

Java goupang 프로젝트

휴먼교육센터

개발기간/0411~0422

이태형

목차

1.요구사항

2.유스케이스

3.클래스다이어그램

4.테이블 명세서

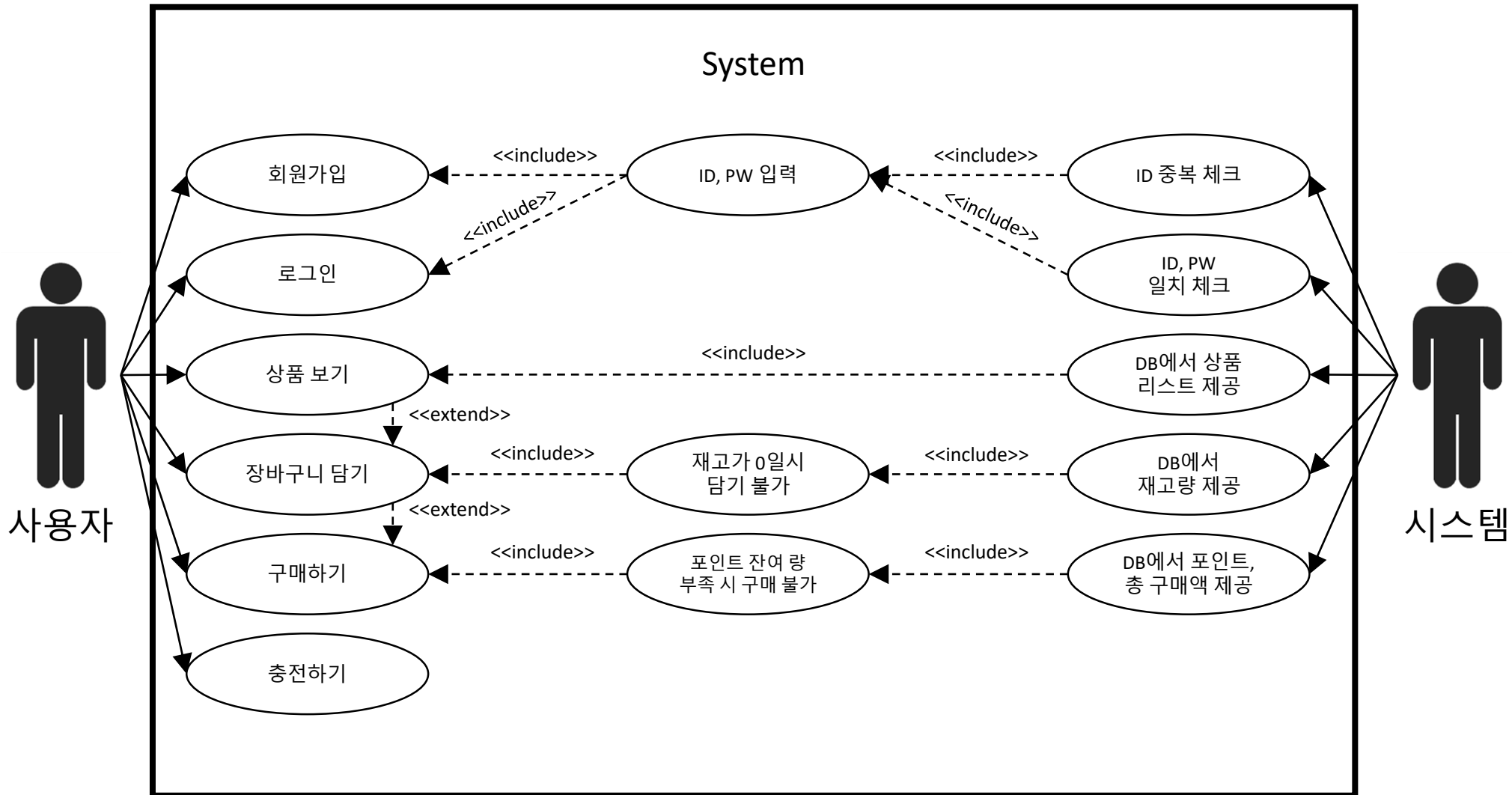
5.ERD

6.기능 별 주요 코드 해설

요구사항

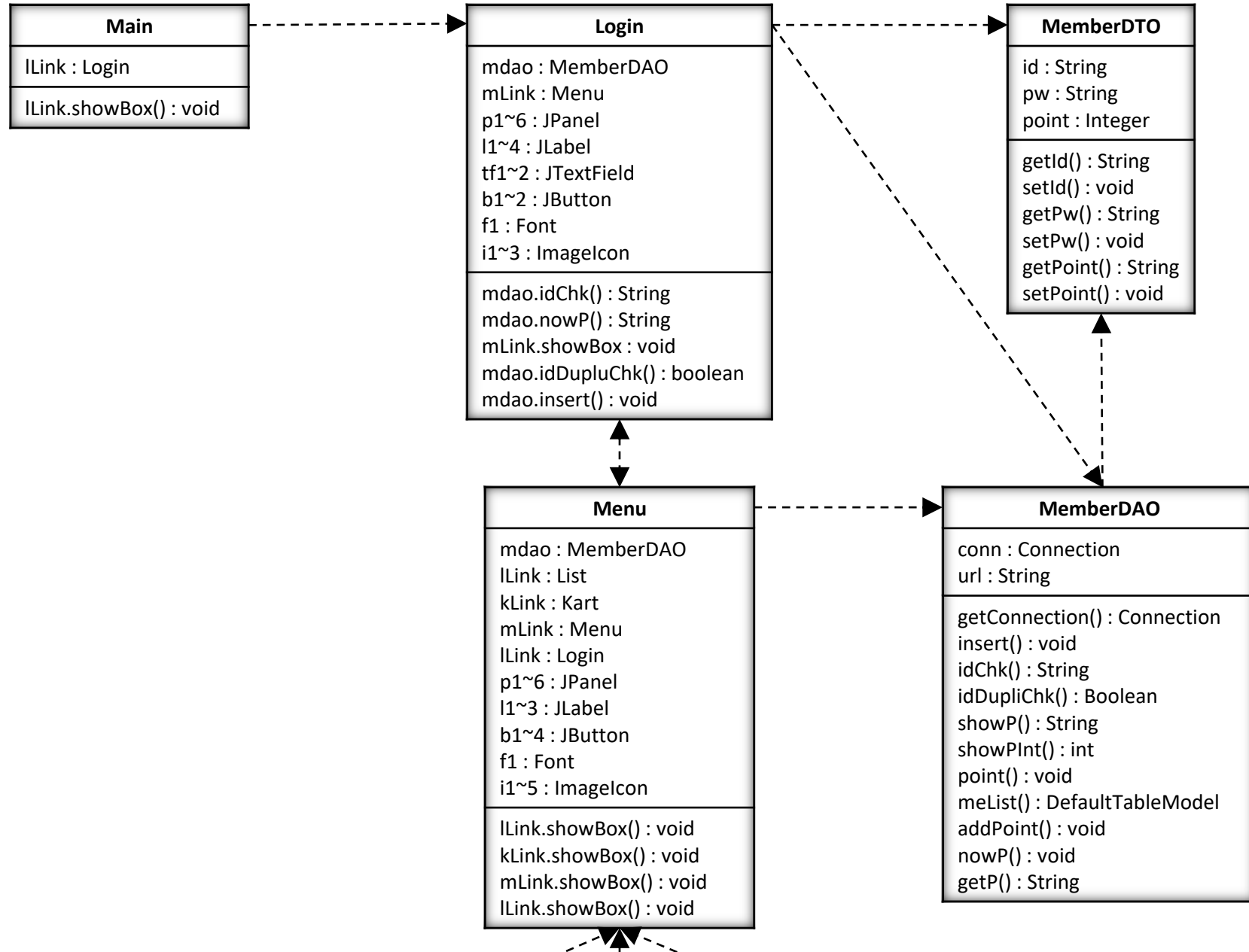
요구사항	필요 기능	기능 설명
회원제로 운영되는 인터넷 쇼핑 프로그램	로그인	ID, PW 확인후 로그인 및 기능 사용 가능
	회원가입	ID, PW 입력 후 회원가입 (DB에 회원정보 저장)
	포인트 충전	회원 포인트에 원하는 만큼 추가
	장바구니	원하는 상품 장바구니 추가 (재고 소진 시 추가 불가)
	결제	장바구니에 담긴 상품 결제 (포인트 부족할 시 결제 불가), (물품 재고, 포인트 차감)

유스케이스



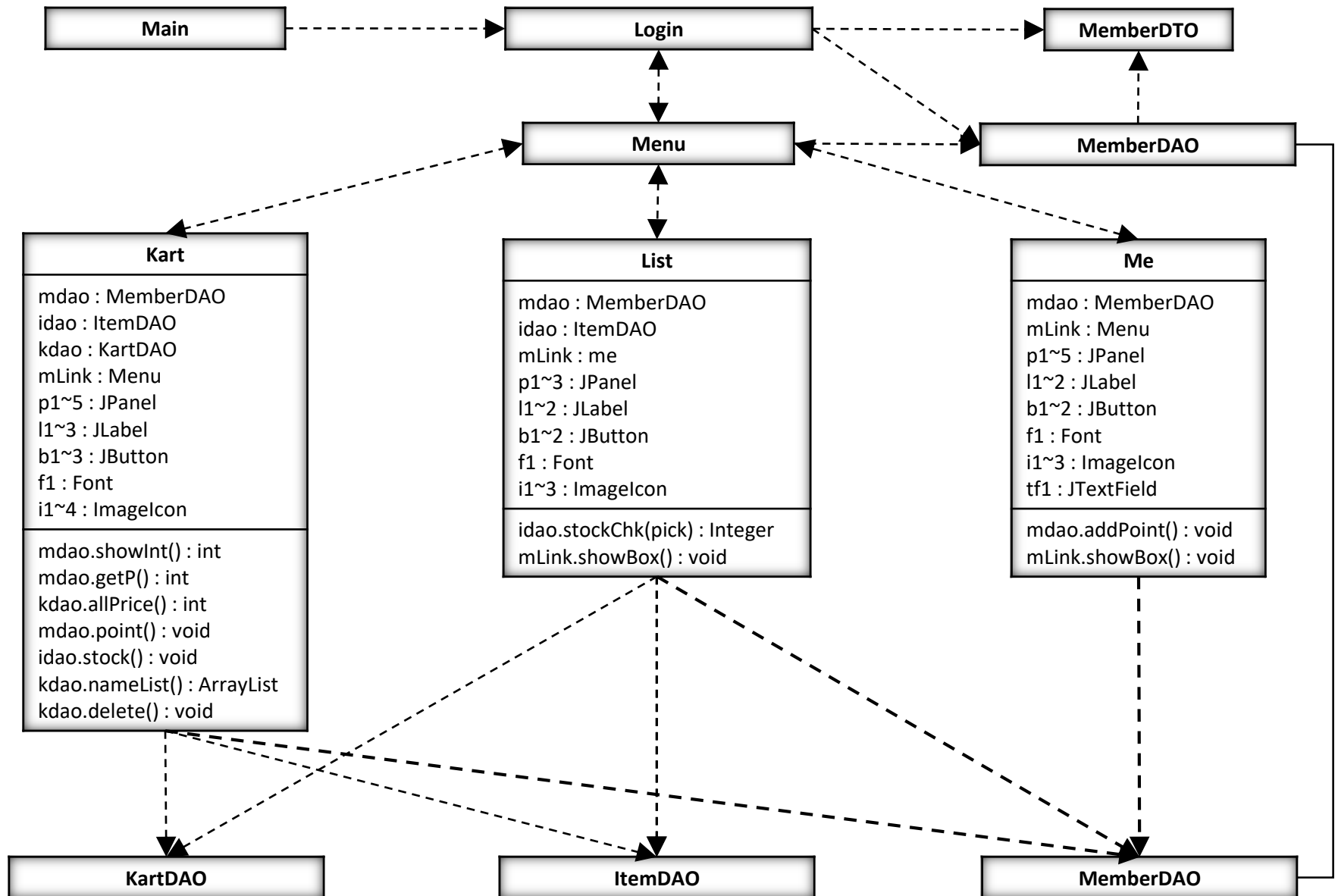
클래스 다이어그램-1

---> 의존



클래스 다이어그램-2

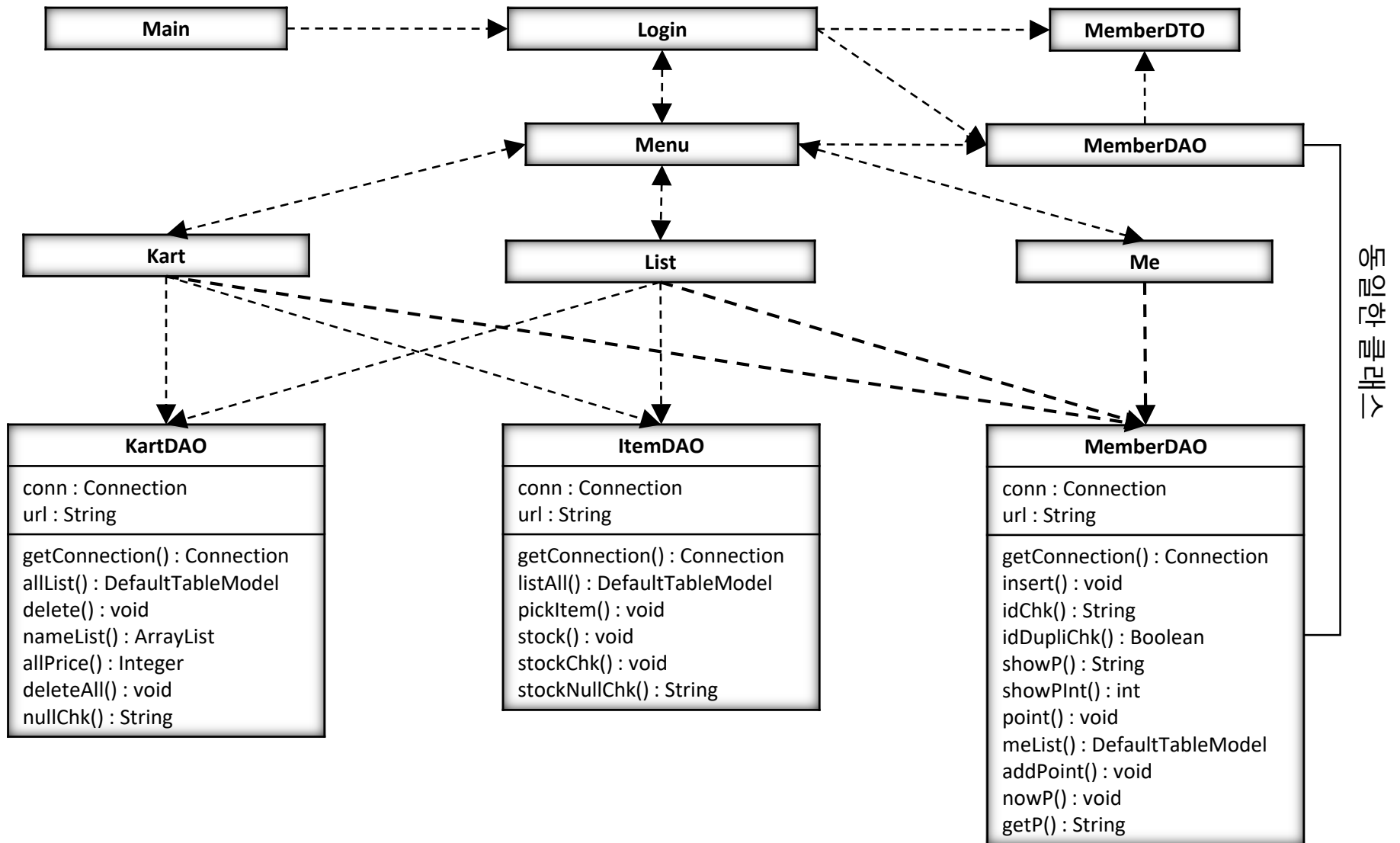
---> 의존



이와 같은 의존

클래스 다이어그램-3

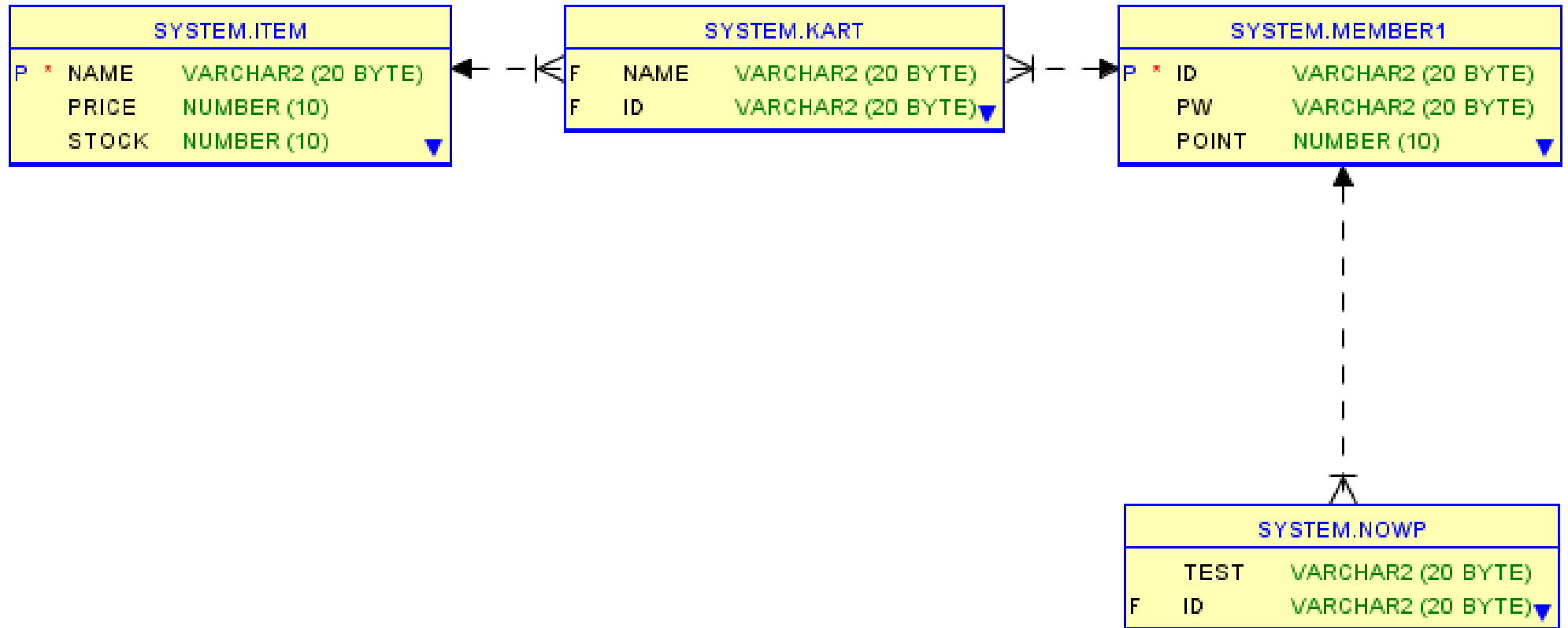
---> 의존



테이블 명세서

	속성명	컬럼명	타입	키
회원 테이블 /member1	아이디	id	varchar2	PK
	비밀번호	pw	varchar2	
	포인트	point	number	
상품 테이블/item	상품명	name	varchar2	PK
	가격	price	number	
	재고	stock	number	
장바구니 테이블 /kart	상품명	name	varchar2	FK
	아이디	id	varchar2	FK
현재 사용자 테이블 /nowp	임시 컬럼	test	varchar2	
	현재 사용자 아이디	id	varchar2	FK

ERD



주요 코드 해설

주요 코드 해설

다음 클래스의 GUI로 넘어가기

Main 클래스

```
package main;

import gui.Login;

public class Main {

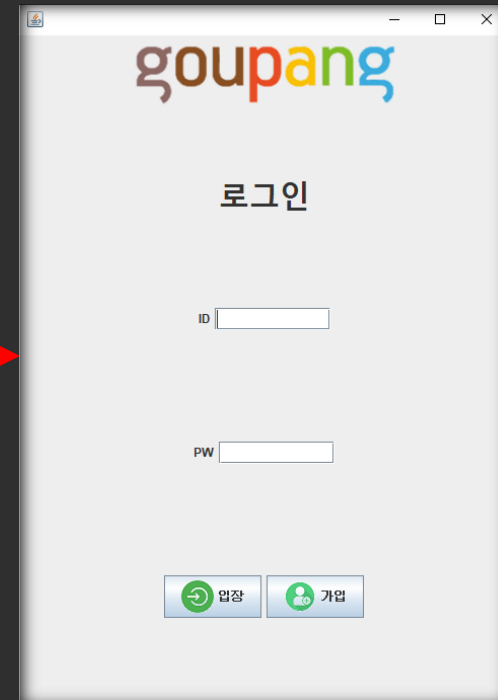
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Login lLink = new Login();
        lLink.showBox();
    }

}
```

Login 클래스

```
public void showBox() {
    // TODO Auto-generated method stub
    this.setVisible(true);
}
```

Login 클래스를 참조하는 객체 생성 후 로그인 화면 GUI를 표시해주는 메서드 .showBox()를 호출한다. 그리고 showBox()는 .setVisible(); 메서드를 통해 생성한 GUI를 보여준다.



첫 화면

주요 코드 해설

버튼에 기능 추가하기

Login 클래스

```
public class Login extends JFrame implements ActionListener
```

버튼클릭 기능을 사용하기 위해 ActionListener를 사용.
.addActionListener() 메서드를 사용하여 버튼(b1)에 기능을 부여.
if문을 사용하여 각 버튼마다 다른 기능을 부여.

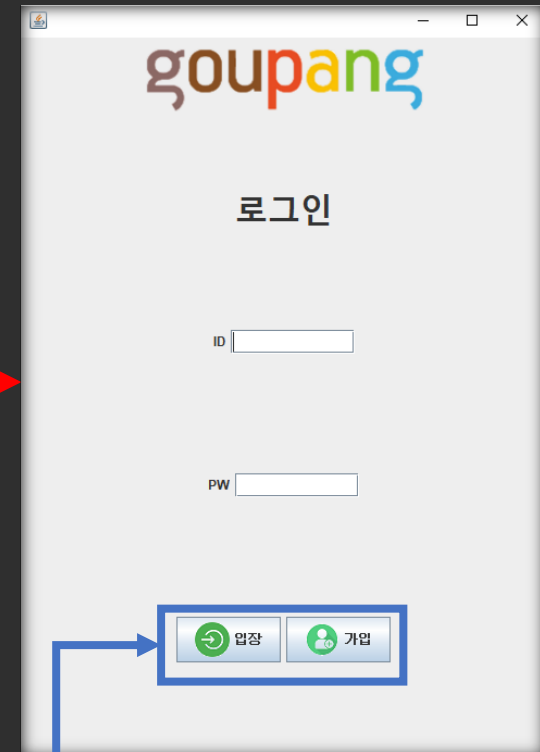
Login 클래스

```
b1.addActionListener(this);  
b2.addActionListener(this);
```

Login 클래스

```
@Override  
public void actionPerformed(ActionEvent e) {  
    // TODO Auto-generated method stub  
    if (e.getSource().equals(b1)) {
```

기능이 생긴 버튼



주요 코드 해설

MemberDAO 클래스

로그인 확인 코드

```
public String idChk(MemberDTO m) { // 로그인
    // TODO: Auto-generated method stub
    String sql = "select * from member1 where id = ? and pw = ?";
    String nowP = null;
    PreparedStatement psmt = null;
    if (getConnection() != null) {
        try {
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, m.getId());
            psmt.setString(2, m.getPw());
            ResultSet rs = null;
            rs = psmt.executeQuery();
            while (rs.next()) {
                if (m.getId().equals(rs.getString("id")) && m.getPw().equals(rs.getString("pw"))) {
                    System.out.println("로그인 성공");
                    nowP = rs.getString("id");
                } else {
                    System.out.println("로그인 실패");
                }
            }
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("로그인 실패");
        }
    }
    return nowP;
}
```

Login 클래스

```
String nowP = mdao.idChk(m);
if (nowP == null) {
    l1.setText("잘못된 아이디 또는 비밀번호");
} else {
    this.setVisible(false);
    mdao.nowP(nowP);
    mLink.showBox();
}
```

입력받은 id와 pw를 매개변수로 .idChk() 메서드에 넣은 후, 쿼리 문을 통해 해당 id와 pw가 있는지 확인 한다. 그리고 해당 아이디가 존재할 시 임시로 선언된 변수 nowP에 저장되며, if문을 통해 다음 GUI클래스로 넘어가고, 해당 아이디가 없을 시, null값으로 Return되어 메시지창을 통해 로그인이 불가능함을 알려준다.

주요 코드 해설

MemberDAO 클래스

회원가입 중복 체크

```
public boolean idDupliChk(String id) { // 로그인
    // TODO: Auto-generated method stub
    String sql = "select * from member1 where id = ?";
    PreparedStatement psmt = null;
    if (getConnection() != null) {
        try {
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, id);
            ResultSet rs = null;
            rs = psmt.executeQuery();
            while (rs.next()) {
                if (id.equals(rs.getString("id"))) {
                    System.out.println("아이디 중복");
                    return true;
                } else {
                    System.out.println("아이디 중복 아님");
                    return false;
                }
            }
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("아이디 중복 체크 실패");
        }
    }
    return false;
}
```

Login 클래스

```
} else if (e.getSource().equals(b2)) {
    String getId = tf1.getText();
    String getpw = tf2.getText();
    MemberDTO m = new MemberDTO();
    m.setId(getId);
    m.setPw(getpw);
    if(mdao.idDupliChk(getId)) {
        l1.setText("중복된 아이디");
    } else {
        mdao.insert(m);
    }
}
```

회원가입시 입력받은 아이디가 이미 존재 하는지 확인한다.

주요 코드 해설

로그인 확인 코드

로그인 실패

The screenshot shows the Goupang login interface. At the top is the 'goupang' logo. Below it, a blue-bordered box contains the text '잘못된 아이디 또는 비밀번호' (Incorrect ID or password). Underneath, there are two input fields: 'ID' with the value 'p4' and 'PW' with the value '3214'. At the bottom, there are two buttons: '입장' (Login) with a green circular arrow icon and '가입' (Sign Up) with a green circular icon containing a person silhouette.

아이디 중복

The screenshot shows the Goupang login interface. At the top is the 'goupang' logo. Below it, a blue-bordered box contains the text '중복된 아이디' (Duplicate ID). Underneath, there are two input fields: 'ID' with the value 'p2' and 'PW' with the value '1234'. At the bottom, there are two buttons: '입장' (Login) with a green circular arrow icon and '가입' (Sign Up) with a green circular icon containing a person silhouette.

주요 코드 해설

MemberDAO 클래스

현재 접속 중인 아이디 판별

```
public void nowP(String nowP) { //현재 접속중인 아이디 저장
    // TODO Auto-generated method stub
    if (getConnection() != null) {
        try {
            String sql = "update nowp set id = (?) where test = test";
            PreparedStatement pstmt = null;
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, nowP);
            int cnt = pstmt.executeUpdate();
            System.out.println(cnt + "건 현재아이디 입력 성공");
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("현재 아이디 입력 실패");
        }
    }
}

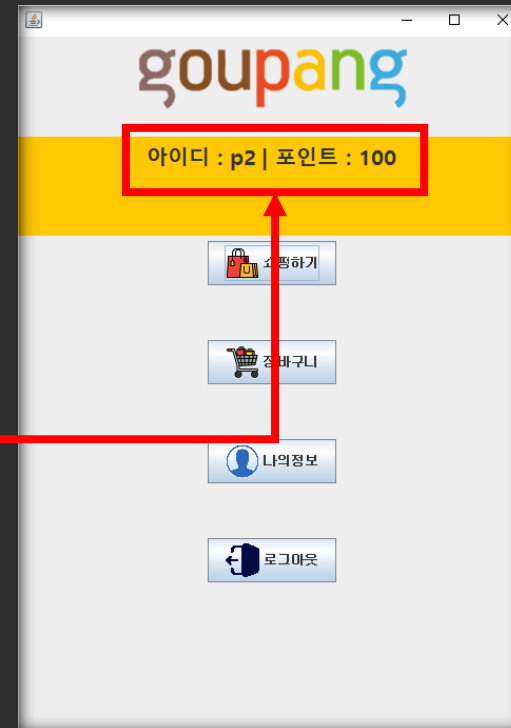
public String getP() { //현재 접속중인 아이디 출력
    // TODO Auto-generated method stub
    String nowP = null;
    if (getConnection() != null) {
        try {
            String sql = "select id from nowp where test = test";
            Statement stmt = null;
            stmt = conn.createStatement();
            ResultSet rs = null;
            rs = stmt.executeQuery(sql);
            while(rs.next()) {
                nowP = rs.getString("id");
                return nowP;
            }
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("아이디 출력 실패");
        }
    }
    return nowP;
}
```

nowp 테이블

TEST	ID
1 test	p2

로그인할때 사용된 id를 .nowP()메서드의 매개변수로 넣어 nowp테이블의 id컬럼에 저장한다.

그리고, 현재 접속중인 사용자의 id를 사용하고자 할때 .getP() 메서드를 사용하여 현재 접속중인 id를 Return받아 사용한다.



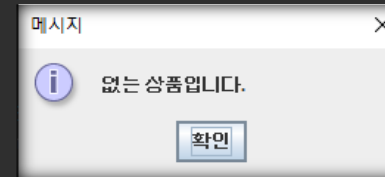
주요 코드 해설

ItemDAO 클래스

장바구니 담기-상품의 존재여부

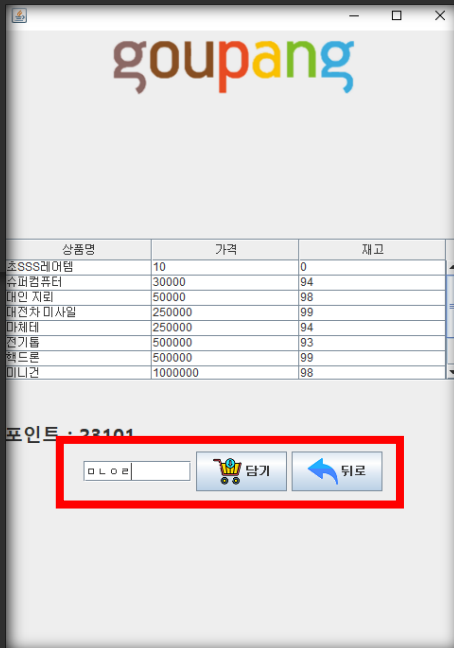
```
public String stockNullChk(String pick) {  
    // TODO Auto-generated method stub  
    String stock = null;  
    if (getConnection() != null) {  
        try {  
            String sql = "select stock from item where name = (?)";  
            PreparedStatement pstmt = null;  
            pstmt = conn.prepareStatement(sql);  
            pstmt.setString(1, pick);  
            ResultSet rs = null;  
            rs = pstmt.executeQuery();  
            while(rs.next()) {  
                stock = rs.getString("stock"+"");  
                return stock;  
            }  
        }  
    }  
}
```

.stockNullChk()에서 쿼리문 입력 결과값이 null일 경우 없는 상품으로 나온다.



List 클래스

```
public void actionPerformed(ActionEvent e) {  
    // TODO Auto-generated method stub  
    if (e.getSource().equals(b1)) {  
        String pick = tf1.getText();  
        if (idao.stockNullChk(pick) == null) {  
            JOptionPane.showMessageDialog(null, "없는 상품입니다.");  
        } else {  
            if (idao.stockChk(pick) <= 0) {  
                JOptionPane.showMessageDialog(null, "재고가 부족합니다.");  
            } else {  
                idao.pickItem(pick, mdao.getP());  
            }  
        }  
    } else if (e.getSource().equals(b2)) {  
        this.setVisible(false);  
        Menu mLink = new Menu();  
        mLink.showBox();  
    }  
}
```



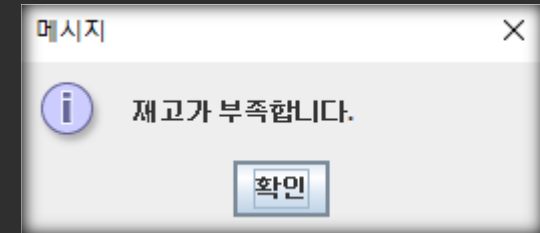
주요 코드 해설

ItemDAO 클래스

장바구니 담기-재고량

```
public int stockChk(String pick) {
    // TODO Auto-generated method stub
    Integer stock = null;
    if (getConnection() != null) {
        try {
            String sql = "select stock from item where name = (?)";
            PreparedStatement psmt = null;
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, pick);
            ResultSet rs = null;
            rs = psmt.executeQuery();
            while(rs.next()) {
                stock = rs.getInt("stock");
                return stock;
            }
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("재고량 출력 실패");
        }
    }
    return stock;
}
```

.stockChk() 메서드를 통하여 재고가 0일시 담을 수가 없다.



List 클래스

```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    if (e.getSource().equals(b1)) {
        String pick = tf1.getText();
        if (idao.stockNullChk(pick) == null) {
            JOptionPane.showMessageDialog(null, "없는 상품입니다.");
        } else {
            if (idao.stockChk(pick) <= 0) {
                JOptionPane.showMessageDialog(null, "재고가 부족합니다.");
            } else {
                idao.pickItem(pick, mdao.getP());
            }
        }
    }
}
```

상품명	가격	재고
초SSS레머템	10	0
슈퍼컴퓨터	30000	94
대인 지뢰	50000	98
대전차 미사일	250000	99
마체테	250000	94
전기톱	500000	93
핵드론	500000	99
미니건	1000000	98

주요 코드 해설

List 클래스

장바구니 담기-상품명 저장

```
public void actionPerformed(ActionEvent e) {  
    // TODO Auto-generated method stub  
    if (e.getSource().equals(b1)) {  
        String pick = tf1.getText();  
        if (idao.stockNullChk(pick) == null) {  
            JOptionPane.showMessageDialog(null, "없는 상품입니다.");  
        } else {  
            if (idao.stockChk(pick) <= 0) {  
                JOptionPane.showMessageDialog(null, "재고가 부족합니다.");  
            } else {  
                idao.pickItem(pick, mdao.getP());  
            }  
        }  
    }  
}
```

상품존재여부, 재고량 체크를 전부 통과하고 나면, kart테이블에 상품명과 현재 접속중인 아이디가, 함께 저장된다.

ItemDAO 클래스

```
public void pickItem(String pick, String nowP) { // 장바구니 담기  
    // TODO Auto-generated method stub  
    if (getConnection() != null) {  
        try {  
            String sql = "insert into kart values (?, ?)";  
            PreparedStatement psmt = null;  
            psmt = conn.prepareStatement(sql);  
            psmt.setString(1, pick);  
            psmt.setString(2, nowP);  
            int cnt = psmt.executeUpdate();  
            System.out.println(cnt + "건 장바구니 입력 완료");  
        } catch (Exception e) {  
            // TODO: handle exception  
            System.out.println("장바구니 입력 실패");  
        }  
    }  
}
```

상품명	가격	재고
슈퍼컴퓨터	30000	96
대인 지뢰	50000	99

포인트 : 100

총액 : 80000

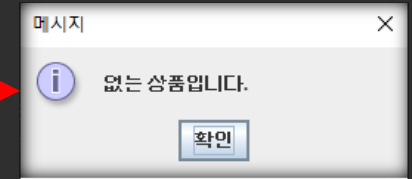
NAME	ID
1 대인 지뢰	p2
2 슈퍼컴퓨터	p2

주요 코드 해설

장바구니 삭제

Kart 클래스

```
} else if (e.getSource().equals(b3)) {  
    String name = tf1.getText();  
    if (kdao.nullChk(name) == null) {  
        JOptionPane.showMessageDialog(null, "없는 상품입니다.");  
    } else {  
        kdao.delete(name, mdao.getP());  
    }  
}
```



삭제하고자 하는 상품이 존재 하는지
.nullChk() 메서드를 통해 확인 후,
.delete() 메서드에 상품명과 현재접속
중인 id를 매개변수로 넣고 삭제한다.

상품명	가격	재고
대인 지리	50000	98

포인트 : 23100

총액 : 50000

```
public void delete(String name, String nowP) { // 삭제  
    // TODO Auto-generated method stub  
    if (getConnection() != null) {  
        try {  
            String sql = "delete from kart where name = (?) and id = (?)";  
            PreparedStatement psmt = null;  
            psmt = conn.prepareStatement(sql);  
            psmt.setString(1, name);  
            psmt.setString(2, nowP);  
            int cnt = psmt.executeUpdate();  
            System.out.println(cnt + "건 삭제 완료");  
        } catch (Exception e) {  
            // TODO: handle exception  
            System.out.println("삭제 실패");  
        }  
    }  
}
```

KartDAO 클래스

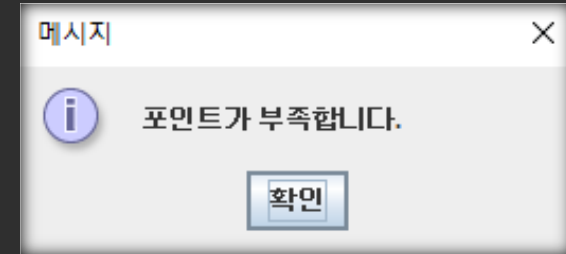
주요 코드 해설

MemberDAO 클래스

구매하기 기능-포인트 확인

```
public int showPInt(String nowP) { // 포인트 출력
    // TODO: Auto-generated method stub
    String sql = "select point from member1 where id = ?";
    int showP = 0;
    PreparedStatement psmt = null;
    if (getConnection() != null) {
        try {
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, nowP);
            ResultSet rs = null;
            rs = psmt.executeQuery();
            while (rs.next()) {
                showP = rs.getInt("point");
            }
            System.out.println("포인트 출력 성공");
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("포인트 출력 실패");
        }
    }
    return showP;
}
```

현재 사용자의 잔여포인트 - 장바구니의 총 상품가격이 0보다 작을 시 구매가 불가능하다.



```
public Integer allPrice(String nowP) { // 가격 총합 가져오기
    // TODO: Auto-generated method stub
    String sql = "select price from item inner join kart on kart.name = item.name where kart.id = (?)";
    int sum = 0;
    if (getConnection() != null) {
        try {
            PreparedStatement psmt = null;
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, nowP);
            ResultSet rs = null;
            rs = psmt.executeQuery();
            while (rs.next()) {
                sum = sum + rs.getInt("price");
            }
            System.out.println("가격 불러오기 성공");
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("가격 불러오기 실패");
        }
    }
    return sum;
}
```

Kart 클래스

```
if (e.getSource().equals(b1)) { // 구매하기 버튼 클릭 시
    if (mdao.showPInt(mdao.getP()) - kdao.allPrice(mdao.getP()) < 0) {
        JOptionPane.showMessageDialog(null, "포인트가 부족합니다.");
    } else {
        mdao.point(kdao.allPrice(mdao.getP()), mdao.getP());
        idao.stock(kdao.nameList(mdao.getP()));
        kdao.deleteAll(mdao.getP());
    }
}
```

KartDAO 클래스

주요 코드 해설

Kart 클래스

구매하기 기능-포인트 차감

```
if (e.getSource().equals(b1)) { // 재고량 감소, 포인트 감소, 장바구니에서 삭제
    if (mdao.showPInt(mdao.getP()) - kdao.allPrice(mdao.getP()) < 0) {
        JOptionPane.showMessageDialog(null, "포인트가 부족합니다.");
    } else {
        mdao.point(kdao.allPrice(mdao.getP()), mdao.getP());
        idao.stock(kdao.nameList(mdao.getP()));
        kdao.deleteAll(mdao.getP());
    }
}
```

현재 사용자의 잔여 포인트에서
장바구니 상품의 총액을 뺀다.

```
public void point(Integer sum, String name) { // 포인트 차감
    // TODO Auto-generated method stub
    if (getConnection() != null) {
        try {
            String sql = "update member1 set point = point - (?) where id = (?)";
            PreparedStatement psmt = null;
            psmt = conn.prepareStatement(sql);
            psmt.setInt(1, sum);
            psmt.setString(2, name);
            int cnt = psmt.executeUpdate();
            System.out.println(cnt + "건 포인트 차감 성공");
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("포인트 차감 실패");
        }
    }
}
```

MemberDAO 클래스

주요 코드 해설

Kart 클래스

구매하기 기능-재고차감

```
if (e.getSource().equals(b1)) { // 재고량 감소, 포인트 감소, 장바구니에서 삭제
    if (mdao.showPInt(mdao.getP()) - kdao.allPrice(mdao.getP()) < 0) {
        JOptionPane.showMessageDialog(null, "포인트가 부족합니다.");
    } else {
        mdao.point(kdao.allPrice(mdao.getP()), mdao.getP());
        idao.stock(kdao.nameList(mdao.getP()));
        kdao.deleteAll(mdao.getP());
    }
}
```

장바구니에 담긴 상품들의 상품명을 가져와, 해당 상품의 재고를 -1씩 차감한다.

```
public ArrayList<String> nameList(String nowP) { // 이름 가져오기
    // TODO Auto-generated method stub
    ArrayList<String> name = new ArrayList<>();
    if (getConnection() != null) {
        try {
            String sql = "select * from kart where id = (?)";
            PreparedStatement psmt = null;
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, nowP);
            ResultSet rs = null;
            rs = psmt.executeQuery();
            while (rs.next()) {
                name.add(rs.getString("name"));
            }
        } catch (Exception e) {
            // TODO: handle exception
        }
    }
    return name;
}
```

ItemDAO 클래스

```
public void stock (ArrayList<String> iList) { // 재고 차감
    // TODO Auto-generated method stub
    if (getConnection() != null) {
        try {
            String sql = "update item set stock = stock -1 where name = (?)";
            PreparedStatement psmt = null;
            psmt = conn.prepareStatement(sql);
            for (int i = 0; i < iList.size(); i++) {
                psmt.setString(1, iList.get(i));
                int cnt = psmt.executeUpdate();
                System.out.println(cnt + "건 재고조정 완료");
            }
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("재고 조정 실패");
        }
    }
}
```

KartDAO 클래스

주요 코드 해설

Kart 클래스

구매하기 기능

```
if (e.getSource().equals(b1)) { // 재고량 감소, 포인트 감소, 장바구니에서 삭제
    if (mdao.showPInt(mdao.getP()) - kdao.allPrice(mdao.getP()) < 0) {
        JOptionPane.showMessageDialog(null, "포인트가 부족합니다.");
    } else {
        mdao.point(kdao.allPrice(mdao.getP()), mdao.getP());
        idao.stock(kdao.nameList(mdao.getP()));
        kdao.deleteAll(mdao.getP());
    }
}
```

구매 한 후에는, 장바구니를 초기화 시킨다.

```
public void deleteAll(String nowP) { // 장바구니 전체 삭제
    // TODO Auto-generated method stub
    if (getConnection() != null) {
        try {
            String sql = "delete from kart where id = (?)";
            PreparedStatement psmt = null;
            psmt = conn.prepareStatement(sql);
            psmt.setString(1, nowP);
            psmt.executeQuery();
            System.out.println("삭제 성공");
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("삭제 실패");
        }
    }
}
```

KartDAO 클래스

goupang

나의 정보

아이디	포인트
p2	20100

3000 + 충전

주요 코드 해설

포인트 충전 기능

아이디	포인트
p2	23100

뒤로

+ 충전

```

public void addPoint(String point, String id) { // 포인트 충전
    // TODO Auto-generated method stub
    if (getConnection() != null) {
        try {
            String sql = "update member1 set point = point + (?) where id = (?)";
            PreparedStatement pstmt = null;
            pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, Integer.parseInt(point));
            pstmt.setString(2, id);
            int cnt = pstmt.executeUpdate();
            System.out.println(cnt + "건 포인트 충전 성공");
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("포인트 충전 실패");
        }
    }
}
MemberDAO 클래스
    
```

현재 접속중인 사용자의 id와 충전할 포인트를 매개변수로 받아 DB의 데이터를 업데이트하여 포인트 충전이 가능하다.

감사합니다