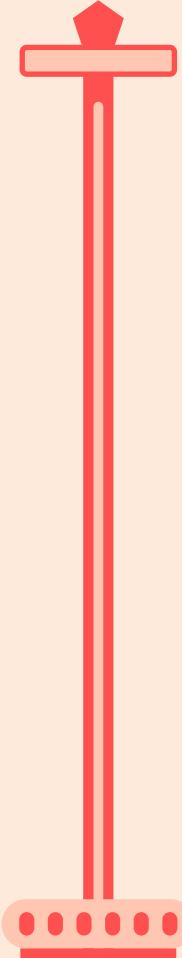


1차 프로젝트
놀이동산 스마트 예약



민희원
이혜련
정병수
임정통

목차

1) 개발과정

- 1-1) 보유/사용 기술 목록
- 1-2) 개발 기간
- 1-3) DB테이블
- 1-4) 순서도(기능분담)
- 1-5) UML

2) 기술설명

- 2-1) 민희원 회원관리 구현
- 2-2) 이혜련 놀이기구 관리 구현
- 2-3) 정병수 놀이기구 예약 구현
- 2-4) 임정통 게시판 구현

1) 개발과정

1-1) 보유기술



1] 개발과정

1-2] 개발기간

	단계	8/19	8/20	8/21	8/22
기획	프로젝트 기획				
설계	데이터베이스 설계				
코딩	레이아웃 작업				
	코딩				
구현	디버깅				

	단계	8/23	8/24	8/25	8/26
기획	프로젝트 기획				
설계	데이터베이스 설계				
코딩	레이아웃				
	코딩				
구현	디버깅				



1) 개발과정

1-3] DB테이블

Members table

속성	속성타입	제약조건	내용
M_id	varchar(100)	Not null, primary	회원 아이디
Pw	varchar(100)	Not null	비밀번호
M_name	varchar(100)		회원 이름
Gender	varchar(100)		성별 (M or F)
Birth	varchar(100)		출생년도
Tall	Int(11)	Not null	키 (cm)
Tel	varchar(100)		전화번호
Email	varchar(100)		이메일



1) 개발과정

1-3] DB테이블

Board table

속성	속성타입	제약조건	내용
B_id	Int(11)	Not null, primary	게시글 아이디
M_id	varchar(100)	Foreign key, references Members	회원 아이디
title	varchar(100)		제목
content	text		내용
uDate	datetime		게시일시
hits	Int(11)		조회수



1) 개발과정

1-3] DB테이블

Ride table

속성	속성타입	제약조건	내용
R_id	varchar(100)	Not null, primary	놀이기구 아이디
R_name	varchar(100)		놀이기구 이름
Location	varchar(100)		놀이기구 위치
people	Int(11)		수용인원
R_Ltall	Int(11)		탑승가능 최소 키
R_Htall	Int(11)		탑승가능 최대 키
ITime	Int(11)		놀이기구 소요 시간
R_content	text		놀이기구 내용
Availability	varchar(100)	Not null	예약가능 여부
Img	varchar(100)		놀이기구 사진

1-3) DB테이블

1) 개발과정

Ptable table

속성	속성타입	제약조건	내용
P_id	varchar(100)	Not null, primary	입장권 아이디
P_name	varchar(100)		입장권 종류 이름
price	Int(11)		가격

ttable table

속성	속성타입	제약조건	내용
T_id	varchar(100)	Not null	타임테이블 아이디
R_id	varchar(100)	Foreign key, references Ride	놀이기구 아이디
R_name	varchar(100)		놀이기구 이름
sTime	varchar(100)		시작시간
fTime	varchar(100)		종료시간
bTime	varchar(100)		탑승시간

1) 개발과정

1-3] DB테이블

SmartR table

속성	속성타입	제약조건	내용
SR_id	varchar(100)	Not null, primary	상품예약번호
P_id	varchar(100)	Foreign key, references Ptable	상품번호
P_name	varchar(100)		상품명
rDate	varchar(100)		예약날짜
price	Int(11)		상품가격
M_id	varchar(100)	Foreign key, references Members	회원 아이디
M_name	varchar(100)		회원 이름
kdate	datetime		데이터 적재시간



1) 개발과정

1-3] DB테이블

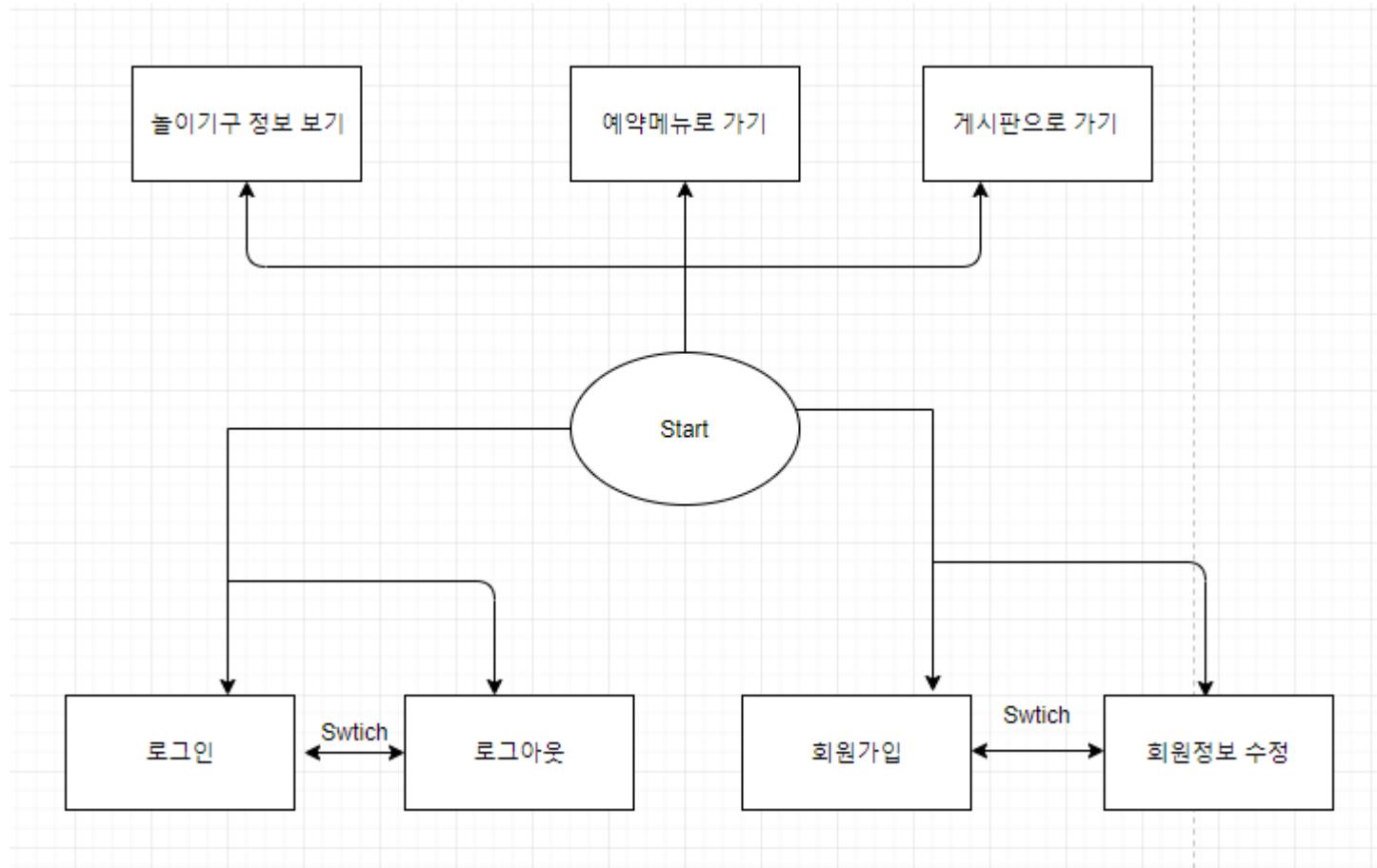
Rider table

속성	속성타입	제약조건	내용
Rr_id	varchar(100)	Not null, primary	놀이기구 예약번호
R_name	varchar(100)		놀이기구 이름
Sr_id	varchar(100)	Foreign key, references Smartr	예약번호
rdate	varchar(100)		탑승날짜
sTime	varchar(100)		회차 시작시간
fTime	varchar(100)		회차 종료시간
bTime	varchar(100)		탑승시간
M_id	varchar(100)	Foreign key, references Members	회원 아이디
M_name	varchar(100)		회원 이름
R_id	varchar(100)	Foreign key, references Ride	놀이기구 아이디
T_id	varchar(100)	Foreign key, references ttable	타임테이블 아이디
kdate	datetime		데이터 적재시간

1) 개발과정

1-4) 순서도

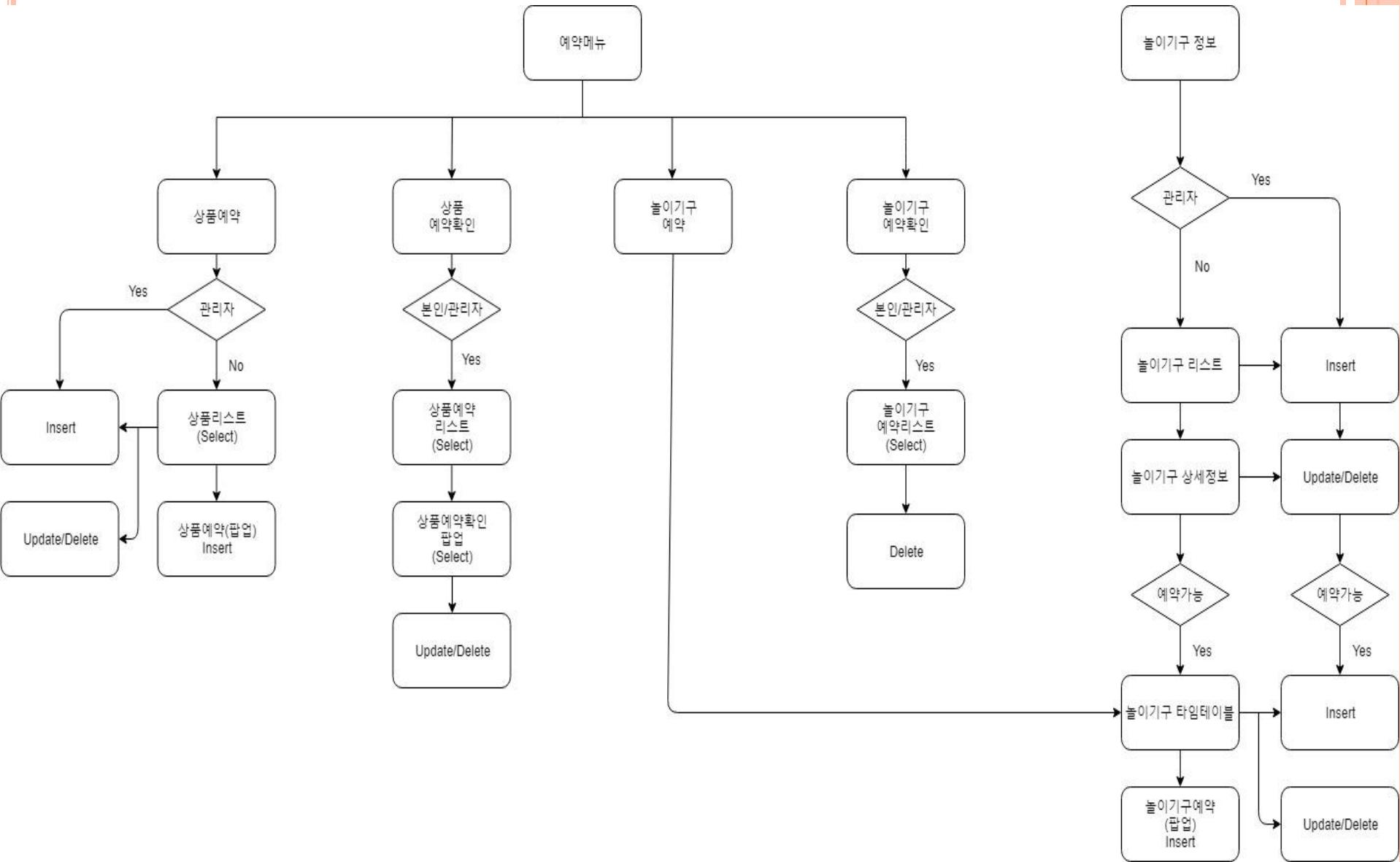
첫 화면 메인 순서도



1) 개발과정

1-4) 순서도

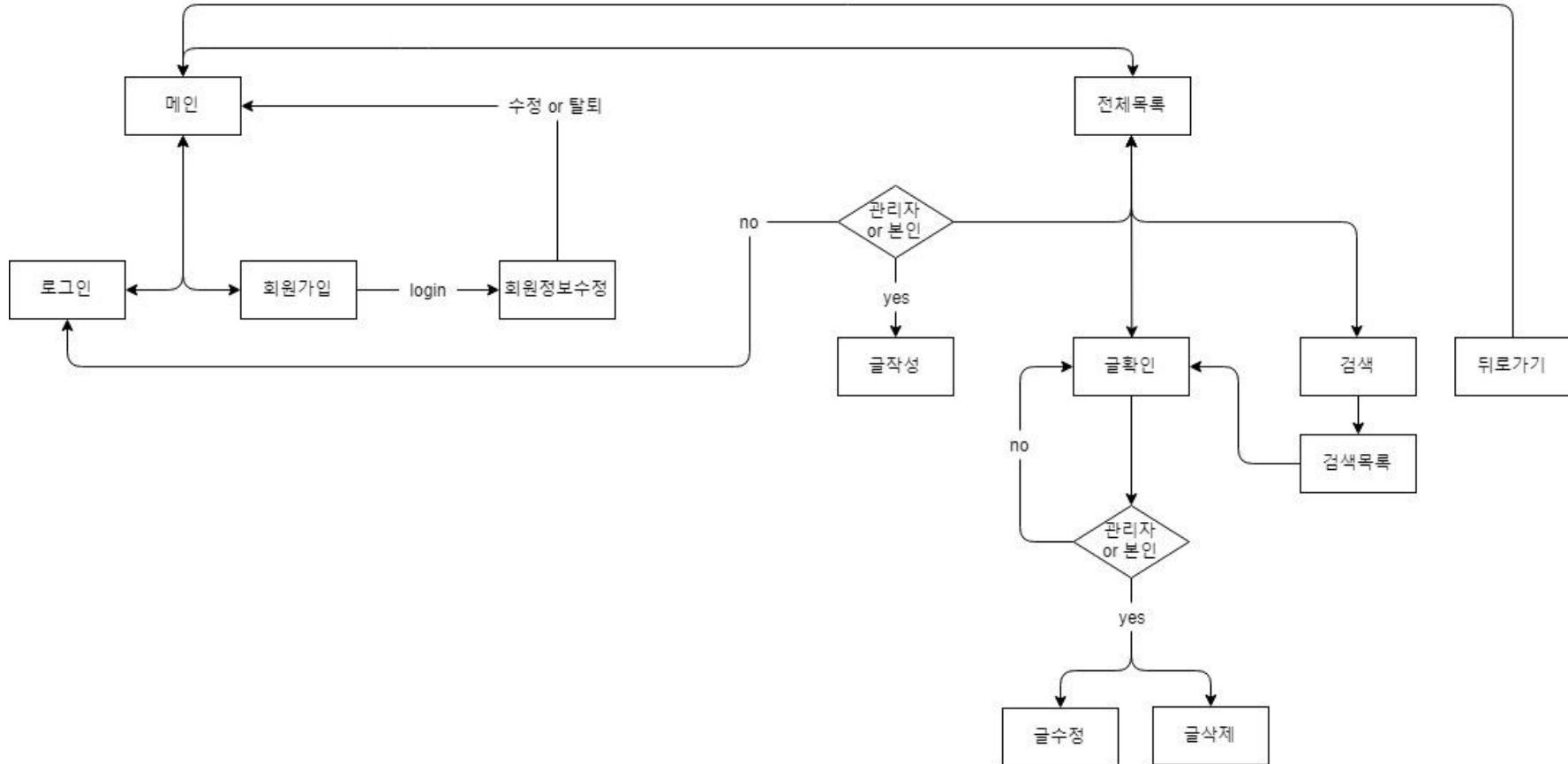
예약메뉴로 가기 & 놀이기구 정보 보기 선택 후, 상세 순서도



1) 개발과정

1-4) 순서도

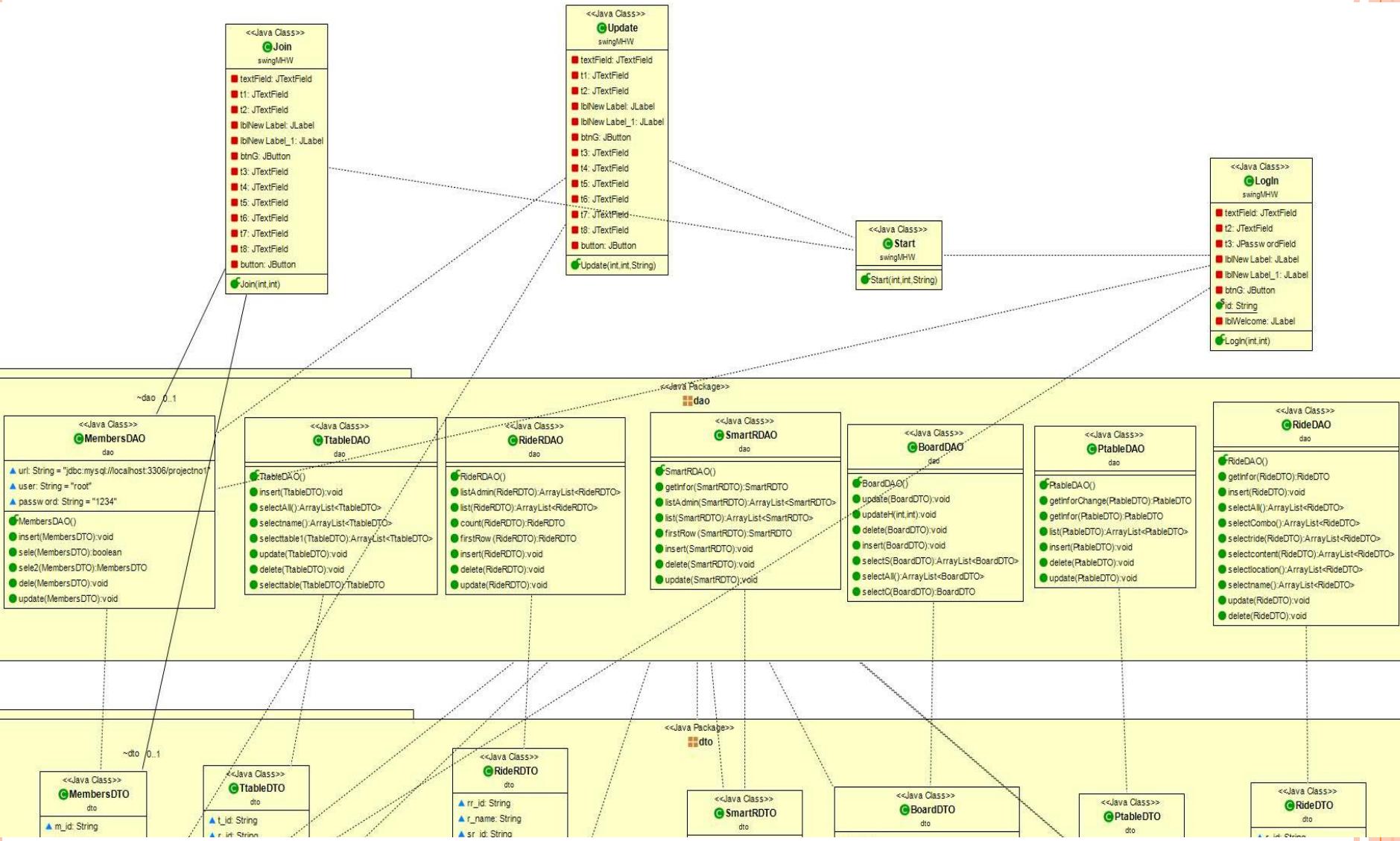
게시판으로 가기 선택 후, 상세 순서도



1) 개발과정

1-5) UML

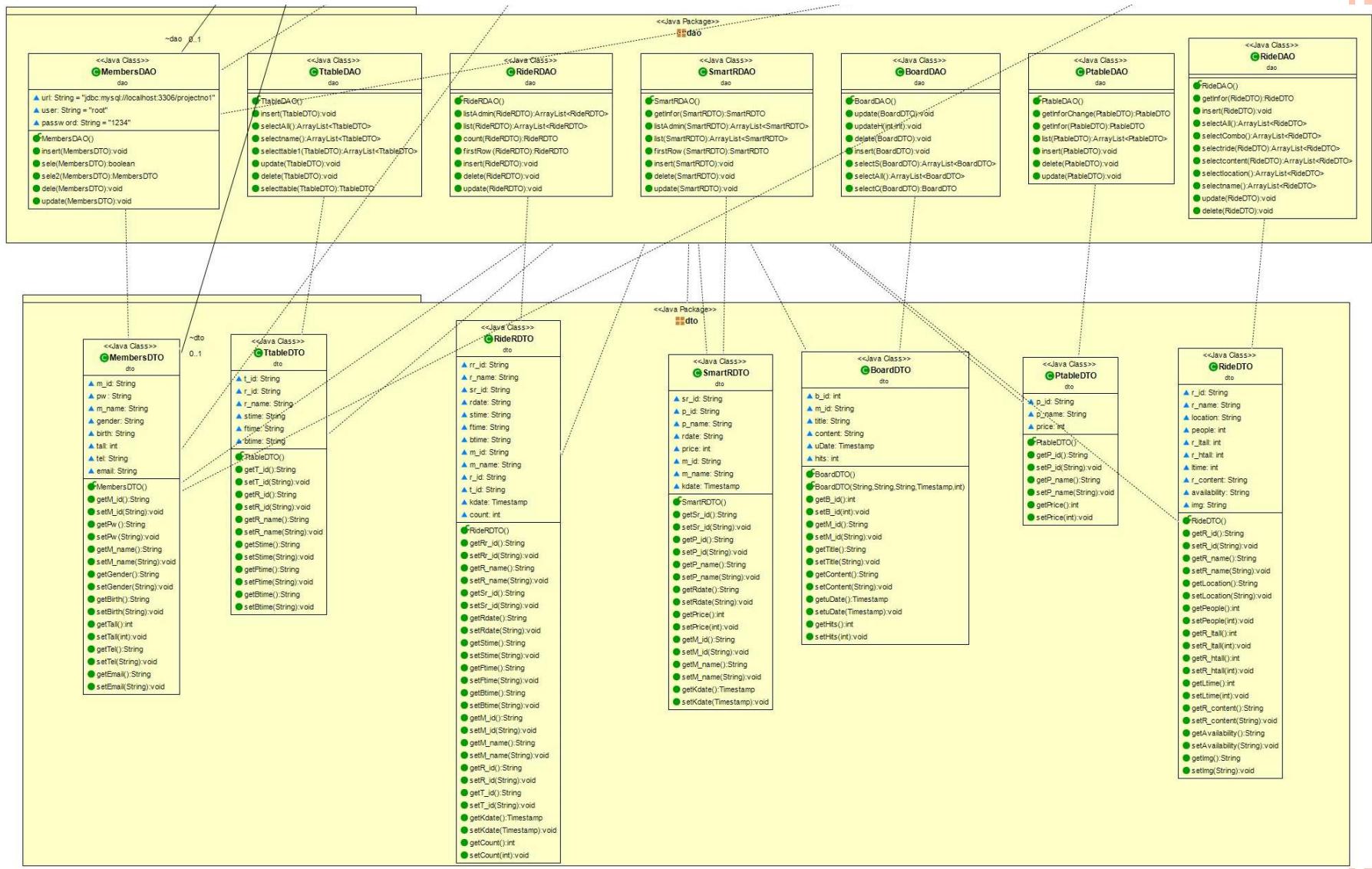
(1) SwingMHW, Join, Update, Login



1) 개발과정

1-5] UML

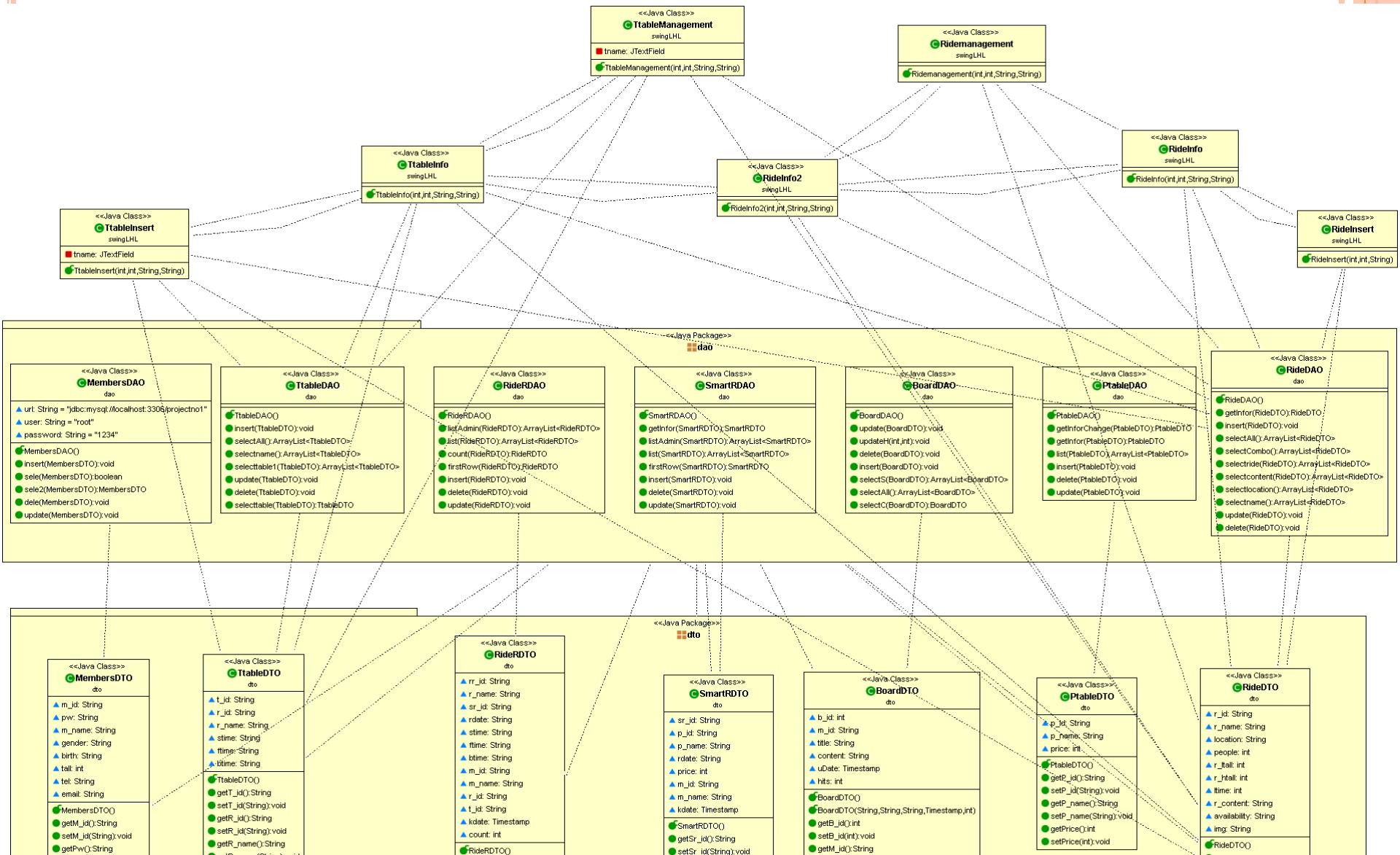
(1) SwingMHW, Join, Update, Login – DTO, DAO 연결상태



1) 개발과정

1-5] UML

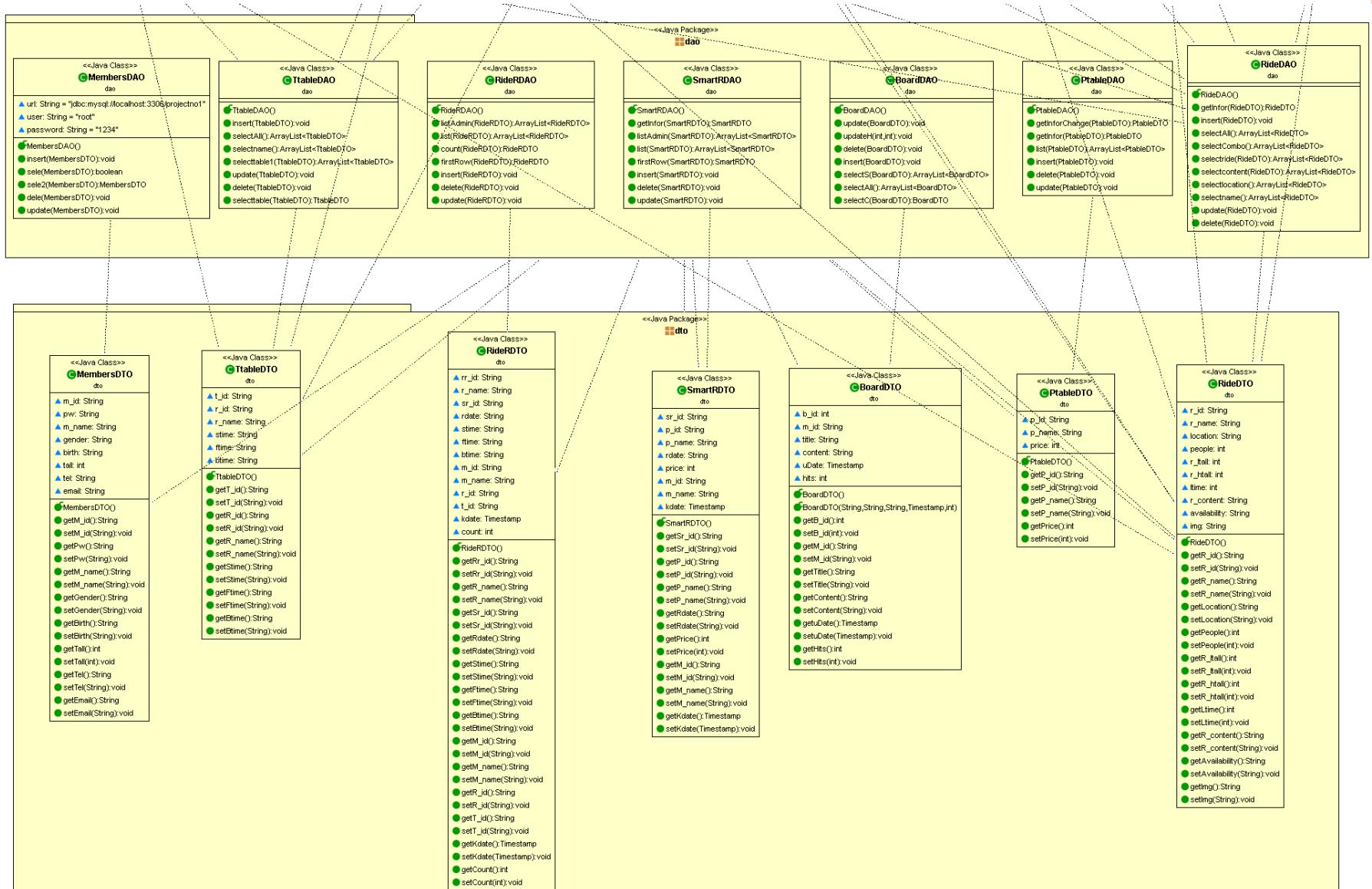
(2) SwingLHL, 놀이기구 정보 열람 시스템



1) 개발과정

1-5] UML

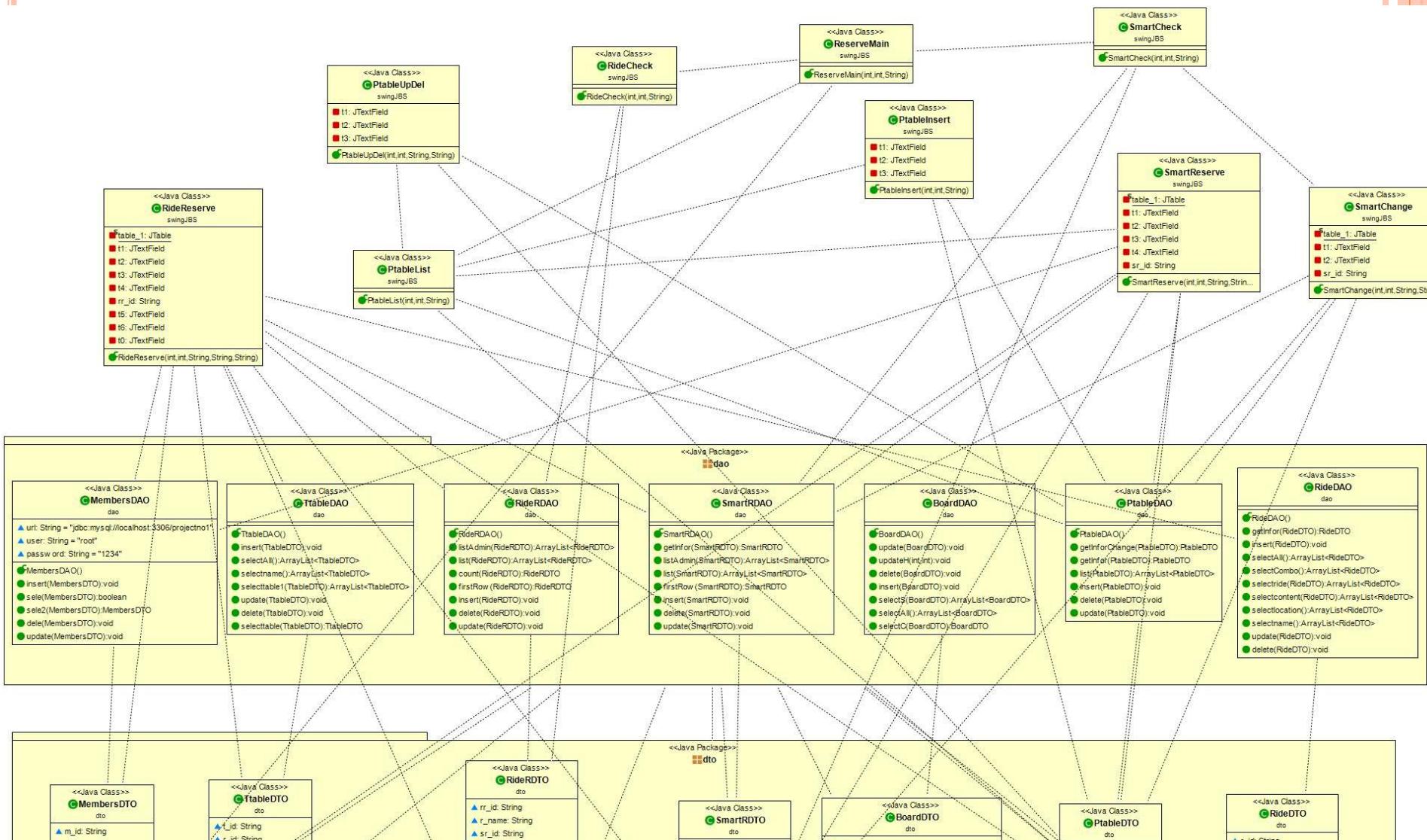
(2) SwingLHL, 놀이기구 정보 열람 시스템 – DTO, DAO 연결상태



1] 개발과정

1-5] UML

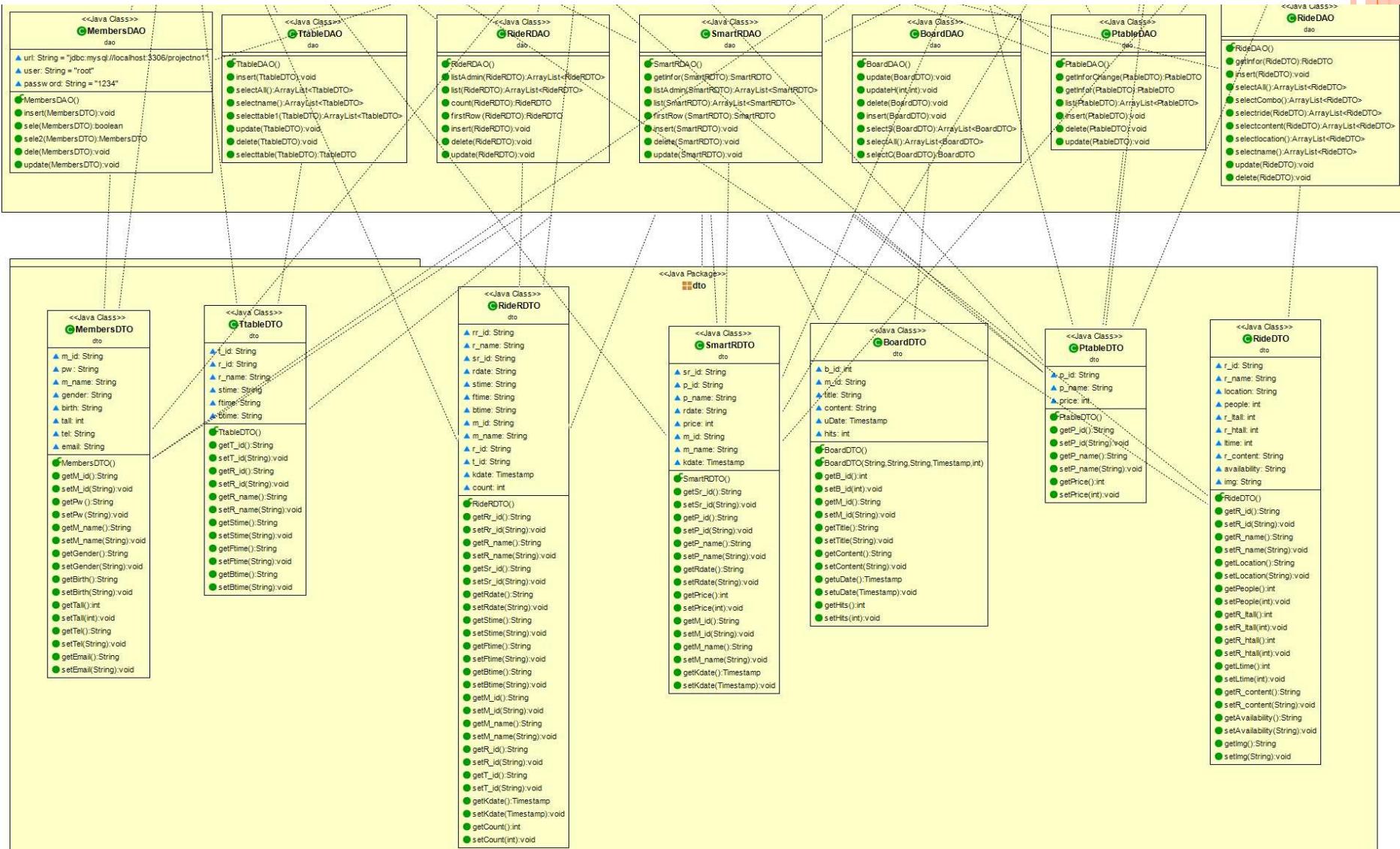
(3) SwingJBS, 놀이기구 예약 시스템



1) 개발과정

1-5] UML

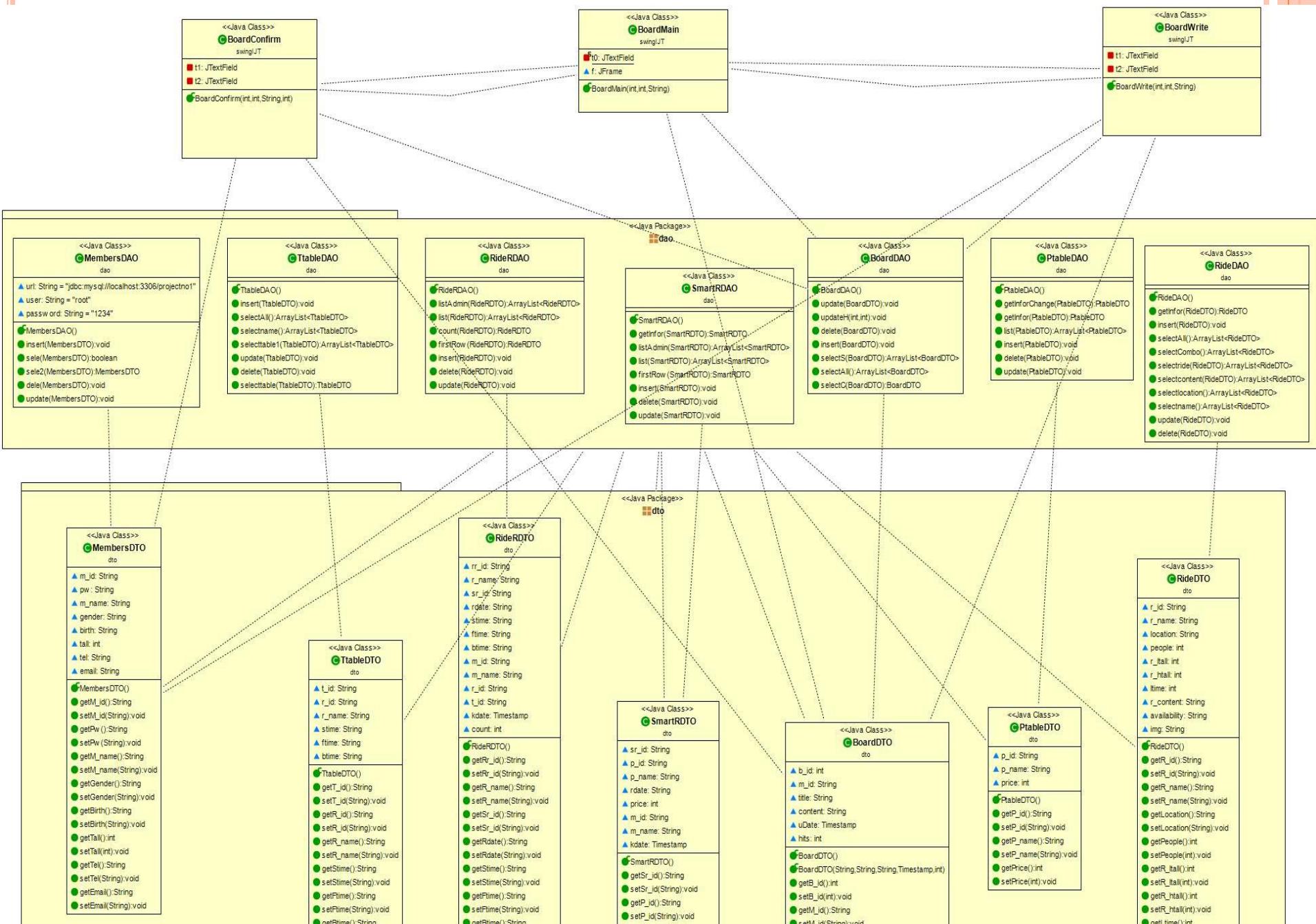
(3) SwingJBS, 놀이기구 예약 시스템 – DTO, DAO 연결상태



1] 개발과정

(4) SwingLJT, 게시판

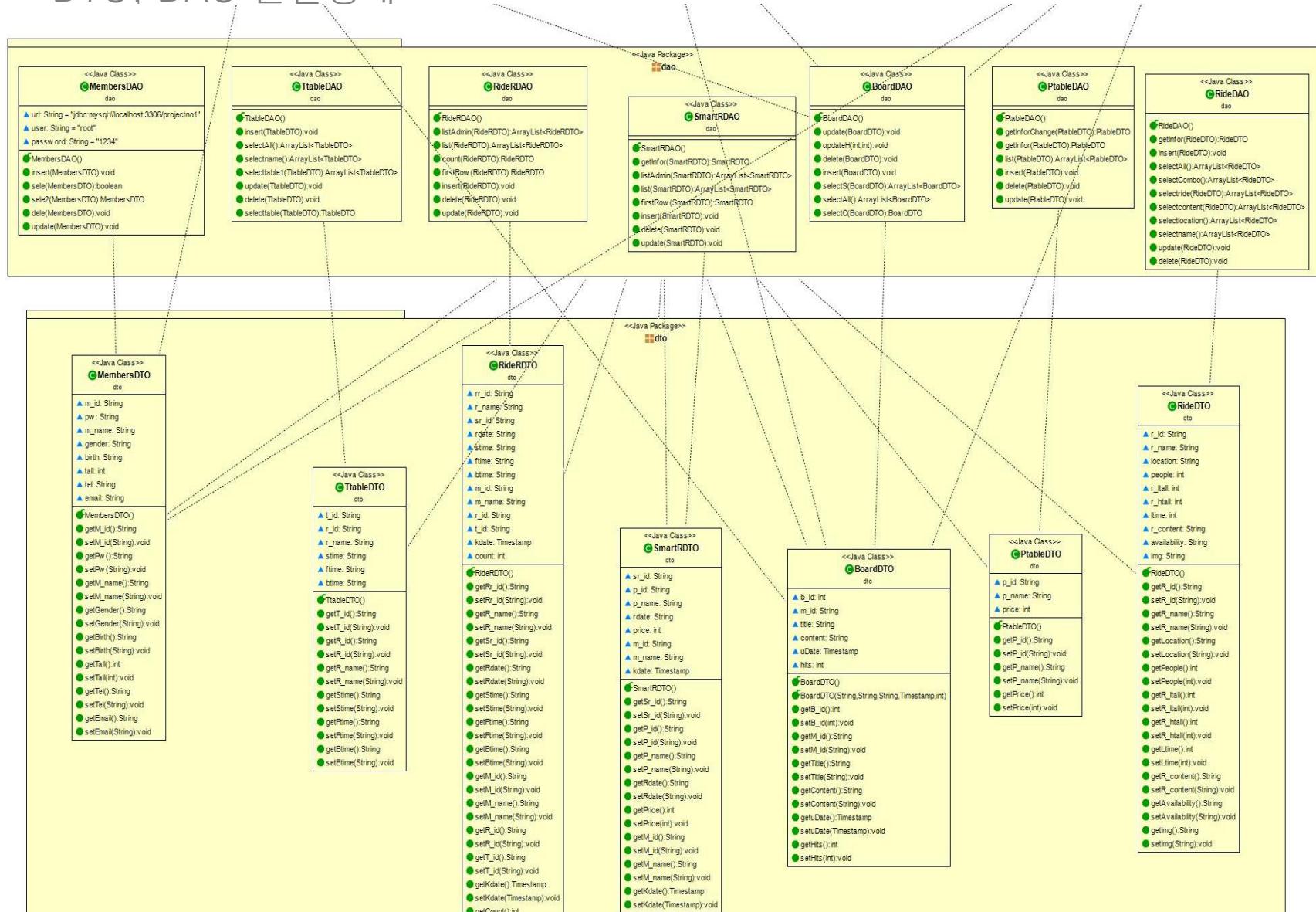
1-5] UML



1) 개발과정

(4) SwingLJT, 게시판 – DTO, DAO 연결상태
– DTO, DAO 연결상태

1-5] UML



2] 기술설명

2-1] CRUD 회원가입 – 민희원

회원가입 화면

회원가입 진행

ID:	<input type="text"/>	*필수 입력값
패스워드:	<input type="password"/>	*필수 입력값
이름:	<input type="text"/>	
성별 (M 혹은 F):	<input type="text"/>	
출생년도:	<input type="text"/>	
키 (cm):	<input type="text"/>	*필수 입력값
전화번호:	<input type="text"/>	
이메일 주소:	<input type="text"/>	

확인 **홈으로**

Join.java

Try, Catch를 사용하여 예러(not null로 설정된 column의 값을 입력하지 않았을 시에 캐치하고 팝업 띄우기)

```
try {
    dao.insert(dto); //DAO의 insert 메소드에 dto
    값을 넣어 실행
    f.dispose(); //확인이 눌려서 정보가 넘어가면
    꺼지고 다시 Start(메인)화면 실행
    Start main = new Start(f.getLocation().x,
    f.getLocation().y, "");
} catch (Exception e1) {
    e1.printStackTrace();
    JOptionPane.showMessageDialog(null, "필수
    값을 입력하지 않으셨습니다!");
}
```

2] 기술설명

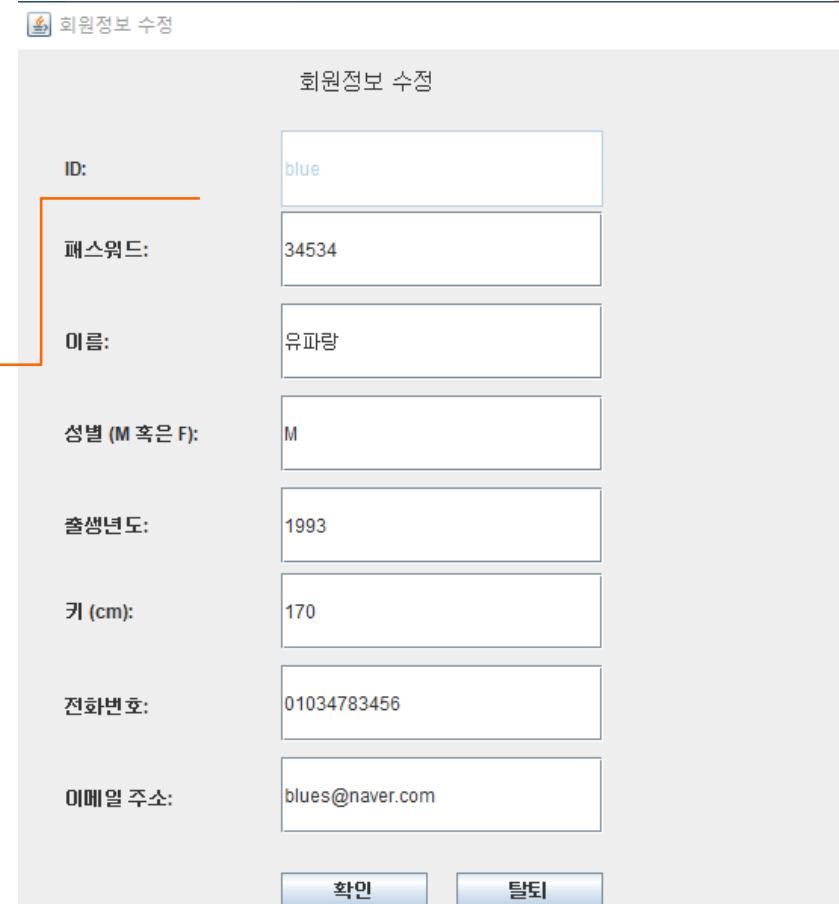
```
MembersDAO dao = new MembersDAO();
//dao, dto 불러오기
MembersDTO dto = new MembersDTO();
dto.setM_id(id); //dto에서 id값 가져오기
dao.sele2(dto); //dao에 있는 sele2실행
//sele메소드로 얻은 로그인 아이디 값을
line32의 String id로 받아 저장함
//sele2는 sele에서 가져온 데이터와 비교하여
id를 조건으로 나머지 정보를 다 가져옴
```

```
t1 = new JTextField(dto.getM_id());
t1.setEnabled(false); //아이디는 수정 불가
하게 설정
t1.setText(dto.getM_id());
t1.setColumns(10);
t1.setBounds(176, 52, 213, 49);
f.getContentPane().add(t1);
```

Update.java

Sele2메소드를 통해서 ID 제외 모든 정보를 수정 가능하게 가져오고 그것을 원래 dto로 보내서 데이터 베이스에 저장함

2-1] CRUD 설명 - 민희원



ID:	blue
패스워드:	34534
이름:	유파랑
성별 (M 혹은 F):	M
출생년도:	1993
키 (cm):	170
전화번호:	01034783456
이메일 주소:	blues@naver.com

확인 탈퇴

2] 기술설명

회원정보 수정

회원정보 수정

ID:	blue
패스워드:	34534
이름:	유파랑
성별 (M 혹은 F):	M
출생년도:	1993
키 (cm):	170
전화번호:	01034783456
이메일 주소:	blues@naver.com

확인 **탈퇴**

Try, Catch를 사용하여 여러(not null)로 설정된 column의 값을 입력하지 않았을 시에 캐치하고 팝업 띄우기

2-1] CRUD 회원정보 수정 – 민희원

Update.java

```
btnG = new JButton("확인"); //수정 완료 후 dto 통해  
서 업데이트 된 정보를 다시 보냄
```

```
btnG.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        dto.setM_id(t1.getText());  
        dto.setPw(t2.getText());  
        dto.setM_name(t3.getText());  
        dto.setGender(t4.getText());  
        dto.setBirth(t5.getText());  
        dto.setTall(Integer.parseInt(t6.getText()));  
        dto.setTel(t7.getText());  
        dto.setEmail(t8.getText());  
    }  
});
```

```
try {  
    dao.update(dto); //dao에 있는 sql 업데이트 문이 dto  
    저장값을 가지고 실행됨  
    f.dispose(); //업데이트 완료 후 창 끄고 다시 메인으로  
    전환
```

```
Start main = new Start(f.getLocation().x,  
f.getLocation().y, id);  
} catch (Exception e1) {  
    // TODO Auto-generated catch block  
    e1.printStackTrace();  
}
```

2] 기술설명

회원정보 수정

회원정보 수정

ID:	blue
패스워드:	34534
이름:	유파랑
성별 (M 혹은 F):	M
출생년도:	1993
키 (cm):	170
전화번호:	01034783456
이메일 주소:	blues@naver.com

확인 **탈퇴**

2-1) CRUD 회원 탈퇴 – 민희원

Update.java

```
button = new JButton("탈퇴");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            dao.delete(dto);
            Start main = new Start(f.getLocation().x,
f.getLocation().y, "");
            f.dispose();
        } catch (Exception e2) {
        }
    }
});
```

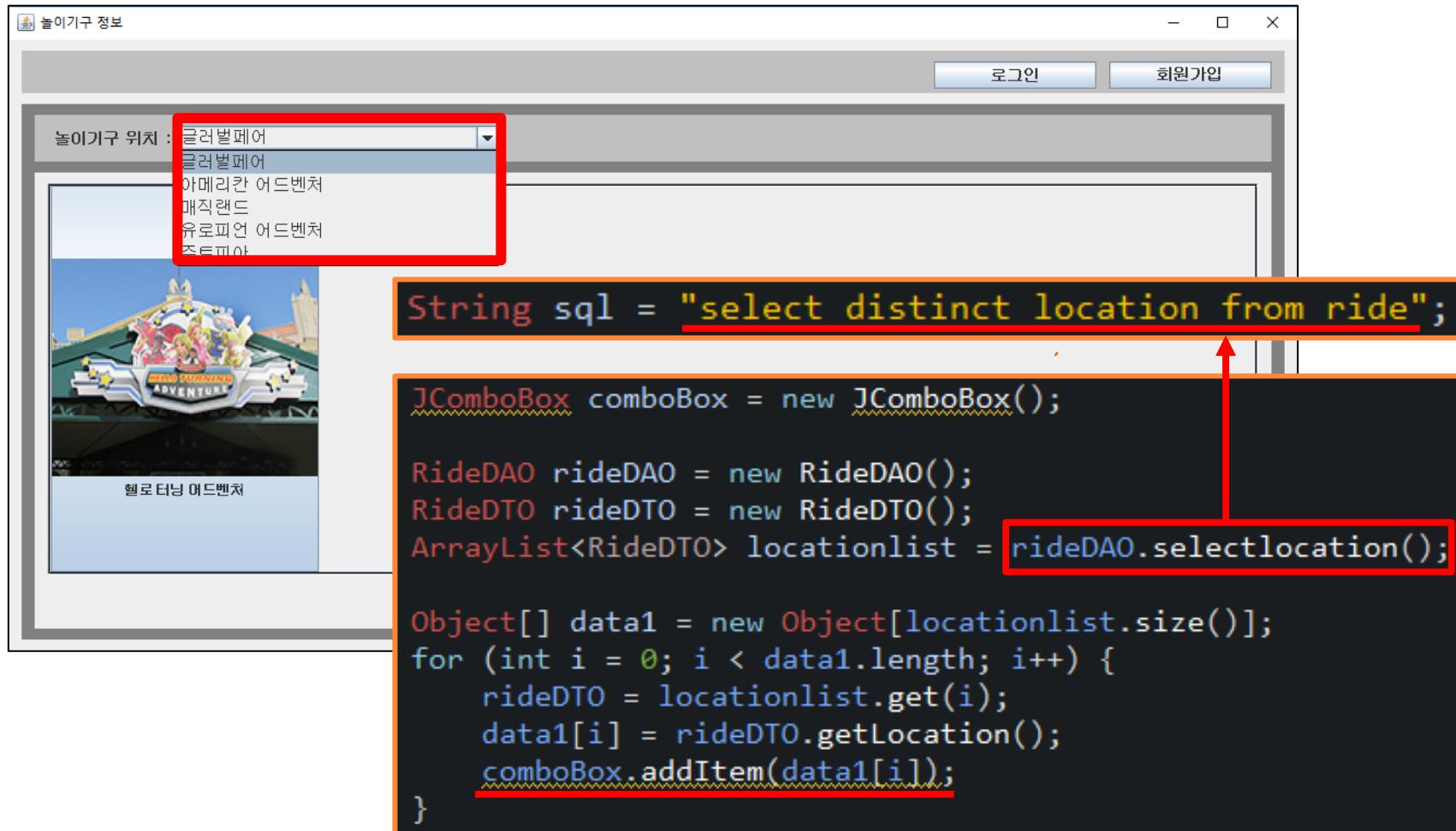
MembersDao.java

```
public void delete(MembersDTO dto) throws Exception {
    Class.forName("com.mysql.jdbc.Driver");
    try {
        Connection con = DriverManager.getConnection(url,
user, password);
        String sql = "delete from members where m_id=?";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, dto.getM_id());
        ps.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련

- ride 테이블에서 놀이기구 위치를 중복제거하고 가져와 콤보박스에 넣기

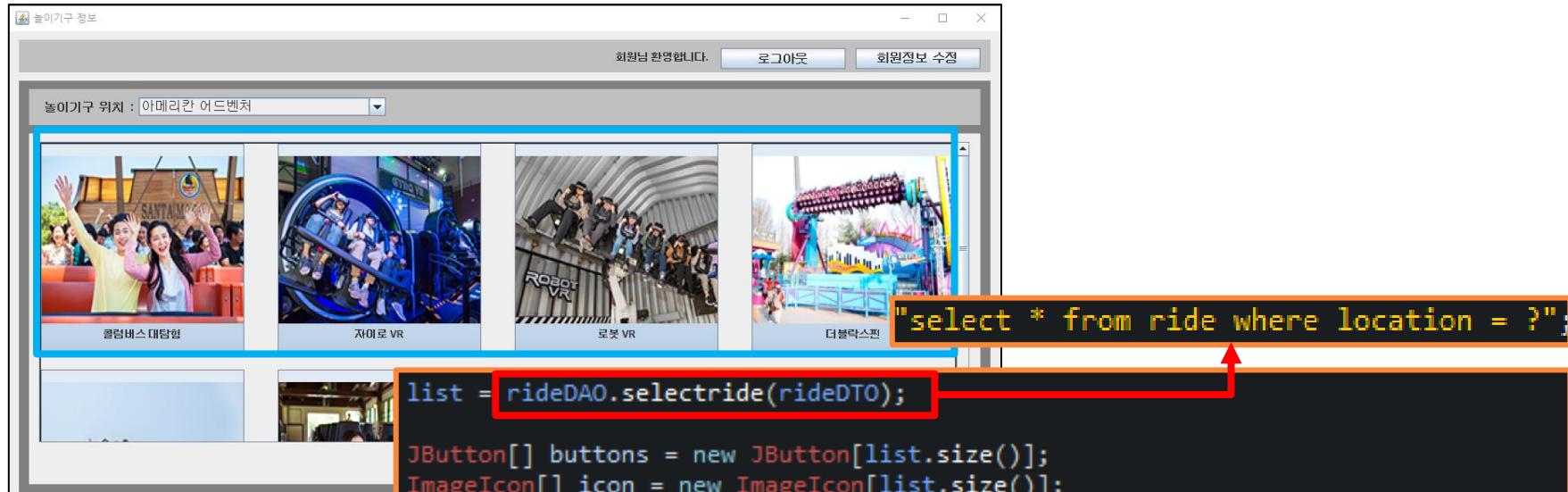


```
String sql = "select distinct location from ride";  
  
JComboBox comboBox = new JComboBox();  
  
RideDAO rideDAO = new RideDAO();  
RideDTO rideDTO = new RideDTO();  
ArrayList<RideDTO> locationlist = rideDAO.selectlocation();  
  
Object[] data1 = new Object[locationlist.size()];  
for (int i = 0; i < data1.length; i++) {  
    rideDTO = locationlist.get(i);  
    data1[i] = rideDTO.getLocation();  
    comboBox.addItem(data1[i]);  
}
```

2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련

– 콤보박스에 일치하는 놀이기구 이미지와 이름을 버튼에 넣기



```
list = rideDAO.selectride(rideDTO);  
  
JButton[] buttons = new JButton[list.size()];  
ImageIcon[] icon = new ImageIcon[list.size()];  
  
for (int i = 0; i < buttons.length; i++) {  
    panel_4.revalidate(); 패널 재배치  
  
    rideDTO = list.get(i);  
    icon[i] = new ImageIcon(rideDTO.getImg());  
    Image originimg = icon[i].getImage();  
    Image changedImg = originimg.getScaledInstance(250, 200, Image.SCALE_SMOOTH);  
    ImageIcon Icon1 = new ImageIcon(changedImg);  
  
    buttons[i] = new JButton(rideDTO.getR_name() + "", Icon1); 버튼에 이미지와  
    buttons[i].setPreferredSize(new Dimension(150, 250));  
    buttons[i].setVerticalTextPosition(AbstractButton.BOTTOM); 레이아웃 넣기  
    buttons[i].setHorizontalTextPosition(AbstractButton.CENTER);
```

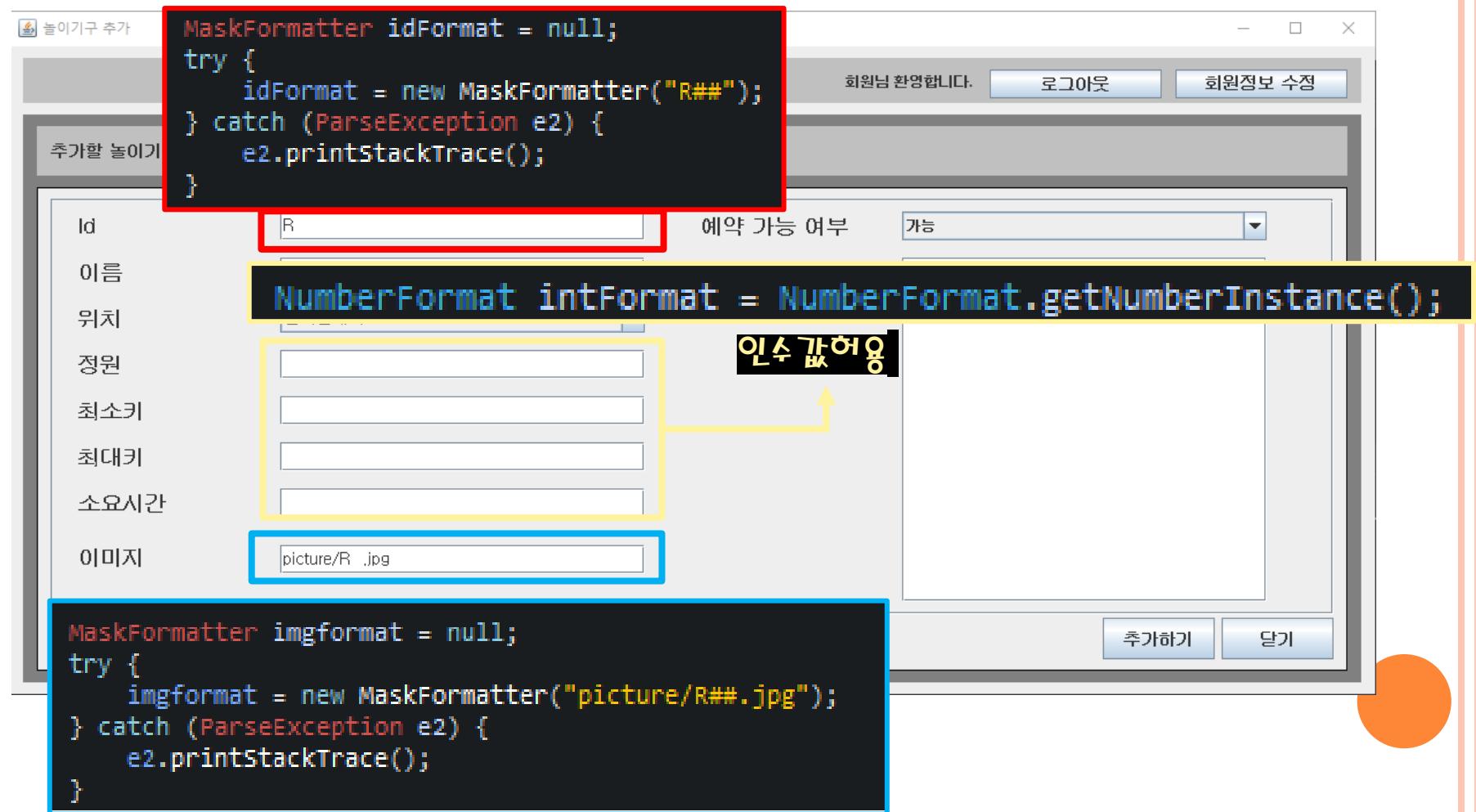
이미지 크기 조절

버튼에 이미지와 레이아웃 넣기

2] 기술설명

2-1] 어플리케이션 기능 설명 – 이해련

-Format을 사용하여 텍스트값 제한설정



2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련

`JOptionPane.showMessageDialog(f, "추가되었습니다.", "추가확인", JOptionPane.INFORMATION_MESSAGE);`

Code snippet from the application:

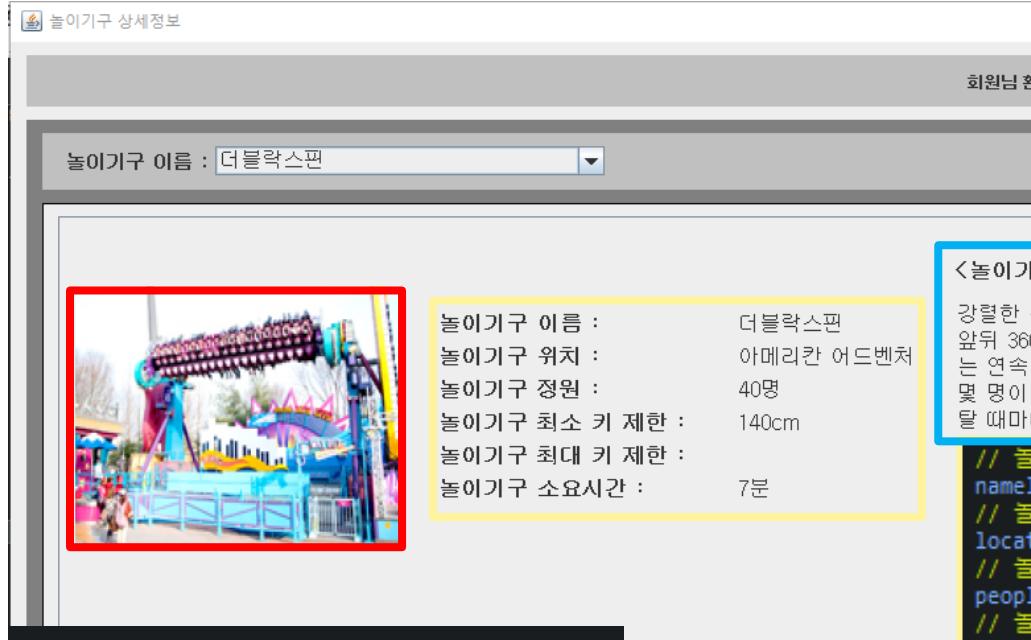
```
 JButton cbtn = new JButton("닫기");
cbtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String location = (String) combolocation.getSelectedItem();
        try {
            f.dispose();
            RideInfo name = new RideInfo(f.getLocation().x, f.getLocation().y, m_id, location);
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }
});
```

Buttons at the bottom of the application windows:

- 권리 (Rights)
- 뒤로가기 (Back)

2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련



```
list = rideDAO.selectcontent(rideDTO);
rideDTO = list.get(0);
```

```
// 해당 놀이기구 이미지
ImageIcon icon = new ImageIcon(rideDTO.getImg());
Image originimg = icon.getImage();
Image changedImg = originimg.getScaledInstance(250, 200, Image.SCALE_SMOOTH);
ImageIcon Icon1 = new ImageIcon(changedImg);
imglabel.setIcon(Icon1);
imglabel.setBounds(12, 12, 250, 200);
panel_4.add(imglabel);
```

```
// 놀이기구 설명
textPane.setText(rideDTO.getR_content());

if(rideDTO.getAvailability().equals("가능")) {
    panel_3.add(tbtn);
    tbtn.setVisible(true);
} else {
    tbtn.setVisible(false);
}
```

<놀이기구 설명>

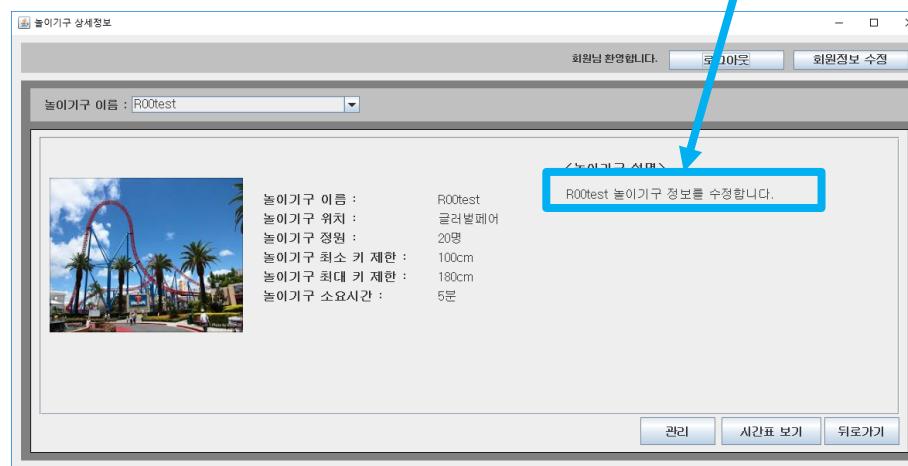
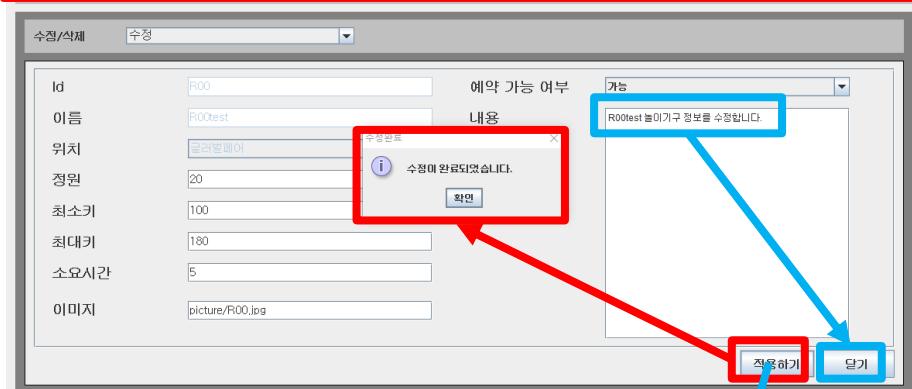
강렬한 족큰을 리듬과 함께 둘고 또 둘고~
앞뒤 360도 회전하는 의자에 앉아 20미터 높이에서 느끼는 연속 4회전의 짜릿함!
몇 명이 탑승하느냐에 따라 스판 횟수와 강도가 달라져
달 때마다 다른 재미까지!!

```
// 놀이기구 이름
namelabel.setText(rideDTO.getR_name());
// 놀이기구 위치
locationlabel.setText(rideDTO.getLocation());
// 놀이기구 정원
peoplelabel.setText(rideDTO.getPeople()+"명");
// 놀이기구 최소 키 제한
if(rideDTO.getR_ltall()==0) {
    ltalllabel.setText("");
} else {
    ltalllabel.setText(rideDTO.getR_ltall()+"cm");
}
// 놀이기구 최대 키 제한
if(rideDTO.getR_htall()==0) {
    htalllabel.setText("");
} else {
    htalllabel.setText(rideDTO.getR_htall()+"cm");
}
// 놀이기구 소요시간
if(rideDTO.getLtime()==0) {
    ltimelabel.setText("");
} else {
    ltimelabel.setText(rideDTO.getLtime()+"분");
}
```

2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련

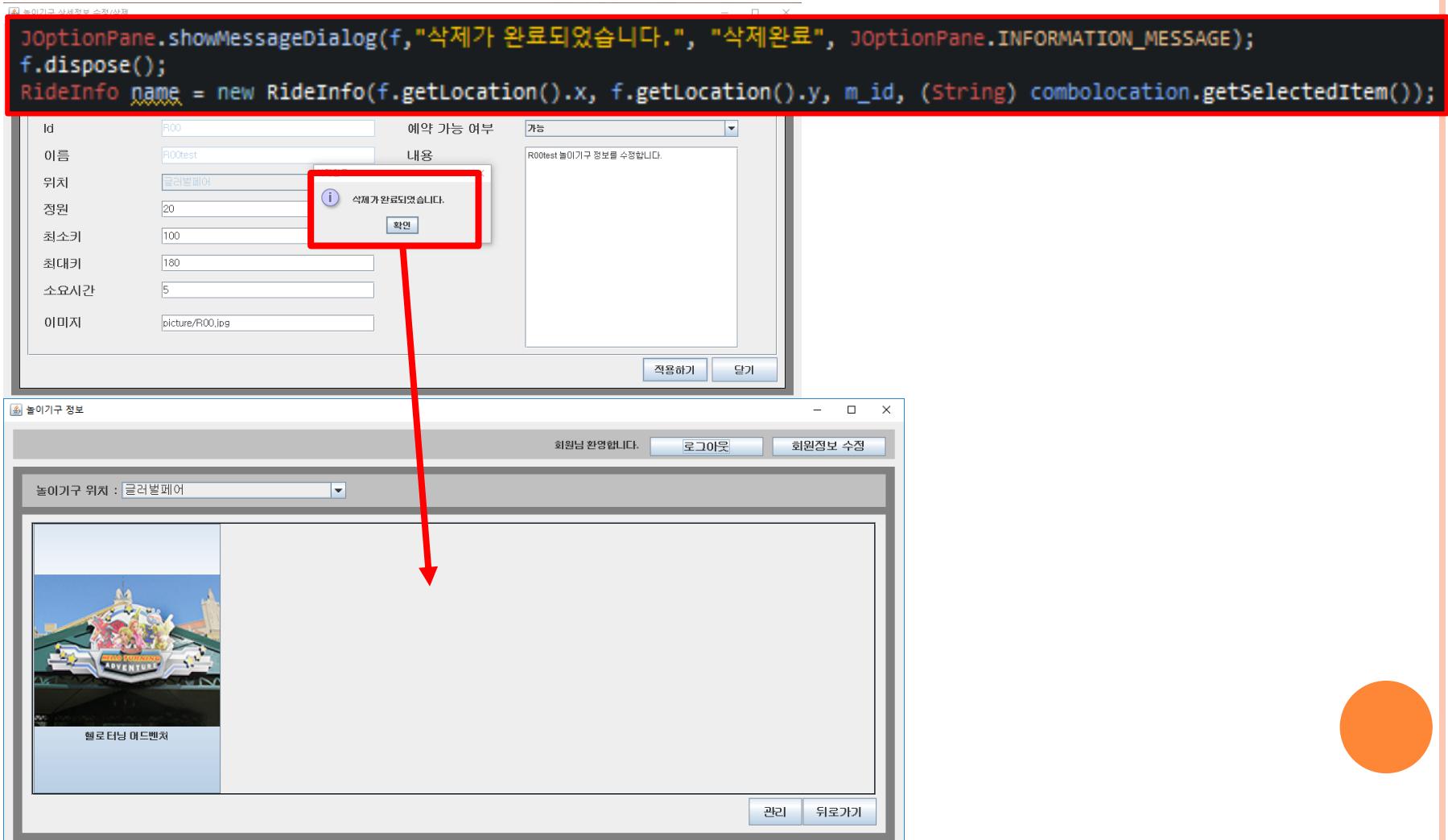
```
JOptionPane.showMessageDialog(f, "수정이 완료되었습니다.", "수정완료", JOptionPane.INFORMATION_MESSAGE);
```



```
dto.setR_id(tid.getText());  
dto.setR_name(tname.getText());  
dto.setLocation((String) combolocation.getSelectedItem());  
dto.setPeople(Integer.parseInt(tpeople.getText()));  
System.out.println("&&&" + tltall.getText() + "&&");  
if(tltall.getText().equals("")) {  
    dto.setR_ltall(0);  
} else {  
    dto.setR_ltall(Integer.parseInt(tltall.getText()));  
}  
if(thall.getText().equals("")) {  
    dto.setR_htall(0);  
} else {  
    dto.setR_htall(Integer.parseInt(thall.getText()));  
}  
if(tltime.getText().equals("")) {  
    dto.setLtime(0);  
} else {  
    dto.setLtime(Integer.parseInt(tltime.getText()));  
}  
dto.setR_content(tcontent.getText());  
dto.setAvailability((String) comboability.getSelectedItem());  
dto.setImg(timg.getText());  
  
dao.update(dto);
```

2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련



2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련

T_id	stime	ftime	btime
R06001	10시 00분	10시 07분	9시 50분
R06002	10시 20분	10시 27분	10시 10분
R06003	10시 40분	10시 47분	10시 30분
R06004	11시 00분	11시 07분	10시 50분
R06005	11시 20분	11시 27분	11시 10분
R06006	11시 40분	11시 47분	11시 30분
R06007	12시 00분	12시 07분	11시 50분
R06008	12시 20분	12시 27분	12시 10분
R06009	12시 40분	12시 47분	12시 30분
R06010	13시 00분	13시 07분	12시 50분
R06011	13시 20분	13시 27분	13시 10분
R06012	13시 40분	13시 47분	13시 30분
R06013	14시 00분	14시 07분	13시 50분
R06014	14시 20분	14시 27분	14시 10분
R06015			
R06016			
R06017			
R06018			
R06019			
R06020			
R06021			

```
String[] items = { "T_id", "stime", "ftime", "btime" };
DefaultTableModel model = new DefaultTableModel(items, 0);

comboBox.setSelectedItem(r_name);

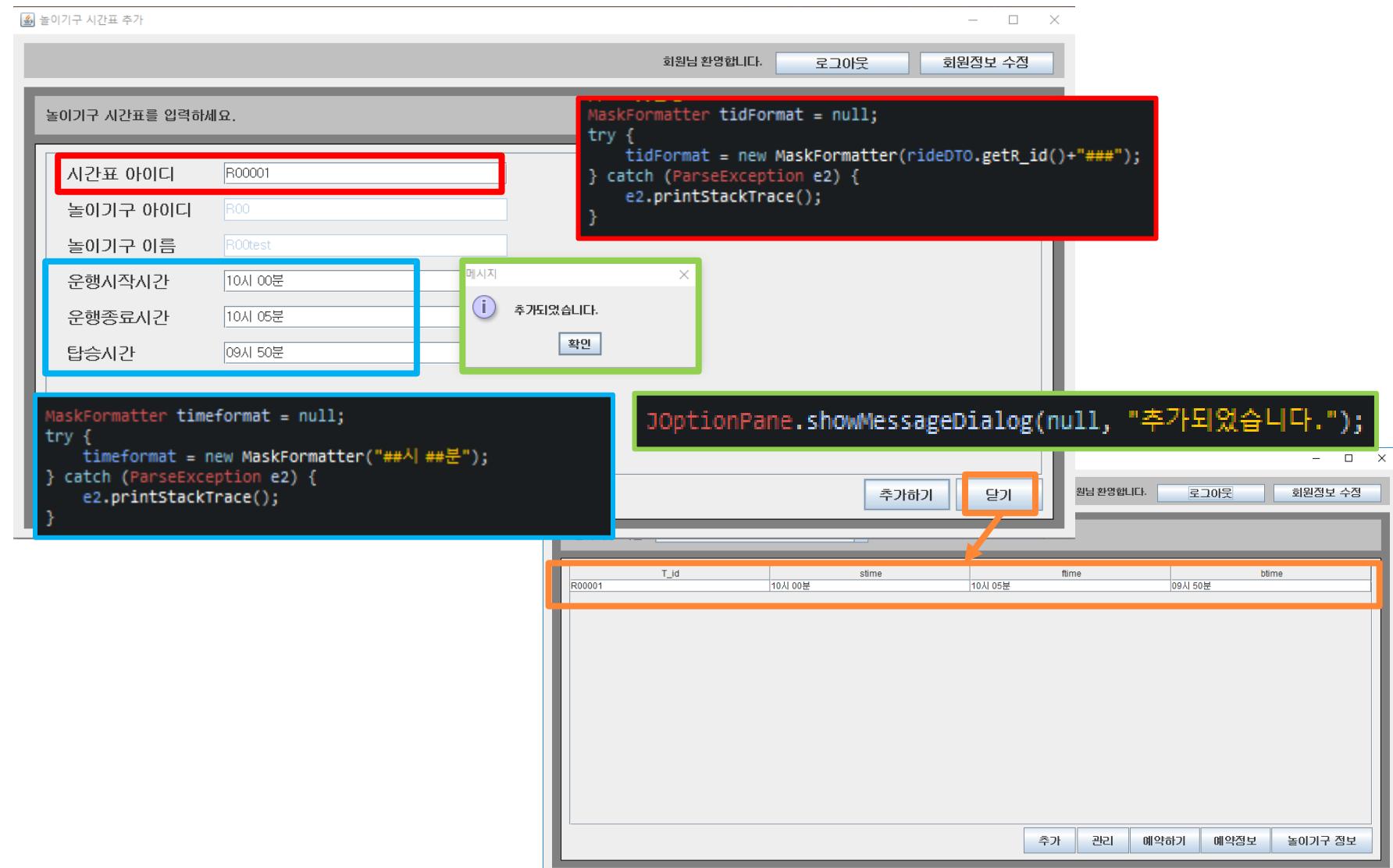
TtableDTO ttableDTO = new TtableDTO();
TtableDAO ttableDAO = new TtableDAO();

String index = (String) comboBox.getSelectedItem();
ttableDTO.setR_name(index);
ArrayList<TtableDTO> list = ttableDAO.selecttable1(ttableDTO);

String[][] data = new String[list.size()][];
for (int i = 0; i < data.length; i++) {
    ttableDTO = list.get(i);
    String[] r = { ttableDTO.getT_id(), ttableDTO.getStime(), ttableDTO.getFtime(), ttableDTO.getBtime() };
    data[i] = r;
    model.addRow(r);
}
```

2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련



2] 기술설명

2-1) 어플리케이션 기능 설명 – 이혜련

The screenshot displays two windows of a Java Swing application. The top window is titled '놀이기구 시간표 수정/삭제' (Amusement Park Schedule Modification/Delete). It contains fields for '시간표 아이디' (Schedule ID), '놀이기구 아이디' (Attraction ID), '놀이기구 이름' (Attraction Name), '운행시작시간' (Operation Start Time), '운행종료시간' (Operation End Time), and '탑승시간' (Boarding Time). A red box highlights the '탑승시간' field, which has the value '09시 55분'. A message dialog box is overlaid on the window, containing the message '수정되었습니다.' (Modified successfully.) with a '확인' (Confirm) button. Red arrows point from the '수정되었습니다.' message to the '수정하기' (Modify) button at the bottom of the main window and to the 'btime' column in the table of the bottom window.

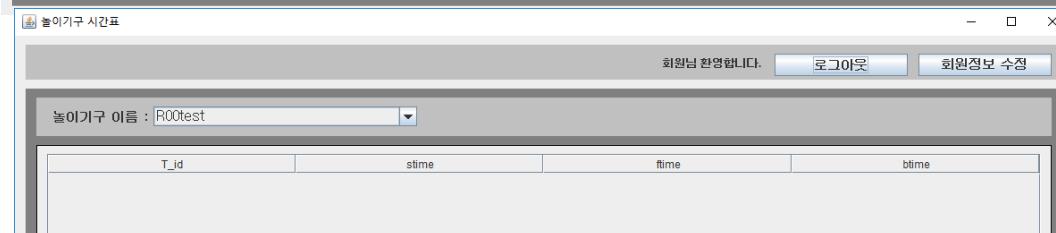
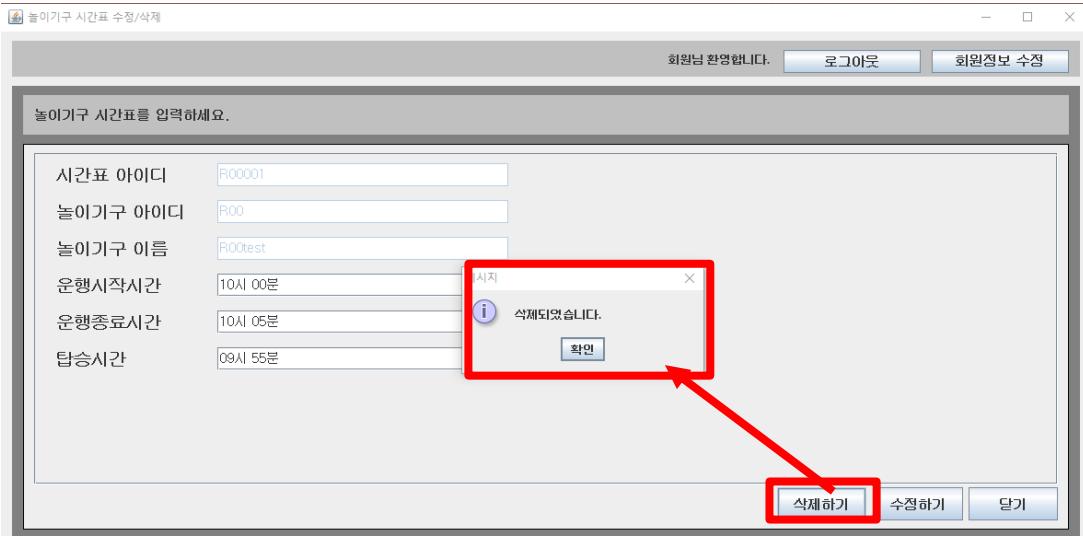
The bottom window is also titled '놀이기구 시간표' (Amusement Park Schedule). It shows a table with columns 'T_Id', 'tstime', 'ftime', and 'btime'. The 'btime' column for the first row (T_Id: R00001) is highlighted with a red box and contains the value '09시 55분'. A red box also highlights the code in the bottom window's text area:

```
ttableDTO.setStime(tstime.getText());
ttableDTO.setFtime(ftime.getText());
ttableDTO.setBtime(btime.getText());

try {
    ttableDAO.update(ttableDTO);
    JOptionPane.showMessageDialog(null, "수정되었습니다.");
    TtableInfo name = new TtableInfo(f.getLocation().x, f.getLocation().y, m_id, ttableDTO.getR_name());
    f.dispose();
}
```

2] 기술설명

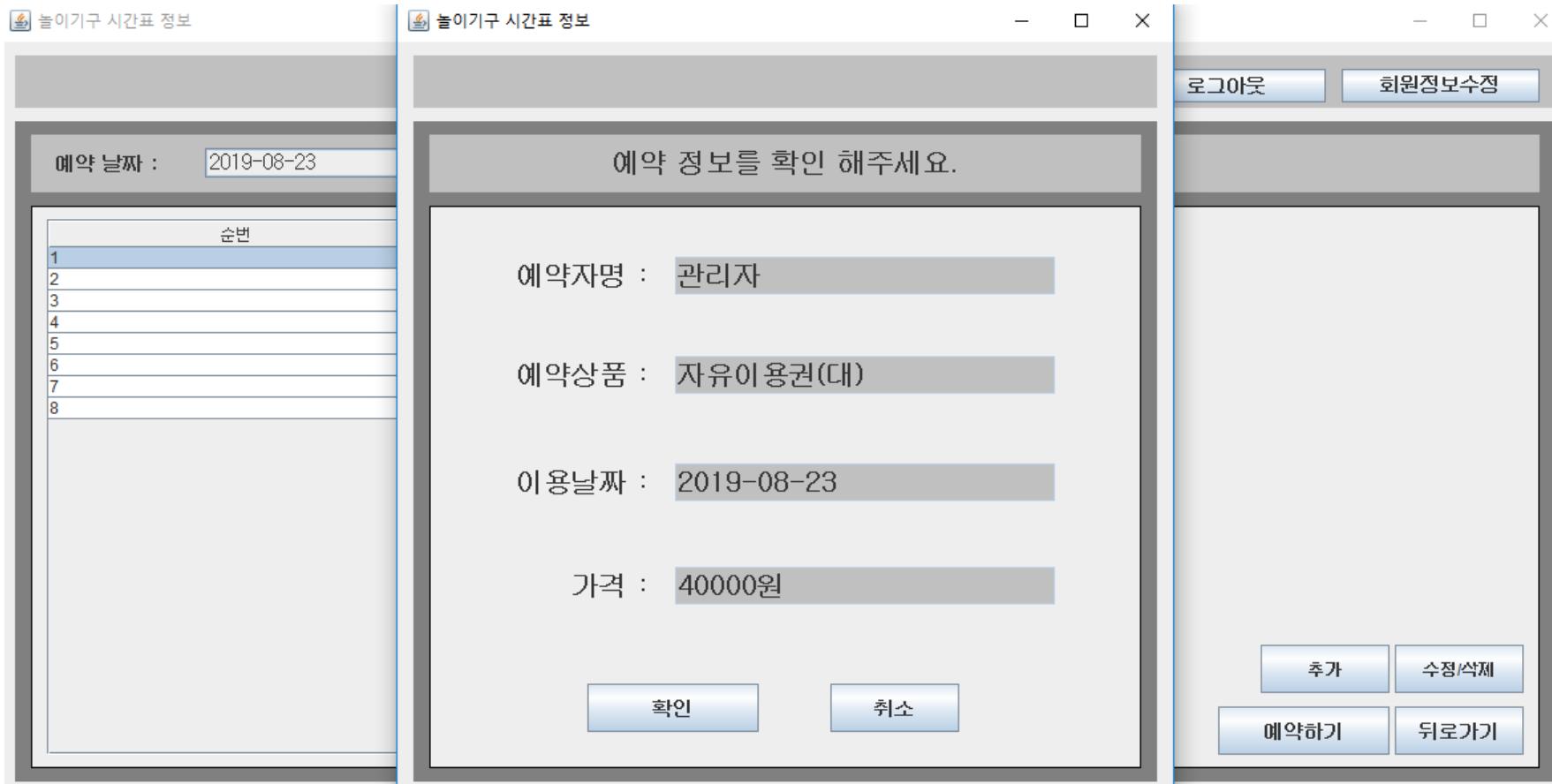
2-1) 어플리케이션 기능 설명 – 이혜련



```
ttableDAO.delete(ttableDTO);
 JOptionPane.showMessageDialog(null, "삭제되었습니다.");
TtableInfo name = new TtableInfo(f.getLocation().x, f.getLocation().y, m_id, ttableDTO.getR_name());
f.dispose();
```

2] 기술설명

2-3) CRUD - 정병수



텍스트 필드에서 DTO를 통해 넘겨 받은 값으로 Insert문 실행하여 예약하는 화면

2] 기술설명

2-3) CRUD - 정병수

The screenshot shows a Java Swing application window titled "놀이기구 시간표 정보". The window has a title bar with icons for minimize, maximize, and close. In the top right corner, there are buttons for "로그아웃" (Logout) and "회원정보수정" (Edit Member Information). The main content area contains a search bar with placeholder "예약번호 :" and a "조회" (Search) button. Below the search bar is a table with 13 rows of reservation data. The table has columns for 예약번호 (Reservation ID), 상품명 (Product Name), 예약날짜 (Reservation Date), 가격 (Price), 예약자ID (Reservee ID), and 예약자 (Reservee Name). The data in the table is as follows:

예약번호	상품명	예약날짜	가격	예약자ID	예약자
20190823000001	자유미용권(대)	2019-08-23	40000	test	테스트
20190823000002	입장권(대)	2019-08-23	15000	test	테스트
20190823000003	워터파크(소)	2019-08-27	24000	test	테스트
20190823000005	자유미용권(소)	2019-08-23	32000	test2	테스트2
20190823000006	자유미용권(대)	2019-08-25	40000	test	테스트
20190823000007	워터파크(대)	2019-08-29	30000	test	테스트
20190823000008	자유미용권(대)	2019-08-24	40000	test	테스트
20190823000009	자유미용권(대)	2019-08-23	40000	test2	테스트2
20190823000010	자유미용권(대)	2019-08-23	40000	test10	테스트10
20190823000011	자유미용권(대)	2019-08-29	40000	test2	테스트2
20190823000012	입장권(대)	2019-08-25	15000	test2	테스트2
20190823000013	자유미용권(대)	2019-08-25	40000	test	테스트

In the bottom right corner of the window, there are two buttons: "예약변경" (Reservation Change) and "뒤로가기" (Back).

Select문 결과의 각 레코드를 DTO에 넣고 전체를 ArrayList로 받아주어 DefaultTableModel에 Row추가하여 조회결과를 보여주는 화면

2] 기술설명

2-3) CRUD - 정병수

예약번호 :

예약번호	상품명
20190823000001	자유미용권(대)
20190823000002	입장권(대)
20190823000003	워터파크(소)
20190823000005	자유미용권(소)
20190823000006	자유미용권(대)
20190823000007	워터파크(대)
20190823000008	자유미용권(대)
20190823000009	자유미용권(대)
20190823000010	자유미용권(대)
20190823000011	자유미용권(대)
20190823000013	자유미용권(대)
20190823000014	자유미용권(대)

예약 정보를 확인 해주세요.

예약번호 : 20190823000014

예약자 : 테스트2

예약상품 : ▼

이용날짜 : 2019-08-25

로그아웃 회원정보수정

예약변경 뒤로가기

텍스트 필드에 있는 예약정보를 받아 Update문을 실행하여 예약을 변경하는 화면

2] 기술설명

2-3) CRUD - 정병수

The image displays three sequential screenshots of a web application interface, likely a booking system for a playground (놀이기구).

Screenshot 1: A search results page titled "놀이기구 시간표 정보". It shows a table with columns "예약번호" (Reservation Number) and "상품명" (Product Name). The table contains 14 rows of data, each showing a reservation number and a product name.

예약번호	상품명
20190823000001	자유미용권(대)
20190823000002	입장권(대)
20190823000003	워터파크(소)
20190823000005	자유미용권(소)
20190823000006	자유미용권(대)
20190823000007	워터파크(대)
20190823000008	자유미용권(대)
20190823000009	자유미용권(대)
20190823000010	자유미용권(대)
20190823000011	자유미용권(대)
20190823000013	자유미용권(대)
20190823000014	자유미용권(대)

Screenshot 2: A confirmation dialog box titled "예약 정보를 확인 해주세요." (Please check the reservation information). It displays the reservation details: "예약번호 : 20190823000014", "예약자 : 테스트2", "예약상품 : 자유미용권(대)", and "이용날짜 : 2019-08-25". At the bottom are three buttons: "예약변경" (Change Reservation), "예약취소" (Cancel Reservation), and "닫기" (Close).

Screenshot 3: A right-hand sidebar with two buttons: "예약변경" (Change Reservation) and "뒤로가기" (Go Back).

텍스트 필드에 있는 예약번호(키값)를 받아 Delete문을 실행하여 해당 예약 내역을 취소하는 화면

2] 기술설명

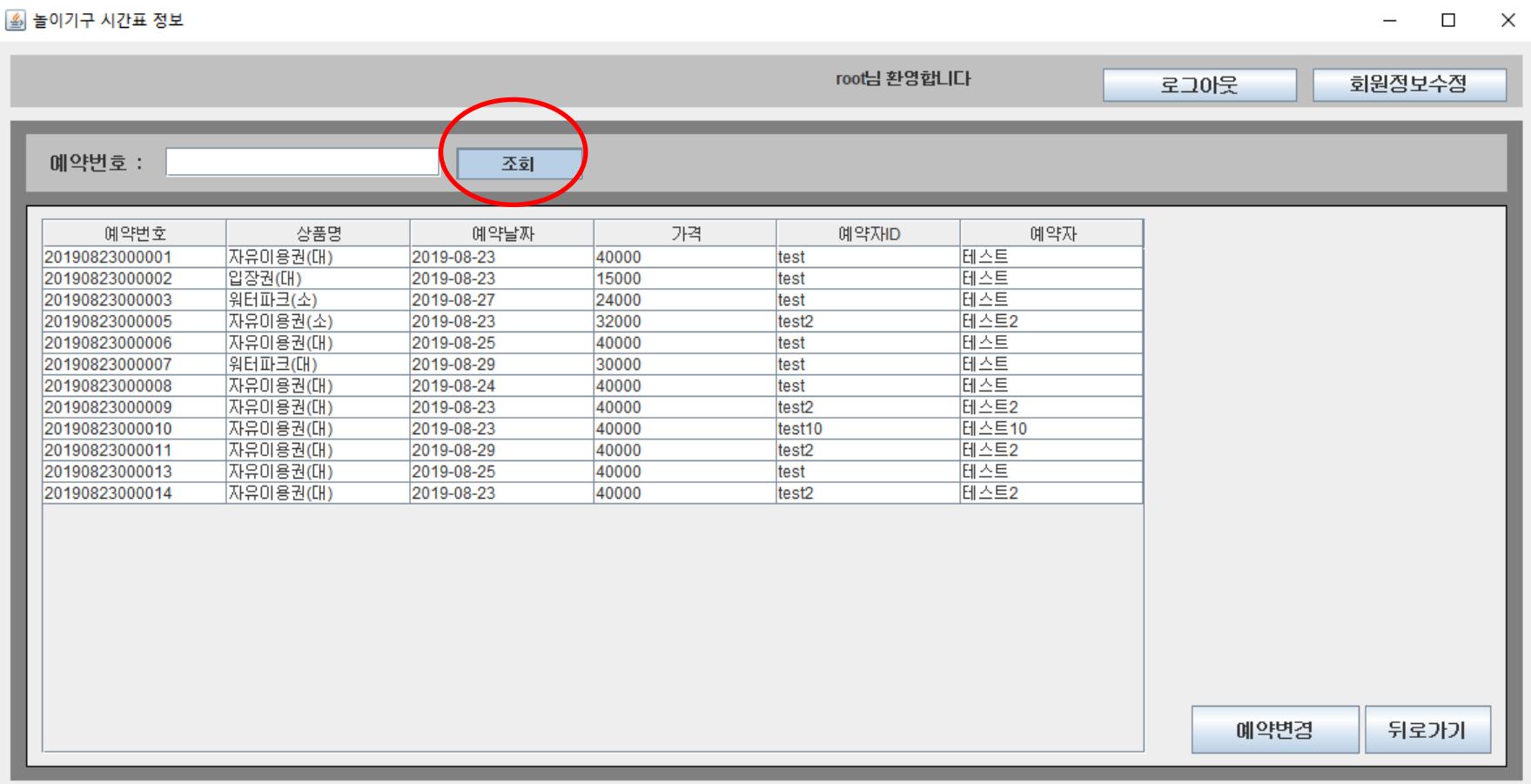
2-3] 예약번호 알고리즘 - 정병수

ABC SR_id	ABC P_id	ABC P_name	ABC rDate	123 price	ABC M_id	ABC M_name	⌚ kdate
20190823000001	▣ a708	자유이용권(대)	2019-08-23	40,000	▣ test	테스트	2019-08-23 11:48:31
20190823000002	▣ b708	입장권(대)	2019-08-23	15,000	▣ test	테스트	2019-08-23 11:48:37
20190823000003	▣ d708s	워터파크(소)	2019-08-27	24,000	▣ test	테스트	2019-08-23 11:48:44
20190823000005	▣ a708s	자유이용권(소)	2019-08-23	32,000	▣ test2	테스트2	2019-08-23 11:49:06
20190823000006	▣ a708	자유이용권(대)	2019-08-25	40,000	▣ test	테스트	2019-08-23 12:03:20
20190823000007	▣ d708	워터파크(대)	2019-08-29	30,000	▣ test	테스트	2019-08-23 12:03:24
20190823000008	▣ a708	자유이용권(대)	2019-08-24	40,000	▣ test	테스트	2019-08-23 12:22:21
20190823000009	▣ a708	자유이용권(대)	2019-08-23	40,000	▣ test2	테스트2	2019-08-23 12:24:10
20190823000010	▣ a708	자유이용권(대)	2019-08-23	40,000	▣ test10	테스트10	2019-08-23 12:27:22
20190823000011	▣ a708	자유이용권(대)	2019-08-29	40,000	▣ test2	테스트2	2019-08-23 12:33:39
20190823000013	▣ a708	자유이용권(대)	2019-08-25	40,000	▣ test	테스트	2019-08-23 12:34:38
20190823000014	▣ a708	자유이용권(대)	2019-08-23	40,000	▣ test2	테스트2	2019-08-23 18:02:58

DateTimeFomatter로 날짜형식과 순차적으로 증가하는 6자리 format을 조합하여 예약번호 생성. 예약번호 생성 시 최근 예약번호를 불러와 당일 예약이 있으면 +1, 없으면 초기값(1)로 예약번호 생성

2] 기술설명

2-3) 팝업 변경 Reflash – 정병수



The screenshot shows a Java Swing application window titled "놀이기구 시간표 정보". The window has a title bar with icons for minimize, maximize, and close. It features a header bar with "root님 환영합니다", "로그아웃", and "회원정보수정" buttons. Below the header is a search bar with a placeholder "예약번호 :" and a blue "조회" button, which is circled in red. The main content area contains a table with 14 rows of reservation data. The columns are labeled: 예약번호, 상품명, 예약날짜, 가격, 예약자ID, and 예약자. The data includes various items like "자유이용권(대)" and "워터파크(소)" with dates ranging from August 23rd to 25th and prices from 15,000 to 40,000. At the bottom right of the main area are two buttons: "예약변경" and "뒤로가기".

예약번호	상품명	예약날짜	가격	예약자ID	예약자
20190823000001	자유이용권(대)	2019-08-23	40000	test	테스트
20190823000002	입장권(대)	2019-08-23	15000	test	테스트
20190823000003	워터파크(소)	2019-08-27	24000	test	테스트
20190823000005	자유이용권(소)	2019-08-23	32000	test2	테스트2
20190823000006	자유이용권(대)	2019-08-25	40000	test	테스트
20190823000007	워터파크(대)	2019-08-29	30000	test	테스트
20190823000008	자유이용권(대)	2019-08-24	40000	test	테스트
20190823000009	자유이용권(대)	2019-08-23	40000	test2	테스트2
20190823000010	자유이용권(대)	2019-08-23	40000	test10	테스트10
20190823000011	자유이용권(대)	2019-08-29	40000	test2	테스트2
20190823000013	자유이용권(대)	2019-08-25	40000	test	테스트
20190823000014	자유이용권(대)	2019-08-23	40000	test2	테스트2

조회 버튼 클릭 시 Select문 조건을 like로 주어 텍스트 필드가 빈 값일 때
모든 예약 내역 출력. 조회 완료 후 텍스트 필드(검색창) 초기화.
WindowFocusListener매서드로 화면 선택 시 조회 버튼 클릭 되도록 정의

2] 기술설명 [게시판]

: 글 작성,수정,삭제 권한

BoardConfirm.java

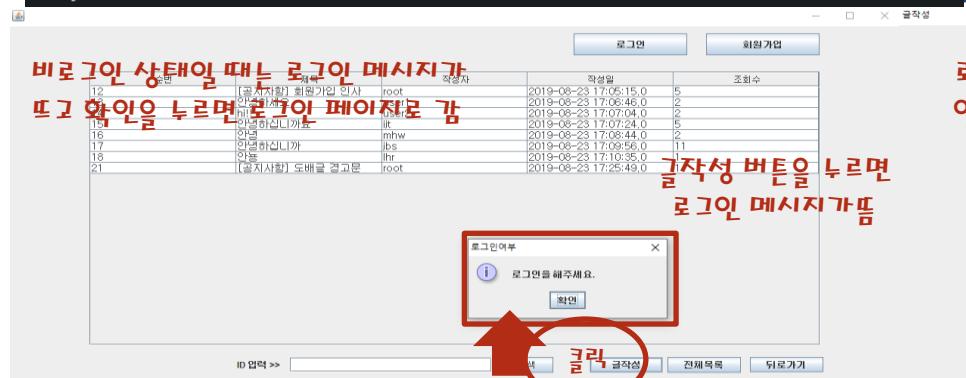
BoardConfirm.java

비로그인 상태이거나 본인이 아닌 사용자는 수정,삭제 버튼을 못보게 하고 본인이거나 관리자(root)일 경우에는 수정,삭제 버튼을 보이게 하여 권한을 부여.

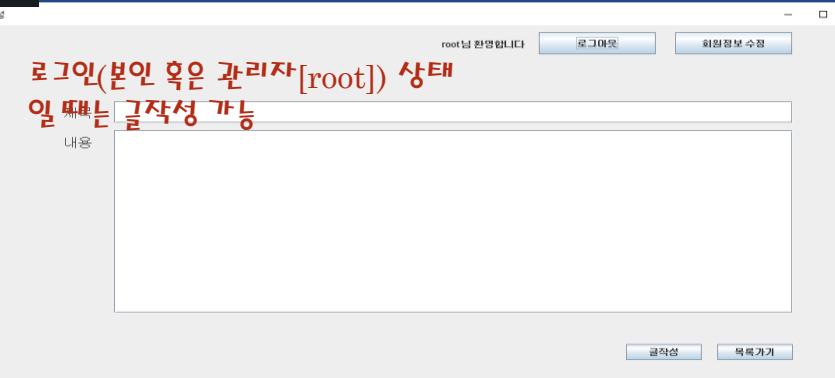
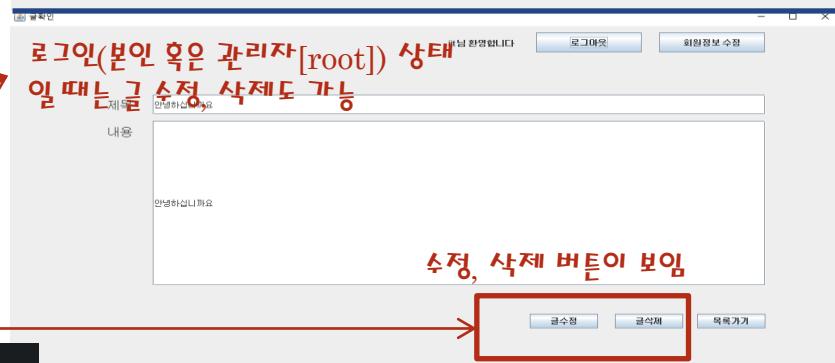
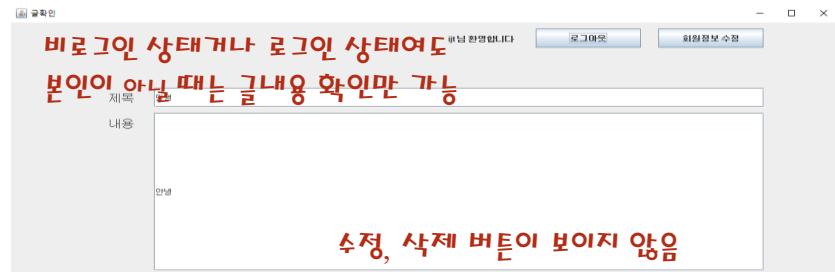
글작성 또한 로그인시에만 사용가능

```
// 수정,삭제 권한 부여
//로그인한 아이디 m_id가 관리자(root) or dto에 입력되어 있는 m_id와 같다면 수정버튼을 보이게 아니라면 안보이게
if (m_id.equals("root") || m_id.equals(boardDTO.getM_id())) {
    panel.add(b_w); //글수정버튼 조건에 맞으면 보이게
}

// 글작성 권한 부여
//m_id가 null같이 아니라면(즉 로그인 상태라면) 글작성화면으로 이동, 다르다면 로그인메시지 띄우고 로그인화면으로
if (!m_id.equals("...")) {
    //글작성화면으로 이동
    BoardWrite name = new BoardWrite(f.getLocation().x, f.getLocation().y, m_id);
    f.dispose();
} else {
    JOptionPane.showMessageDialog(null, "로그인을 해주세요.", "로그인여부", 1);
    LogIn login = new LogIn(f.getLocation().x, f.getLocation().y);
    f.dispose();
}
```



2-4] 작성,수정,삭제 권한 설명 - 임정통



2] 기술설명 [게시판]

: 글 전체목록 보여주기[select 사용]

BoardDAO.java

```
//select문 (JTable위에 있는 model 위에 '전체리스트'를 뿌려주기 위해 생성)
public ArrayList<BoardDTO> selectAll() throws Exception { // 예외처리
    ArrayList<BoardDTO> list = new ArrayList();

    // 3. SQL문 결정
    String sql = "select b_id, title, m_id, uDate, hits from board";
    PreparedStatement ps = conn.prepareStatement(sql);
    System.out.println("3. SQL문 객체화 ok...");

    // 4. SQL문 실행 요청
    ResultSet rs = ps.executeQuery();

    while (rs.next()) {
        BoardDTO dto = new BoardDTO();

        int b_id = rs.getInt(1);
        String title = rs.getString(2);
        String m_id = rs.getString(3);
        Timestamp uDate = rs.getTimestamp(4);
        int hits = rs.getInt(5);

        dto.setB_id(b_id);
        dto.setTitle(title);
        dto.setM_id(m_id);
        dto.setuDate(uDate);
        dto.setHits(hits);

        list.add(dto);
    }

    System.out.println("4. SQL문 실행 요청 OK...");

    rs.close();
    ps.close();
    con.close();
}

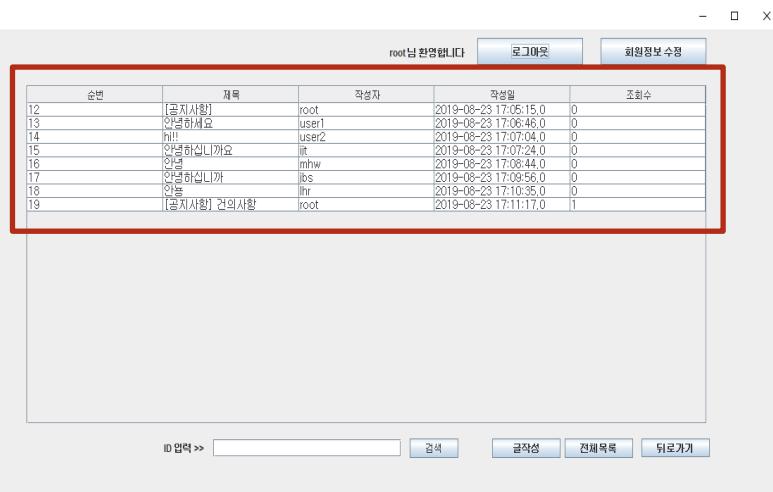
return list; // list로 둘어서 반환
```

BoardMain.java

```
String[] items = { "순번", "제목", "작성자", "작성일", "조회수" };
DefaultTableModel model = new DefaultTableModel(items, 0);
BoardDAO dao = new BoardDAO();

ArrayList<BoardDTO> list = dao.selectAll();
Object[][] data = new Object[list.size()][];
for (int i = 0; i < list.size(); i++) {
    BoardDTO dto = list.get(i);
    Object[] row = { dto.getB_id(), dto.getTitle(), dto.getM_id(), dto.getuDate(), dto.getHits() };
    model.addRow(row);
}
```

2-4) CRUD(select전체) – 임정통



BoardDAO.java

테이블 전체항목을 BoardMain(전체 목록화면)에 나타내주기 위해
ArrayList타입 SelectAll메소드 생성

BoardMain.java

SelectAll메소드를 불러와 2차원 배열 list를 생성하여 model에 나타내고 for반복문을 사용하여 list로 추가했었던 dto의 각 항목값들을 가져오고 record로 계속 추가함

2] 기술설명 [게시판]

: 글 검색 목록 보여주기 [selectS 사용]

BoardDAO.java

```
// select문 BoardMain에 정의해놓음
public ArrayList<BoardDTO> selectS(BoardDTO dto) throws Exception { // 예외처리
    ArrayList<BoardDTO> list = new ArrayList<>();

// 3. SQL문 작성
String sql = "select b_id, title, m_id, uDate, hits from board where m_id like concat('%',?, '%')";
//유사한 아이디입니다. ? , % 와 %% 같은식으로
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, dto.getM_id());
System.out.println("3. SQL문 객체화 ok...");

// 4. SQL문 실행 요청
ResultSet rs = ps.executeQuery();
while (rs.next()) {
    dto = new BoardDTO();
    int b_id = rs.getInt(1);
    String title = rs.getString(2);
    String m_id = rs.getString(3);
    Timestamp uDate = rs.getTimestamp(4);
    int hits = rs.getInt(5);

    dto.setB_id(b_id);
    dto.setTitle(title);
    dto.setM_id(m_id);
    dto.setuDate(uDate);
    dto.setHits(hits);
    |
    list.add(dto);
}
System.out.println("4. SQL문 실행 요청 OK...");

rs.close();
ps.close();
con.close();

return list; / list로 둘어서 반환
}
```

BoardMain.java

```
// 버튼(검색) 클릭하기, 검색할 id를 입력하고 누르면 해당하는 글을 검색해서 띄우게 해야함
JButton b_d = new JButton("검색");
b_d.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        model.setRowCount(0);

        BoardDTO dto = new BoardDTO();
        dto.setM_id(t0.getText());
        ArrayList<BoardDTO> listS;
        try {
            listS = dao.selectS(dto);

            Object[][] dataS = new Object[listS.size()][];
            for (int i = 0; i < listS.size(); i++) {
                dto = listS.get(i);
                Object[] rowS = { dto.getB_id(), dto.getTitle(), dto.getM_id(), dto.getuDate(), dto.getHits() };
                model.addRow(rowS);

                t0.setText("");
            }
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }
});
```

2-4) CRUD(selectS검색) – 임정통

순번	제목	작성자	작성일	조회수
12	[공지사항]	root	2019-08-23 17:05:15.0	0
13	안녕하세요	user1	2019-08-23 17:06:46.0	0
14	hi!	user2	2019-08-23 17:07:04.0	0
15	안녕하십니까요	jih	2019-08-23 17:07:24.0	0
16	안녕	mhw	2019-08-23 17:08:44.0	0
17	안녕하십니까	jbs	2019-08-23 17:09:56.0	0
18	안녕	ihm	2019-08-23 17:10:35.0	0
19	[공지사항] 검의사항	root	2019-08-23 17:11:17.0	1



순번	제목	작성자	작성일	조회수
12	[공지사항]	root	2019-08-23 17:05:15.0	0
19	[공지사항] 검의사항	root	2019-08-23 17:11:17.0	1

BoardDAO.java

검색된 테이블 항목을 나타내주기 위해 ArrayList타입 SelectS메소드 생성

BoardMain.java

SelectS메소드를 불러와 2차원 배열 list를 생성하여 model에 나타냄

2] 기술설명 [게시판]

: 글 작성기능(insert 사용)

BoardDAO.java

```
//insert(BoardWrite에 '글작성'에서 사용)
    public void insert(BoardDTO dto) throws Exception {

// 3. SQL문 결정
    String sql = "insert into board (m_id, title, content, uDate, hits) values (?, ?, ?, now(), 0)";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, dto.getM_id());
    ps.setString(2, dto.getTitle());
    ps.setString(3, dto.getContent());
    System.out.println("3. SQL문 결정 성공...");

// 4. SQL문 전송 요청
    ps.executeUpdate();
    System.out.println("4. SQL문 전송 요청 성공...");

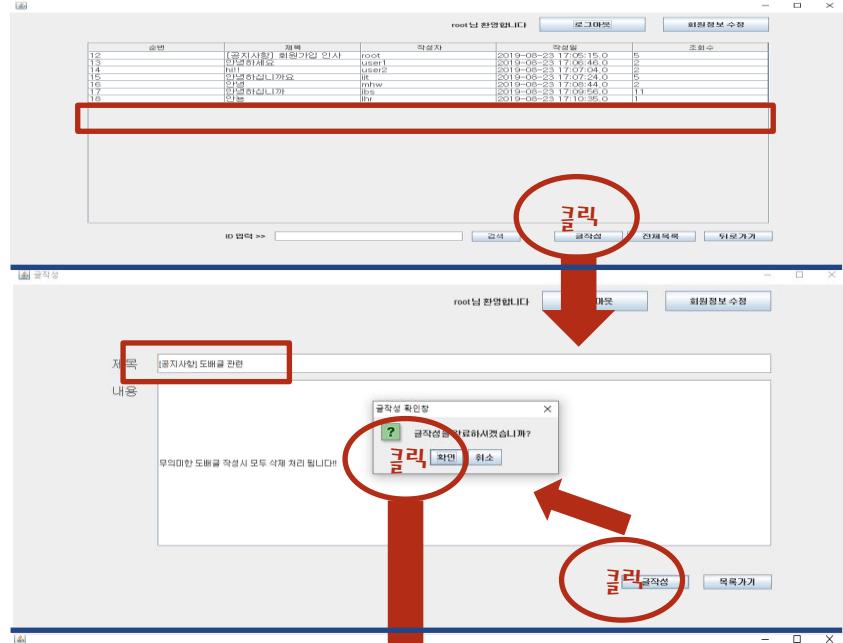
    ps.close();
    con.close();
}
```

BoardWrite.java

```
//버튼(글작성) 블러오기
    JButton b_w = new JButton("글작성");
    b_w.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            //글작성 확인창 띄우기
            int data = JOptionPane.showConfirmDialog
            (null, "글작성을 완료하시겠습니까?", "글작성 확인창", JOptionPane.OK_CANCEL_OPTION);
            //컨펌창에서 확인 누를 때
            if (data == 0) {
                //t1에 입력된 텍스트 값을 변수 title로 할당
                String title = t1.getText();
                //t2에 입력된 텍스트 값을 변수 content로 할당
                String content = t2.getText();

                BoardDTO dto = new BoardDTO();
                try {
                    BoardDAO dao = new BoardDAO();
                    dto.setM_id(m_id);
                    dto.setTitle(title);
                    dto.setContent(content);
                    |
                    dao.insert(dto);
                }
                catch (Exception e2) {
                    e2.printStackTrace();
                }
            }
        }
    });
}
```

2-4) CRUD(insert)설명 - 임정통



BoardDAO.java

테이블의 각 항목들에 값을 입력해주기 위해 insert메소드 생성

BoardWrite.java

컨펌창에서 글작성확인 버튼을 누르면 입력받은 값들을 insert메소드의 dto에 입력

2] 기술설명 (게시판)

: 글 수정기능(update 사용)

BoardDAO.java

```
//update(BoardConfirm의 '글수정'에서 사용)
public void update(BoardDTO dto) throws Exception {
// 3. SQL문 결정
String sql = "update board set title = ?, content = ? where b_id = ?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, dto.getTitle());
ps.setString(2, dto.getContent());
ps.setInt(3, dto.getB_id());

System.out.println("3. SQL문 결정 성공...");

// 4. SQL문 전송 요청
ps.executeUpdate();
System.out.println("4. SQL문 전송 요청 성공...");

ps.close();
con.close();
```

BoardConfirm.java

```
/A 버튼(글수정) 클릭하기
JButton b_w = new JButton("글수정");
b_w.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // 글수정 확인창 띄우기
        int data = JOptionPane.showConfirmDialog(null, "글수정을 원료하시겠습니까?", "글수정 확인창",
                JOptionPane.OK_CANCEL_OPTION);
        // 컨펌창에서 확인을 누를경우
        if (data == 0) {
            boardDTO.setTitle(t1.getText());
            boardDTO.setContent(t2.getText());
            boardDTO.setM_id(m_id);

            BoardDAO dao = new BoardDAO();
            try {
                dao.update(boardDTO);
            } catch (Exception e2) {
                e2.printStackTrace();
            }
        }
    }
});
```

2-4) CRUD(update제목,내용)-임정통



BoardDAO.java

테이블 title, content의 값을 수정하기 위해 update메소드 생성

BoardConfirm.java

컨펌창에서 글수정확인 버튼을 누르면 입력받은 값들을 update메소드의 dto에 입력

2] 기술설명 (게시판)

: 조회수 기능(update 사용)

BoardDAO.java

```
//update(hits, '조회수'에 사용)
//변수 2개만 있어도 된다니까 변수에 b_id, hits 입력
public void updateH(int b_id, int hits) throws Exception {

// 3. SQL문 결정
String sql = "update board set hits = ? where b_id = ?";
PreparedStatement ps = con.prepareStatement(sql);

ps.setInt(1, hits);
ps.setInt(2, b_id);

System.out.println("3. SQL문 결정 성공...");

// 4. SQL문 전송 요청
ps.executeUpdate();
System.out.println("4. SQL문 전송 요청 성공...");

ps.close();
con.close();
}
```

BoardConfirm.java

```
public void mouseClicked(MouseEvent e) {
    int index = (int) model.getValueAt(table.getSelectedRow(), 0);

//hits(조회수)
    int hits = (int) model.getValueAt(table.getSelectedRow(), 4);

    hits++;

    BoardDAO dao = new BoardDAO();
    try {
        BoardConfirm name = new BoardConfirm(f.getLocation().x, f.getLocation().y, m_id, index);

        dao.updateH(index, hits);
    } catch (Exception e1) {
        e1.printStackTrace();
    }
}

f.dispose();
```

2-4) CRUD(update 조회수) - 임정통

The screenshot shows a Java Swing application window titled 'root님 환영합니다'. It contains a table with columns: 순번 (Index), 제목 (Title), 작성자 (Author), 작성일 (Date), and 조회수 (Views). The table has 21 rows. Row 17 is highlighted with a red box and circled, showing the value '0' in the 조회수 column. Row 18 is also highlighted with a red box.

순번	제목	작성자	작성일	조회수
12	[공지사항] 회원가입 인사	root	2019-08-23 17:05:15.0	3
13	안녕하세요	user1	2019-08-23 17:06:46.0	1
14	hi!	user2	2019-08-23 17:07:04.0	1
15	안녕하십니까요	root	2019-08-23 17:07:24.0	1
16	안녕하세요	lrrw	2019-08-23 17:08:44.0	1
17	안녕하십니까	lrs	2019-08-23 17:09:56.0	0
18	안녕	lrr	2019-08-23 17:10:35.0	0
21	[공지사항] 도배글 경고문	root	2019-08-23 17:25:49.0	1

The screenshot shows the same Java application window after the update. The table now shows row 17 with a value of 10 in the 조회수 column, indicating the view count has been updated.

순번	제목	작성자	작성일	조회수
12	[공지사항] 회원가입 인사	root	2019-08-23 17:05:15.0	3
13	안녕하세요	user1	2019-08-23 17:06:46.0	1
14	hi!	user2	2019-08-23 17:07:04.0	1
15	안녕하십니까요	root	2019-08-23 17:07:24.0	1
16	안녕하세요	lrrw	2019-08-23 17:08:44.0	1
17	안녕하십니까	lrs	2019-08-23 17:09:56.0	10
18	안녕	lrr	2019-08-23 17:10:35.0	0
21	[공지사항] 도배글 경고문	root	2019-08-23 17:25:49.0	1

BoardDAO.java

글 확인 시마다 조회수를 증가시키기 위해 updateH 메소드 생성

BoardConfirm.java

글을 클릭하여 확인할 때마다 조회수가 증가되게 구현.
Dao에서 생성한 updateH 메소드를 불러와 index값을 기준으로 hits를 증가시키도록 함

2] 기술설명 (게시판)

: 글 삭제기능(delete 사용)

BoardDAO.java

```
//delete(BoardConfirm에 '글삭제'에서 사용)
public void delete(BoardDTO dto) throws Exception {
    // 1. connector(driver) 설정
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("1. connector설정 성공...");

    String url = "jdbc:mysql://localhost:3306/projectno1?characterEncoding=utf8";
    String user = "root";
    String password = "1234";

    // 2. DB연결
    Connection con = DriverManager.getConnection(url, user, password); // 오버로딩 중 3개짜리 선택
    System.out.println("2. db연결 성공...");

    // 3. SQL문 설정
    String sql = "delete from board where b_id = ?"; // 쿼리문에 따옴표x
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setInt(1, dto.getB_id());
    System.out.println("3. SQL문 결정 성공...");

    // 4. SQL문 전송 요청
    ps.executeUpdate();
    System.out.println("4. SQL문 전송 요청 성공...");

    ps.close();
    con.close();
}
```

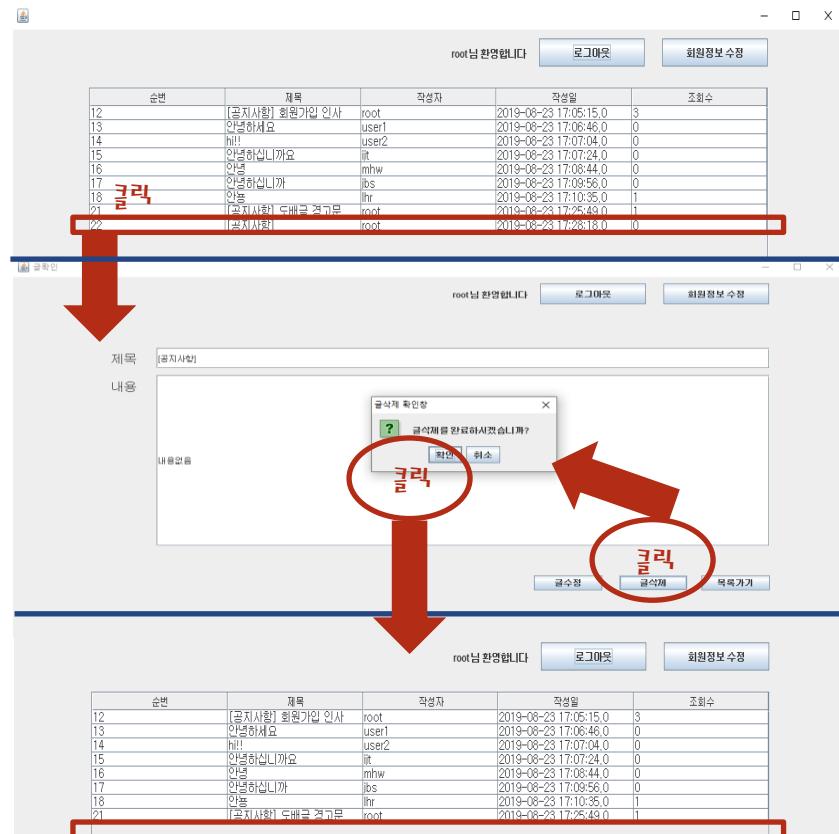
BoardConfirm.jav

```
//글삭제 버튼(글삭제) 불러오기
JButton b_d = new JButton("글삭제");
b_d.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // 글삭제 확인창 띄우기
        int data = JOptionPane.showConfirmDialog(null, "글삭제를 완료하시겠습니까?", "글삭제 확인창",
                JOptionPane.OK_CANCEL_OPTION);
        // 컨펌창에서 확인을 누른경우
        if (data == 0) {
            boardDTO.setB_id(b_id);
            boardDTO.setM_id(m_id);

            BoardDAO dao = new BoardDAO();

            try {
                dao.delete(boardDTO);
                BoardMain name = new BoardMain(f.getLocation().x, f.getLocation().y, m_id);
                f.dispose();
            } catch (Exception e2) {
                e2.printStackTrace();
            }
        }
    }
});
```

2-4) CRUD(delete)설명 - 임정통



BoardDAO.java

테이블 레코드를 삭제하기 위해
delete메소드 생성

BoardConfirm.java

컨펌창에서 글삭제 확인 버튼을 누르
delete메소드를 사용하여 Board dto
의 값들을 삭제

“ 감사합니다.”
Q & A ?

