreply");  
  
    return list;  
}
```

플래너게시판 댓글 조회

컨트롤러 & 서비스호출

```
/**좋아요*/
@RequestMapping("/like")
@ResponseBody
public LikesDTO likes(Long pboardId, String userId) {
    LikesDTO result = planBoardService.saveLikes(pboardId, userId);
    System.out.println(result);
    return result;
}

/**상세보기 + 좋아요*/
@RequestMapping("/board/Planboard_Detail/{pboardId}")
public String read(@PathVariable Long pboardId, Model model) {

    //Users users = (Users)SecurityContextHolder.getContext().getAuthentication().getPrincipal();

    PlanBoard p = planBoardService.selectById(pboardId);
    //boolean isChecked = planBoardService.selectByBoardId(pboardId, users.getUserId());

    //model.addAttribute("isChecked", isChecked);

    //DTO변환
    DateTimeFormatter format = DateTimeFormatter.ofPattern("yyyy년 MM월 dd일");
    PlanBoardDTO planBoard= new PlanBoardDTO(p.getPboardId(), p.getUserPlan().getPlannerId(),p.getUser().getUserId(), p.getUserPlan().getPlannerCount(),
        p.getPboardCategory(), p.getPboardTitle(), p.getPboardContent(), p.getPboardAttach(), p.getPboardRegdate().format(format), p.getLikesCount());

    model.addAttribute("planBoard", planBoard);
    return "/board/Planboard_Detail";
}

@RequestMapping("/checkLike")
@ResponseBody
public int checkLike(Long pboardId, String userId) {
    if(userId==null) {
        return 2; //비회원
    }else {
        boolean isChecked = planBoardService.selectByBoardId(pboardId, userId);
        if(isChecked) {
            return 1; //좋아요 눌렀음
        }else {
            return 0; //좋아요 안눌렀음
        }
    }
}
```

플래너게시판 상세보기 및 좋아요버튼

컨트롤러 & 서비스호출

```
/**전체검색 & 카테고리 & 페이징 - 관리자*/
@RequestMapping("/admin/Planboard_Admin")
public void planListAdmin(Model model, @RequestParam(defaultValue="1") int nowPage, String pboardCategory) {

    System.out.println("category???" + pboardCategory);

    Page<PlanBoard> pagelist = planBoardService.selectByCate(pboardCategory , nowPage, PAGE_COUNT);

    int temp = (nowPage-1)%BLOCK_COUNT;
    int startPage = nowPage - temp;

    model.addAttribute("pagelist", pagelist);
    model.addAttribute("blockCount", BLOCK_COUNT);
    model.addAttribute("startPage", startPage);
    model.addAttribute("nowPage", nowPage);
    model.addAttribute("pboardCategory", pboardCategory);

}
```

관리자페이지 – 플래너게시판 전체조회, 카테고리, 페이징처리

```
/**삭제하기 - Admin*/
@RequestMapping("/admin/deleteAdmin")
public String deleteAdmin(Long pboardId) {

    System.out.println("삭제1");

    planBoardService.deletePlanBoard(pboardId);

    System.out.println("삭제2");

    return "redirect:/admin/Planboard_Admin";

}
```

관리자페이지 – 게시판에 어울리지 않으면 관리자가 삭제가능

흐름도 - 1:1문의 작성 및 상세보기

1:1 문의

순서	카테고리	제목	날짜	답변여부
136	상품문의	상품문의드립니다	2022년 06월 18일	N

123

1:1 문의 작성하기



공지사항을 참고하시면 자주 올라오는 질문과 답변을 확인하실 수 있습니다.

주말 및 공휴일(임시 공휴일, 법정 공휴일 포함)은 영업 휴무일 인 관계로, 다음 날 신속하게 답변 드리도록 하겠습니다.

상당유형

Category

제목

제목을 입력해주세요

내용

내용을 입력해주세요

첨부파일

파일 선택

선택된 파일 없음

문의 등록하기

1:1 문의 상세보기

카테고리

상품문의

제목

상품 문의드립니다

내용

씨워킹 상품을 원했는데 보이지 않네요.....

이 상품은 들어올 계획이 없나요?

목록으로 돌아가기

삭제

답변

흐름도 - 관리자 1:1 답변작성 및 답변완료



관리자 페이지 로그인 상품검색



홈페이지

회원

전체회원조회

1:1 문의

관리자 1:1 문의 목록

순서	카테고리	제목	날짜	답변여부	삭제
137	상품문의	상품 문의드립니다	2022년 06월 18일	N	삭제

1 2 3

관리자 1:1 문의 상세보기

답변완료

카테고리

상품문의

제목

상품 문의드립니다

내용

씨워킹 상품을 원했는데 보이지 않네요.....
이 상품은 들어올 계획이 없나요?

답변

안녕하세요 고객님, 제주잇다는 현지상황에 따라 상품을 제휴하고 있습니다.
죄송하지만 현재 씨워킹 상품은 제휴되고 있지 않아 고객님의 요구를 해결해드리지 못하여 죄송합니다.

삭제하기

관리자 1:1 문의 목록

순서	카테고리	제목	날짜	답변여부	삭제
136	상품문의	상품문의드립니다	2022년 06월 18일	Y	삭제

1 2 3

흐름도 - 1:1문의 답변여부 확인 및 삭제하기

1:1 문의

순서	카테고리	제목	날짜	답변여부
136	상품문의	상품문의드립니다	2022년 06월 18일	Y

작성하기

1 2 3

1:1 문의 상세보기

카테고리

상품문의

제목

상품문의드립니다

내용

씨워킹 상품을 원했는데 보이지 않네요.....
이 상품은 들어올 계획이 없나요?

목록으로 돌아가기

삭제

답변

안녕하세요 고객님, 제주 있다는 현지 상황에 따라 상품을 제휴하고 있습니다.
최송하지만 현재 씨워킹 상품은 제휴되고 있지 않아 고객님께 불편을 드린 점 죄송합니다.

1:1 문의

순서

카테고리

제목

날짜

답변여부

작성하기

1 2 3

관리자 1:1 문의 목록

순서

카테고리

제목

날짜

답변여부

삭제

138

회원문의

ㅅㅅㅅㅅㅅ

2022년 06월 18일

N

삭제

1 2 3

관리자 1:1 문의 목록

순서

카테고리

제목

날짜

답변여부

삭제

1 2 3

흐름도 - 플래너게시판 카테고리별 확인

Board

공지사항

제주잇다의 소식을 확인하세요

플래너 게시판

좋았던 여행을 공유하세요

소통 게시판

정보를 나누고 교류하세요

동행 구하기 게시판

여러 사람과 더 즐거운 여행을 즐기세요

플래너 게시판

나홀로

가족/부모님

친구

연인

나홀로

힐링여행 min****

연인

연인이랑 여행갈때 추천해요! jeo****

친구

친구 5명과의 여행 !!! cha****

작성하기

1

Board

공지사항

제주잇다의 소식을 확인하세요

플래너 게시판

좋았던 여행을 공유하세요

소통 게시판

정보를 나누고 교류하세요

동행 구하기 게시판

여러 사람과 더 즐거운 여행을 즐기세요

플래너 게시판

나홀로

가족/부모님

친구

연인

나홀로

힐링여행 min****

작성하기

1

흐름도 - 플래너게시판 글 작성 및 유효성검사



localhost:8888 내용:
카테고리를 선택해주세요

확인

카테고리
Category

제목
제목을 입력해주세요


플래너
Planner

내용
내용을 입력해주세요

첨부파일
파일 선택 선택된 파일 없음

등록하기

흐름도 - 플래너게시판 상세보기 및 수정 유효성검사



♡ 좋아요 0

내용

오랜만에 혼자서 제주도 다녀왔어요
여유롭게 다니면서 생각정리도 하고 바다도 실컷 보고 왔습니다
처음으로 한라산도 다녀왔는데 다들 왜 추천하는지 알겠더라고요
그리고 제 사랑 중문색달해변은 여전히 이뽕습니다!!!!

목록으로 돌아가기

수정

삭제

localhost:8888 내용:

자신이 작성한 게시물만 수정 가능합니다.

확인

♡ 좋아요 1

내용

여자친구랑 여름휴가로 여행다녀왔습니다
플래너로 편하게 숙소랑 여행장소를 정했고
둘다 모두 만족하는 여행을 다녀왔습니다.

목록으로 돌아가기

수정

삭제

힐링여행 / 2022-06-01

▼


내용

오랜만에 혼자서 제주도 다녀왔어요
여유롭게 다니면서 생각정리도 하고 바다도 실컷 보고 왔습니다
처음으로 한라산도 다녀왔는데 다들 왜 추천하는지 알겠더라고요
그리고 제 사랑 중문색달해변은 여전히 이뽕습니다!!!!

첨부파일

파일 선택

선택된 파일 없음



수정하기

흐름도 - 플래너게시판 삭제 유효성 검사



♡ 좋아요 0

내용

오랜만에 혼자서 제주도 다녀왔어요
여유롭게 다니면서 생각정리도 하고 바다도 실컷 보고 왔습니다
처음으로 한라산도 다녀왔는데 다들 왜 추천하는지 알겠더라고요
그리고 제 사랑 중문색달해변은 여전히 이쁠습니다!!!!

목록으로 돌아가기

수정 삭제

localhost:8888 내용:

자신이 작성한 게시물만 삭제 가능합니다.

확인

♡ 좋아요 1

내용

여자친구랑 여름휴가로 여행다녀왔습니다
플래너로 편하게 숙소랑 여행장소를 정했고
둘다 모두 만족하는 여행을 다녀왔습니다.

목록으로 돌아가기

수정

삭제

흐름도 - 좋아요 기능



♡ 좋아요 0

내용

오랜만에 혼자서 제주도 다녀왔어요
여유롭게 다니면서 생각정리도 하고 바다도 실컷 보고 왔습니다
처음으로 한라산도 다녀왔는데 다들 왜 추천하는지 알겠더라고요
그리고 제 사랑 중문색달해변은 여전히 이뻐했습니다!!!!



♡ 좋아요 1

내용

오랜만에 혼자서 제주도 다녀왔어요
여유롭게 다니면서 생각정리도 하고 바다도 실컷 보고 왔습니다
처음으로 한라산도 다녀왔는데 다들 왜 추천하는지 알겠더라고요
그리고 제 사랑 중문색달해변은 여전히 이뻐했습니다!!!!

흐름도 - 댓글작성 및 삭제

minseo님

저도 중문색달해변을 좋아해요~ 너무 부럽습니다

댓글 등록하기

Comments : 0개

댓글이 없습니다

minseo님

댓글을 입력해주세요

댓글 등록하기

Comments : 1개



minseo님

사진보니까 제주도 가고싶네요~ [삭제](#)

minseo님

댓글을 입력해주세요

댓글 등록하기

Comments : 0개

댓글이 없습니다

감사합니다