

1차 프로젝트

코로나 바이러스 현황

3조 박경민, 박창호

목차

1. 개발과정

- 1-1) 보유/ 사용 기술 설명
- 1-2) 개발 기간
- 1-3) DB 테이블
- 1-4) 순서도/UML

2. 기술설명

- 2-1) 박창호 로그인, 회원탈퇴/정보수정 홈,
국내 확진자 현황 화면 구성
- 2-2) 박경민 회원가입, 시·도별 확진자 현황,
게시판 구성

3. 소감문

- 3-1) 박창호
- 3-2) 박경민

1. 개발과정

1-1) 보유/ 사용 기술 설명



1. 개발과정

1-2) 개발 기간

		8월18일	8월19일	8월20일	8월21일
공동		프로젝트 기획 주제 설정	프로젝트 기획 마감	개인분량 설정 UI 샘플 제작	UI 샘플 제작
	8월24일	8월25일	8월26일	8월27일	8월28일
박경민	회원가입 UI 제작	회원가입 코딩, DAO,VO	회원가입 DB설계/연동	회원가입 추가 코딩	회원가입 추가 코딩
박창호	로그인 UI 제작 홈 UI 제작	로그인 UI제작, 코딩 홈 UI제작, 코딩	국내확진자 UI 제작 DAO,VO 제작	UI,VO 제작	코드 간결화, 수정
	9월07일	9월08일	9월09일	9월10일	9월11일
박경민	시도별 확진자 UI 제작	시도별 확진자 크롤링	게시판 UI 제작 (목록, 글쓰기, 수정, 삭제)	게시판DAO,VO	게시판 기능 구현 (목록, 글쓰기, 수정, 삭제)
박창호	UI,DB연결	DAO, VO 코드 수정	로그인 UI 기능 (회원정보수정, 탈퇴, 아이디찾기)	UI, DB 연결 테스트 및 수정	최종 구현 테스트
	9월14일	9월15일	9월16일	9월17일	
박경민	게시판 추가 코딩	게시판 추가 코딩 PPT 발표자료 제작	PPT 발표자료 제작, 완성	발표	
박창호	최종 구현 테스트	1차 합치기 PPT 발표자료 제작	동영상 제작 발표자료 완성		

Login 테이블

속성	속성타입	제약 조건	내용
User_id	Varchar(20)	Not null Primary	회원 아이디
User_pw	Varchar(20)	Not null	회원 비밀번호
User_name	Varchar(20)	Not null	회원 이름
User_tel	Varchar(20)	Not null	회원 전화번호
User_birth	Varchar(20)	Not null	회원 생년월일

Board 테이블

속성	속성타입	제약 조건	내용
No	int(10)	Not null, Primary Auto Increment	게시판 번호 자동 증가
Title	Varchar(20)	Not null	게시판 제목
Writer	Varchar(10)	Not null	게시판 작성자
Content	Varchar(1000)	Not null	게시판 내용
Create_day	Varchar(10)		게시판 작성 날짜
Create_time	Varchar(10)		게시판 작성 시간

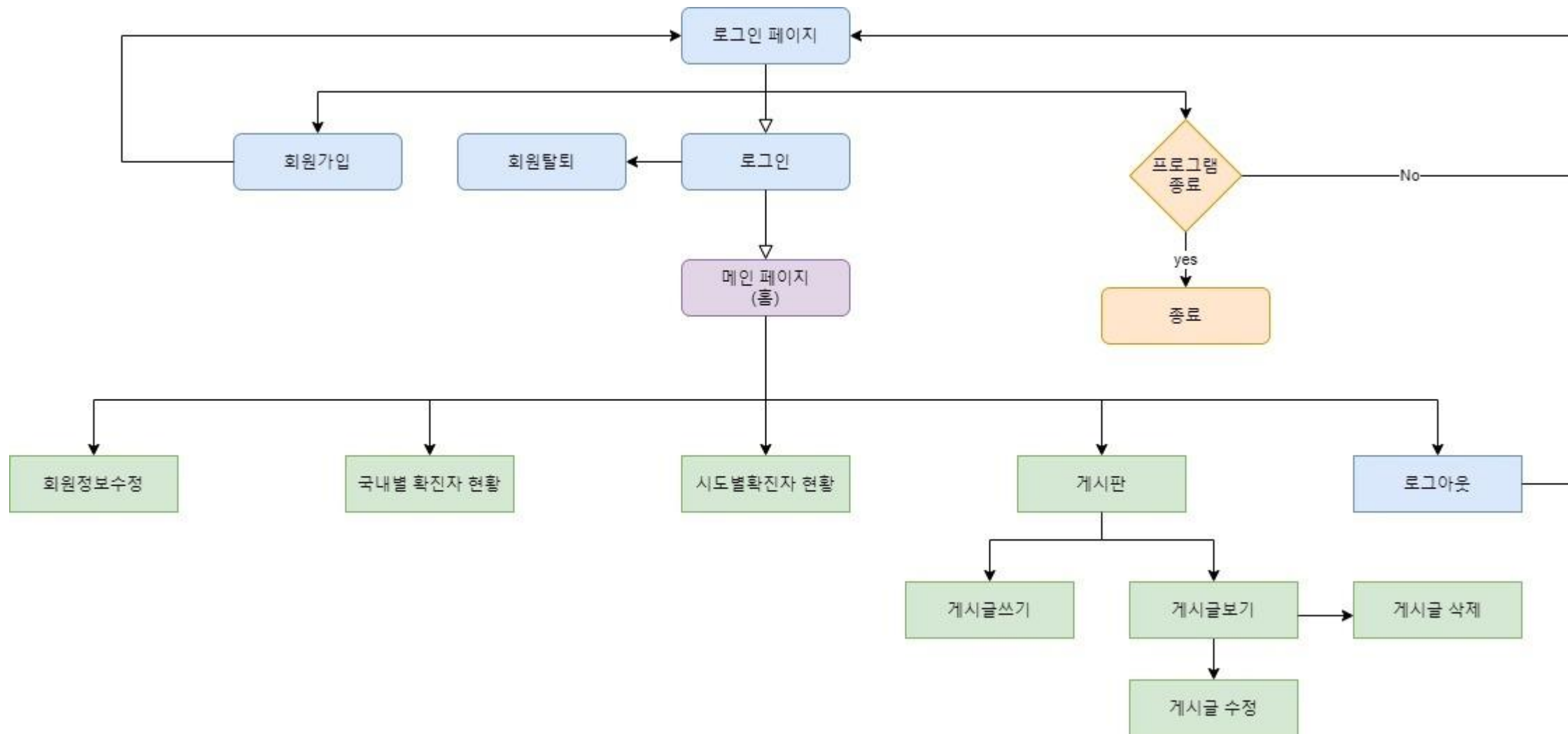
1. 개발과정

1-4) tree



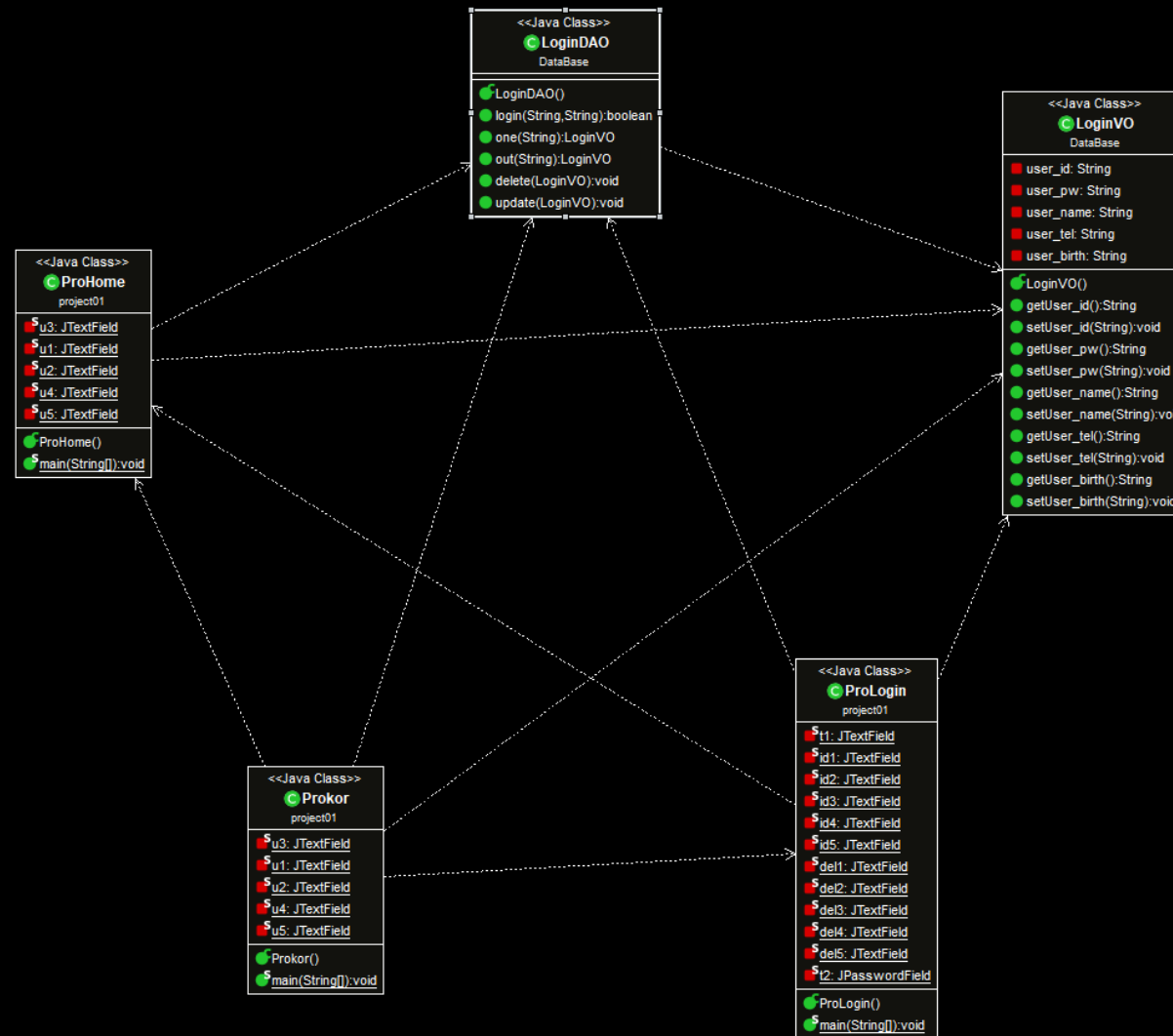
1. 개발과정

1-4) 순서도



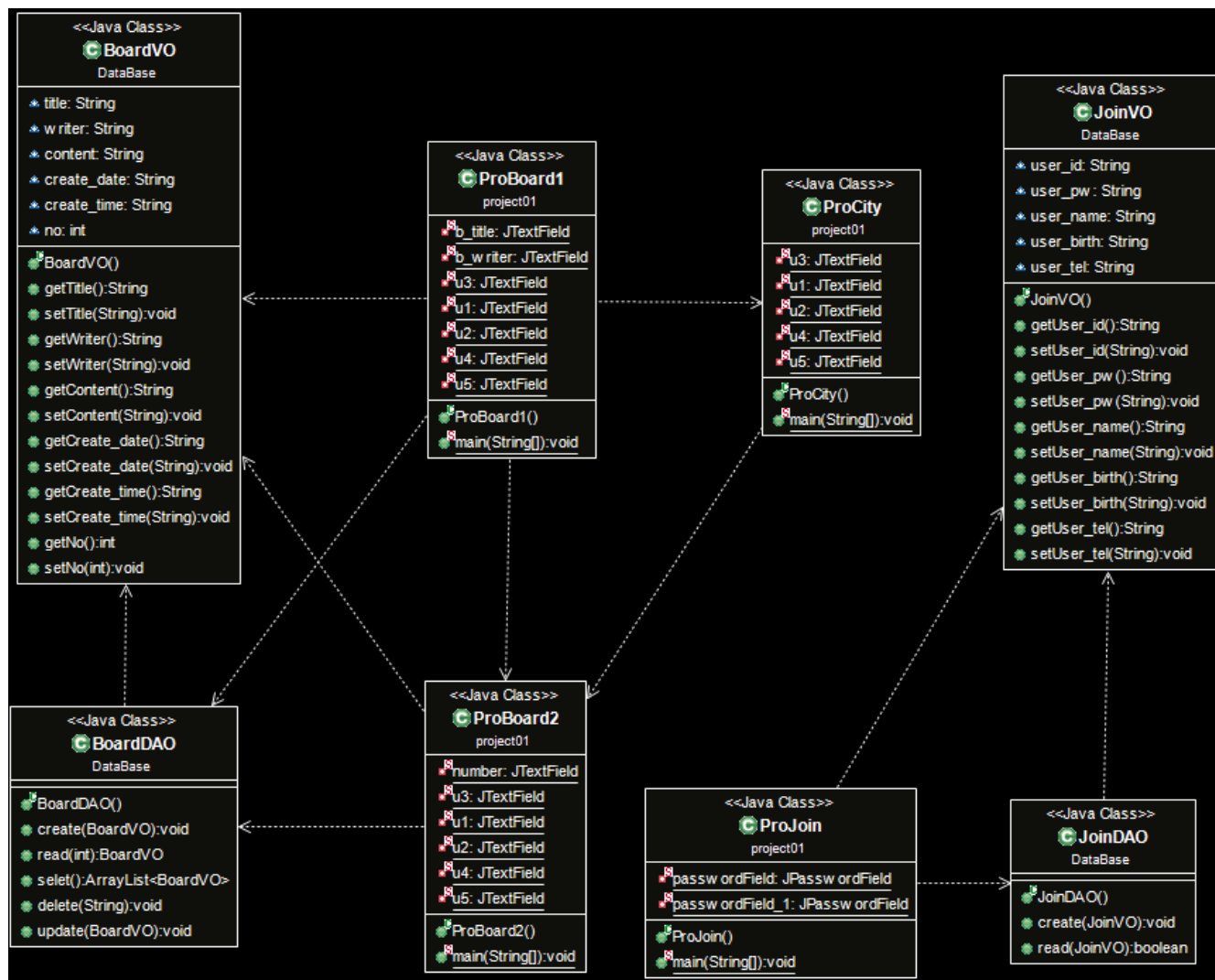
1. 개발과정

1-4) 박창호 UML



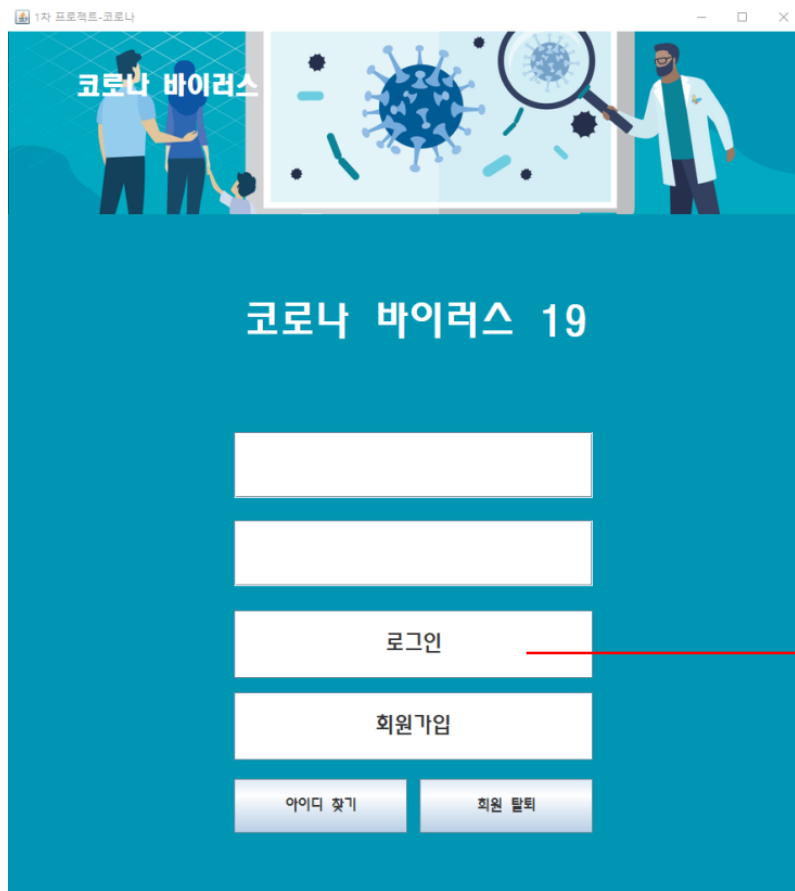
1. 개발과정

1-4)박경민 UML



2. 기술 설명

2-1) 박창호 로그인



<DAO>

```
// 3. sql문
String sql = "select * from login where user_id = ? and user_pw = ? ";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, user_id);
ps.setString(2, user_pw);
System.out.println("3. SQL생성 성공");

// 4. sql문을 전송
ResultSet rs = ps.executeQuery();
System.out.println("4. SQL문 전송 성공");

boolean result = false;
if (rs.next()) {
    System.out.println("로그인 ok.");
    result = true;
} else {
    System.out.println("로그인 not.");
}
return result;
}
```

로그인 처리는 DAO에서 기본 boolean = false로 선언해
결과값을 true 또는 false로 값을 받은 후 return

2. 기술 설명

2-1) 박창호 로그인



DB에서 id, pw를 가져옴

<UI>

```
String user_id = t1.getText();
String user_pw = t2.getText();
LoginDAO dao = new LoginDAO();

try {
    boolean res = dao.login(user_id, user_pw);
    if (res) {
        JOptionPane.showMessageDialog(f, "로그인성공");

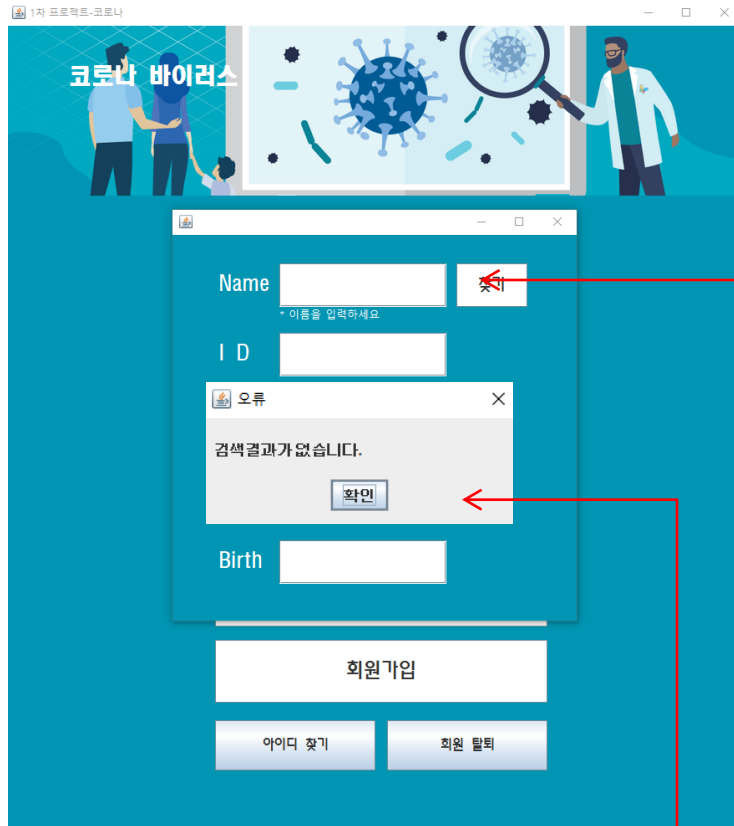
        ProHome t1 = new ProHome();
        t1.main(null);
        f.setVisible(false);
    } else {
        JOptionPane.showMessageDialog(f, "로그인실패");
    }
} catch (Exception e1) {
    e1.printStackTrace();
}
```

DAO에서 return 받은 DB가 true일 경우
로그인 성공하여 메인페이지로 이동

false이면 로그인 실패로 메인페이지 이동불가

2. 기술 설명

2-1) 박창호 아이디 찾기



```
// 4. sql문을 전송
ResultSet rs = ps.executeQuery();
System.out.println("4. SQL문 전송 성공");
ReadVO bag = new ReadVO();
if (rs.next()) { // 결과가 있는지 없는지 체크해주는 메서드
    System.out.println("검색결과가 있어요.");

    String user_id = rs.getString("user_id");
    String user_pw = rs.getString("user_pw");
    String user_tel = rs.getString("user_tel");
    String user_birth = rs.getString("user_birth");

    // 가방에 넣기
    bag.setUser_id(user_id);
    bag.setUser_pw(user_pw);
    bag.setUser_tel(user_tel);
    bag.setUser_birth(user_birth);

} else {
    JOptionPane.showMessageDialog(null, "검색결과가
}
return bag;
}
```

<DAO>

<UI>

```
//아이디찾기 in 정보찾기
JButton read = new JButton("찾기");
read.setBackground(Color.WHITE);
read.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String user_name = id1.getText();
        ReadDAO dao = new ReadDAO();

        try {
            ReadVO bag = dao.one(user_name);
            id2.setText(bag.getUser_id());
            id3.setText(bag.getUser_pw());
            id4.setText(bag.getUser_tel());
            id5.setText(bag.getUser_birth());

        } catch (Exception e1) {
            JOptionPane.showMessageDialog(f, "검색결과가 없습니다.");
        }

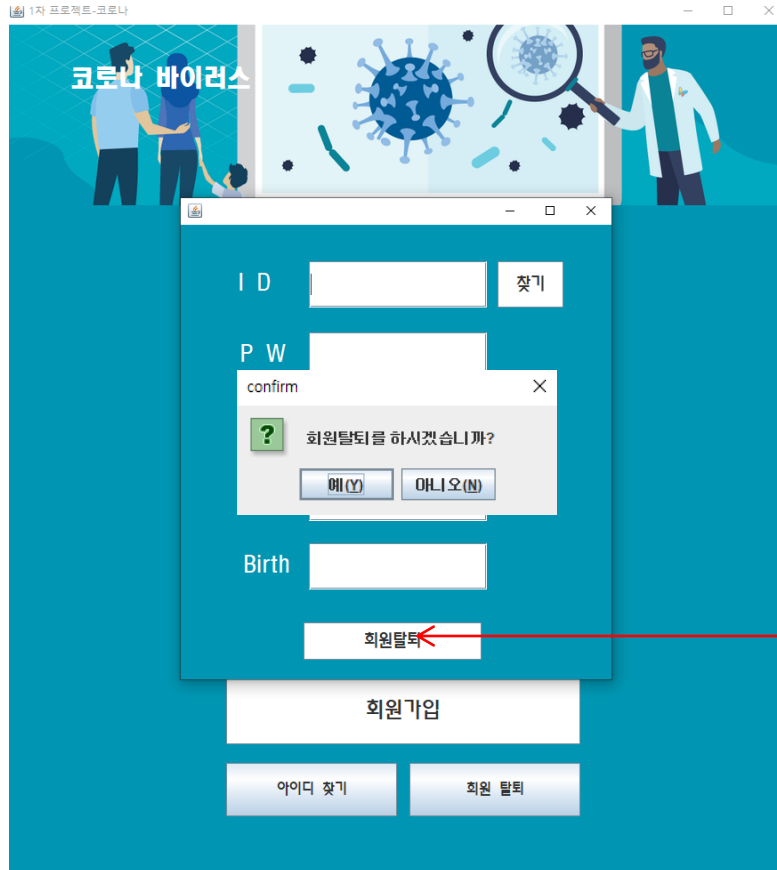
    }
});
```

DB에 저장된 Name과 입력값이 동일하면
그에 해당 하는 정보값을 불러옴.

DB에 저장된 값이 없을 시 오류창이 뜸

2. 기술 설명

2-1) 박창호 회원탈퇴



<UI>

```
// 회원 탈퇴
public void delete(LoginVO bag) throws Exception {
    // 1. connector 설정
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("1. connector 연결 성공");

    // 2. db 연결
    String url = "jdbc:mysql://localhost:3366/corona?useSSL=false";
    String user = "root";
    String password = "1234";
    Connection con = DriverManager.getConnection(url, user, password);
    System.out.println("2. db 연결 성공");

    // 3. sql문 (delete)
    String sql = "delete from login where user_id = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, bag.getUser_id());
    System.out.println("3. SQL 생성 성공");

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

JButton read_1 = new JButton("회원탈퇴");
read_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String user_id = dell.getText();
        LoginDAO dao = new LoginDAO();
        LoginVO bag = new LoginVO();
        bag.setUser_id(user_id);

        try {
            dao.delete(bag);
            int result = JOptionPane.showConfirmDialog(f, "회원탈퇴를 하시겠습니까?", "confirm",
                JOptionPane.YES_NO_OPTION);
            if (result == JOptionPane.CLOSED_OPTION) {
                return;
            } else if (result == JOptionPane.YES_NO_OPTION) {
                JOptionPane.showMessageDialog(f, "회원탈퇴가 완료되었습니다.");
                f2.setVisible(false);
            } else {
                return;
            }
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }
});
```

<DAO>

확인 메시지에서 Yes 클릭 시 이전의 검색을 통한 ID를 기준으로 탈퇴를 시도해 DB에서 해당 ID정보는 삭제가 된다.

2. 기술 설명

2-1) 박창호 홈



확진환자	완치 (격리해제)	치료중 (격리중)	사망
누적)22,504 =	19,310 +	2,827 +	367

국내 환자현황을 기존의 존재하는 사이트에서 크롤링을 통해 UI제작

```
// 하단 일일 확진자 크롤링

Connection con = Jsoup.connect("http://ncov.mohw.go.kr/");
String total = null;
String complete = null;
String ing = null;
String dead = null;

try {
    Document doc = con.get();

    Elements list = doc.select("li > span.num");
    total = list.get(0).text();
    complete = list.get(1).text();
    ing = list.get(2).text();
    dead = list.get(3).text();

} catch (Exception e) {
    e.printStackTrace();
}
```

2. 기술 설명

2-1) 박창호 국내확진자 현황

1차 프로젝트-코로나

코로나 바이러스

회원정보 수정 | 로그아웃

홈 | 국내확진자 현황 | 시도별확진자 | 게시판

◇국내 발생 현황 (9.16. 00시 기준)

□누적 확진자 현황

확진현황				국내발생		국외유입		사망	
누적	전일대비			누적	전일대비	누적	전일대비	누적	전일대비
22,504	소계	해외유입	국내발생	19,310	+ 432	2,827	- 319	367	+ 0

□누적 검사현황

검사 완료							
확진환자수				검사중		합계	
격리중	격리해제	사망	소계	격리중	소계		
2,827	19,310	367	22,504	2,130,486	2,152,990	25,842	2,178,832

```
// 크롤링
Connection con = Jsoup.connect(
    "http://ncov.mohw.go.kr/bdBoardList_Real.do?brdId=1&brdGubun=11&ncvContSeq=&contSeq=&board_id=&gubun=");
// 누적확진자 현황
String 누적 = null; // 확진환자 -누적
String 누적2 = null; // 격리해제 -누적
String 누적3 = null; // 격리중 -누적
String 누적4 = null; // 사망 -누적

String 전일 = null; // 격리해제-전일
String 전일2 = null; // 격리중-전일
String 전일3 = null; // 사망-전일

String 확진 = null; // 확진환자-소계
String 확진2 = null; // 확진환자-해외유입
String 확진3 = null; // 확진환자-국내발생

// 누적검사 현황
String m1 = null;
String m2 = null;
String m3 = null;
String m4 = null;
String m5 = null;
String m6 = null;
String m7 = null;
String m8 = null;

String z1 = null;

try {
    Document doc = con.get();
    // 누적확진자현황 - 누적
    Elements list = doc.select("dd");
    Elements list2 = list.select(".ca_value");

    누적 = list2.get(0).text(); // 확진환자 -누적
    누적2 = list2.get(2).text(); // 격리해제 -누적
    누적3 = list2.get(4).text(); // 격리중 -누적
    누적4 = list2.get(6).text(); // 사망 -누적

    전일 = list2.get(3).text(); // 격리해제 -전일대비
    전일2 = list2.get(5).text(); // 격리중 -전일대비
    전일3 = list2.get(7).text(); // 사망 -전일대비

    Elements list3 = doc.select("dd");
    Elements list4 = list3.select(".inner_value");
    확진 = list4.get(0).text(); // 확진환자 -소계
    확진2 = list4.get(1).text(); // 확진환자-해외유입
    확진3 = list4.get(2).text(); // 확진환자-국내발생

    Elements li = doc.select("tbody");
    Elements list2 = li.select("td");
    m1 = list2.get(32).text(); // 확진환자수-격리중
    m2 = list2.get(33).text(); // 확진환자수-격리해제
    m3 = list2.get(34).text(); // 확진환자수-사망
    m4 = list2.get(35).text(); // 확진환자수-소계
    m5 = list2.get(36).text(); // 검사중
    m6 = list2.get(37).text(); // 소계
    m7 = list2.get(38).text(); // 검사중
    m8 = list2.get(39).text(); // 합계
}
```

국내 누적 확진자 현황을 기존의 존재하는 사이트에서 크롤링을 통해 각 라벨에 대입했음.

2. 기술 설명

2-1) 공통 페이지 이동



```
// 홈버튼
JButton homeButton = new JButton("홈");
homeButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        ProHome t1 = new ProHome();
        t1.main(null);
        home.setVisible(false);
    }
});
homeButton.setFont(new Font("휴먼모듬T", Font.PLAIN, 20));
homeButton.setBounds(24, 128, 202, 75);
home.getContentPane().add(homeButton);

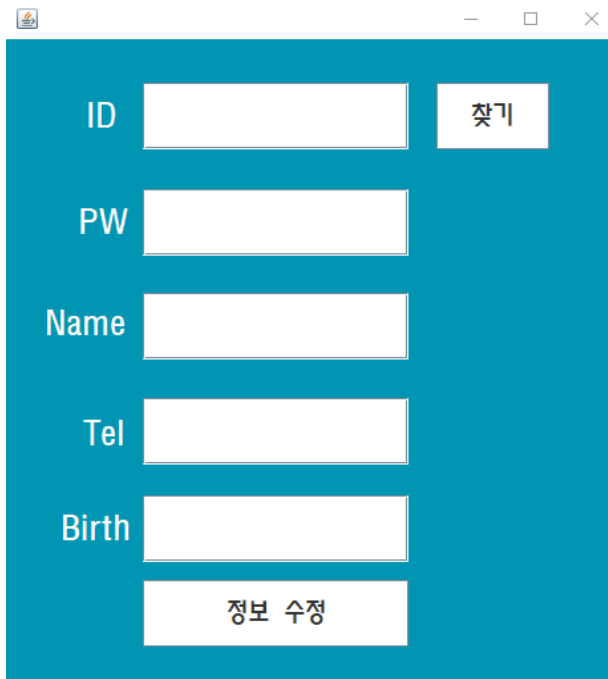
// 국내확진자 현황 버튼
JButton dayButton = new JButton("국내 확진자 현황");
dayButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Prokor t1 = new Prokor();
        t1.main(null);
        home.setVisible(false);
    }
});
```

페이지 버튼 클릭 시 기존의 화면은 false 이동 클래스는 main이 되어 true가 됨.

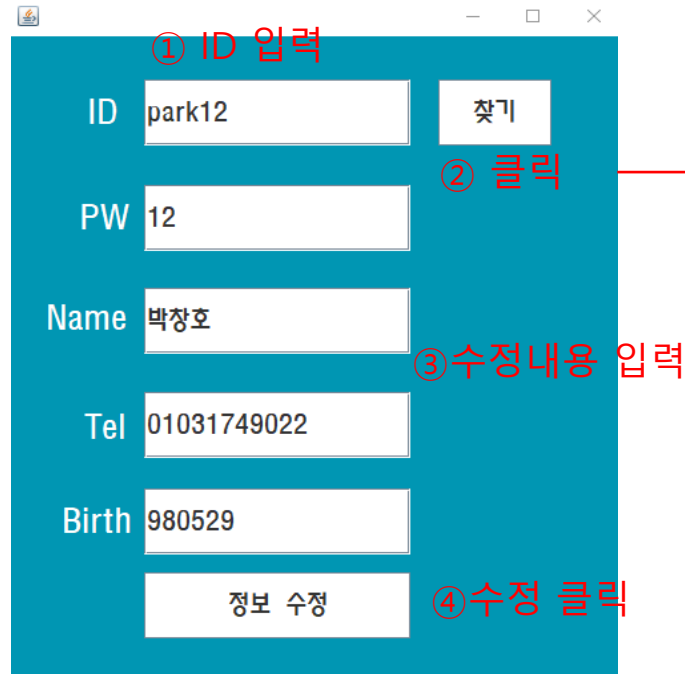
2. 기술 설명

2-1) 박창호 회원정보 수정

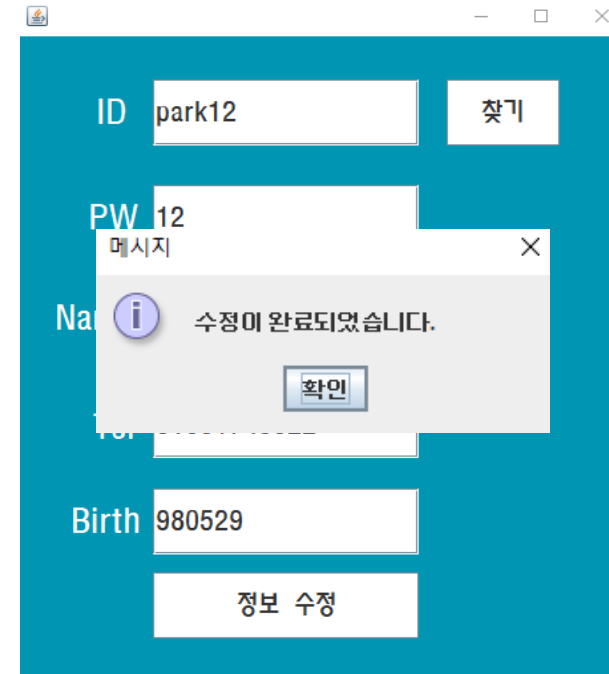
<1> 정보수정



<2> 수정 입력



<3> 수정 완료



	ABC user_id	ABC user_pw	ABC user_name	ABC user_tel	ABC user_birth
1	cagho2000	12	박창순	0101122	980524
2	park12	12	박창호	01031749022	980529

수정 전 DB

	ABC user_id	ABC user_pw	ABC user_name	ABC user_tel	ABC user_birth
1	cagho2000	12	박창순	0101122	980524
2	park12	34	홍길동	010	980529

수정 후 DB

2. 기술 설명

2-1) 박창호 로그아웃



Yes 클릭 시 초기화면 로그인 화면으로 이동

```
// 상단 로그아웃 버튼
JButton logout = new JButton("로그아웃");
logout.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int result = JOptionPane.showConfirmDialog(home, "로그아웃 하시겠습니까?", "confirm", JOptionPane.YES_NO_OPTION);
        if (result == JOptionPane.CLOSED_OPTION) {

        } else if (result == JOptionPane.YES_NO_OPTION) {
            ProLogin t1 = new ProLogin();
            t1.main(null);
            home.setVisible(false);
        } else {

        }
    }
});
```

2. 기술 설명

2-2) 회원가입(박경민)

1차 프로젝트-코로나

코로나 바이러스

ID 중복확인

비밀번호

비밀번호 확인

이름

생년월일
→ 생년월일 6자리 입력 ex) 900305

연락처
→ 010 부터 '-' 없이 입력해주세요.

메시지
중복된 ID가 있습니다.
확인

ID 중복확인 UI에 입력 후 DAO에서 결과 값을 가져와 같은 값이 있는 true일 경우 “중복” 메시지, 값이 없는 false일 경우 “사용가능” 메시지 출력

<DAO>

```
String sql = "select * from login where user_id = ?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, vo.getUser_id()); //중복확인 할 ID

ResultSet rs = ps.executeQuery();

boolean result = false;
if (rs.next()) { // 결과가 있는지 없는지 체크해주는 메서드

    result = true;
} else {

}

return result;
```

<UI>

```
String user_id = joinid.getText();

JoinDAO dao = new JoinDAO();
JoinVO vo = new JoinVO();
vo.setUser_id(user_id);

try {
    boolean result = dao.read(vo);
    if (result) { //결과값이 true 일 경우
        JOptionPane.showMessageDialog(f, "중복된 ID가 있습니다.");
    } else { // 결과값이 false 일 경우
        JOptionPane.showMessageDialog(f, "사용가능한 ID 입니다.");
    }
} catch (Exception e1) {
    e1.printStackTrace();
}
```

2. 기술 설명

2-2) 회원가입(박경민)

<UI>

코로나 바이러스

I D kyungmin 중복확인

비밀번호

비밀번호 확인

이름

생년월일

연락처

회원가입 완료

<DAO>

```
String sql = "insert into login values (?, ?, ?, ?, ?)";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, vo.getUser_id()); //회원가입 ID
ps.setString(2, vo.getUser_pw()); //회원가입 PW
ps.setString(3, vo.getUser_name()); //회원가입 이름
ps.setString(4, vo.getUser_tel()); //회원가입 전화번호
ps.setString(5, vo.getUser_birth()); //회원가입 생년월일
```

<DB>

user_id	user_pw	user_name	user_tel	user_birth
kyungmin	1234	박경민	010010	908888888
kyungmin11	123	박경민	010010010	900305

```
String user_id = joinid.getText();
String user_pw = new String(passwordField.getText());
String user_pwchk = new String(passwordField_1.getText());
String user_name = joinname.getText();
String user_birth = joinbirth.getText();
String user_tel = joinhp.getText();

//회원가입 빈칸 & 비밀번호 확인 유무
if (user_id.isEmpty()) { //isEmpty, 빈칸인 경우 확인
    JOptionPane.showMessageDialog(f, "아이디를 입력해주세요.");
} else if (user_pw.isEmpty()) {
    JOptionPane.showMessageDialog(f, "비밀번호를 입력해주세요.");
} else if (user_name.isEmpty()) {
    JOptionPane.showMessageDialog(f, "이름을 입력해주세요.");
} else if (user_birth.isEmpty()) {
    JOptionPane.showMessageDialog(f, "생일을 입력해주세요.");
} else if (user_tel.isEmpty()) {
    JOptionPane.showMessageDialog(f, "전화번호를 입력해주세요.");
} else if (!user_pw.equals(user_pwchk)) { // String, equals로 비교 / 입력pw와 pwchk가 다를 경우
    JOptionPane.showMessageDialog(f, "입력하신 비밀번호가 다릅니다. 비밀번호를 확인해주세요.");
}

else {
    JOptionPane.showMessageDialog(f, "회원가입 완료");

    JoinDAO dao = new JoinDAO();
    JoinVO vo = new JoinVO();
    vo.setUser_id(user_id);
    vo.setUser_pw(user_pw);
    vo.setUser_name(user_name);
    vo.setUser_birth(user_birth);
    vo.setUser_tel(user_tel);
}
```

isEmpty로 입력칸의 빈칸인 경우 확인
비밀번호와 비밀번호 확인 !equals로 비교하여 일치 여부 확인
모든 조건이 맞을 경우 회원 가입 완료
모든 입력 정보 DB로 전송

2. 기술 설명

2-2) 시도별확진자(박경민)

1차 프로젝트-코로나

코로나 바이러스

회원정보수정

홈 국내별확진자 현황 시도별확진자 게시판

시도별 확진자
(9.16. 00시 기준, 1.3 이후 누계)

지역	누적 확진환자	전일 대비	지역	누적 확진환자	전일 대비
서울	4,794	(+51)	경기	3,998	(+26)
인천	852	(+9)	부산	365	(+4)
대구	7,116	(+4)	광주	484	(+2)
대전	346	(+1)	울산	141	(0)
세종	70	(0)	강원	217	(0)
충북	151	(+2)	충남	447	(+2)
전북	106	(+5)	전남	166	(0)
경북	1,491	(+2)	경남	274	(+3)
제주	56	(0)	검역	1,430	(+2)

```
Connection con = Jsoup.connect("http://ncov.mohw.go.kr/"); // 코로나 19 사이트
Document doc = con.get();
Elements nujuck_list = doc.select("div.rpsa_map span.num"); // 지역별 누적 확진자 크롤링
Elements yseter_list = doc.select("div.rpsa_map span.before"); // 지역별 전일대비 확진자 크롤링
Elements date = doc.select("a span.livedate"); // 2020.x.x x시 기준 크롤링
String date1 = date.get(0).text(); //date에서 크롤링 하여 가져온 텍스트

String[] city = new String[nujuck_list.size()]; //크롤링으로 추출한 누적확진자 텍스트 배열
String[] city_day = new String[yseter_list.size()]; //크롤링으로 추출한 전일대비 확진자 텍스트 배열

for (int i = 0; i < nujuck_list.size(); i++) { //for문으로 반복하여 지역별 누적확진자 텍스트를 배열에 넣는다
    String nujuck = nujuck_list.get(i).text();
    city[i] = nujuck;
}
for (int i = 0; i < yseter_list.size(); i++) { //for문으로 반복하여 지역별 전일대비 확진자 텍스트를 배열에 넣는다
    String yester = yseter_list.get(i).text();
    city_day[i] = yester;
}
```

크롤링한 값을 UI 각 라벨에 넣어줌

2. 기술 설명

2-2) 게시판 글쓰기(박경민)

<DAO>

```
Date today = new Date();
SimpleDateFormat date = new SimpleDateFormat("yyyy/MM/dd");
SimpleDateFormat time = new SimpleDateFormat("HH:mm");

String creat_day = date.format(today);
String creat_time = time.format(today);

String sql = "insert into board values (null, ?, ?, ?, ?, ?)";
PreparedStatement ps = con.prepareStatement(sql);

ps.setString(1, vo.getTitle());
ps.setString(2, vo.getWriter());
ps.setString(3, vo.getContent());
ps.setNString(4, creat_day);
ps.setNString(5, creat_time);
```

<UI>

```
String title = b_title.getText();
String writer = b_writer.getText();
String content = b_content.getText();

BoardDAO dao = new BoardDAO();
BoardVO vo = new BoardVO();
vo.setTitle(title);
vo.setWriter(writer);
vo.setContent(content);
try {
    dao.create(vo);
} catch (Exception e1) {
    e1.printStackTrace();
}
```

<DB>

no	title	writer	content	create_day	create_time
1	게시판 테스트 1	박경민	게시판 테스트 1	2020/09/14	10:14
2	게시판 테스트2	박경민	게시판 테스트2	2020/09/14	14:03
3	게시판 테스트3	박경민	게시판 테스트입니다.	2020/09/14	14:15
4	게시판 테스트4	박경민	게시판 테스트4	2020/09/14	16:04
5	게시판 테스트5	박경민	테스트다	2020/09/15	09:40
6	안녕하세요	가나다	가나다라마가나다라마	2020/09/16	12:37
7	ㅎㅇ	하위	하위하위하위	2020/09/16	12:37
8	코로나 바이러스	ㅎㅎ	ㅎㅎㅎㅎㅎㅎ	2020/09/16	12:37
9	마스크 착용	마스크	제발좀써라	2020/09/16	12:38
10	2.5단계...	어디가냐	갈대가없음...어디가냐	2020/09/16	12:38
11	후후후후	하하하	호호호호후후후후	2020/09/16	12:39
12	시도별확진자	시도	시도별 확진자는??	2020/09/16	12:39
13	배고프다	우우	배고픈거같음	2020/09/16	12:40
14	첫번째 프로젝트	박경민	머리가 아프다!힘머리	2020/09/16	12:40
15	게시판 활성화	박창호	게시판 좀 많이쓰자	2020/09/16	12:40
16	코딩코딩코딩	창호	코딩코딩코딩	2020/09/16	12:41
17	COVID - 19	corona	virus	2020/09/16	12:41
18	Hi!!!	hello	hello it's me	2020/09/16	12:41

제목, 작성자, 내용 입력 후 DB 전송
게시판 번호 자동 입력, 증가
작성날짜, 시간을 String으로 DB에 전송

2. 기술 설명

2-2) 게시판(박경민)



<DAO>

```
while (rs.next()) {
    BoardVO vo2 = new BoardVO();
    int no1 = rs.getInt(1);
    String title = rs.getString(2);
    String writer = rs.getString(3);
    String date = rs.getString(4);

    vo2.setNo(no1);
    vo2.setTitle(title);
    vo2.setWriter(writer);
    vo2.setCreate_date(date);

    list.add(vo2);
}
return list;
```

<UI>

```
// 게시판 테이블
BoardDAO dao = new BoardDAO();
ArrayList<BoardVO> list = dao.selet();

String titles[] = { "글번호", "제목", "작성자", "게시일" };
Object contents[][] = new Object[list.size()][4];

for (int i = 0; i < list.size(); i++) {
    BoardVO vo = list.get(i);

    Object row[] = new Object[4];
    row[0] = vo.getNo();
    row[1] = vo.getTitle();
    row[2] = vo.getWriter();
    row[3] = vo.getCreate_date();
    contents[i] = row;
}
```

<DB>

no	title	writer	content	create_day	create_time
1	게시판 테스트 1	박경민	게시판 테스트 1	2020/09/14	10:14
2	게시판 테스트2	박경민	게시판 테스트2	2020/09/14	14:03
3	게시판 테스트3	박경민	게시판 테스트 입니다.	2020/09/14	14:15

for문으로 반복하며 배열에 값을 게시판 내용에 넣고,
Jtable을 사용하여 띄워줌
글번호(no), 제목(title), 작성자(writer), 게시일(create_day)

2. 기술 설명

2-2) 게시판 내용보기(박경민)

<DAO>

```
BoardVO bag = new BoardVO();

if (rs.next()) { // 결과가 있는지 없는지 체크
    int no1 = rs.getInt("no");
    String title = rs.getString("title");
    String content = rs.getString("content");
    String writer = rs.getString("writer");
    String date = rs.getString("create_day");
    String time = rs.getString("create_time");

    bag.setNo(no1);
    bag.setTitle(title);
    bag.setContent(content);
    bag.setWriter(writer);
    bag.setCreate_date(date);
    bag.setCreate_time(time);
} else {
    System.out.println("검색결과가 없어요.");
}
return bag;
```

<UI>

```
String no2 = number.getText(); //게시판 번호 입력
BoardDAO dao = new BoardDAO();
try {
    BoardVO vo = dao.read(Integer.parseInt(no2)); //입력된 번호 String->int로 변환 후
                                                    //입력된 번호 DB 출력

    no.setText(String.valueOf(vo.getNo())); //글번호
    writer.setText(vo.getWriter());         //작성자
    title.setText(vo.getTitle());           //제목
    content.setText(vo.getContent());       //내용
    cr_date.setText(vo.getCreate_date());  //작성일
    cr_time.setText(vo.getCreate_time());  //작성 시간
```

123 no	ABC title	ABC writer	ABC content	ABC create_day	ABC create_time
1	게시판 테스트 1	박경민	게시판 테스트 1	2020/09/14	10:14

게시판

글번호 1 작성자 박경민 게시일 2020/09/14 10:14

제 목 게시판 테스트 1

게시판 테스트 1

게시글 입력 번호에 입력한 번호
String->int로 변환 후 입력된 번호의 값이 true이면 DB출력
(글번호, 작성자, 제목, 내용,작성일, 작성시간)

2. 기술 설명

게시판 수정

글번호 1 작성자 박경민 게시일 2020/09/14 10:14

제목 게시판 테스트 1

내용 게시판 테스트 1

확인

글쓰기

2-2) 게시판 내용 수정 전(박경민)

<UI>

```
vo.setNo(Integer.parseInt(no2)); //처음 입력된 게시판 번호를 통해 수정  
vo.setTitle(TitleText.getText()); //제목  
vo.setWriter(WriterText.getText()); //작성자  
vo.setContent(content.getText()); //내용
```

```
BoardDAO dao = new BoardDAO();  
try {  
    dao.update(vo);  
} catch (Exception e1) {  
    e1.printStackTrace();  
}
```

<DAO>

```
String sql = "update board set writer=?, title=?, content=? where no =?";  
PreparedStatement ps = con.prepareStatement(sql);  
ps.setString(1, vo.getWriter());  
ps.setString(2, vo.getTitle());  
ps.setString(3, vo.getContent());  
ps.setString(4, String.valueOf(vo.getNo())); //입력번호  
ps.executeUpdate();
```

게시글 보기에 입력했던 게시글 번호를 통해
작성자, 제목, 내용만 수정 가능

2. 기술 설명

2-2) 게시판 내용 수정 후(박경민)

<게시판 수정 후> UI, DB 변화

The first screenshot shows the '게시판 수정' (Board Modification) form. It includes fields for '글번호' (1), '작성자' (경민박), '게시일' (2020/09/14 10:14), and '제 목' (게시판 테스트 123456789). A message box indicates '수정 완료' (Modification Complete). The second screenshot shows the '게시판' (Board) list table with a red box around the first row. The third screenshot shows the '게시판' (Board) form with a red box around the content area.

글번호	제목	작성자	게시일
1	게시판 테스트 123456789	경민박	2020/09/14
2	게시판 테스트2	박경민	2020/09/14
3	게시판 테스트3	박경민	2020/09/14
4	게시판 테스트4	박경민	2020/09/14
5	게시판 테스트5	박경민	2020/09/15
6	안녕하세요	카나다	2020/09/16
7	ㅇㅇ	허위	2020/09/16
8	코로나 바이러스	ㅇㅇ	2020/09/16
9	마스크 착용	마스크	2020/09/16
10	2.5단계...	어디가나	2020/09/16
11	부부부부	허위	2020/09/16
12	시도별 확진자	시도	2020/09/16
13	해고피다	우우	2020/09/16
14	첫번째 프로젝트	박경민	2020/09/16
15	게시판 활성화	박장호	2020/09/16
16	코로나당국공방	장호	2020/09/16
17	COVID - 19	corona	2020/09/16
18	Hi!!!	hello	2020/09/16

<DB>

	123 no	ABC title	ABC wr	ABC content	ABC create_day	ABC create_time
1	1	게시판 테스트 123456789	경민박	게시판 테스트 1123456789	2020/09/14	10:14

2. 기술 설명

2-2) 게시판 삭제(박경민)

<DAO>

```
String sql = "delete from board where no = ?";  
PreparedStatement ps = con.prepareStatement(sql);  
ps.setString(1, no);
```

<UI>

```
public void actionPerformed(ActionEvent e) {  
    String no2 = number.getText(); //게시판 보기 시 입력 번호  
    BoardDAO dao = new BoardDAO();  
    try {  
        dao.delete(no2);  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
}
```

게시물보기 입력 번호를 통해
해당 번호 데이터를 찾아 DB에서 삭제

글번호	제목	작성자	게시일
1	게시판 테스트 123456789	박경민	2020/09/14
2	게시판 테스트2	박경민	2020/09/14
3	게시판 테스트3	박경민	2020/09/14
4	게시판 테스트4	박경민	2020/09/14
5	게시판 테스트5	박경민	2020/09/15
6	안녕하세요	박경민	2020/09/16
7	ㅎㅇ	박경민	2020/09/16
8	코로나 바이러스	후후	2020/09/16
9	마스크 착용	마스크	2020/09/16
10	2.5단 계...	여디가나	2020/09/16
11	후후후후	하하하	2020/09/16
12	시도별확진자	시도	2020/09/16
13	메고프다	후후	2020/09/16
14	첫번째 프로젝트	박경민	2020/09/16
15	게시판 활성화	박정호	2020/09/16
16	코딩코딩코딩	장호	2020/09/16

3. 소감문

3-1) 소감문(박창호)

박창호

비전공인 상태로 자바에 입문하여 처음으로 프로젝트를 진행하는 과정에서 첫 단계인 계획, 설계, 팀원과 파트분배 등 어려움과 문제점이 많았습니다.

팀원과 프로젝트를 진행해나가면서도 필요한 DAO, DTO(VO), 크롤링 등을 이해하지 못하여 진행이 힘들었으나 차츰 직접 검색해보고 팀원한테 도움을 받으며 직접 부딪히다 보니 어느새 자연스럽게 사용하고 있어 원만하게 프로젝트를 잘 마무리 할 수 있었습니다.

결과적으로, 이번 프로젝트로 인해 수업 중에 듣는 이론과 코딩을 귀로만 듣고 이해하는 것보다 많은 코드와 DB, Spring 등 을 직접 사용할 수 있어서 이해하기 좋았고 팀원과 많은 의견을 조율하고 소통하며 서로 완성된 프로젝트를 합치는 과정에 일어나는 오류를 직접 찾아 해결할 수 있어서 재밌었습니다.