

택배 예약 프로그램

휴먼교육센터

개발자: 전민정

개발 기간:

2024-04-11~2024-04-19

목차

- 1 주제 소개
- 2 요구사항 정의서
- 3 Use Case
- 4 ERD
- 5 클래스 다이어그램
- 6 기능 관련 코드 증빙 및 해설
- 7 시연 영상

1. 주제 소개

주제: 택배 예약 프로그램



1. 개발 동기

택배시스템을 자주 이용한 경험이 있으며 택배 예약을 할 때 본인의 정보와 타인의 정보 등을 입력 받아 받은 정보를 관리하는 프로그램을 만들고자 하였다.

2. 프로젝트 목표

- 회원 가입, 로그인 시 기능을 세분화하여 구현
- 회원의 택배 예약 정보 관리
- 데이터베이스와 연동하여 정보를 CRUD 하고 영구적으로 관리하도록 구현

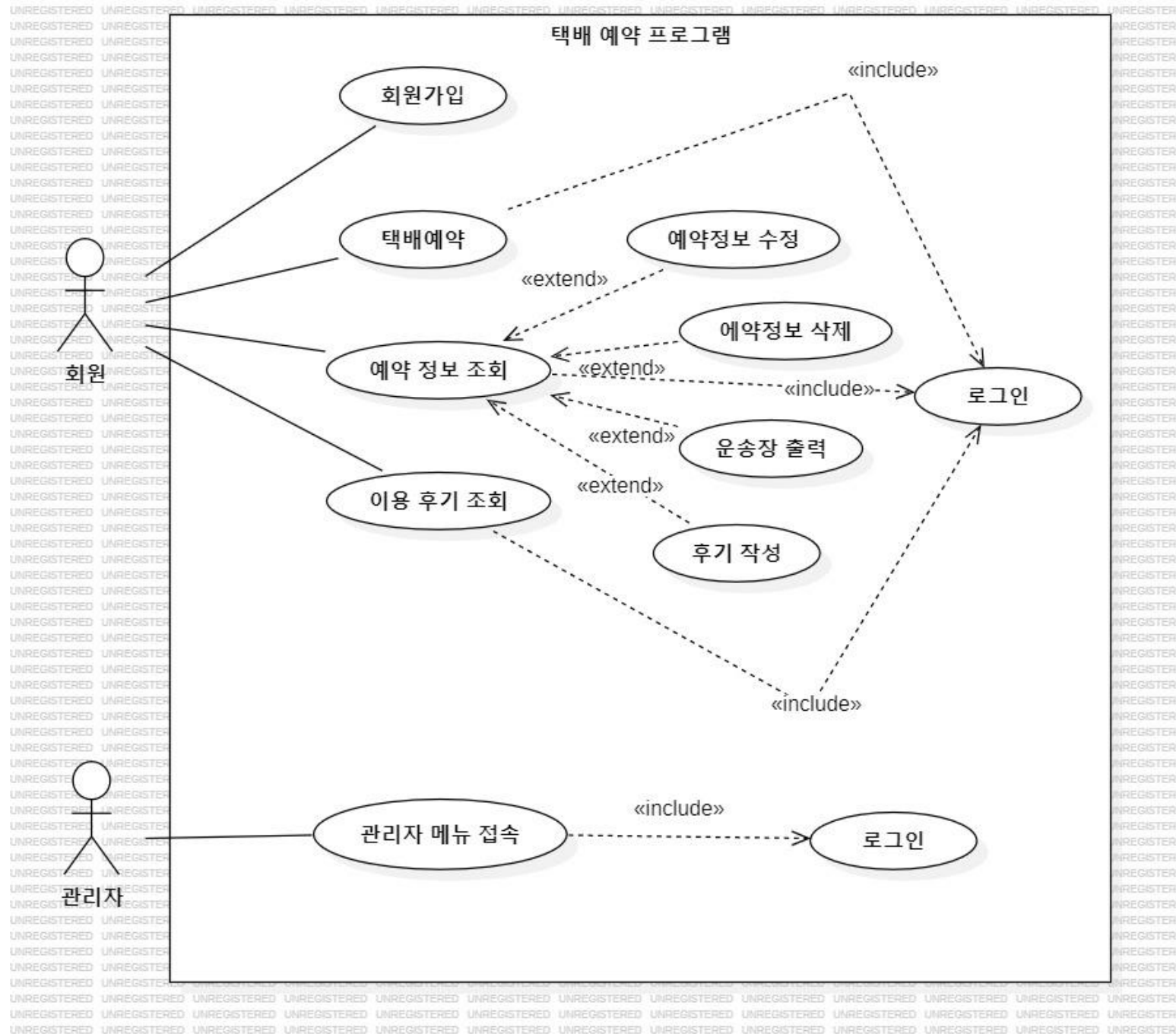
3. 개발 환경

- JDK: JDK 8
- IDE: Eclipse


2. 요구사항 정의서

요구사항 정의서					
프로젝트명	택배 예약 프로그램		작성자	전민정	작성일 2024-04-18
번호	요구사항명	요구사항내용			
1-1	회원가입	1.아이디,비밀번호,비밀번호힌트,이름,전화번호,주소를 입력받는다.			
1-2	유효성	1.중복된 아이디는 허용되지 않는다.			
		2.전화번호 형식은 010-xxxx-xxxx으로 지정하고 자릿수는 13자리이다.			
		3.전화번호는 고유하며 같은 전화번호로 두 번 가입할 수 없다.			
		4.모든 정보를 입력할 시 가입된다.			
2-1	로그인	1.아이디와 비밀번호를 입력하여 로그인한다.			
2-2	아이디 찾기	1.사용자의 이름과 전화번호를 입력받아 찾기 버튼을 누르면 아이디 정보를 표시한다.			
2-3	비밀번호 찾기	1.사용자의 아이디와 비밀번호힌트를 입력 받아 찾기 버튼을 누르면 비밀번호 정보를 표시한다.			
3-1	택배 예약	1. 품목,가격,등록명칭,보내는 사람의 이름,전화번호,주소 받는 사람의 이름,전화번호,주소,배송요청사항을 입력받는다.			
3-2	유효성	1. 모든 정보를 입력할시 접수되며, 배송요청사항은 선택이다.			
		2.가격에는 숫자만 입력할 수 있다.			
		3.전화번호 자릿수는 13자리로 지정한다.			
4-1	프로그램 접속 화면	1.사용자의 아이디,포인트 정보가 표시되고 로그아웃,택배예약,마이페이지,이용후기 버튼이 있다.			
5-1	예약 정보 조회	1.사용자는 메뉴의 '마이페이지'를 눌러 자신의 예약 정보를 조회할 수 있다.			
5-2	예약 정보 수정	1.사용자가 테이블에서 예약정보가 담긴 각각의 셀을 선택하고 수정 버튼을 누른다.			
		2.수정창에는 예약한 정보가 입력창에 미리 입력되어 있으며 그 위에서 수정할 수 있다.			
5-3	예약 정보 삭제	3.운송장 출력되면 예약 정보를 수정할 수 없다.			
5-3	예약 정보 삭제	1.사용자가 테이블에서 예약정보가 담긴 각각의 셀을 선택하고 삭제버튼을 누르면 해당 항목이 삭제된다.			
		1.사용자가 테이블에서 예약정보가 담긴 각각의 셀을 선택하고 운송장 출력버튼을 누르면 운송장 출력여부가 '출력완료'로 변경된다.			
6-1	운송장 출력				
7-1	후기 작성	1.사용자가 테이블에서 예약정보가 담긴 각각의 셀을 선택하고 후기작성 버튼을 누르면 후기를 작성한다.			
		2.후기는 예약한 횟수만큼 작성할 수 있고 최초 작성창과 수정창은 동일하다.			
7-2	후기 조회	1.메뉴에서 '이용후기'버튼을 누르면 모든 사용자의 후기를 조회할 수 있다.			
8-1	관리자 메뉴	1.관리자는 관리자 아이디와 비밀번호를 입력하여 관리자 메뉴에 접속할 수 있다.			


3.Use Case



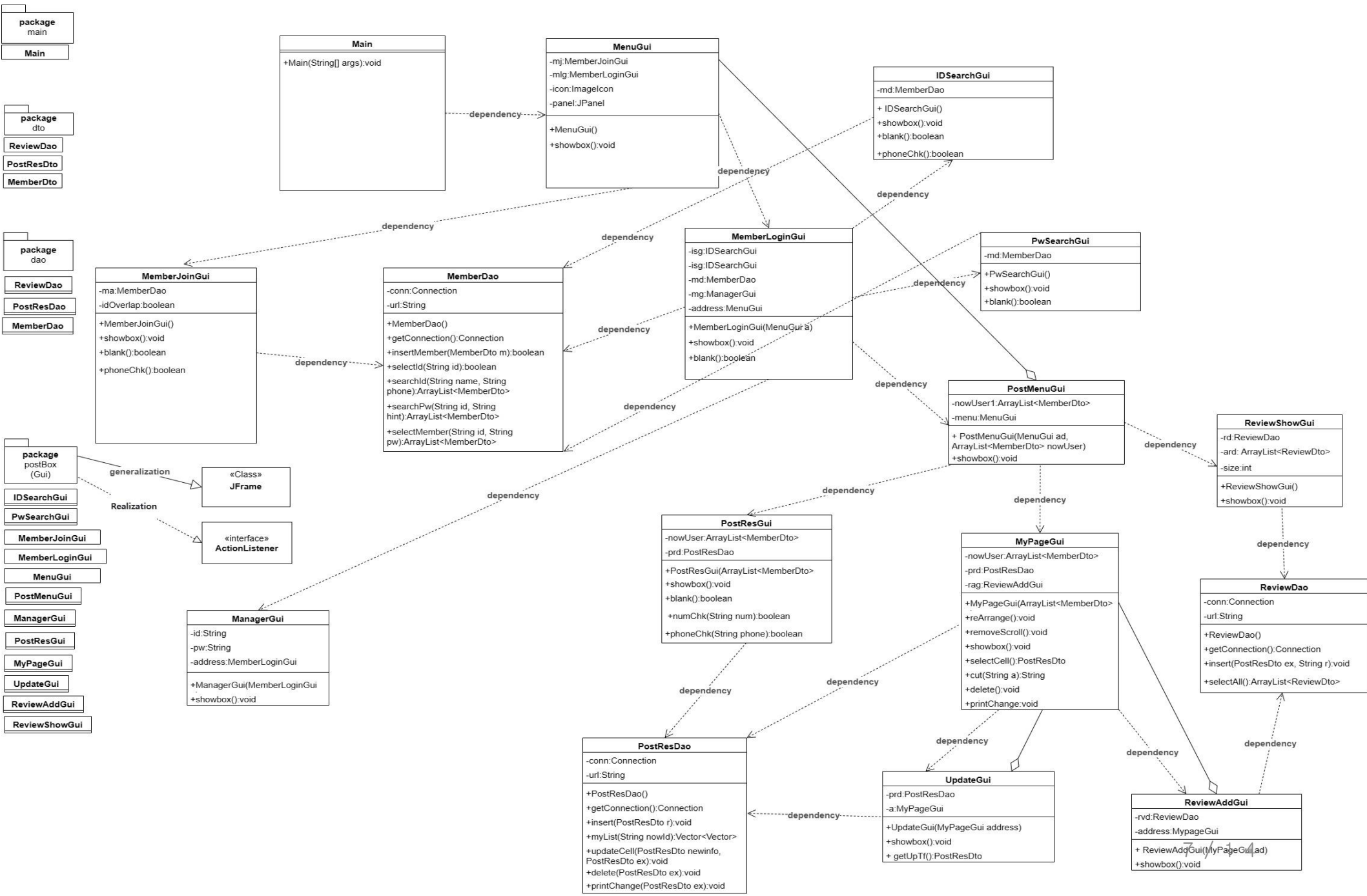
4.ERD

회원			member			
	아이디	id	PK	varchar2(10)	NOT NULL	Default value
	비밀번호	pw	Domain	varchar2(10)	NULL	Default value
	비밀번호 힌트	pw_hint	Domain	varchar2(10)	NULL	Default value
	이름	name	Domain	varchar2(8)	NULL	Default value
	전화번호	phone	UK,CK	varchar2(13)	NULL	Default value
	주소	address	Domain	varchar2(100)	NULL	Default value
	포인트	point	Domain	number	NULL	100



예약정보			post_res			
	예약날짜	res_date	Domain	varchar2(10)	NULL	Default value
	품목	p_category	Domain	varchar2(10)	NULL	Default value
	물품가격	p_price	Domain	varchar2(7)	NULL	Default value
	등록명칭	p_name	Domain	varchar2(14)	NULL	Default value
	보내는 사람	s_name	Domain	varchar2(8)	NULL	Default value
	보내는 사람의 전화번호	s_phone	Domain	varchar2(13)	NULL	Default value
	보내는 사람의 주소	s_address	Domain	varchar2(100)	NULL	Default value
	받는 사람	g_name	Domain	varchar2(8)	NULL	Default value
	받는 사람의 전화번호	g_phone	Domain	varchar2(13)	NULL	Default value
	받는 사람의 주소	g_address	Domain	varchar2(100)	NULL	Default value
	배송 요청사항	g_request	Domain	varchar2(50)	NULL	Default value
	운송장 출력여부	track_print	CK	varchar2(1)	NULL	Default value
	이용 후기	review	Domain	varchar2(100)	NULL	Default value
	아이디	n_id	Domain	varchar2(10)	NOT NULL	Default value

5. 클래스 다이어그램



6. 기능 관련 코드 증빙 및 해설-

1) 회원가입 시 아이디 중복 확인 코드 증빙

The screenshot shows a web form titled '회원가입' (Sign Up) with the following fields and labels:

- 아이디 (ID): d
- 비밀번호 (Password): [empty]
- 비밀번호 힌트 (Password hint): [empty]
- 기억에 남는 추억의 장소는? (Where is the memorable memory?): [empty]
- 이름 (Name): [empty]
- 전화번호 (Phone number): [empty] ex) 010-xxxx-xxxx
- 주소 (Address): [empty]

Buttons: '사용가능' (Available), '가입' (Join).

Messages: '중복된 아이디입니다' (ID already exists), '사용가능한 아이디입니다' (Available ID).

MemberDao 클래스

```
public boolean selectId(String id) {
    PreparedStatement p = null;
    String sql = "select id from member where id=?";
    int a = 0;
    if (getConnection() != null) {
        try {
            ResultSet r = null;
            p = conn.prepareStatement(sql);
            p.setString(1, id);
            r = p.executeQuery();
            a = p.executeUpdate();
            System.out.println(a + "개 결과값 찾기 성공");
        } catch (Exception e) {
            System.out.println("결과값 찾기 실패");
        }
    }
    if (a == 0) {
        return false;
    } else {
        return true;
    }
}
```

DB의 쿼리문 실행 결과에 따라
결과값 리턴

MemberJoinGui 클래스

```
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource().equals(yes)) {
        if (ma.selectId(idtf.getText())) {
            System.out.println(ma.selectId(idtf.getText()));
            yestext.setText("중복된 아이디입니다");
            idOverlap = true;
        } else {
            System.out.println(ma.selectId(idtf.getText()));
            yestext.setText("사용가능한 아이디입니다");
            idOverlap = false;
        }
    }
}
```

사용자가 '사용가능'버튼을 누르고
유효성확인을 해야 회원가입 성공

6. 기능 관련 코드 증빙 및 해설-

2) 로그인 시 아이디&비밀번호 찾기 코드 증빙

JsPostBox

회원 로그인

id

pw

아이디 찾기비밀번호 찾기로그인

JsPostBox

아이디 찾기

이름

전화번호

찾기

JsPostBox

비밀번호 찾기

아이디:

비밀번호 힌트: 기억에 남는 추억의 장소는?

찾기

```
public ArrayList<MemberDto> searchId(String name, String phone) {
    ArrayList<MemberDto> seari = new ArrayList<>();
    PreparedStatement p = null;
    String sql = "select id from member where name=? and phone=? ";
    if (getConnection() != null) {
        try {
            p = conn.prepareStatement(sql);
            p.setString(1, name);
            p.setString(2, phone);
            ResultSet r = p.executeQuery();
            while (r.next()) {
                MemberDto newd = new MemberDto();
                newd.setId(r.getString("id"));
                seari.add(newd);
                return seari;
            }
        } catch (Exception e) {
            System.out.println("아이디 찾기 오류 발생");
        }
    }
    return null;
}
```

아이디 찾기

```
public ArrayList<MemberDto> searchPw(String id, String hint) {
    ArrayList<MemberDto> searp = new ArrayList<>();
    PreparedStatement p = null;
    String sql = "select pw from member where id=? and pw_hint=? ";
    if (getConnection() != null) {
        try {
            p = conn.prepareStatement(sql);
            p.setString(1, id);
            p.setString(2, hint);
            ResultSet r = p.executeQuery();
            while (r.next()) {
                MemberDto newd = new MemberDto();
                newd.setPw(r.getString("pw"));
                searp.add(newd);
                return searp;
            }
        } catch (Exception e) {
            System.out.println("비밀번호 찾기 오류 발생");
        }
    }
    return null;
}
```

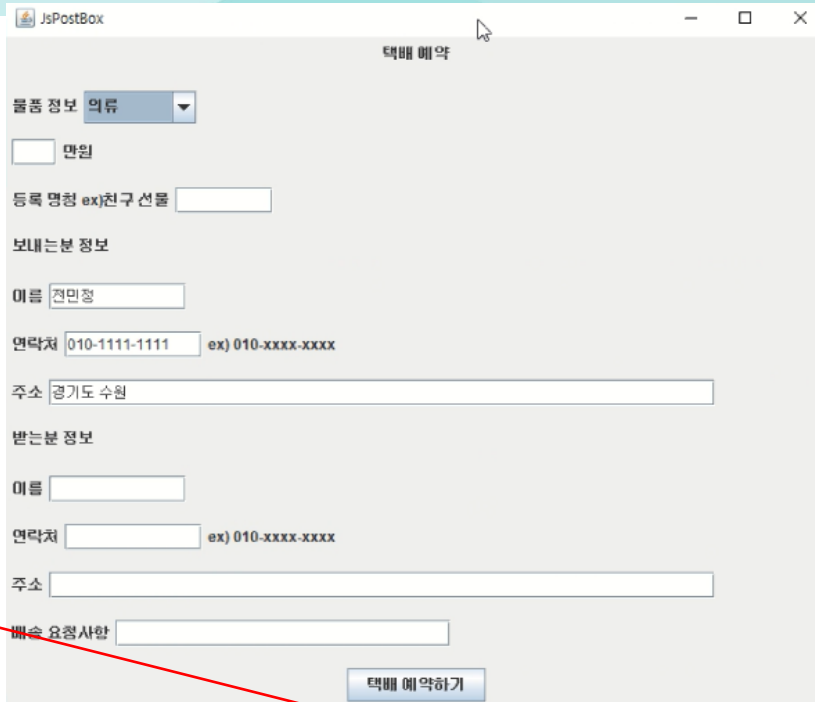
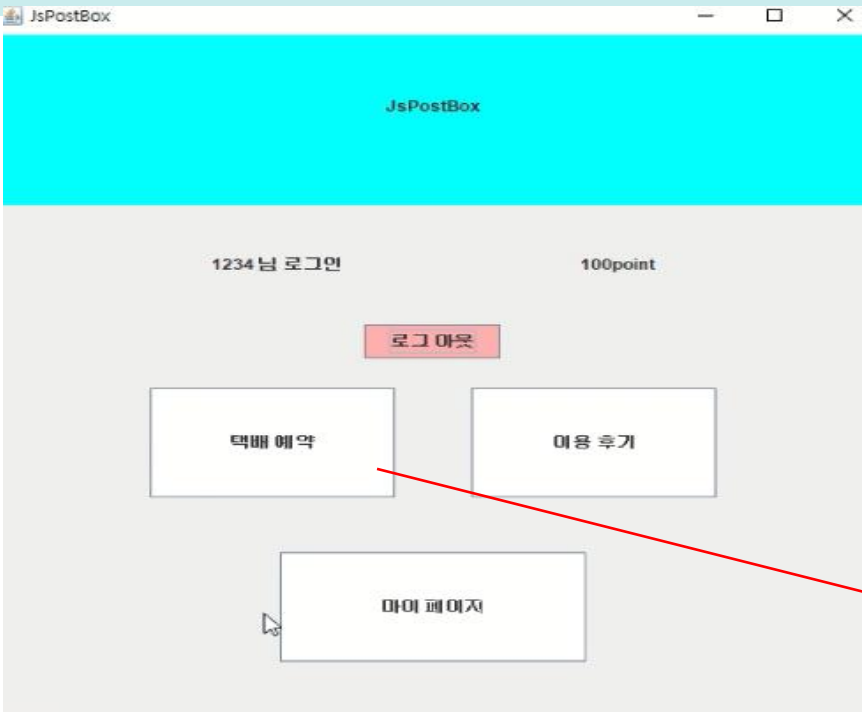
결과값 존재할 시 주소값 리턴

MemberDao 클래스

비밀번호 찾기

6. 기능 관련 코드 증빙 및 해설-

3) 택배 예약에서 사용자 정보를 알 수 있도록 하는 코드 증빙



```
public void actionPerformed(ActionEvent e) {
    ...
} else {
    ArrayList<MemberDto> nowUser = md.selectMember(idtf.getText(), pwtf.getText());
    if (nowUser != null) {
        menuGuiAddress.setVisible(false);
        this.setVisible(false);
        idtf.setText("");
        pwtf.setText("");
        PostMenuGui prg = new PostMenuGui(menuGuiAddress, nowUser);
        prg.showbox();
    } else {
        JOptionPane.showMessageDialog(null, "로그인 실패");
    }
}
```

MemberLoginGui 클래스

```
public void actionPerformed(ActionEvent e) {
    ...
} else if (e.getSource().equals(res)) {
    PostResGui prg=new PostResGui(nowUser1);
    prg.showbox();
}

private ArrayList<MemberDto> nowUser;

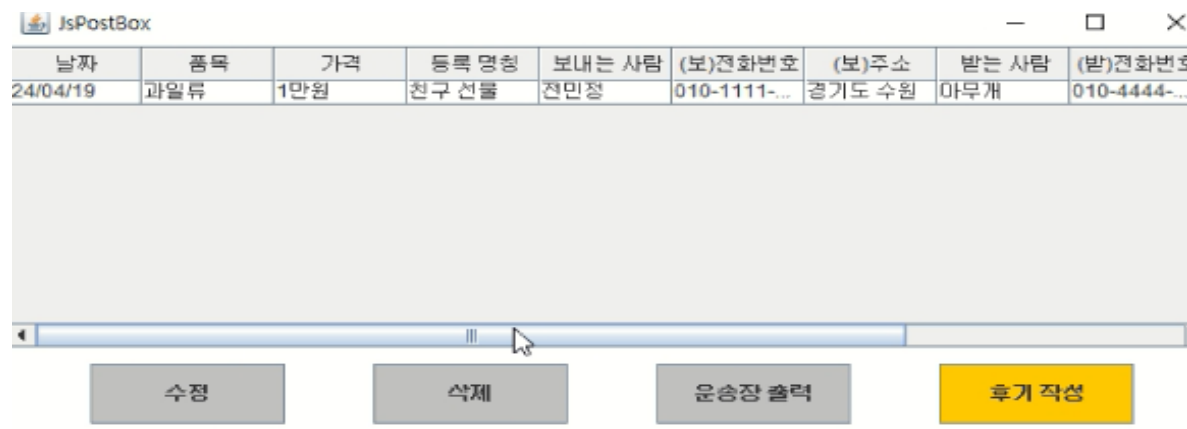
public PostResGui(ArrayList<MemberDto> ad) {
    nowUser = ad;
}
```

PostMenuGui 클래스

PostResGui 클래스

6. 기능 관련 코드 증빙 및 해설-

4) 마이페이지에서 택배 예약한 목록 조회 코드 증빙



PostResDao 클래스

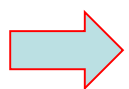
```
public Vector<Vector> myList(String nowId) {
    Vector<Vector> setRow = new Vector<>();
    String sql = "select res_date, p_category, p_price, p_name, s_name, s_phone,"
        + " s_address, g_name, g_phone, g_address, g_request, track_print" + " from post_res where n_id=?";
    PreparedStatement p = null;
    if (getConnection() != null) {
        try {
            p = conn.prepareStatement(sql);
            p.setString(1, nowId);
            ResultSet r = p.executeQuery();
            while (r.next()) {
                Vector<String> row = new Vector<String>();
                row.addElement(r.getString("res_date"));
                row.addElement(r.getString("p_category"));
                row.addElement(r.getString("p_price") + "만원");
                row.addElement(r.getString("p_name"));
                row.addElement(r.getString("s_name"));
                row.addElement(r.getString("s_phone"));
                row.addElement(r.getString("s_address"));
                row.addElement(r.getString("g_name"));
                row.addElement(r.getString("g_phone"));
                row.addElement(r.getString("g_address"));
                row.addElement(r.getString("g_request"));
                if (r.getString("track_print").equals("0")) {
                    row.addElement("-");
                } else {
                    row.addElement("출력 완료");// 12
                }
                setRow.addElement(row);
            }
        } catch (Exception e) {
            System.out.println("출력 작업 오류");
        }
    }
    return setRow;
}
```

Vector 객체를 생성하여 값을 저장하고 Vector 객체의 주소값을 리턴하여 받은 값으로 jTable의 행 데이터를 추가

MyPageGui 클래스

```
public void reArrange() {
    Vector<String> column = new Vector<String>(Arrays.asList("날짜", "품목", "가격", "등록명칭", "보내는 사람",
        "(보)주소", "받는 사람", "(받)전화번호", "(받)주소", "배송요청사항", "운송장 출력"));
    model = new DefaultTableModel(column, 0) {
        @Override
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };
    Vector<Vector> list = prd.myList(nowUser.getId());
    for (int i = 0; i < list.size(); i++) {
        model.addRow(list.get(i));
    }
    table = new JTable(model);
    js = new JScrollPane(table);
    add(js, BorderLayout.CENTER);
    js.setSize(688, 200);
    table.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
    this.setDefault(model);
    this.setJScroll(js);
    this.setTable(table);
    private DefaultTableModel model;
    private JTable table;
    private JScrollPane js;
}
```

메소드 호출 시마다 새 jTable 생성



6. 기능 관련 코드 증빙 및 해설-

5) 예약 정보 수정 후 Jtable의 값 업데이트 관련 코드 증빙



MyPageGui 클래스

```
}else if(e.getSource().equals(mypage)) {
    MyPageGui mpg =new MyPageGui(nowUser1);
    mpg.showbox();
    mpg.reArrange();

    private MyPageGui a;

    public UpdateGui(MyPageGui address) {
        a = address;

        setTitle("예약정보 수정");

        setBounds(100, 100, 600, 500);
        PostResDto info = a.selectCell();
        JPanel p = new JPanel(new GridLayout(11, 2));
        JPanel p1 = new JPanel(new FlowLayout(FlowLayout.LEFT));
        ptf.setText(info.getpPrice());
        atf.setText(info.getpName());
        sntf.setText(info.getsName());
        sptf.setText(info.getsPhone());
        satf.setText(info.getsAddress());
        gntf.setText(info.getgName());
        gp tf.setText(info.getgPhone());
        gatf.setText(info.getgAddress());
        grtf.setText(info.getgRequest());
```

사용자가 수정하고자 하는 행의 수정창에서 JTextField 에 해당 정보를 미리 보여지게 구현

UpdateGui 클래스

```
public PostResDto selectCell() {
    int row = table.getSelectedRow();
    TableModel data = table.getModel();

    PostResDto select = new PostResDto();
    select.setDate((String) data.getValueAt(row, 0));
    select.setCategory((String) data.getValueAt(row, 1));
    String price = cut((String) data.getValueAt(row, 2));
    select.setpPrice(price);
    select.setpName((String) data.getValueAt(row, 3));
    select.setsName((String) data.getValueAt(row, 4));
    select.setsPhone((String) data.getValueAt(row, 5));
    select.setsAddress((String) data.getValueAt(row, 6));
    select.setgName((String) data.getValueAt(row, 7));
    select.setgPhone((String) data.getValueAt(row, 8));
    select.setgAddress((String) data.getValueAt(row, 9));
    select.setgRequest((String) data.getValueAt(row, 10));
    select.setPrint((String) data.getValueAt(row, 11));
    select.setId(nowUser.getId(0).getId());

    return select;
}
```

MyPageGui 클래스

```
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource().equals(nUp)) {
        PostResDto exinfo = a.selectCell();
        PostResDto newinfo = getUpTf();
        prd.updateCell(newinfo, exinfo);
        JOptionPane.showMessageDialog(null, "수정되었습니다");
        a.removeScroll();
        a.reArrange();
        this.dispose();
    }
}
```

UpdateGui 클래스

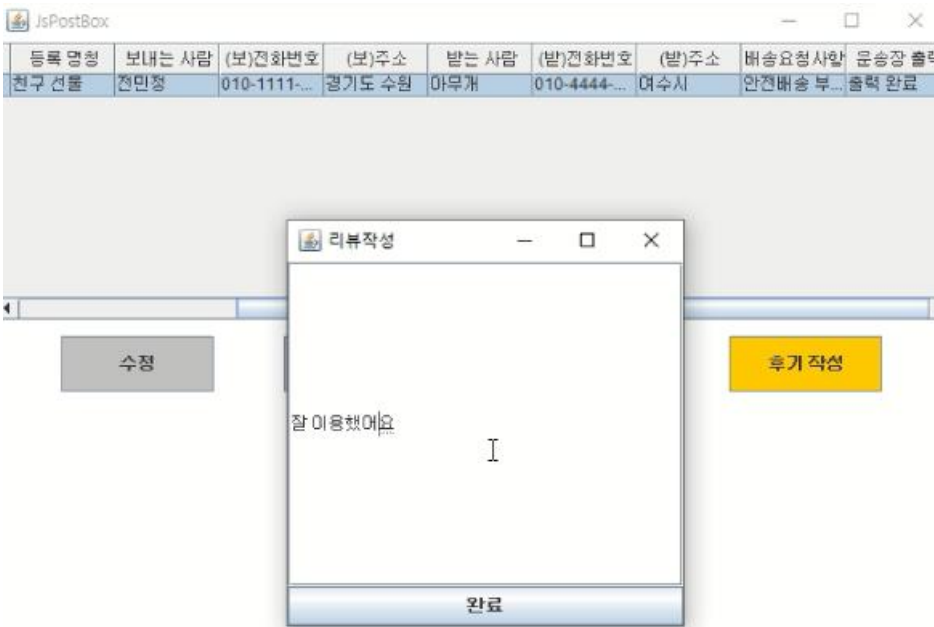
```
public void removeScroll() {
    this.remove(js);
}
```

MyPageGui 클래스

Jscrollpane을 제거하고 새 Jtable 생성

6. 기능 관련 코드 증빙 및 해설-

6) 이용후기 작성 및 조회 코드 증빙



```
private MyPageGui address;

public ReviewAddGui(MyPageGui ad) {
    address=ad;
}
```

ReviewAddGui 클래스

리뷰 개수에 따라 JLabel을 생성하여 Jpanel에 추가

ReviewShowGui 클래스



```
private ArrayList<ReviewDto> ard = rd.selectAll();
private int size = ard.size();
private JPanel panel = new JPanel(new GridLayout(size, 2));
private JButton complete = new JButton("완료");

public ReviewShowGui() {
    setTitle("후기");
    setBounds(100, 100, 300, 600);
    JLabel[] label = new JLabel[size];
    JLabel[] label2 = new JLabel[size];

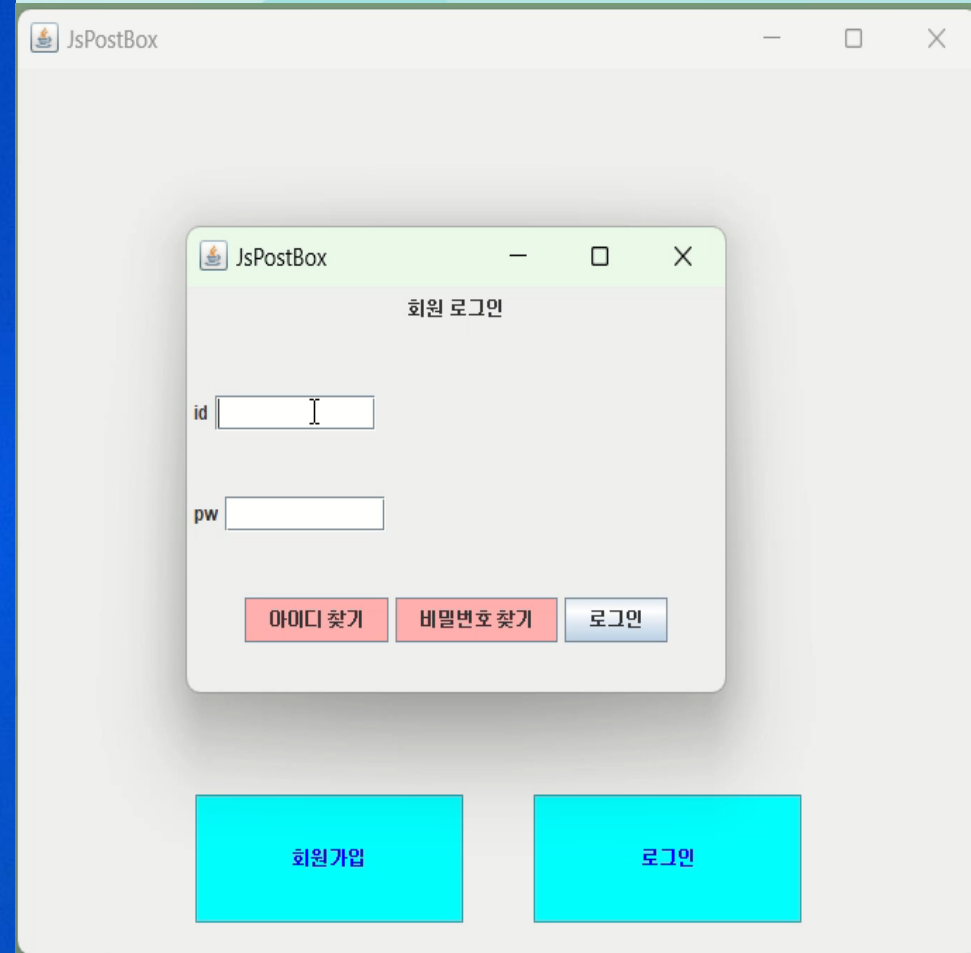
    for (int i = 0; i < size; i++) {
        label[i] = new JLabel();
        label[i].setText("아이디: " + ard.get(i).getId());
        panel.add(label[i]);

        label2[i] = new JLabel();
        label2[i].setText("내용: " + ard.get(i).getContents());
        panel.add(label2[i]);
    }
}
```

7. 시연 영상



사용자 모드



관리자 모드