

트래블
하우스

목 차

앱 개요

01

팀원 별 역할

02

개발 시 사용한
언어 및 기술

0
3

기능정의서

0
4

TABLE 설계도

05

앱 주요화면

06

핵심기능의
Source Code

07

앱 개요

CONTENTS

프로젝트명

호텔예약시스템

프로젝트 설명

최근에는 가족 단위로 여행을 많이 감으로써 숙박에 대한 수요가 많이 늘어남에 따라 , 호텔 예약 시스템을 개발하게 되었다 .

팀원 및 역할

- 프로젝트 총괄
- 예약추가, 목록, 내용 구현
- 예약취소, 환불목록, 내용 구현

김광수

팀장

- 회원목록, 회원조회 구현
- 로그인 저장, 아이디찾기, 비밀번호찾기 구현
- 문의에서 답변하기 구현
- 페이징 구현

김태연

개발자

- 객실조회, 추가, 수정 구현
- 객실타입목록 조회 구현

이윤정

개발자

- 객실타입 조회, 추가, 수정, 삭제 구현
- 파일첨부 구현

박혜원

개발자

- 후기목록, 조회, 수정, 삭제 구현

박병국

개발자

- 프로젝트 디자인 총괄
- 문의목록, 조회, 수정, 삭제 구현

김이현

디자이너 및 개발자

- 프로젝트 문서작성 총괄
- 후기삭제, 문의삭제 구현

하태창

문서작성자 및 개발자

- 예약취소, 환불조회 구현

김광호

개발자

개발 시 사용한 언어 및 기술

언어



기술



개발환경

Oracle SQL Developer



www.download.zone



Apache
Tomcat



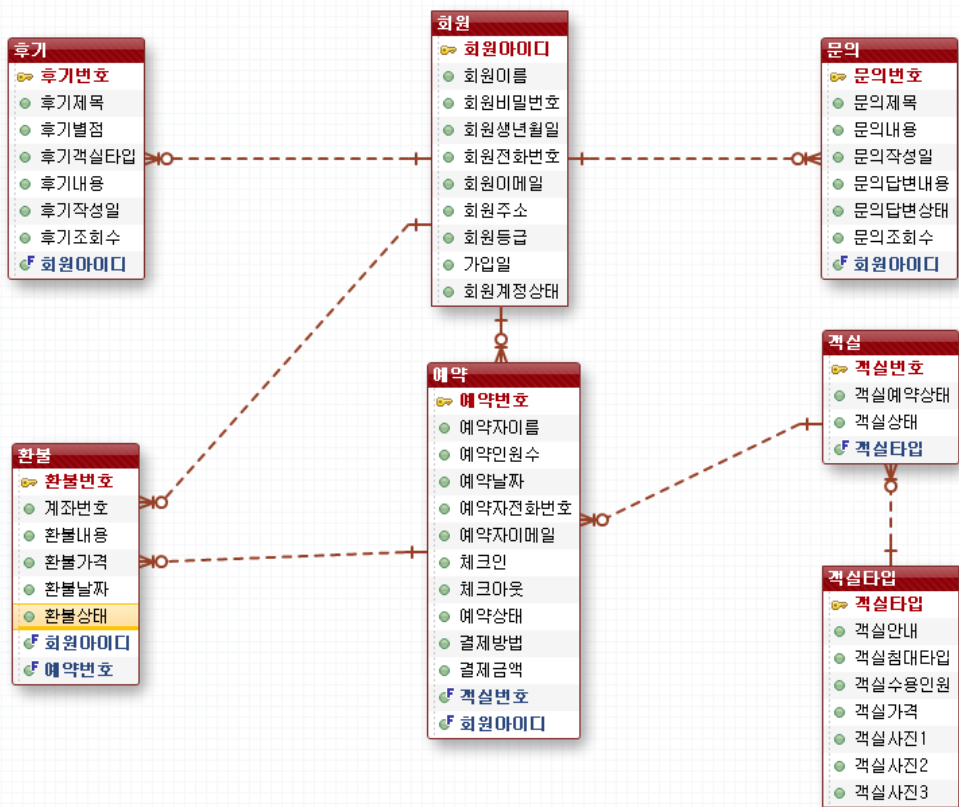
eclipse

기능정의서

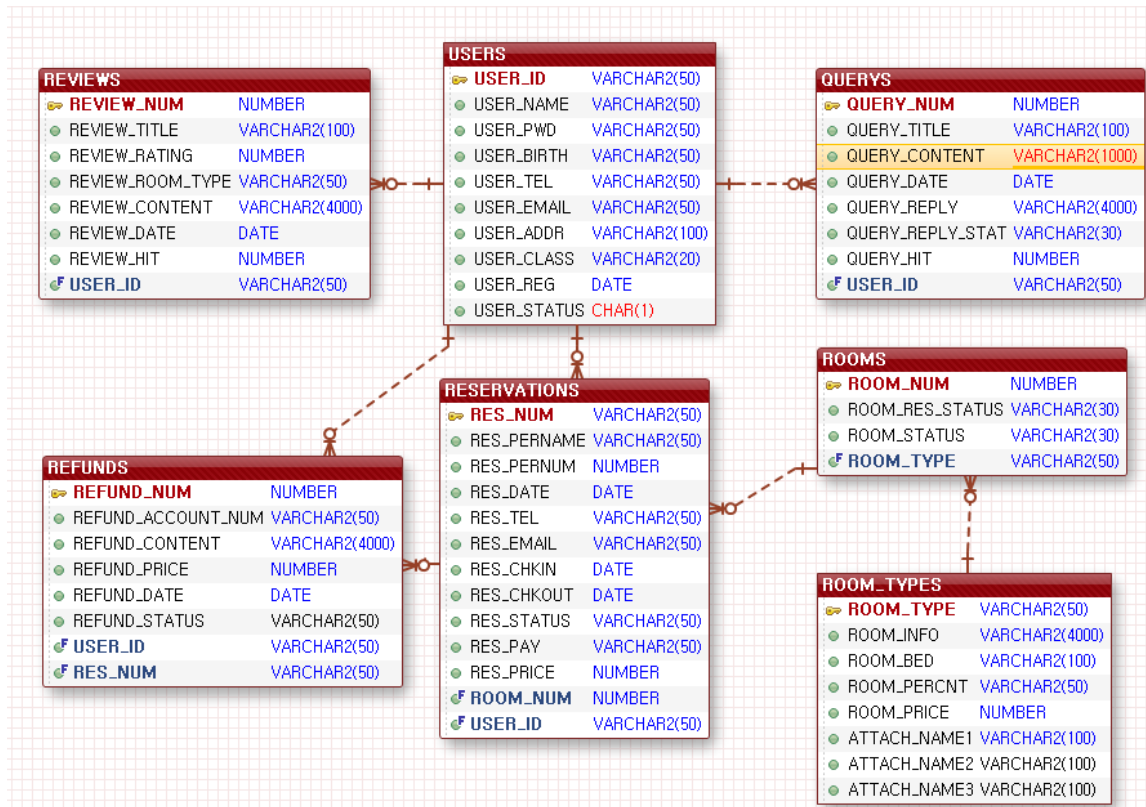
업무명	기능명	요구사항	담당자	업무명	기능명	요구사항	담당자
회원	회원 가입	회원이입을 한다 .	김광수	예약	예약 결제	예약을 할 때 결제를 한다 .	김광수
	회원 수정	회원이입을 할 때 입력한 내용을 수정한다 .	김태연		예약목록 조회	예약 목록을 확인한다 .	김광수
	로그인	로그인을 한다 .	김광수		예약 조회	예약을 확인한다 .	하태창
	로그인 저장	로그인창에 아이디를 저장한다 .	김태연		예약 취소	예약을 취소한다 .	김광호
	로그아웃	로그아웃을 한다 .	김광수	환불	환불 추가	회원에게 환불을 한다 .	김태연
	아이디 찾기	회원의 아이디를 찾는다 .	김태연		환불목록 조회	환불 목록을 확인한다 .	김광수
	비밀번호 찾기	회원의 비밀번호를 찾는다 .	김태연		환불조회	환불을 확인한다 .	김광호
	회원 탈퇴	회원탈퇴를 한다 .	김태연	후기	후기 등록	후기를 등록한다 .	박병국
	회원목록 조회	회원 목록을 확인한다 .	김태연		후기목록 조회	후기 목록을 확인한다 .	박병국
	회원 조회	회원을 조회한다 .	김태연		후기내용 조회	후기 내용을 확인한다 .	박병국
객실	객실목록 조회	객실 목록을 확인한다 .	박혜원		후기 수정	후기를 수정한다 .	박병국
	객실 조회	객실을 조회한다 .	이윤정		후기 삭제	후기를 삭제한다 .	하태창
	객실 추가	객실을 추가한다 .	이윤정	문의	문의 등록	문의를 등록한다 .	김이현
	객실 수정	객실을 수정한다 .	이윤정		문의목록 조회	문의 목록을 확인한다 .	김이현
	객실타입목록 조회	객실 타입 목록을 확인한다 .	이윤정		문의내용 조회	문의 내용을 확인한다 .	김이현
	객실타입 조회	객실 타입을 조회한다 ,	박혜원		문의 수정	문의 내용을 수정한다 .	김이현
	객실타입 추가	객실 타입을 추가한다 .	박혜원		문의 삭제	문의를 삭제한다 .	김이현
	객실타입 수정	객실 타입을 수정한다 .	박혜원		답변 등록	답변을 등록한다 .	김태연
	객실타입 삭제	객실 타입을 삭제한다 ,	박혜원		답변내용 조회	답변내용을 조회한다 .	김태연
				공통	배너 등록	메인 화면이 배너사진을 등록한다	하태창

TABLE 설계도

LOGICAL

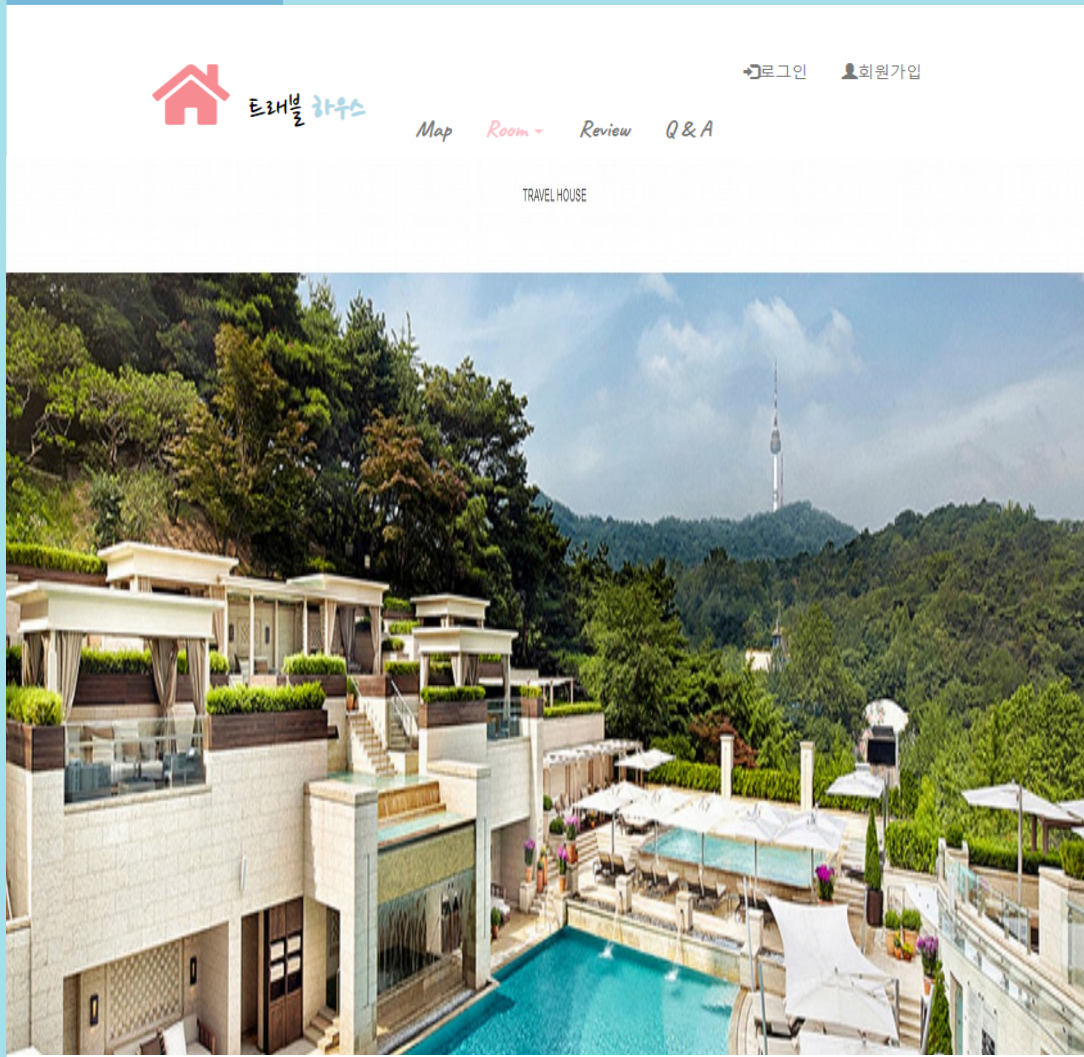


PHYSICAL

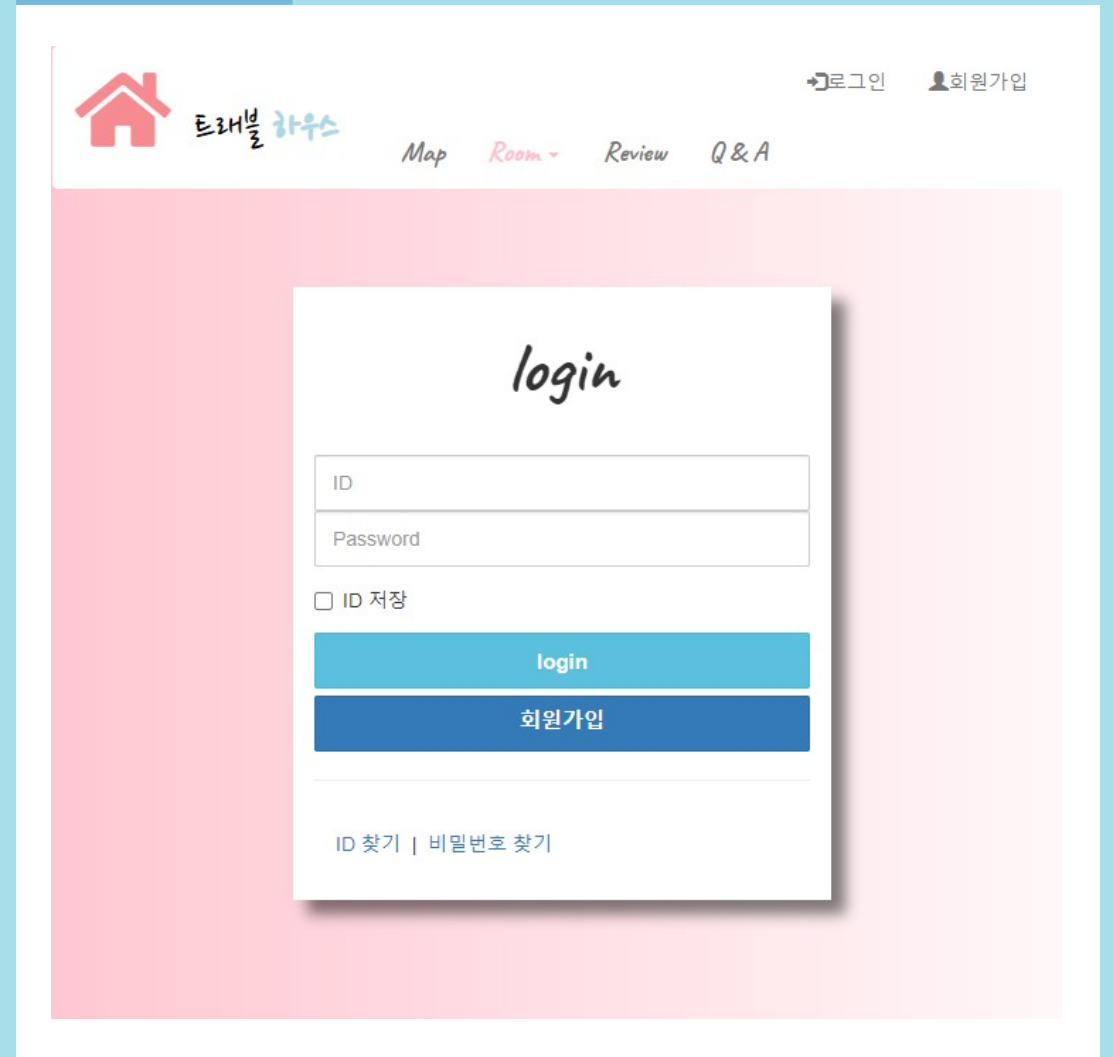


앱 주요화면

Main



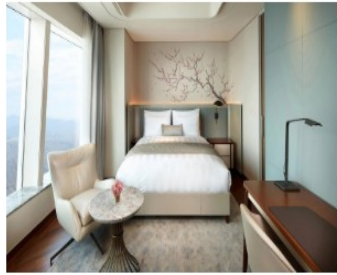
Login



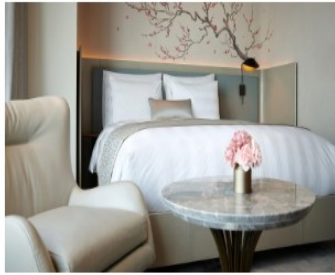
앱 주요화면

객실등록

> 객실 타입 등록



파일 선택 DE1.jpg



파일 선택 DE2.jpg



파일 선택 DE3.jpg

객실 타입

디럭스

중복 확인

사용가능한 객실 타입 명입니다.

객실 안내

트래블 하우스의 디럭스 룸은 넓은 30~40 m²으로 여유롭게 생활하실 수 있습니다. 모던한 인테리어의 아늑한 공간으로 고객을 위해 편안한 휴식과 편의를 제공합니다.

침대 타입

더블*1,싱글*1

인원 수

2~3 명

가격

195000

객실 타입 목록

객실 타입 등록

객실조회

디럭스



[디럭스 룸]

트래블 하우스의 디럭스 룸은 넓은 30~40 m²으로 여유롭게 생활하실 수 있습니다. 모던한 인테리어의 아늑한 공간으로 고객을 위해 편안한 휴식과 편의를 제공합니다.

개요

침대 타입

• 더블*1,싱글*1

체크인 / 체크아웃

• 15:00 / 12:00

투숙 인원

• 2~3 명

어메니티

일반

• 편안한 숙면을 도와주는 최고급사양 침대와 침구류
• 55인치 LED TV
• 에어컨
• 개인 금고
• 전기포트

욕실

• 세면대 2개
• 욕조
• 비데
• 샤워즈 별 고급 타일
• Luxury Bath Amenity (샴푸, 린스, 바디 클렌저)

기타

• 케이블 위성 TV재널
• 무료 생수 2병 및 티백
• 무료 일간 신문

앱 주요화면

예약



user님 환영합니다. [로그아웃](#) [마이페이지](#)

[Map](#) [Room](#) [Review](#) [Q & A](#)

예약 신청

신청자 성명

이지호

이메일

bbc926@naver.com

연락처

01084861633

체크인

2020-07-16

📅

체크아웃

2020-07-30

📅

인원수

3

↑ ↓

예약신청 진행

예약신청 취소

환불



user님 환영합니다. [로그아웃](#) [마이페이지](#)

[Map](#) [Room](#) [Review](#) [Q & A](#)

환불작성

예약 정보

아이디	user
예약자 성명	이지호
이메일	bbc926@naver.com
연락처	01084861633
객실번호	202호
객실종류	스탠다드
체크인	2020-07-16
체크아웃	2020-07-30
인원수	3명
결제방식	신용카드
총 환불금액	150000원

환불 정보

계좌번호

110396906158

상세사유

코로나 때문에 못가요 ㅠ

환불 신청

환불 취소

앱 주요화면

예약 정보

예약 정보

- 예약자 정보

아이디	user
예약자 성명	이울
이메일	onesock@gle.com
연락처	000-000-0025

- 예약상세 정보

객실번호	201
객실종류	디럭스
체크인	2020-07-13
체크아웃	2020-07-13
인원수	2명
결제방식	신용카드
총 결제금액	165000원

[예약내역](#)[예약취소 신청](#)

회원 정보

회원정보

회원님의 정보입니다.

sejong1443

이도

1984-10-09

02-1235-5458

korea@naver.com

서울시 서초구 서초동

서울병원 3층

[수정](#)

앱 주요화면

후기

후기목록

번호	후기제목	작성자	등록일자	조회수
3	여기 너무 깨끗해서 좋아요 ㅠㅠ	user	2020-07-13	0
2	czesc, Dzień dobry	loon0214	2020-07-10	0
1	que estas hacidon ?	loon0214	2020-07-10	1

글쓰기

1 2 3 4 5

문의

Q&A

문의번호 10

작성자 user

등록일 2020-07-13

조회수 0

제목 예약 변경

예약 날짜 변경하고 싶어요 @.@

수정

삭제

목록

핵심기능의 Source Code

```
@RequestMapping("/queryList")
public String getQueryys(Model model, Page page, HttpSession session,
    @RequestParam(value="nowPage", required=false)String nowPage,
    @RequestParam(value="cntPerPage", required=false)String cntPerPage)
{
    //roomlist
    int total1 = roomService.getRoomTypesCnt();
    if(nowPage ==null && cntPerPage ==null) {
        nowPage = "1";
        cntPerPage = "5";
    }else if(nowPage ==null) {
        nowPage = "1";
    }else if(cntPerPage ==null) {
        cntPerPage = "5";
    }
    page = new Page(total1, Integer.parseInt(nowPage),
    Integer.parseInt(cntPerPage));

    model.addAttribute("roomTypeList", roomService.getRoomTypes(page));

    int total = queryService.getQueryysCnt();
    if(nowPage ==null && cntPerPage ==null) {
        nowPage = "1";
        cntPerPage = "5";
    }else if(nowPage ==null) {
        nowPage = "1";
    }else if(cntPerPage ==null) {
        cntPerPage = "5";
    }
    page = new Page(total, Integer.parseInt(nowPage),
    Integer.parseInt(cntPerPage));
    session.setAttribute("paging", page);
    session.setAttribute("query", queryService.listQueryys(page));

    return "query/queryList";
}
```

문의목록 & 페이징

회원들이 문의글을 작성하고, 그에 대한 문의목록을 확인할 수 있다. 페이징의 경우, 한 화면에는 1 부터 5 까지 5 개의 페이지 버튼들과 이전 페이지 버튼, 다음 페이지 버튼이 나타난다. 이전 페이지 버튼과, 다음 페이지 버튼을 사용하여 5 개씩 건너뛸 수 있다.

핵심기능의 Source Code

```
public @ModelAttribute("room")RoomType roomInfo(Page page, Model model,
    @RequestParam(value="roomType", required=false) String roomTypeText,
    @RequestParam(value="nowPage", required=false)String nowPage
    ,@RequestParam(value="cntPerPage",required=false)String cntPerPage){
    int total = roomService.getRoomTypesCnt();
    if(nowPage ==null && cntPerPage ==null) {
        nowPage = "1";
        cntPerPage = "5";
    }else if(nowPage ==null) {
        nowPage = "1";
    }else if(cntPerPage ==null) {
        cntPerPage = "5";
    }
    page = new Page(total, Integer.parseInt(nowPage), Integer.parseInt(cntPerPage));
    model.addAttribute("roomTypeList", roomService.getRoomTypes(page));

    RoomType chooseroom = new RoomType();
    chooseroom = roomService.getRoomType(roomTypeText);

    model.addAttribute("chooseRoom", chooseroom);

    RoomType showRoomType = new RoomType();
    showRoomType.setRoomType(roomTypeText);
    showRoomType.setRoomInfo(chooseroom.getRoomInfo());
    showRoomType.setRoomBed(chooseroom.getRoomBed());
    showRoomType.setRoomPerCnt(chooseroom.getRoomPerCnt());
    showRoomType.setRoomPrice(chooseroom.getRoomPrice());
    showRoomType.setAttachName1(chooseroom.getAttachName1());
    showRoomType.setAttachName2(chooseroom.getAttachName2());
    showRoomType.setAttachName2(chooseroom.getAttachName3());

    model.addAttribute("showRoomType", showRoomType);

    Room findRoom = new Room();
    findRoom = roomService.findRoom(roomTypeText);

    findRoom.getRoomNum();
    model.addAttribute("findRoom", findRoom);

    return roomService.getRoomType(roomTypeText);
}
```

객실정보

관리자가 객실타입과, 객실을 등록하고, 회원들은 현재 존재하는 객실을 확인할 수 있다. 객실타입에 대한 데이터들과, 객실을 소개하는 이미지 파일들 3 개를 차례대로 로드한다. 그리고, 객실이 존재하는지의 여부를 검사하여 객실타입이 존재하더라도, 해당 객실타입에 대한 객실이 없으면, 회원이 객실을 확인할 수 없다.

핵심기능의 Source Code

```
<bean id='LoginInterceptor' class='kimkwangsoo.hotel.user.controller.LoginInterceptor' />
<mvc:interceptors>
    <mvc:interceptor>
        <!-- 인터셉터 부분 -->
        <mvc:mapping path="/myPage" />
        <mvc:mapping path="/admin/**" />
        <mvc:mapping path="/review/reviewAdd" />
        <mvc:mapping path="/query/queryAdd" />
        <mvc:mapping path="/admin" />
        <mvc:mapping path="/user/modUser" />
        <mvc:mapping path="/user/userInfo" />
        <mvc:mapping path="/reservation/resInfo" />
        <mvc:mapping path="/refund/refundInfo" />

        <!-- 인터셉터 예외부분 -->
        <mvc:exclude-mapping path="/" />
        <ref bean='LoginInterceptor' />
    </mvc:interceptor>
</mvc:interceptors>
```

```
public class LoginInterceptor extends HandlerInterceptorAdapter{
    public boolean preHandle(HttpServletRequest request,
        HttpServletResponse response, Object handler)throws Exception{
        boolean isLogin = true;

        if(!request.getRequestURI().equals("/hotel/user/login")) {
            User user = (User)request.getSession().getAttribute("user");
            if(user==null) {
                response.sendRedirect("/hotel/user/logon");
                isLogin = false;
            }
        }
        return isLogin;
    }
}
```

로그인 인터셉터

인터셉터를 사용하여, 로그인 상태가 아닐 시, 로그인이 반드시 필요한 페이지에 접속할 수 없게 설정한다. 스프링 컨텍스트에서 인터셉터를 사용할 url 주소들을 설정하고, 컨트롤러에서 로그인의 여부를 검사하여 로그인이 필요한 사이트에서 로그인이 되어 있지 않으면, 로그인 화면으로 강제로 이동하게 한다.