

# JAVA GUI 프로젝트

## 호텔 예약 프로그램

---

발표자 : 문형욱

개발기간 : 2024.4.15~2024.4.22

# CONTENTS

## 01 프로젝트 소개   02 개발 및 환경구현   03 코드 증빙 및 해설   04 시연 영상

- 기능 설계
- 메인 페이지

- 사용프로그램
- 유스케이스
- 클래스 다이어그램
- DATABASE

- UI 및 기능
- 회원가입
- 로그인
- 예약가능 룸리스트
- 예약

- 시연 영상

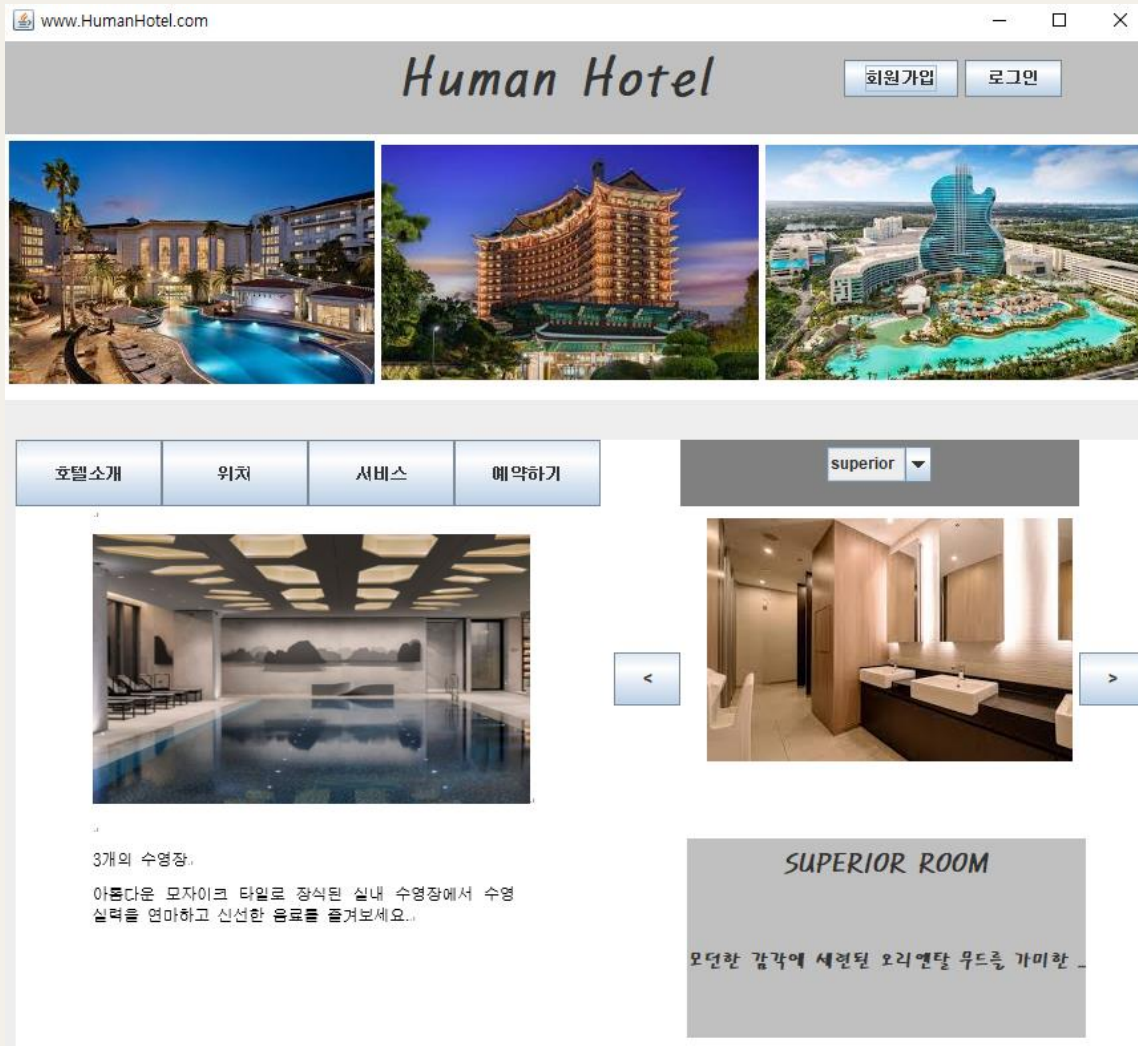


## 01 프로젝트 소개

# 01 프로젝트 소개 – 기능 설계

호텔홈페이지	자바 스윙을 통해 호텔GUI를 구성하고 콤보박스과 텍스트필드를 이용하여 정보를 오라클 데이터베이스와 주고 받는다.
회원가입 기능	사용자가 예약정보를 입력하고 예약을 할 수 있는 페이지에 접근하기 위한 로그인 정보를 오라클 데이터베이스 Guest 테이블에 저장한다.
로그인 기능	오라클 데이터베이스에 있는 저장해 놓은 로그인 정보들과 내가 입력한 아이디와 패스워드 정보를 비교하여 일치하면 예약페이지에 접근 할 수 있게 한다.
예약 기능	예약자 정보를 입력하고 예약하기 버튼을 누르면 오라클 데이터베이스의 Guest 테이블의 해당 아이디의 튜플의 정보를 업데이트한다. Room 테이블의 해당 룸넘버의 사용가능 변수를 변경시켜 다른 예약자가 접근 할 수 없게 한다.

# 01 프로젝트 소개 – 메인 페이지



호텔소개	위치	서비스	예약하기
예약자명	<input type="text" value="민경훈"/>		
입실	<input type="text" value="2024"/> ▼	<input type="text" value="2"/> ▼	<input type="text" value="3"/> ▼
퇴실	<input type="text" value="2024"/> ▼	<input type="text" value="2"/> ▼	<input type="text" value="4"/> ▼
객실	<input type="text" value="delux"/> ▼		
인원수	<input type="text" value="2"/> ▼		
<input type="button" value="예약하기"/>			



## 02 개발 및 환경구현



## 02 개발 및 환경구현 - 사용 프로그램



기능

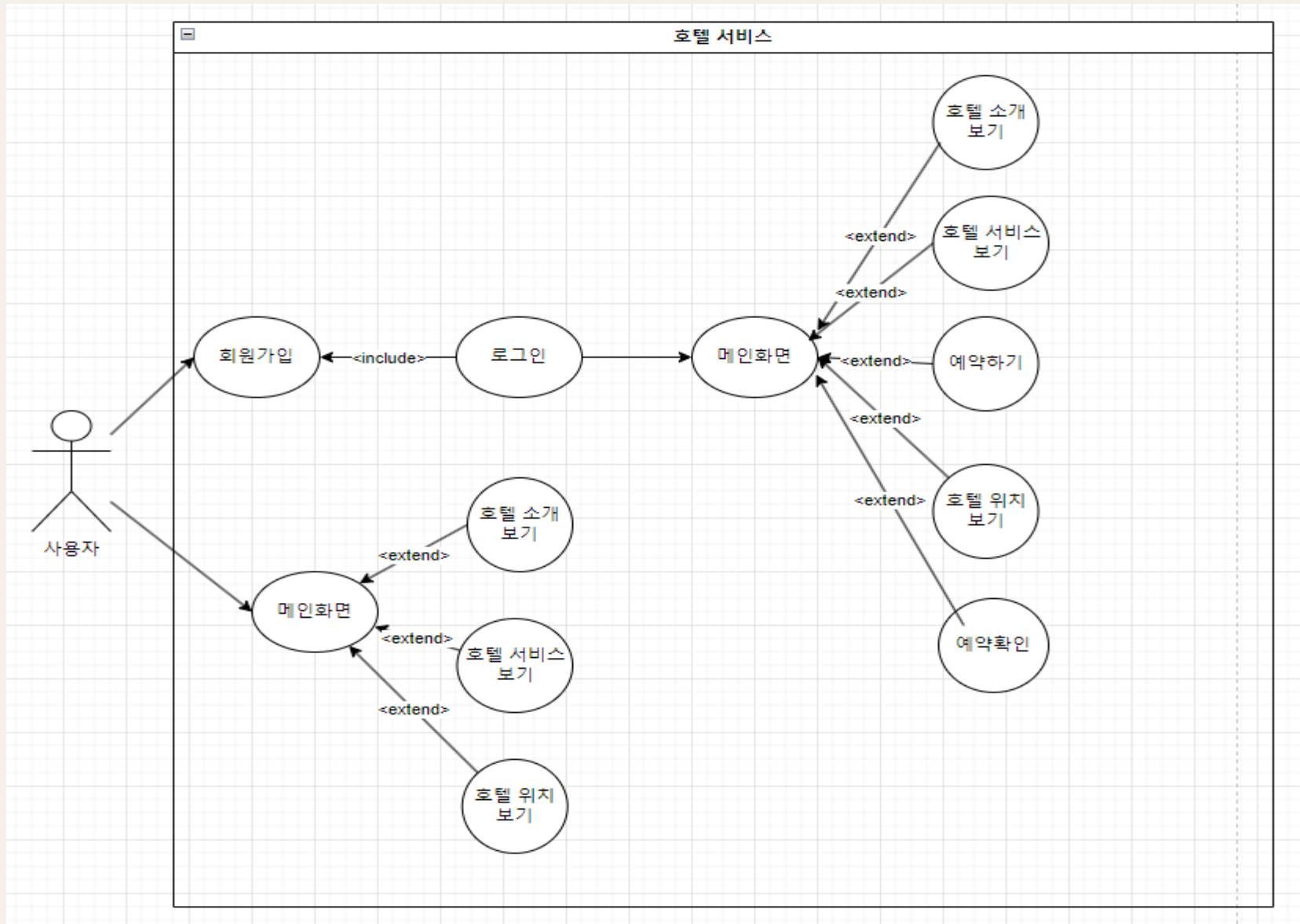


DB 구축



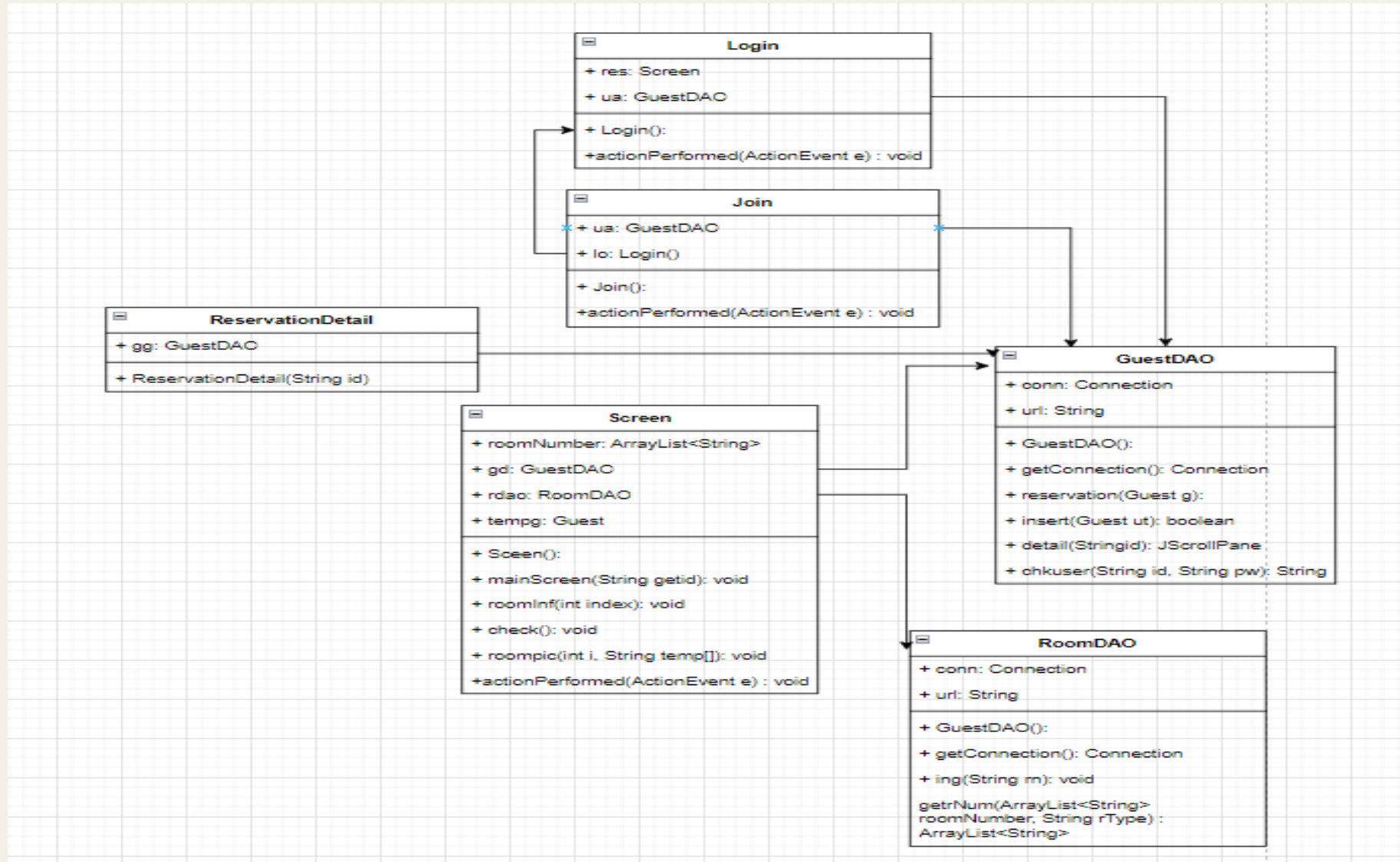
GUI 구현

## 02 개발 및 환경구현 - 유스케이스





## 02 개발 및 환경구현 - 클래스 다이어그램

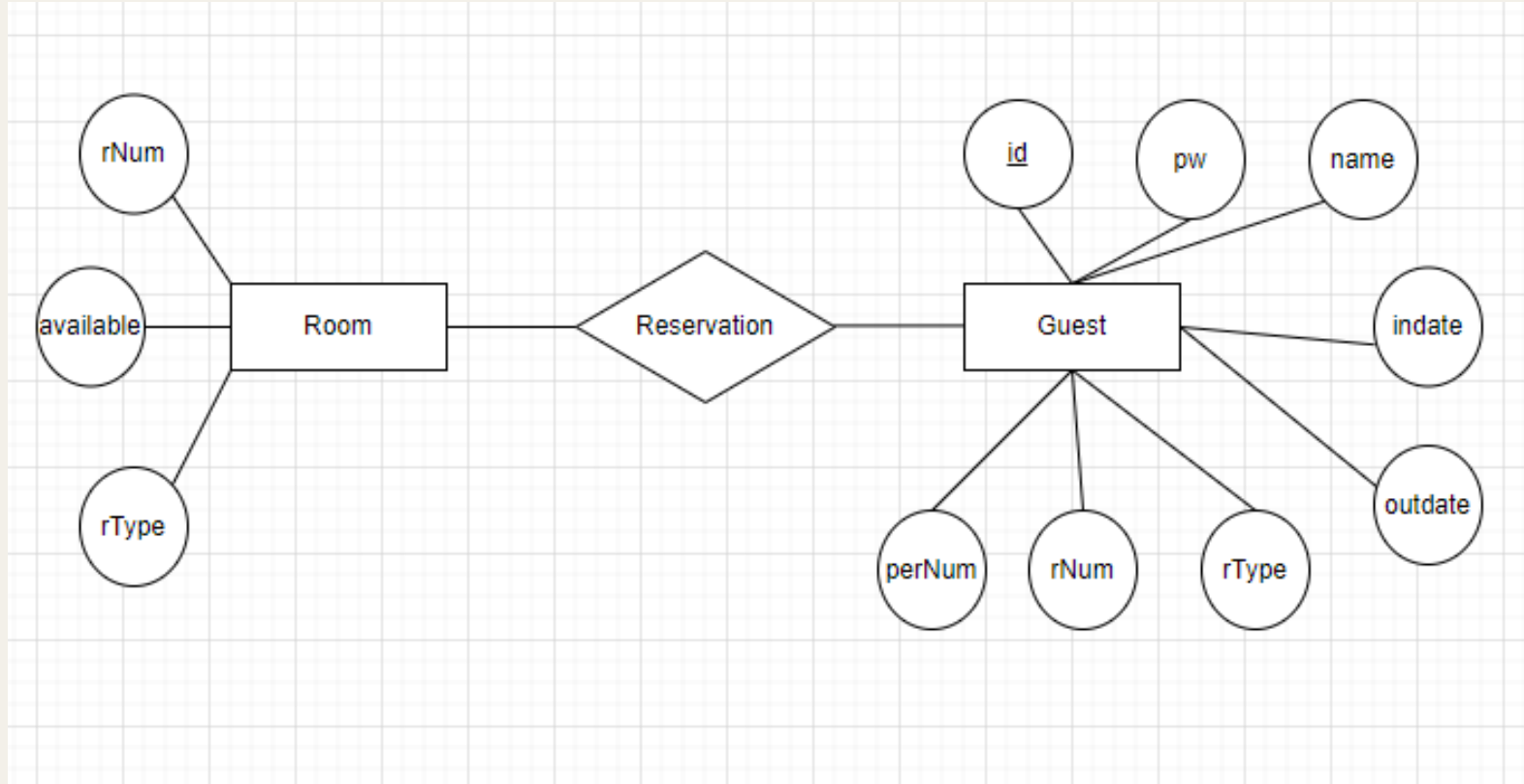


```
create table Room (  
  rNum number(5),  
  rType varchar2(10),  
  available number(2));
```

예 약

```
create table guest(  
  id varchar2(30) primary key,  
  pw varchar2(50),  
  name varchar2(10),  
  indate varchar2(30),  
  outDate varchar2(30),  
  rType varchar2(30),  
  rNum varchar2(20),  
  perNum varchar2(10));
```

## 02 개발 및 환경구현 - 데이터베이스 구조





## *03 코드 증빙 및 해설*



# 03 코드 증빙 및 해설- UI 및 기능



## 03 코드 증빙 및 해설- 회원가입 기능

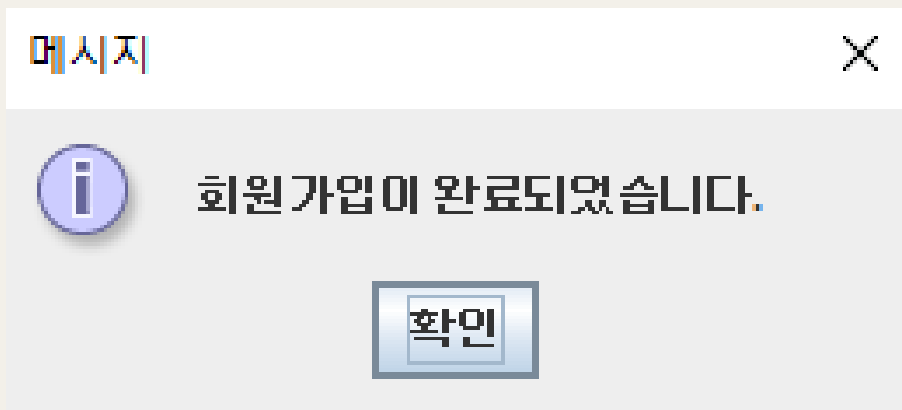


회원가입

ID user

pw \*\*\*\*

회원가입



### FRONT

- 가입 할 아이디와 비밀번호를
- TextField에 입력
- 회원가입 버튼을 눌러 회원가입

### BACK

- TextField를 통해 데이터 저장
- GuestDTO를 선언하고 저장된 데이터 저장
- GuestDAO 클래스의 insert() 매개변수에 GuestDTO 주소 저장
- 오라클 데이터베이스 Guest 테이블에 INSERT 구문을 통해 데이터 insert

## 03 코드 증빙 및 해설- 회원가입 코드

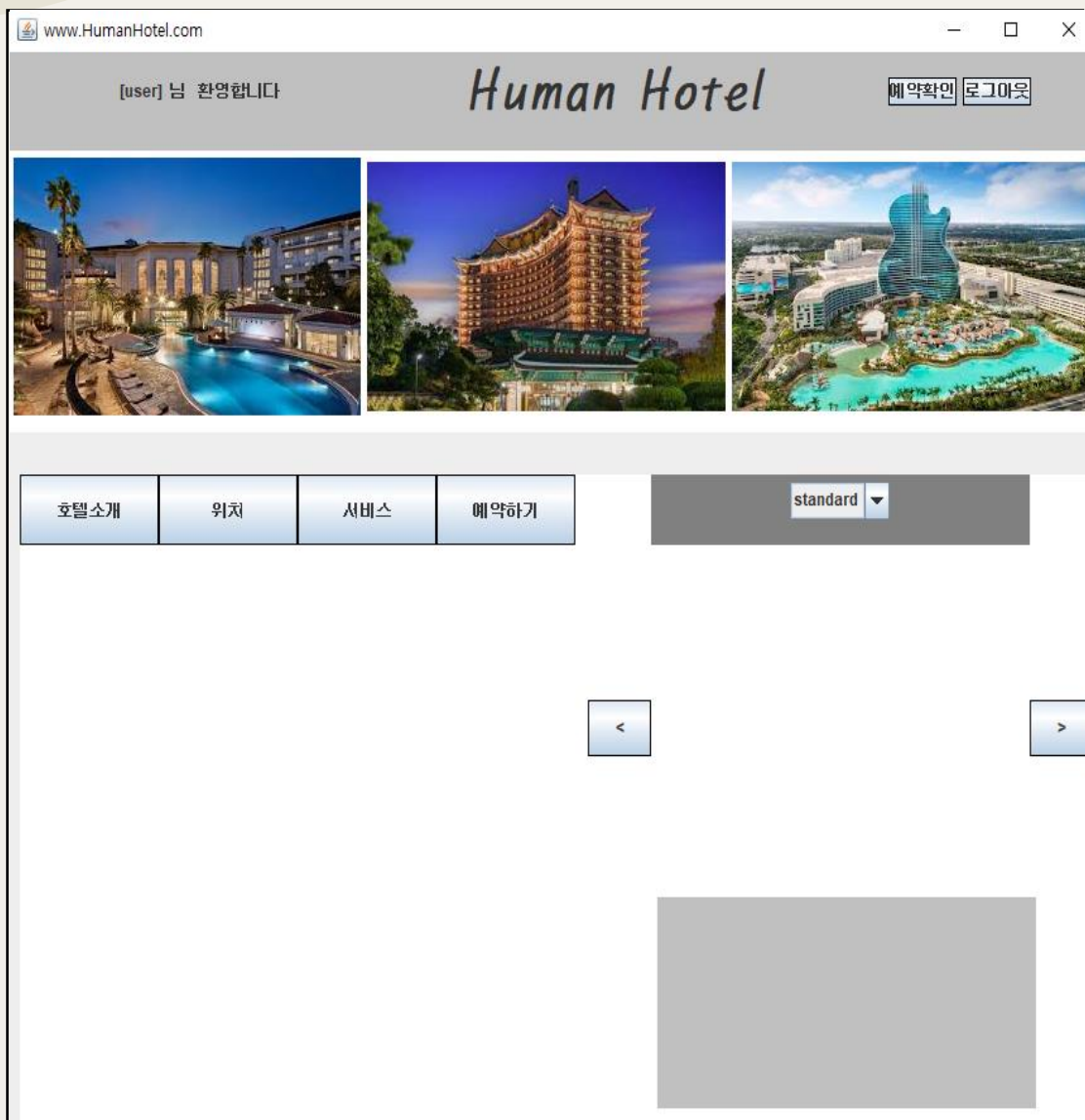
입력받은 Text를 GuestDTO에 저장 후 주소 주입

```
@Override
public void actionPerformed(ActionEvent e) {
    String getid = idt.getText();
    String getpw = pwt.getText();
    private Guest ut = new Guest();
    if (e.getSource().equals(joinButton)) {
        ut.setId(getid);
        ut.setPw(getpw);
        if (ua.insert(ut)) {
            this.setVisible(false);
            JOptionPane.showMessageDialog(null, "회원가입이 완료되었습니다.");
        } else {
            JOptionPane.showMessageDialog(null, "중복된 아이디입니다.");
        }
    }
}
```

GuestDTO의 값을 SQL문에 매핑 후 INSERT

```
public boolean insert(Guest ut) {
    PreparedStatement psmt = null;
    if (getConnection() != null) {
        try {
            String sql = "insert into guest values (?, ?, null, null, null, null, null, null)";
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, ut.getId());
            psmt.setString(2, ut.getPw());
            psmt.executeUpdate();
            return true;
        } catch (Exception e) {
            System.out.println("입력작업에서 예외 발생");
        }
    }
    return false;
}
```

## 03 코드 증빙 및 해설- 로그인 기능



### FRONT

- 아이디와 패스워드를 TextField에 입력
- 로그인 버튼을 눌러 로그인

### BACK

- getText() 메소드를 통해 데이터 저장
- GuestDAO 클래스의 chkuser() 매개변수에 저장된 아이디와 비밀번호 대입
- 오라클 데이터베이스에서 SELECT문을 통해 Guest 테이블의 튜플들과 비교 후 일치하는 튜플이 있다면 ID 데이터 리턴
- Chkuser()의 리턴값이 null 이 아니라면 즉 로그인이 되었다면 메인스크린으로 이동



## 03 코드 증빙 및 해설- 로그인 코드

입력받은 Text를 chkuser 메소드의 매개변수로 대입

```
@Override
public void actionPerformed(ActionEvent e) {
    String getid = idt.getText();
    String getpw = pwt.getText();
    if (e.getSource().equals(joinButton)) {
        this.setVisible(false);
        new Join();
    } else if (e.getSource().equals(loginButton)) {
        wel.setText("아이디 혹은 패스워드가 일치하지 않습니다.");
        if (ua.chkuser(getid, getpw) != null) {
            this.setVisible(false);
            res.mainScreen(getid);
        } else {
            System.out.println("실패");
        }
    }
}
```

Geust 테이블의 모든 튜플과 입력받은 text 비교

```
public String chkuser(String id, String pw) {
    if (getConnection() != null) {
        try {
            Statement stmt = null;
            ResultSet rs = null; // DB에서 결과
            String sql = "select * from guest";
            stmt = conn.createStatement();
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                String cid = rs.getString("id");
                String cpw = rs.getString("pw");
                if (cid.equals(id) && cpw.equals(pw)) {
                    System.out.println(id + "님 로그인하셨습니다. ");
                    return id;
                }
            }
        } catch (Exception e) {
            return null;
        }
    }
    return null;
}
```

## 03 코드 증빙 및 해설- 예약 기능

호텔소개

위치

서비스

예약하기

예약자명

민경훈

입실

2024

4

29

퇴실

2024

4

30

객실

standard

standard

superior

delux

suite

인원수

예약하기

[user] 님의 예약내역

이름	입실 날짜	퇴실 날짜	룸 타입	룸 번호	인원
민경훈	2024/5/2	2024/5/3	standard	401	2

### FRONT

- 예약자 정보를 TextFeild와 ComboBOX에 입력
- 예약하기 버튼을 눌러 예약

### BACK

- 현재 로그인한 ID의 테이블 정보를 업데이트 하기 위해 DTO객체를 생성하여 현재 ID데이터를 저장
- 예약자 정보를 DTO에 저장 후 주소를 GuestDAO 클래스의 reservation() 의 매개변수로 주입
- 오라클에서 UPDATE구문을 통해 해당 아이디 튜플의 정보 업데이트

## 03 코드 증빙 및 해설- 예약 코드

새로운 GuestDTO 객체에 예약정보 SET 후 주소 주입

```
tempg.setName(gName);
tempg.setInDate(in);
tempg.setOutDate(out);
tempg.setrType(roomType);
tempg.setPerNum(person);
tempg.setrNum(roomnum);
if (tempg.getName() != null && iyear != "YEAR" && imonth != "MONTH" && iday != "DAY" && oyear != "YEAR"
    && oMonth != "MONTH" && oDay != "DAY" && roomType != "RoomType" && person != "인원수") {
    gd.reservation(tempg);
    rdao.ing(tempg.getrNum());
    JOptionPane a = new JOptionPane();
    JOptionPane.showMessageDialog(null, gName + " 님 예약 되었습니다.");
} else {
    JOptionPane a = new JOptionPane();
    JOptionPane.showMessageDialog(null, "예약정보를 전부 입력해주세요");
}
```

주입 받은 DTO의 예약정보를 SQL에 매핑 후 UPDATE


```
public void reservation(Guest g) {
    PreparedStatement psmt = null;
    if (getConnection() != null) {
        try {
            String sql = "update guest set name = ?, inDate = ?, outDate = ?, rType = ?, perNum = ?, rNum = ? where id = ?";
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, g.getName());
            psmt.setString(2, g.getInDate());
            psmt.setString(3, g.getOutDate());
            psmt.setString(4, g.getrType());
            psmt.setString(5, g.getPerNum());
            psmt.setString(6, g.getrNum());
            psmt.setString(7, g.getId());

            psmt.executeUpdate();
        } catch (Exception e) {
            System.out.println("입력작업에서 예외 발생");
        }
    }
}
```

## 03 코드 증빙 및 해설- 이용가능 룸 리스트

호텔소개	위치	서비스	예약하기
<p>예약자명 <input type="text" value="민경훈"/></p> <p>입실 <input type="text" value="2024"/> <input type="text" value="4"/> <input type="text" value="29"/></p> <p>퇴실 <input type="text" value="2024"/> <input type="text" value="4"/> <input type="text" value="30"/></p> <p>객실 <input type="text" value="standard"/></p> <p>인원수 <input type="text" value="standard"/> <input type="text" value="superior"/> <input type="text" value="delux"/> <input type="text" value="suite"/></p> <p><input type="button" value="예약하기"/></p>			

103 ▼  
103  
203  
303  
404



< >

SUPERIOR ROOM

모던한 감각에 세련된 오리엔탈 무드를 가미한

### FRONT

- 객실 타입 콤보박스를 통해 이용하고자 하는 객실타입 선택
- 선택한 객실타입 중 이용가능한 룸번호 확인 후 선택

### BACK

- ArrayList를 선언하고 저장된 객실타입과 available 변수가 0인 튜플을 SELECT
- 해당 튜플의 룸번호를 ArrayList에 저장
- 리스트를 통해 예약가능한 콤보박스 생성

## 03 코드 증빙 및 해설- 이용가능 룸 리스트 코드

선택 된 객실타입과 ArrayList를 매개변수로 대입

```
ArrayList<String> roomNumber = new ArrayList<String>();
```

```
if (index == 0) {
    rT.setVisible(false);
    rT.removeAll();
    temp = standard;
    String st = "standard";
    rdao.getrNum(roomNumber, st);
    JComboBox<String> rrrrr = new JComboBox<String>(roomNumber.toArray(new String[roomNumber.size()]));
    rT.add(rrrrr); // 룸넘버 들어갈 곳
    rrrrr.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            JComboBox crrrr = (JComboBox) e.getSource();
            roomnum = rrrrr.getSelectedItem().toString();
        }
    });
}
```

DB에서 SELECT문을 통해 객실번호를 ArrayList에 ADD

```
public ArrayList<String> getrNum(ArrayList<String> roomNumber, String rType) {
    PreparedStatement psmt = null;
    if (getConnection() != null) {
        try {

            ResultSet rs = null; // DB에서 결과
            String sql = "select * from Room where rType = ? and available = 0";
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, rType);
            psmt.executeUpdate();
            rs = psmt.executeQuery(sql);
            roomNumber.clear();
            while (rs.next()) {
                String rNum = rs.getString("rNum");
                roomNumber.add(rNum);
            }

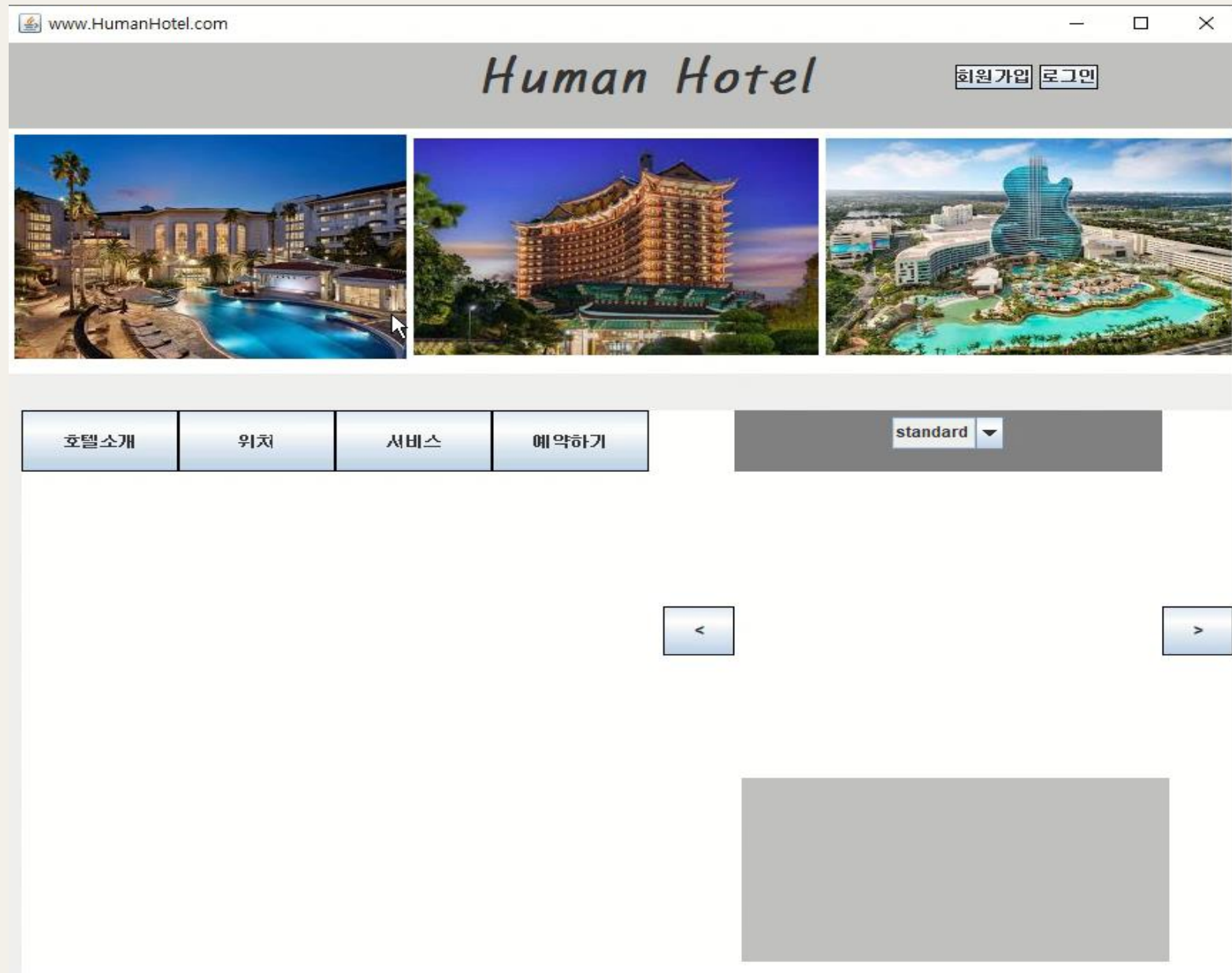
        } catch (Exception e) {
            // TODO: handle exception
        }
    }
    return roomNumber;
}
```



## 04 시연 영상



# 04 시연영상



*Thank you*

---

