

소프트 웨어 개발 방법론 최종 보고서

목차

1. 과제 개요
2. 사용사례 분석
3. DB설계
4. UI설계
5. SSD 및 클래스도
6. 코딩 및 실행결과
7. 결론.
8. Q&A

1 과제 개요

1.1. 과제의 필요성(배경)

우리 나라에서 장사 혹은 무슨 물건을 팔 때 일 정산, 월 정산 확실한 재고 관리를 하기 나름이다. 사람이 모두 정확할 수는 없지 않은가? 그리고 일손이 부족하거나 매장이 바쁠 경우에는 돈, 재고관리가 제대로 된 관리가 필요한 시점이다.

1.2. 과제의 목적

현금 정산 카드 정산 그리고 부가세 그리고 재고 인수, 인출, 인건비, 식대, 일 매출, 월 매출, 연 매출 까지 관리 할 수 있는 프로그램

1.3. 과제의 범위

매출 관리 금고관리 가계부 관리 매출관리 가 주요 기능이 며 pos시스템이나 판매 쪽은 제외,

1과제 개요

1.4. 사용자 및 주요 관심사항.

사용자 분석표

사용자	주요 관심사	주요기능
근무자	매출 및 재고 확인	매출, 재고, 가계부 관리기능
사장	모든기능 확인	매출, 재고, 가계부, 금고관리기능

1.5. 주요 기능

주요 기능 분석표

기능명	기능설명	주요사용자	사용빈도
매출관리	년,월,일 매출을 볼수있다	근무자,사장	고
재고관리	재고유,무 수량 파악	근무자,사장	고
가계부 관리	부수 자재 매입	근무자,사장	중
금고 관리	모든 물질적 매출 관리	사장	고

1 과제 개요

1.6 제약사항

사용자가 하나하나 입력을 해야하고 그리고 정보가 모이기 전까지는 기능을 모두 활용하기엔 어렵다고 본다. 그리고 금액이 완전 정확하게 관리될 수도 있으나 근무자나 잘못된 입력할 시에 오차가 있을 수도 있다.

1.7. 기대효과

소규모 사장님들이 손쉽게 사용하여 금액 관리나 자재관리를 손쉽게 관리할 수 있다.

그리고 매장 밖에서 매입하는 물건 관리를 철저하게 할 수 있거 금고관리에서 금액누수를 적은 확률로 피할 수 있다.

2.사용사례 분석

2 사용자례분석

2.1. 사용자 분석

프로젝트 개요에서 언급된 사용자 분석표로 요약된 사용자, 주요관심사, 사용할 주요 기능을 상세하게 기술한다. 완성된 소프트웨어를 사용할 사람들을 식별하고, 각 사람들이 소프트웨어를 통해 얻고자 하는 관심사항을 상세하게 기술한다.

액터명	시스템 관심사	사용사례목록	우선순위(사용빈도)
직원	신속하고 빠르게 매출/재고관리 가능	상품/매출 관리	1
		재고 관리	2
		가계부 관리	3
사장	정확하고 편리한 금고 관리 및 모든 기능 관리	금고 관리	1
		상품/매출 관리	2
		재고 관리	3
		가계부 관리	4

2 사용사례분석

2.2 사용사례도



2 사용사례분석

2.2 사용사례 도

번호	사용사례명	사용 액터	설명	관련 사용사례
UC1	매출 확인	직원,사장님	현재 매출을 확인	상품추가/삭제
UC2	상품추가/삭제	직원,사장님	매장에 있는 상품을 관리한다.	재고수량파악
UC3	재고수량파악	직원,사장님	현재 매장에 있는 물품의 수량확인	재고추가 / 삭제
UC4	재고추가/삭제	직원,사장님	자재를 입력하고 삭제한다.	가계부 입력/삭제
UC5	가계부입력/삭제	직원,사장님	자재나 매장용품을 입력하고 삭제하여 금액을 관리한다.	금고 관리
UC6	금고 관리	사장님	금고의 총금액을 관리한다.	

2 사용자 사례 분석

2.3 사용자 사례 정의

사용 사례	UC1. 매출 확인
액터	직원, 사장님
시작 조건	개인 정보 입력 후 매출 관리 버튼을 누름.
종료 조건	매출 과 제품 정보 확인
주요 흐름	<ol style="list-style-type: none"> 1. 직원이 매출 관리 메뉴를 선택한다. 2. 시스템은 테이블과 매뉴판을 보여주면서, 매출을 확인할 날짜와 제품 정보를 확인할 수 있는 메뉴를 선택하게 한다. 3. 근무자는 년 월 일 자기가 원하는 매출 검색을 할 수 있다. 4. 시스템은 직원이 원하는 매출을 검색하여 화면에 출력한다. 5. 근무자는 메장의 매출을 확인하고 종료한다.
대안 흐름	<p>티 조회할 매출이나 제품 정보가 없을 때.(데이터가 많이 저장되지 않을 때)</p> <ol style="list-style-type: none"> 1. 시스템은 년 단위 출력을 하지 못할 시 데이터가 없습니다. 라고 창을 띄운다. 2. 제품 확인 정보 확인 시 데이터가 없을 때 추가 하라고 하는 창을 띄운다. 3. 매출이 실수로 백업 되었을 경우에 다시 복원할 수 있다.
비기능 요구 사항	<ol style="list-style-type: none"> 1. 데이터 베이스 데이터 전송 속도가 3초 이내여야 한다. 2. 매출 정보는 영구적으로 보관해야 한다.(단 망할 시 까지)

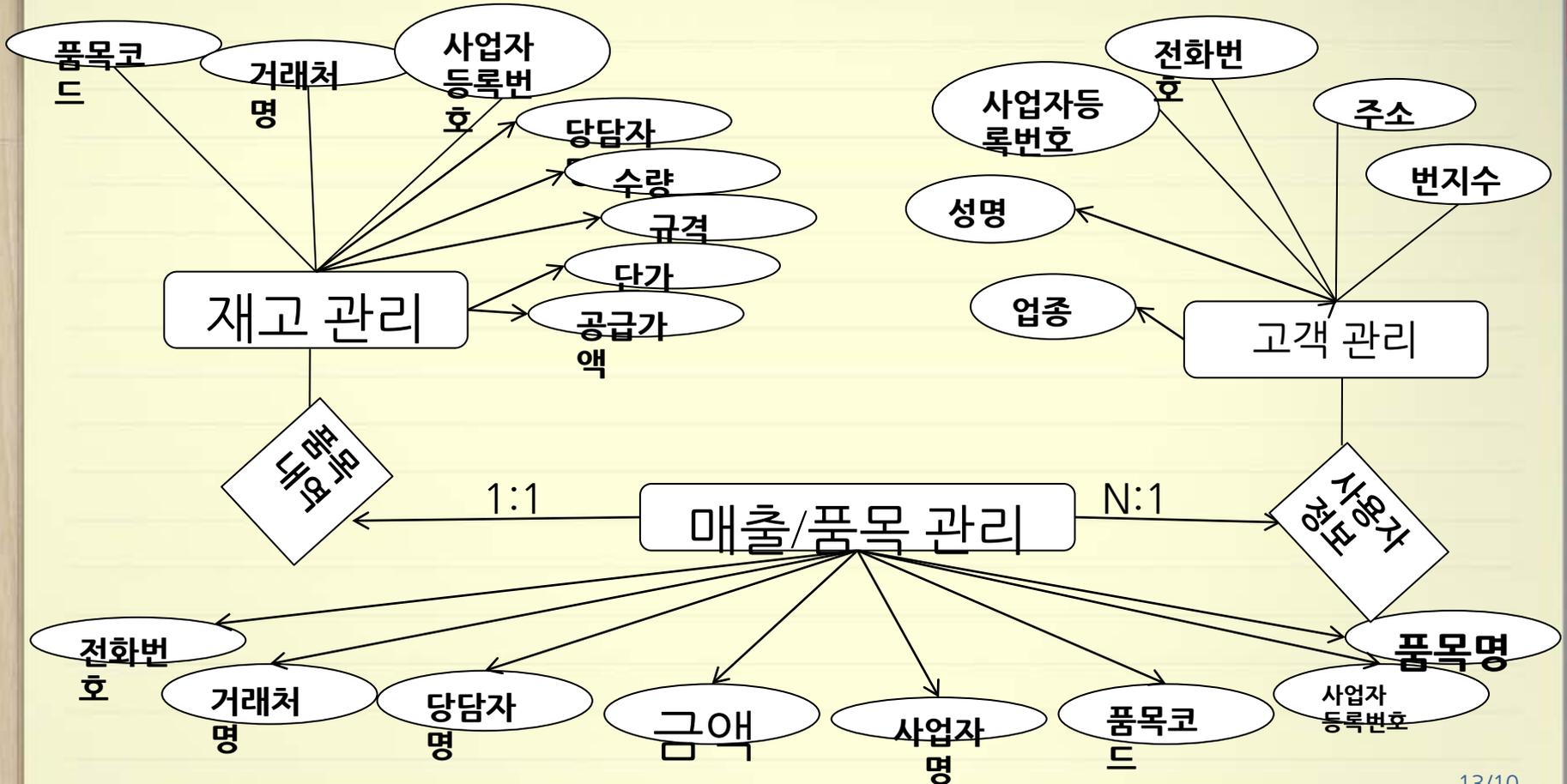
2 사용사례분석

사용사례	UC3. 재고관리
액터	직원,사장님
시작조건	개인정보 입력 후 재고 관리 메뉴를 선택
종료조건	제품 명 ,원가 ,단가, 재고 수량 ,기타정보 확인
주요흐름	<ol style="list-style-type: none"> 1. 직원이 재고 관리 메뉴를 선택한다. 2. 시스템은 테이블과 매뉴판을 보여주면서 , 등록 ,수정,삭제 기능 선택 가능하다. 3. 원하는 상품을 검색 가능하다. 4. 시스템은 직원이 원하는 자제 을 검색하여화면에 출력한다. 5. 근무자는 매장 재고 품목 확인하고 종료한다.
대안흐름	<p>E1 조회할 자재 데이터가 없을 때.(데이터가 많이 저장되지 않을때)</p> <ol style="list-style-type: none"> 1.시스템은 자재 재고 가 1c/s남으면 추가 하라는 창이 뜬 2.원하는 자재 명이 없으면 추가 가능하다.. 3. 매출이 실수로 백업 뒀을경우에 다시 복원할수 있다.
비기능요구사항	<ol style="list-style-type: none"> 1. 검색 후 확인속도가 3초 이내여야 한다. 2. 매출정보는 영구적으로 보관해야 한다.(단 망할 시 까지)

3.DB 설계

3.DB설계

3.1 ER 다이어그램



3.DB설계

3.2 테이블 모델

데이터클래스명	한글명	설명	주요속성
Sale	매출 확인	현재 매출을 확인	판매ID, 판매총액, 할인금액, 세금, 결제정보, 구매회원정보, 판매시각, 판매항목목록
Salesadd	상품	매장에 있는 상품을 관리한다.	물품ID, 단가, 수량, 가격, 세금, 할인 금액 등
stock	재고수량	현재 매장에 있는 물품의 수량확인	품목 ID, 분류, 명칭, 사양, 단가, 재고수량
user	사용자	사용자를 입력하고 삭제한다.	사용자 이름,전화번호,성별,개인정보.
cashmemore	가계부	자재나 매장용품을 입력하고 삭제하여 금액을 관리한다.	품목 명,품목 수량,지출, 입금,총액
stringbox	금고	금고의 총금액을 관리한다.	모든 총액

3.DB설계

3.3 테이블 목록

번호	테이블ID	한글명	설명	주요 속성	비고(용량,빈도)
1	USER	사용자	사용자 정보를 저장	Name, 주소	100사용자이상
2	ARTICLE	상품	품목 자재 정보입력	상품 정보, 단가	2000/상품
3	connection	재고	품목 자재 관리 저장	Name, 수량	50개 고정
4	cashmemore	가계부	매장의 입금 지출을 저장	Name, inputdate	하루에 100 품목
5	stringbox	금고	총매출을 확인 저장	Allmoney	하루 매출 2천 만 이하

3.DB설계

3.4 사용사례 관계 표

번호	사용사례명	사용 액터	설명	관련 사용사례
UC1	사용자확인	직원,사장님	현재 매출을 확인	상품추가/삭제
UC2	매출확인	직원,사장님	매장에 있는 상품을 관리한다.	재고수량파악
UC3	재고수량파악	직원,사장님	현재 매장에 있는 물품의 수량확인	재고추가 / 삭제
UC4	재고추가/삭제	직원,사장님	자재를 입력하고 삭제한다.	가계부 입력/삭제
UC5	가계부입력/삭제	직원,사장님	자재나 매장용품을 입력하고 삭제하여 금액을 관리한다.	금고 관리
UC6	금고 관리	사장님	금고의 총금액을 관리한다	

번호	테이블ID	사용 사례ID ⇒	UC01	UC02	UC03	UC04	UC05	UC06
		사용 사례명 ⇒	사용자 확인	매출 확인	재고 확인	재고 추가 삭제	가계부입력/ 삭제	금고 관리
1	USER	사용자	R	R	R	CRUD	CRUD	
2	ARTICLE	상품			CRUD			CRUD
3	connection	재고관리		RU	R	CRUD	U	
4	cashmemore	가계부관리			R	RU	CRUD	RU
5	stringbox	금고						CRUD

3.DB설계

3.5 테이블 정의서

테이블ID	USER	시스템	사용자 정보
테이블명	사용자 기본정보	업무영역	사용자 관리
작성자	최우형	사용빈도	저 / 중 / 고
생성일	2011-10-24 오전 11:28:46	예상행목수	100명

번호	컬럼ID	한글명	데이터 타입	널허용	키	FK	기본값	비고
1	USER_ID	사용자번호	int	NOT	PK1		IDENTITY(1,1)	
2	KORNAME	한글이름	varchar(50)	NOT				
3	ENGNAME	영문이름	varchar(50)					
4	BIRTHDAY	생년월일	varchar(10)					생일 (형식:1999-01-01)
5	SEX	성별	char(01)					'M' 남, 'F' 여
7	REGDATE	등록일	DATETIME	NOT			entry()	
8	Mobile	휴대폰번호	varchar(14)					
9	Email	메일주소	varchar(64)	NOT				
10	Post	우편번호	char(7)					
11	strAddr1	기본주소	varchar(64)					
12	strAddr2	번지수	varchar(64)					
14	resident_ID	주민번호	int	NOT		FK1		

인덱스 정의 (필요할 때만 정의)

번호	인덱스 명	컬럼명	Ordering	용도
1	IDX_USER1	USER_ID	1	
		KORNAME	2	
2				

테이블 및 인덱스 생성 스크립트

```

CREATE TABLE ST_MEMBER (
  USER_ID INT IDENTITY(1,1) NOT NULL, /* 사용자 테이블 고유키 */
  KORNAME VARCHAR(50) NOT NULL, /* 한글이름 */
  ENGNAME VARCHAR(50), /* 영문이름 */
  BIRTHDAY CHAR(10), /* 생일 (형식:1999-01-01) */
  SEX CHAR(01) DEFAULT 'M' NOT NULL, /* 성별 */
  REGDATE DATETIME DEFAULT GETDATE() NOT NULL, /* 등록일 */
  MOBILE INT, /* 핸드폰번호 ID */
  Email VARCHAR(50) /* E-Mail */
  post char(7) /*우편번호*/
  strAddr1 VARCHAR(70) /*기본주소*/
  strAddr2 VARCHAR(70) /*번지수*/
  resident_ID int NOT NULL /*주민번호*/
  PRIMARY KEY(USER_ID), /* 주 키 지정 */
  FOREIGN KEY(resident_ID) REFERENCES DEPARTMENT(DID) /* 외부 키 지정 */
)
  
```

3.DB설계

3.5 테이블 정의서

테이블ID	ARTICLE						시스템	제품 정보
테이블명	제품정보						업무영역	제품 관리
작성자	최우영						사용빈도	계 / 용 / 고
생성일	2011-10-24 오전 11:28:46						예상양쪽수	2000개
번호	컬럼ID	안글명	데이터 타입	널어용	케	FK	기본값	비고
1	ARTICLE_ID	제품 코드	int	NOT	PK1		product(1.1)	제품번호(01-151)
2	ARTICLE_NAME	제품 이름	varchar(50)	NOT				
3	ARTICLE_ENGNAME	제품 영어이름	varchar(50)					
4	STORE_NAME	거래처 이름	varchar(10)					
5	STORE_NUM	거래처 등록번호	char(15)	NOT		FK1		'151-74588945
7	STORE_PHONE	거래처 전화번호	int					
8	STORE_Mobile	휴대폰번호	int					
9	strAddr1	기본주소	varchar(64)					
10	strAddr2	번지수	varchar(64)					151-19
11	STORE_PART	업종	varchar(20)					ex)롯데리아

인덱스 정의 (필요할 때만 정의)

번호	인덱스명	컬럼명	Ordering	용도
1				
2				

테이블 및 인덱스 생성 스크립트

```

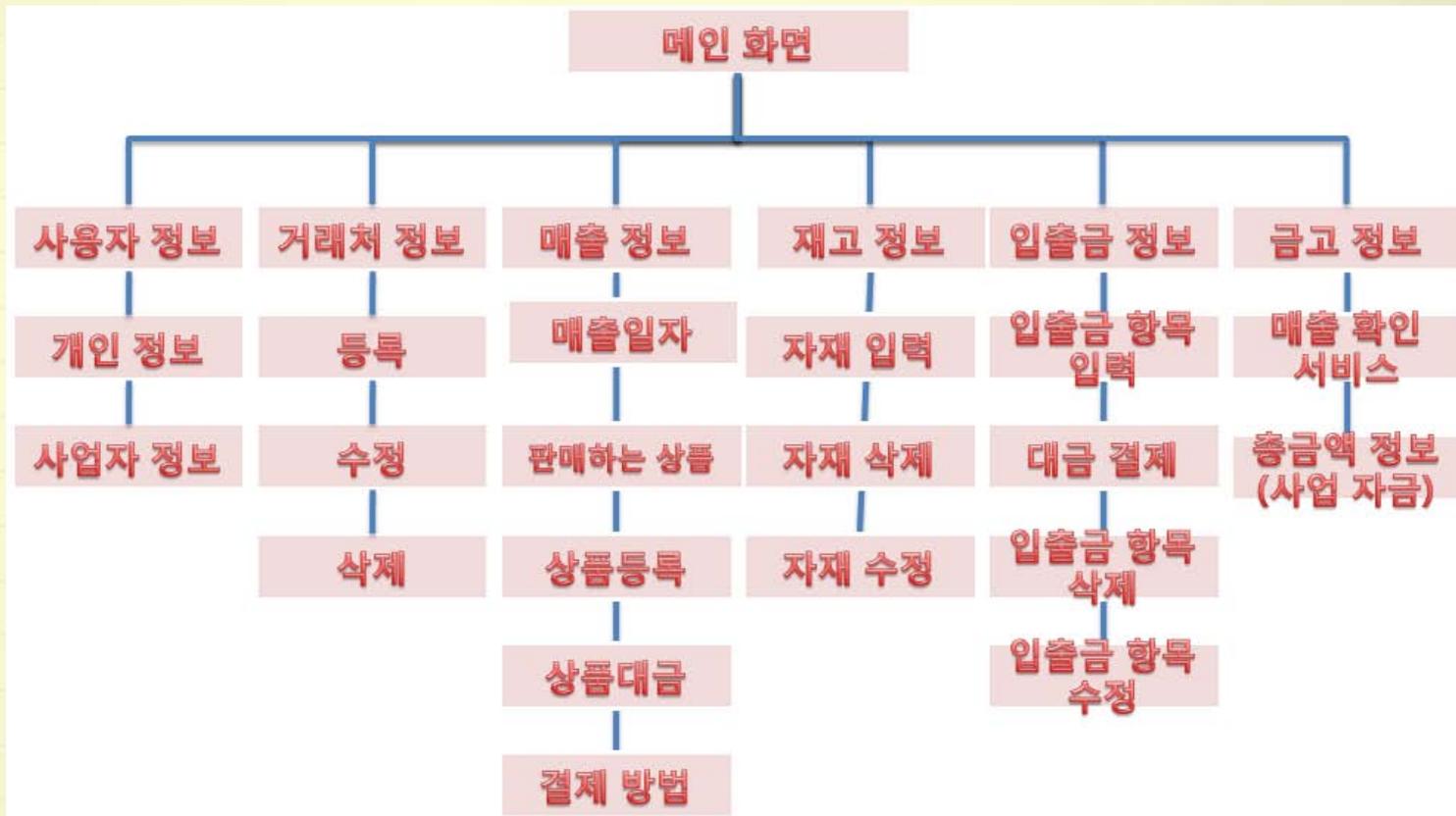
/*****
* 제품의 기본 정보
*****/
CREATE TABLE AT_MEMBER (
ARTICLE_ID INT product(1,1) NOT NULL, /* 제품 테이블 고유키 */
ARTICLE_NAME VARCHAR(50) NOT NULL, /* 제품이름 */
ARTICLE_ENGNAME VARCHAR(50), /* 제품영어이름 */
STORE_NAME CHAR(10), /* 거래처이름 */
STORE_NUM CHAR(10) DEFAULT 'M' NOT NULL, /* 거래처 번호 */
STORE_PHONE INT, /* 거래처 핸드폰번호 */
STORE_Mobile VARCHAR(50) /* 거래처 폰번호 */
STORE_Addr1 VARCHAR(70) /* 거래처 주소 */
STORE_Addr2 VARCHAR(70) /* 거래처 주소 */
STORE_PART int NOT NULL /* 업종 */
PRIMARY KEY(ARTICLE_ID), /* 주 키 지정 */
FOREIGN KEY(STORE_NUM) /* 외부 키 지정 */

```

4.UI 설계

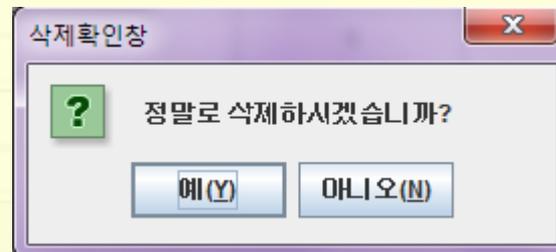
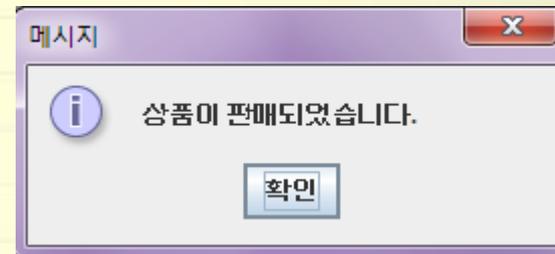
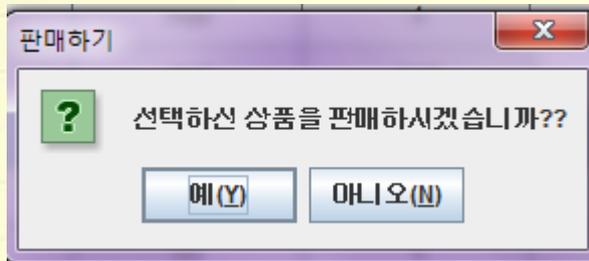
4UI 설계

4.1 메뉴 구조도



4UI 설계

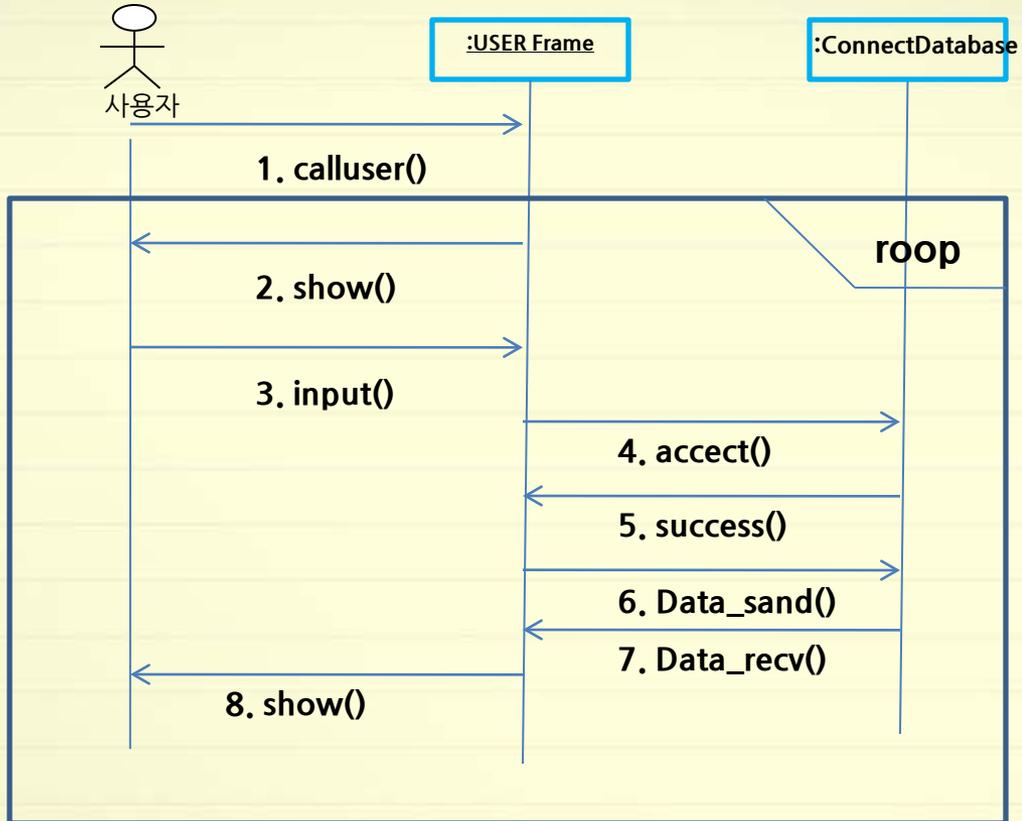
4.3 공통 인터페이스 설계



5.SSD & 클래스도

5.SSD&클래스도

5.1 SSD 시퀀스 다이어그램 ex) 사용자 정보.



5.SSD&클래스도

5.3 쪽지 보내기 동적 설계

약정 1: inputuser_set

Operation: inputuser_set()

Cross References : Use Cases :user 정보 입력 /삭제 /확인/수정.

Preconditions: 유저 정보 테이블로 확인중

Postconditions: - 한 user 의 정보가 생성.

- inputuser_set 은 Main 과 연관되었다.

- inputuser_set 가 ConnectDatabase 로 찾아진다.

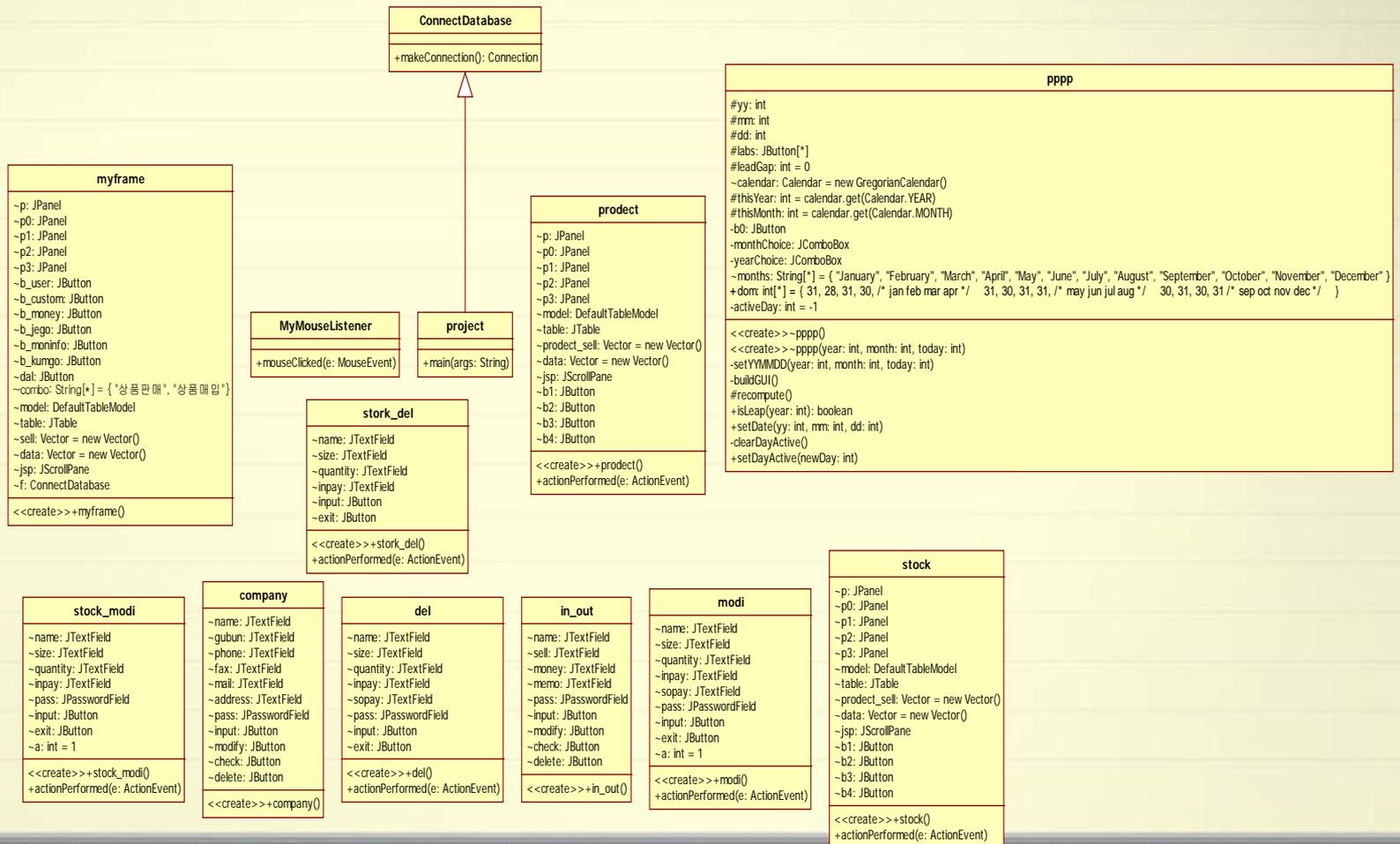
5.SSD&클래스도

5.3 user생성 동적 설계



5.SSD & 클래스도

5.4 클래스도



6 소스 코드

6.1 유저 정보 소스 코드

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.border.Border;
import javax.swing.table.*;
import java.sql.*;
class MemberFrame extends JFrame {
    JTextField name, address, tel, Email, phone, bunjisu;
    JPasswordField pass;
    JButton input, modify, check, delete;

    public MemberFrame() {
        this.setSize(300,300);
        this.setResizable(false);
        this.setTitle("직원정보관리");
        JPanel r0 = new JPanel();
        r0.setLayout(new BorderLayout(r0, BorderLayout.Y_AXIS));

        JPanel r1 = new JPanel();
        JLabel l1 = new JLabel("이름");
        name = new JTextField(15);
        r1.add(l1);
        r1.add(name);

        JPanel r2 = new JPanel();
        JLabel l2 = new JLabel("전화번호");
        tel = new JTextField(15);
        r2.add(l2);
        r2.add(tel);

        JPanel r3 = new JPanel();
        JLabel l3 = new JLabel("핸드폰 번호 ");
        phone = new JTextField(15);
        r3.add(l3);
        r3.add(phone);
```

```
JPanel r4 = new JPanel();
JLabel l4 = new JLabel("E-mail");
Email = new JTextField(15);
r4.add(l4);
r4.add(Email);
```

```
JPanel r5 = new JPanel();
JLabel l5 = new JLabel("주소");
address = new JTextField(15);
r5.add(l5);
r5.add(address);
```

```
JPanel r6 = new JPanel();
JLabel l6 = new JLabel("번지수(기타주소)");
bunjisu = new JTextField(15);
r6.add(l6);
r6.add(bunjisu);
```

```
JPanel r7 = new JPanel();
input = new JButton("입력");
input.addActionListener(new MyListener());
modify = new JButton("수정");
modify.addActionListener(new MyListener());
check = new JButton("조회");
check.addActionListener(new MyListener());
delete = new JButton("삭제");
delete.addActionListener(new MyListener());
```

```
r1.setLayout(new GridLayout(0, 2));
```

```
r1.add(input);
r1.add(modify);
r1.add(check);
r1.add(delete);
r0.add(r1);
```

```
add(r0);
this.setVisible(true);
}
```

6 소스코드

6.1 유저 정보 소스코드

```
JPanel r4 = new JPanel();  
JLabel l4 = new JLabel("E-mail");  
Email = new JTextField(15);  
r1.add(l4);  
r1.add(Email);
```

```
JPanel r5 = new JPanel();  
JLabel l5 = new JLabel("주소");  
address = new JTextField(15);  
r1.add(l5);  
r1.add(address);
```

```
JPanel r6 = new JPanel();  
JLabel l6 = new JLabel("번지수(기타주소)");  
bunjisu = new JTextField(15);  
r1.add(l6);  
r1.add(bunjisu);
```

```
JPanel r7 = new JPanel();  
input = new JButton("입력");  
input.addActionListener(new MyListener());  
modify = new JButton("수정");  
modify.addActionListener(new MyListener());  
check = new JButton("조회");  
check.addActionListener(new MyListener());  
delete = new JButton("삭제");  
delete.addActionListener(new MyListener());
```

```
r1.setLayout(new GridLayout(0, 2));
```

```
r1.add(input);  
r1.add(modify);  
r1.add(check);  
r1.add(delete);  
r0.add(r1);
```

```
add(r0);  
this.setVisible(true);  
}
```

6 소스코드

6.1 유저 정보 소스코드

```
public class checkFrame extends JFrame {
```

```
    Vector member = new Vector();
```

```
    Vector data = new Vector();
```

```
    JScrollPane jsp;
```

```
    JPanel p0,p1;
```

```
    JTable table;
```

```
    DefaultTableModel model;
```

```
    public checkFrame() {
```

```
        this.setSize(700,500);
```

```
        this.setResizable(false);
```

```
        this.setTitle("직원정보조회");
```

```
        p0 = new JPanel();
```

```
        p1 = new JPanel();
```

```
        member.addElement("이름");
```

```
        member.addElement("전화번호");
```

```
        member.addElement("핸드폰");
```

```
        member.addElement("E-mail");
```

```
        member.addElement("address");
```

```
        member.addElement("bunjisu");
```

```
        model=new DefaultTableModel(data,member);
```

```
        table=new JTable(model);
```

```
        table.addMouseListener(new MyMouseListener());
```

```
        table.setRowHeight(30);
```

```
        table.setAutoCreateRowSorter(true);
```

```
        table.setPreferredScrollableViewportSize(new Dimension(680, 430));
```

```
        jsp = new JScrollPane(table);
```

```
        TableColumnModel tm;
```

```
        tm = table.getColumnModel();
```

```
        DefaultTableCellRenderer dtcr = new DefaultTableCellRenderer();
```

```
        dtcr.setHorizontalAlignment(SwingConstants.CENTER);
```

```
        TableColumnModel tcm = table.getColumnModel();
```

```
        for(int i = 0 ; i < tcm.getColumnCount() ; i++){
```

```
            tcm.getColumn(i).setCellRenderer(dtcr);
```

```
        }
```

```
        p1.add(jsp);
```

```
        p0.add(p1);
```

```
        add(p0);
```

```
        this.setVisible(true);
```

```
    }
```

```
    private class MyMouseListener extends MouseAdapter{
```

```
        public void mouseClicked(MouseEvent e){ // 마우스클릭이벤트발생시
```

```
            JTable table = (JTable)e.getSource();
```

```
            int row = table.getSelectedRow();
```

```
            int a = JOptionPane.showConfirmDialog(table, "선택하신 직원정보를 불러올까요?", "불러오기",JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE);
```

```
            if(a == JOptionPane.YES_OPTION){
```

```
                String names = table.getValueAt(row, 0).toString();
```

```
                String tels= table.getValueAt(row, 1).toString();
```

```
                String phones = table.getValueAt(row, 2).toString();
```

```
                String Emails = table.getValueAt(row,3).toString();
```

```
                String addresses = table.getValueAt(row,4).toString();
```

```
                String bunjisu = table.getValueAt(row,5).toString();
```

```
                name.setText(names);
```

```
                tel.setText(tels);
```

```
                phone.setText(phones);
```

```
                Email.setText(Emails);
```

```
                address.setText(addresses);
```

```
                bunjisu.setText(bunjisu);
```

```
            }
```

```
        }
```

```
    }  
  
    public class MyListener implements ActionListener {
```

```
        public void actionPerformed(ActionEvent k){
```

```
            if(input == k.getSource()) {
```

```
                ConnectDatabase c = new ConnectDatabase();
```

```
                Connection con = c.makeConnection();
```

```
                String query = "select *from user1";
```

```
                try {
```

```
                    PreparedStatement stmt = con.prepareStatement(("insert into user1 " +  
                    "(name, tel, phone, Email, address, bunjisu) " + "values(?,?,?,?,?)"));
```

```
                    stmt.setString(1,name.getText());
```

```
                    stmt.setString(2,tel.getText());
```

```
                    stmt.setString(3,phone.getText());
```

```
                    stmt.setString(4,Email.getText());
```

```
                    stmt.setString(5,address.getText());
```

```
                    stmt.setString(6,bunjisu.getText());
```

```
                    stmt.executeUpdate();
```

6 소스코드

6.1 유저 정보 소스코드

```
name.setText("");
tel.setText("");
phone.setText("");
Email.setText("");
address.setText("");
bunjisu.setText("");
```

```
name.requestFocus();
}catch(SQLException e) {
System.err.println("error sql = " +e);
}
}
```

```
if(modify == k.getSource()) {
ConnectDatabase c = new ConnectDatabase();
Connection con = c.makeConnection();
String query = "select *from user1";
try {
PreparedStatement stmt = con.prepareStatement("UPDATE user1 SET tel
= ?,phone = ?,Email = ?,address =? ,bunjisu =? WHERE name LIKE ?");
```

```
stmt.setString(1,name.getText());
stmt.setString(2,tel.getText());
stmt.setString(3,phone.getText());
stmt.setString(4,Email.getText());
stmt.setString(5,address.getText());
stmt.setString(6,bunjisu.getText());
stmt.executeUpdate();
name.setText("");
tel.setText("");
phone.setText("");
Email.setText("");
address.setText("");
bunjisu.setText("");
name.requestFocus();
```

```
}catch(SQLException e) {
System.err.println("error sql = " +e);
}
}
```

```
if(check == k.getSource()) {
ConnectDatabase c = new ConnectDatabase();
checkFrame c1 = new checkFrame();
Connection con = c.makeConnection();
String query = "select *from user1";
try {
Statement ptmt = con.createStatement();
ResultSet rs = ptmt.executeQuery(query);
while(rs.next()) {
Vector Low = new Vector();
Low.addElement(rs.getString("name"));
Low.addElement(rs.getString("tel"));
Low.addElement(rs.getString("phone"));
Low.addElement(rs.getString("Email"));
Low.addElement(rs.getString("address"));
Low.addElement(rs.getString("bunjisu"));
c1_data.add(Low);
}
}
```

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

```
if(delete == k.getSource()) {
ConnectDatabase c = new ConnectDatabase();
Connection con = c.makeConnection();
String query = "select *from user1";
try {
int d = JOptionPane.showConfirmDialog(null, "정말로 삭제하시겠습니까?", "삭제확인창",
JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE);
if(d == JOptionPane.YES_OPTION){
PreparedStatement stmt = con.prepareStatement("delete from user1 where name = ?");
stmt.setString(1,name.getText());
stmt.executeUpdate();
name.setText("");
tel.setText("");
phone.setText("");
Email.setText("");
address.setText("");
bunjisu.setText("");
name.requestFocus();
}
}
```

```
}catch(SQLException e) {
System.err.println("error sql = " +e);
}
}
}
}
}
```

6 소스 코드 & 구동 화면

src/user_tables.java - Eclipse

Source Refactor Navigate Search Project Run Window Help

project.java user_tables.java product.java stock.java stork_del.java

직원정보관리

이름
전화번호
핸드폰 번호
E-mail
주소
번지수(기타주소)

업체명
업체구분
업체번호
팩스번호
E-mail
주소

입력 수정
조회 삭제

매출 재고 관리 프로그램

사용자 정보 거래처 정보 상품정보 재고 정보 입출금정보 금고정보

달력 상품판매

상품 코드	상품 명	상품 규격	상품 수량	입고가	매입가
31	1	1	21312	3	21321
32	bulgogl	max	4	300	5000
33	as	asd	sad	as	dasdas
34	1	11	1321	312	123123123
35	123	213	213	213	123
36	1	11	1	11	1
37	1	1	1	11	1

31	1	1	21312	3	21321
32	bulgogl	max	4	300	5000
33	as	asd	sad	as	dasdas
34	1	11	1321	312	123123123
35	123	213	213	213	123
36	1	11	1	11	1
37	1	1	1	11	1

상품추가 상품삭제 상품수정

재고 정보 뷰

품목 코드	품목 명	품목 규격	품목 수량	입고가
3	1	1	1	1

입출금 정보

품목이름
입출구분
금액
메모

입력 수정
조회 삭제

Calendar

December 2011

S	M	T	W	R	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

7느낌

**이번 소프트웨어 공학 개발방법론 수업에서 많은 것
을 배운 거 같습니다.**

**이것 저것 배우다 보니 벌써 요번학기가 끝나가는데요.
이번학기는 정말 힘들었습니다. 과제도 많고 제실력에
프로젝트 까지 하기는 많이 힘들었지만 차근차근 자바
를 공부 해서 많은 것을 알고 그리고 소프트웨어 쪽으
로 많이 얻어갈 수 있어서 매우 좋았습니다.**

**방학동안에는 저자신의 실력을 좀 쌓으려고 합니다 방
학때 열심히 해서 발전된 모습으로 돌아 오겠습니다.
한학기 동안 고생하셨습니다 송해상 교수님.**

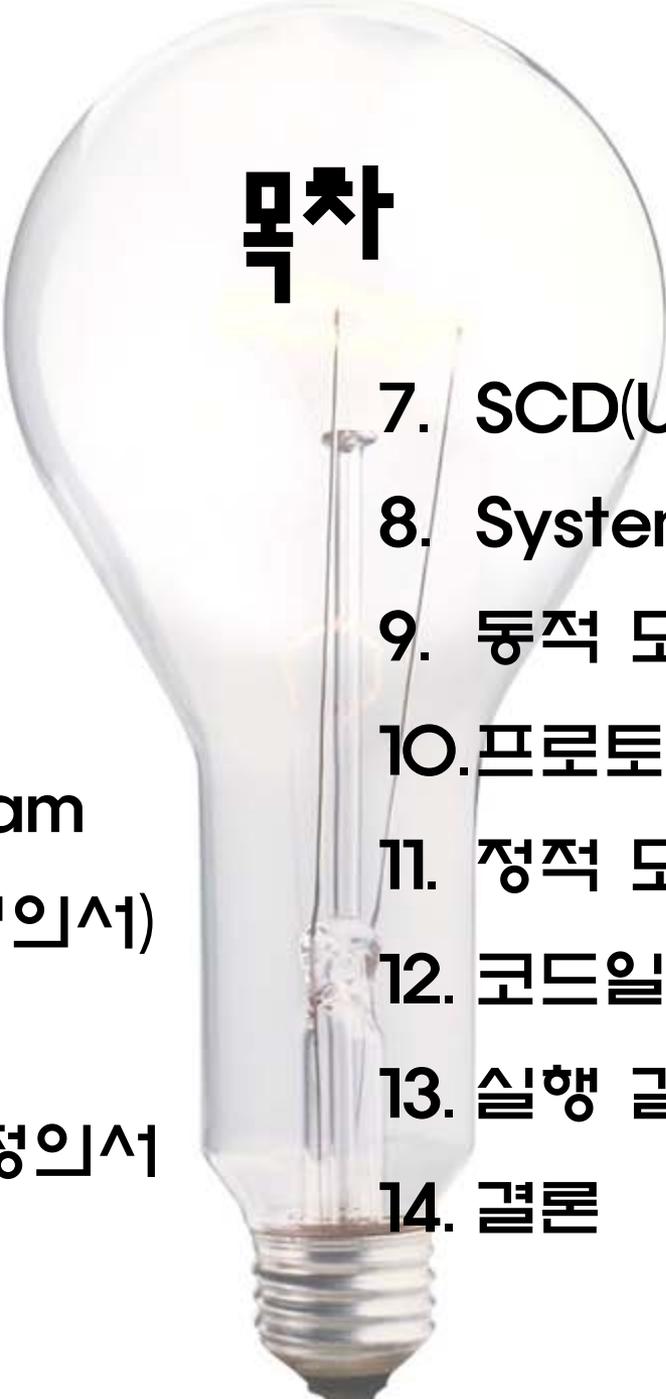
감사합니다!



Twiteon Messenger

(SNS기반 신개념 메신저)

-프로젝트 결과 및 구현-



목차

1. 개요

1. 목표

2. 범위

3. 주요기능

2. UC-Diagram

3. UC-text(정의서)

4. RCD

5. DB Table정의서

6. UI 설계도

7. SCD(UC:트릿글쓰기)

8. SystemOperation co

9. 동적 모델링

10. 프로토콜 모델링

11. 정적 모델링

12. 코드일부

13. 실행 결과

14. 결론

프로젝트 목표

- Twitter의 **트윗 기능을 도입**하여 웹 2.0 시대를 반영한 차별화된 메신저 개발
- 현재 상용화 메신저들의 필요 이상의 기능을 축소하여 보다 **컴팩트한 메신저**로 개발 (최소성)
- 광고 창을 줄이고 메신저의 **디자인을 사용자 중심으로** 개선 (편리성, 디자인)
- 어느 운영체제에서도 사용 가능토록 개발 (호환성)

프로젝트의 범위

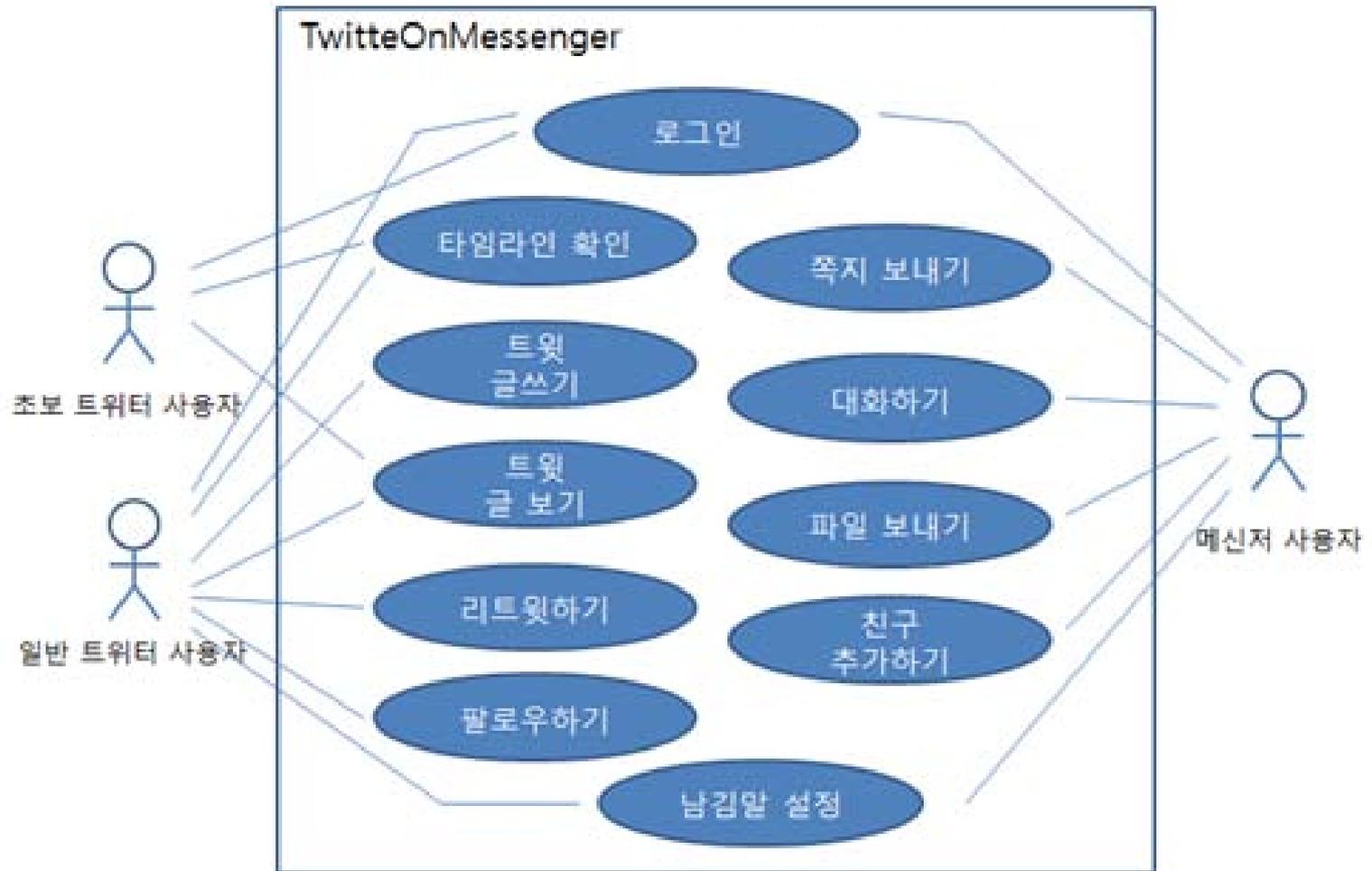
- 일반적인 쪽지, 대화, 파일전송 기능이 있는 메신저를 구현
- 메신저를 트위터와 Open API를 사용하여 연동
- 메신저 속 트윗 게시판을 구현
- 게시 글 작성 시 자동 트윗 기능을 구현
- 서버 측에 메신저 데이터베이스뿐 아니라 트위터 상의 데이터베이스를 읽어와 사용자 관련 트윗 데이터베이스를 기록 관리
- 메신저에 친숙한 인터페이스를 도입한다.

주요기능

기능	내용
트윗 기능	메신저를 통하여 트위터에 글을 올리는 기능
트윗 게시판	트위터의 글들을 메신저의 게시판에서 확인가능하며, 트윗 검색 및 확인 가능
사용자간 쪽지	로그인한 사용자가 친구 에게 text형식의 메시지를 보내는 기능
1:1 대화	로그인한 사용자가 타 로그인한 사용자와 1:1로 대화 가능
다자간 대화	3인 이상이 한 대화 창에서 대화 가능
(미구현)파일전송	로그인한 사용자가 타 로그인한 사용자에게 파일 전송 가능

이판
메신저와
유사

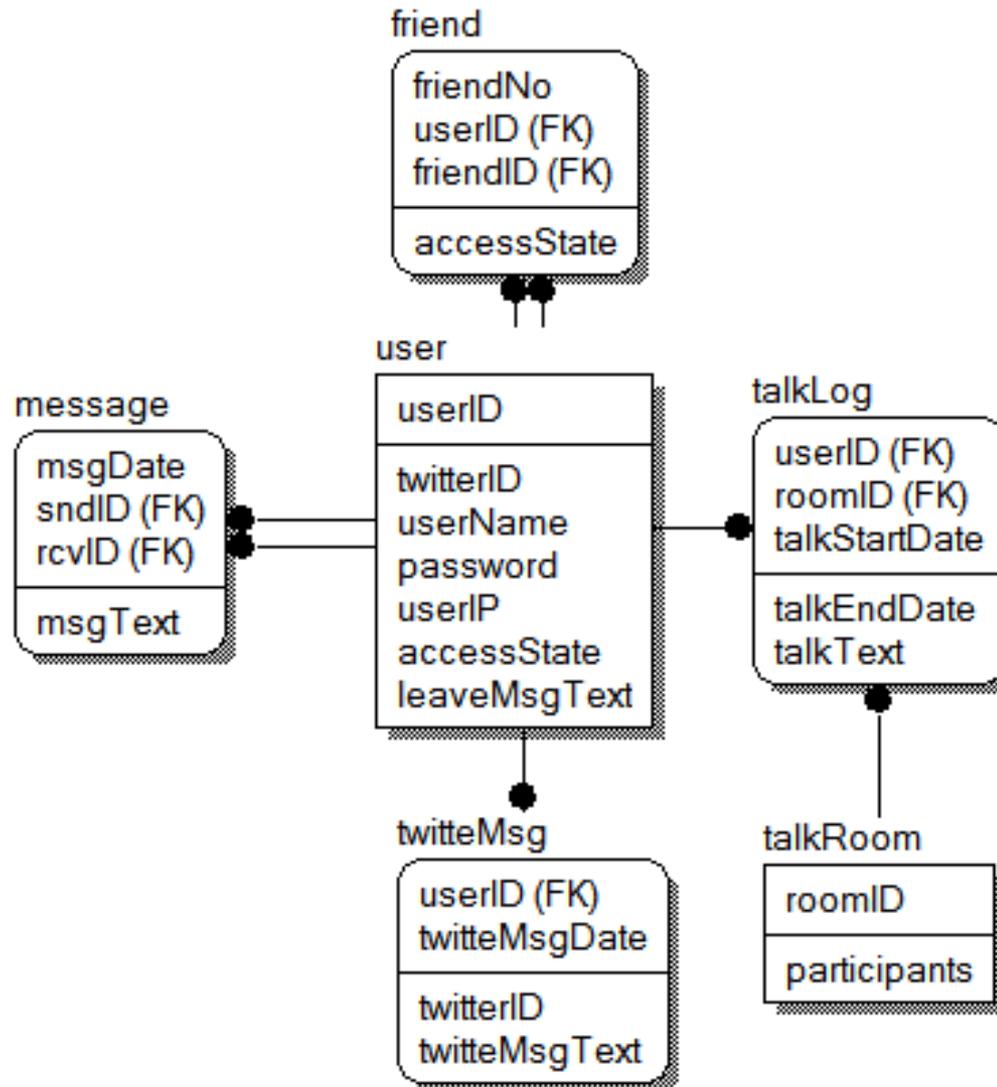
UC-Diagram



UC-정의서

사용사례	UC2. 타임라인 확인
액터	트위터 사용자
시작조건	트위터 사용자가 트윗 게시판을 선택
종료조건	사용자가 타임라인을 확인
주요흐름	<ol style="list-style-type: none"> 1. 사용자가 트윗 게시판 메뉴를 선택한다. 2. 시스템은 트윗 게시판 윈도우를 생성한다. 3. 트윗 게시판 윈도우는 기본으로 최근 타임라인을 게시한다. <<E1: 게시글이 적을때>> <ol style="list-style-type: none"> 3.1 게시글을 자세한 내용을 보고싶을 때에는 <<UC3. 트윗 글읽기>>를 실행한다. 3.2 트위터에 트윗(글쓰기)을 올리고 싶을 때에는 <<UC4. 트윗 글쓰기>>를 실행한다.
대안흐름	<p>E1 게시글이 적을때</p> <ol style="list-style-type: none"> 1. 타임라인 게시판 기본 공간보다 게시글의 수가 부족한 부분은 공백으로 나타난다.
비기능요구사항	<ol style="list-style-type: none"> 1. 게시판 로딩 응답시간이 3초 이내여야 한다.

ACD



DB 테이블 정의서

테이블ID	TWITEMSG						시스템	트위트온메신저
테이블명	트윗 글 정보						업무영역	트윗 글 관리
작성자	석준희						사용빈도	저 / 중 / (고)
생성일	2011-10-25 오후 18:28:46						예상항목수	5만 개
번호	컬럼ID	한글명	데이터 타입	널허용	키	FK	기본값	비고
1	userID	사용자아이디	varchar(50)	NOT		FK1		
2	twitteMsgDate	트윗 날짜	varchar(30)	NOT	PK1			
3	twitteMsgText	트윗 내용	TEXT	NOT				
4	twitterID	트위터아이디	varchar(50)	NOT		FK2		
5	reTwitteCount	리트윗 수	int				'0'	
인덱스 정의 (필요할 때만 정의)								
번호	인덱스 명	컬럼명	Ordering	용도				
1								
테이블 및 인덱스 생성 스크립트								
<pre> /***** * 사용자 기본 정보 *****/ CREATE TABLE TWITEMSG (userID VARCHAR(50) NOT NULL, /* 사용자 아이디 */ twitteMsgDate VARCHAR(30) NOT NULL, /* 트윗 날짜(시간) */ twitteMsgText TEXT NOT NULL, /* 트윗 글 내용 */ twitterID VARCHAR(50) NOT NULL, /* 트위터 아이디 */ reTwitteCount int, /* 리트윗 수 */ PRIMARY KEY(userID), /* 주 키 지정 */ FOREIGN KEY(userID) REFERENCES USER(userID), /* 외부 키 지정 */ FOREIGN KEY(twitterID) REFERENCES USER(twitterID), /* 외부 키 지정 */) </pre>								

미션 계획도

트윗 게시판

1 내가쓴글보기 2 내 글쓰기 3 리플레시

4 시간	사용자	내용
10:30	Junhee	Hi my friends~

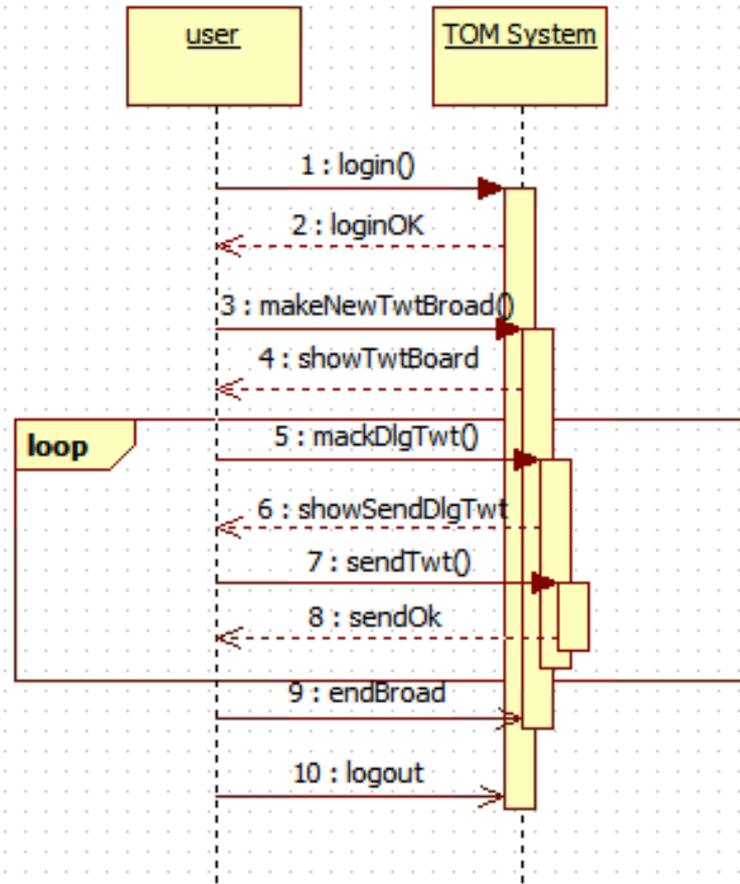
5 더보기 6 트윗 7 8 검색

Seowon University	
시스템명	TwitteOnMessenger
업 무 명	트위터 게시판
작 성 일	2011-10-25
작 성 자	석준희
화 면 명	트윗 게시판 화면
화 면 ID	TwitteBoard

□ DESCRIPTION □

1	자신의 글만 검색하여 보여줌
2	트윗 글쓰기 창이 실행됨
3	게시판을 리플레시하여 세로고침해줌
4	트위터의 글을 게시판형식으로 사용자에게 보여줌
5	한 페이지에 트윗 글들을 20개 더 출력
6	검색 설정 중 트윗/사용자/사진 카테고리 설정
7	검색어 입력 란
8	검색 버튼
9	
10	
11	

SCD(유스케이스: 트위터 글쓰기)



CO(시스템 약정)

약정 C01: makeNewTwtBroad

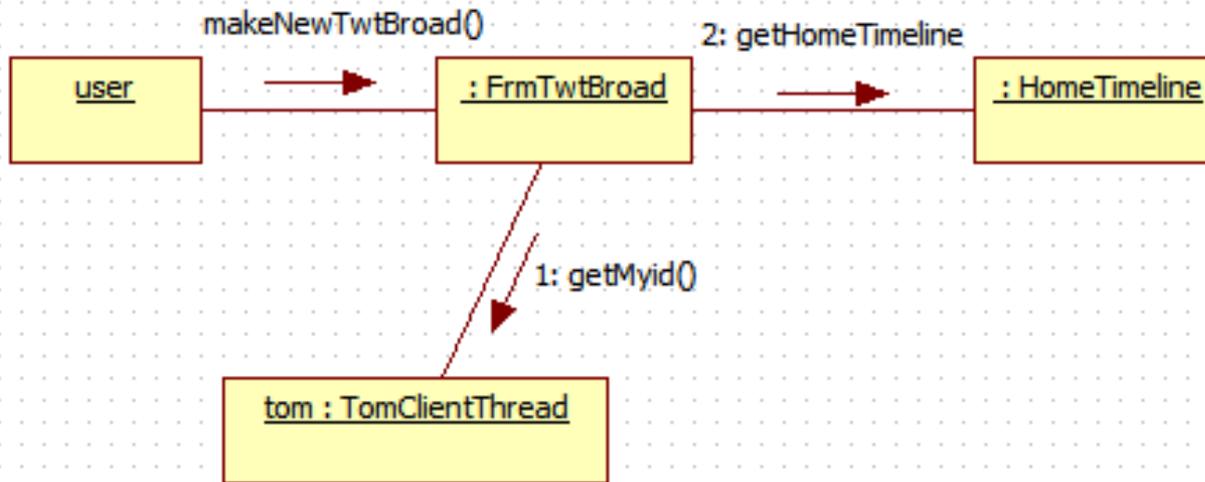
오퍼레이션	makeNewTwtBroad()
상호 참조	유스케이스 : 타임라인 확인
사전조건	로그인 완료후 대기상태이다.
사후조건	FrmTwBroad ftb가 생성됨 ftb는 TomClientThread와 연관 ftb의 테이블을 getHomeTimeline으로 초기화하여 트위터 메시지가 게시판에 출력된다.

CO(시스템 약정)

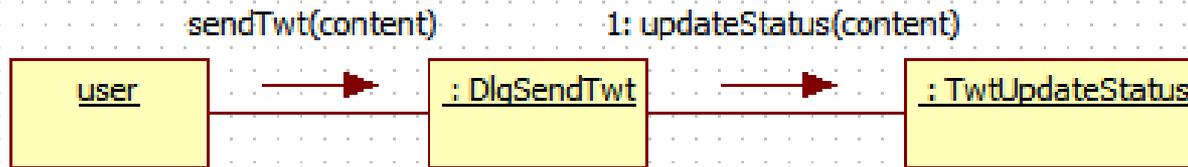
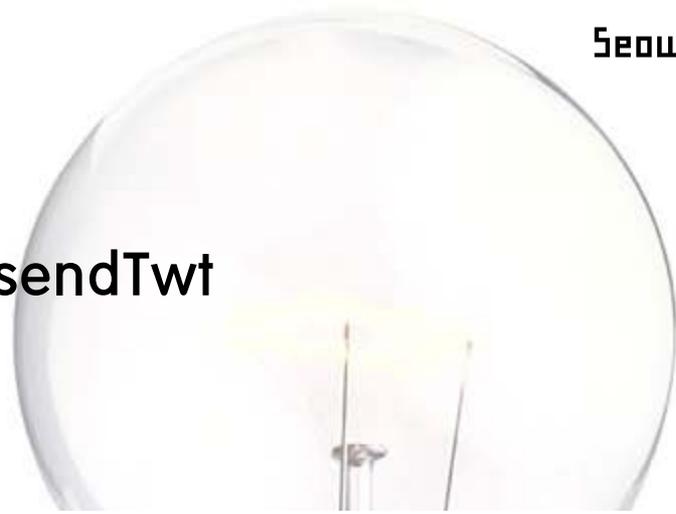
약정 CO2: sendTwt	
오퍼레이션	sendTwt(String content)
상호 참조	유스케이스 : 트윗 글쓰기
사전조건	DlgSendTwt가 생성된 상태
사후조건	updateStatus가 생성되며 매게로 content가 넘어가게된다. DlgSendTwt은 소멸한다.

동적모델링

makeNewTwtBroad

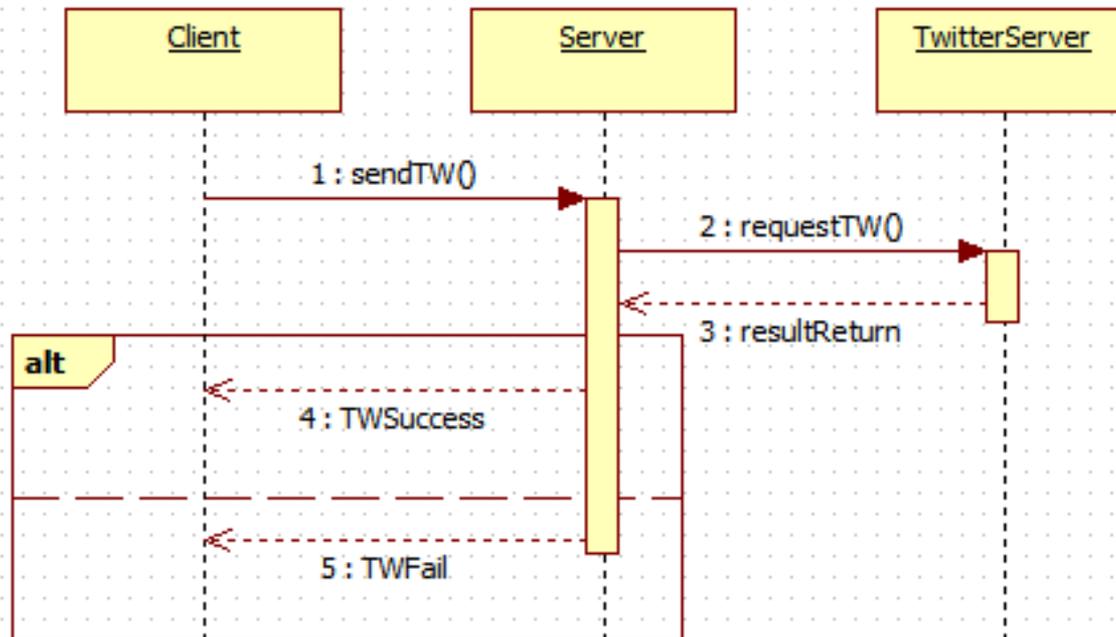


동적모델링 sendTwt



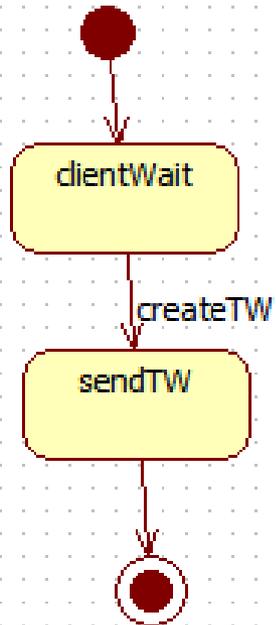
프로토콜모델링

sendTwt

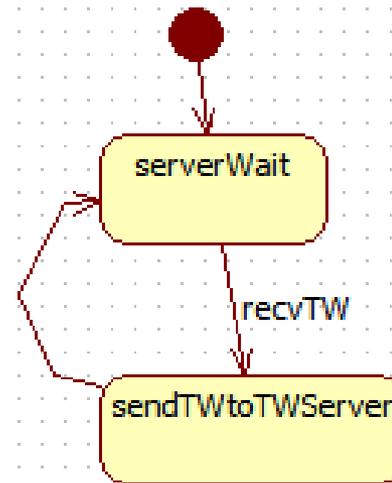


프로토콜모델링 sendTwt

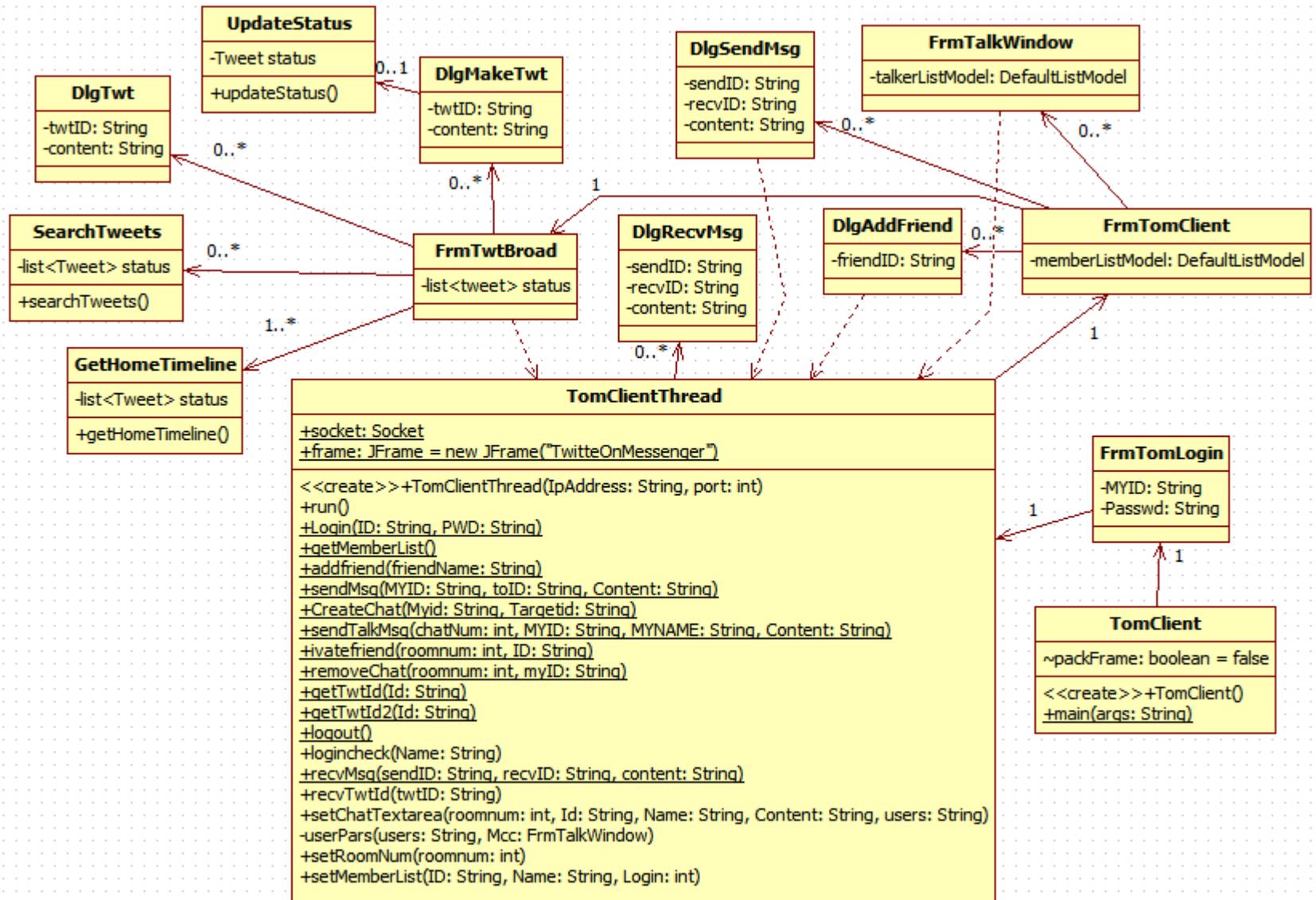
클라이언트 측



서버 측



정적 모델링



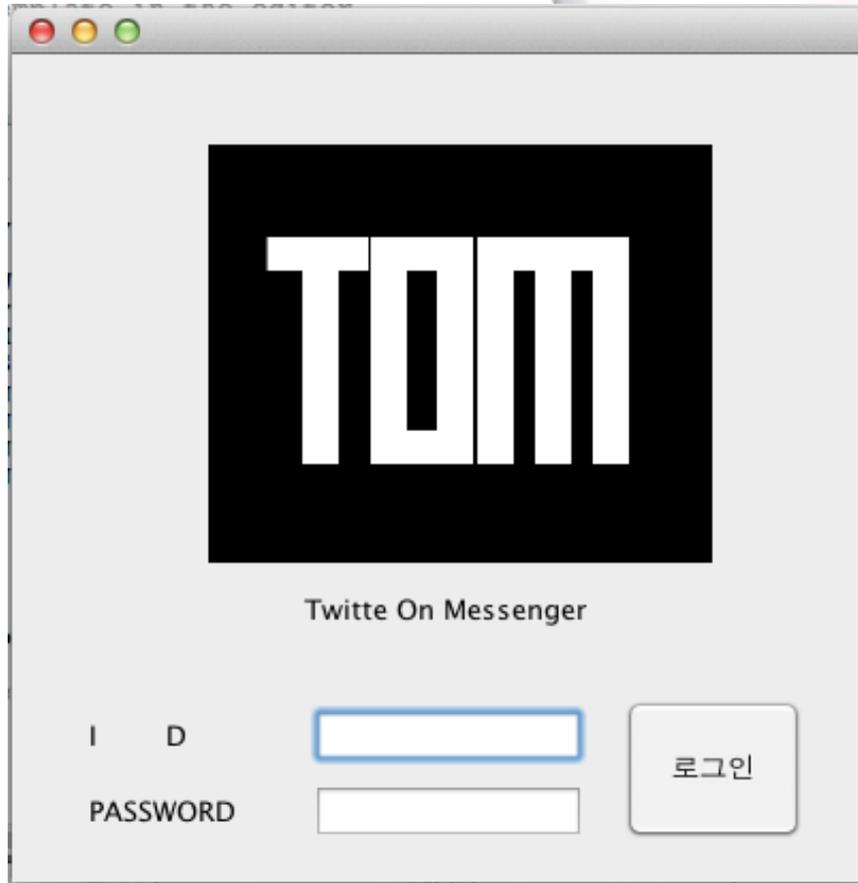
코드 일부

```
public class FrmTwtBroad extends javax.swing.JFrame {  
  
    /** Creates new form FrmTwtBroad */  
    DefaultTableModel model;  
    public String MYID = TomClientThread.MYID;  
    public static String MyTwtId;  
  
    public FrmTwtBroad() {  
        initComponents();  
        model = (DefaultTableModel) jTable1.getModel();  
        jblnit();  
  
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
    }  
}
```

```
private void jblnit() {  
    new GetHomeTimeline(model);  
    this.show();  
    TomClientThread.getTwtId(MYID);  
}
```

본 소스는
FrmTwtBroad 클래스의
생성자 부분이며 게시판이
생성되며
GetHomeTimeline 을
생성해 주고 Jtable의
model을 연관시켜
트위터의 글들을 받아
테이블에 출력하게 해준다.

실행 결과



실행 결과

트윗 게시판 내가쓴글 트윗하기 리프레시

날짜	작성자	내용
1:30 AM Dec 8th	douglasguen	앗 이런거 있었으면
12:57 AM Dec 8th	douglasguen	사업계획의 피드백을
12:36 AM Dec 8th	douglasguen	그림 샀다! #fb
12:33 AM Dec 8th	xguru	#H3 준비단계부터
12:28 AM Dec 8th	GeorgeReese	George will perform
12:27 AM Dec 8th	josanku	하나 하나가 중요하
12:17 AM Dec 8th	abbielundberg	working on 2 new
12:16 AM Dec 8th	josanku	참 자연은 아름답군
12:10 AM Dec 8th	Scobleizer	RT @solete: Looking
11:55 PM Dec 7th	josanku	RT @YoungSeKim:
11:39 PM Dec 7th	douglasguen	한국시간도 아니고
11:02 PM Dec 7th	Iunhee Seok	@seminhi 세민이에

날짜 : 12:33 AM Dec 8th
 작성자 : xguru

#H3 준비단계부터 검색된 모든 트윗 2300여개를 DB에 저장해두었는데, 이걸 이쁘게 정리한후 PDF로 만들었더니 무려 25MB,92페이지가 나오네요. 너무 행복합니다. 관심 가져주신 모든 분들 고맙습니다!

트윗 리트윗(RT)

결론

- 자바 프로그래밍과 Twitter의 OpenAPI를 활용하여 네트워크프로그램(메신저)를 만들었으며, window및 Mac에서도 호환이 가능하다.
- 잘한점은 처음부터 끝까지 소프트웨어 개발방법론에 따라 차근차근 만들었으며 그에따른 부가적 산출물이 있는것이며
- 못한점으로는 MVC모델을 따르려 노력했으나 미숙한 부분이 많아 2티어로 된경우가 종종있으며, 프로토콜을 최적화하지 못한부분이 있다.
- 배운점으로는 Java,network,openAPI,등 많은 프로그래밍 기법에대하여 배웠으며 소프트웨어 개발 방법론의 필요성을 확실히 깨닫게 되었다.
- 향후계획으로는 미숙한 MVC모델의 수정과, 부족한 프로토콜 안정화부분, 몇몇 트위터 기능 추가 및 디자인등을 손볼 계획에 있다.



온라인 의류 쇼핑몰 프로젝트 결과보고서 v1.0



1. 과제개요

(1) 과제의 필요성(배경)	4
(2) 과제의 목적	4
(3) 과제의 범위	5
(4) 사용자 및 주요 관심사항	6
(5) 주요기능	6
(6) 제약사항	7
(7) 타당성 분석(기술적, 경제적, 조직적)	7
(8) SWOT 또는 벤치마크 분석	8
(9) 기대효과	8

2. UCD

9

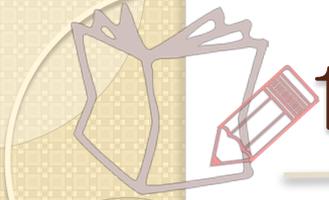
3. UC(text작성)

10

4. DB설계서

(1) E-R다이어그램	11
(2) 테이블정의서	12

5. S.S.D	13
6. S.O에 대한 약정	14
7. S.O – S.D/C.O	15
8. 전체클래스도	16
9. 코드 일부	17
10. 실행결과 UI	19
11. 결론(요약, 배운 것,향후 계획)	20



1. 과제개요

(1) 과제의 필요성(배경)

⇒ 학교 졸업하기 전에 프로젝트를 하게 되면 온라인 쇼핑몰을 제 손으로 직접 만들어 선물하기로 약속했었는데, 그 약속을 지키고 싶어서 프로젝트로 선정하게 되었습니다.

⇒ 예전에 비해 온라인 쇼핑몰들이 급 증가하면서 사용자들이 많아지고, 굳이 옷을 구매하러 밖에 나가지 않아도 손쉽게 옷을 구매할 수 있음으로 인해 온라인 쇼핑몰은 꾸준한 인기를 가지고 있다. 밖에서 본 옷들조차 온라인에서는 원가보다 저렴한 가격으로 구매할 수 있다는 것으로 다양한 연령층에서 많이 이용되는데, 다른 인터넷 쇼핑몰에 없는 기능을 추가함으로써 좀 더 사용자들에게 편리성을 제공할 필요가 있다고 생각된다.

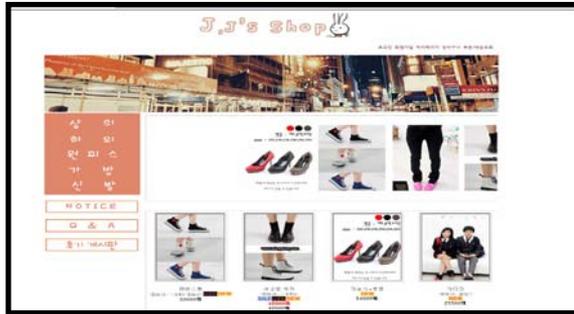
(2) 과제의 목적

⇒ 남의 힘을 빌리지 않고 **스스로** 포기하지 않고 **프로젝트를 완성시키는 것**과, 최대한 완성도 높은 쇼핑몰을 만들어 실제로 이용 가능하게 만드는 것이 목적입니다.

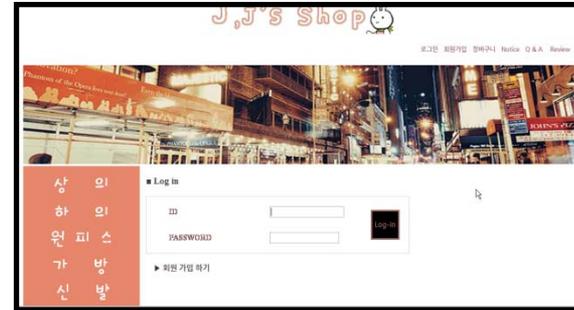
⇒ 사용자들의 불편함을 최소화하고 만족감을 최상으로 하는 쇼핑몰을 만드는 것입니다.

(3) 과제의 범위

<그림1> 화면구성도



메인 화면 ▲



로그인 화면 ▲



회원가입 화면 ▲



장바구니 화면 ▲



Q&A 화면 ▲



상품 화면 ▲

(4) 사용자 및 주요 관심사항

<표1> 사용자 및 주요 관심사항

사용자	주요관심사	사용할 주요기능
관리자	쇼핑몰 관리	상품등록·관리
회원	의류 쇼핑	구매
비회원	의류 쇼핑	구매

(5) 주요기능

<표2> 주요 기능

기능명	기능설명	주요 사용자	사용빈도 (고,중,저)
로그인	사용자 데이터베이스와 확인 후 일치할 경우 접속 확인, 사용자가 접속을 끄고자할 때 로그아웃	회원이입자, 관리자	중
상품등록	관리자는 해당 카테고리에 새로운 상품을 양식에 맞춰 입력 후, 입력된 데이터를 상품테이블 데이터베이스에 저장	관리자	고
상품목록	사용자는 각각의 카테고리를 통해 원하는 상품을 볼 수 있다.	회원 비회원	고
상품 주문	사용자가 원하는 상품을 주문하고자 할 때, 해당 상품에 대한 정보를 사용자에게 보여주고, 사용자는 결제양식에 맞추어 입력한 후, 확인버튼을 누름	회원 비회원	고
장바구니	사용자는 마음에 드는 상품을 선택하고, 선택한 상품은 장바구니 테이블에 저장됨	회원	고
상품조회	사용자가 원하는 상품의 키워드를 입력하면, 키워드와 연관된 상품을 데이터베이스로부터 읽어와 사용자에게 보여줌	회원 비회원	중
공지사항	sale하는 품목이 있거나 회원들에게 알려야 할 사항이 있으면 해당페이지에 글을 올림으로써 회원들은 읽고 답할 수 있다. 비회원은 읽기만 가능하고 댓글은 달 수 없다.	관리자 회원 비회원	하
Q & A	사용자들의 불만과 불편한 점을 조금이나마 개선하기 위해 사용자와 관리자는 해당페이지에 글을 올릴 수 있다. 반면 비회원은 읽기는 가능하나, 글쓰기와 댓글은 달 수 없다.	관리자 회원 비회원	중
주문관리	관리자는 사용자가 구매한 주문을 확인	관리자	고

(6) 제약사항

⇒ 결제기능의 경우 실제 계좌와 연결해야 할 뿐만 아니라, 택배회사와도 사전 제휴를 맺어야 하는데 아직 실제로 사용할 것이 아니기 때문에 제외하였음

(7) 타당성 분석(기술적, 경제적, 조직적)

가. 기술적 타당성

⇒ 팀원 모두 응용 분야에 **미숙**하여, JSP와 JavaScript 등의 언어를 배우는데 있어, 기술적으로 부족 한 점이 많아 서버구축이나, 언어에 대한 것들을 **공부해야 한다.**

나. 경제적 타당성

<표3> 비용수익 분석표

	2011년	2012년	2013년	총합
수익				
매출증가		10,000,000	80,000,000	90,000,000
합계		10,000,000	80,000,000	90,000,000
비용				
개발 비용	3,200,500			3,200,500
- 프린터	500			500
- 소프트웨어	200,000			200,000
- 개발 인건비	3,000,000			3,000,000
운영 비용		100,000	60,099	160,099
- 호스팅 비용		100,000	100,000	110,000
- 운영인건비		6,000,000	7,000,000	13,000,000
- 도메인 비용		30,000	30,000	60,000
합계	3,200,500	60,099	60,099	120,198
수익-비용	3,200,500			

다. 조직적 타당성

⇒ 인원이 2명이다 보니 역할 분담이나 의견에 있어서 어긋날 수 있는데, 이를 극복하기 위해 **공평한 역할 분담**과 서로의 의견을 제시함으로써 어떤 것이 좀 더 낫은 지에 대해 서로 충분한 상의를 하며, 트러블이 나지 않도록 이기적이지 않게 행동하지 않고 **서로의 의견과 역할을 존중**해주어야 할 것이다.

(8) SWOT 또는 벤치마크 분석

<표4> SWOT 분석표

Strength(강점)	Weakness(약점)
-다른 쇼핑몰과는 차별화된 기능 -심플한 디자인	-약한 보안체제 -인력 부족
Opportunity(기회)	Threat(위협)
-다른 쇼핑몰에는 없는 기능 구현 -온라인 쇼핑몰의 상승세	-온라인 시장의 경쟁 -계속 증가되는 쇼핑몰의 상승

(9) 기대효과

⇒ 일반 쇼핑몰과는 다른 기능으로 사용자의 편리함을 제공한다.

⇒ 기존의 쇼핑몰 틀에서 벗어나 새로운 변화의 시작

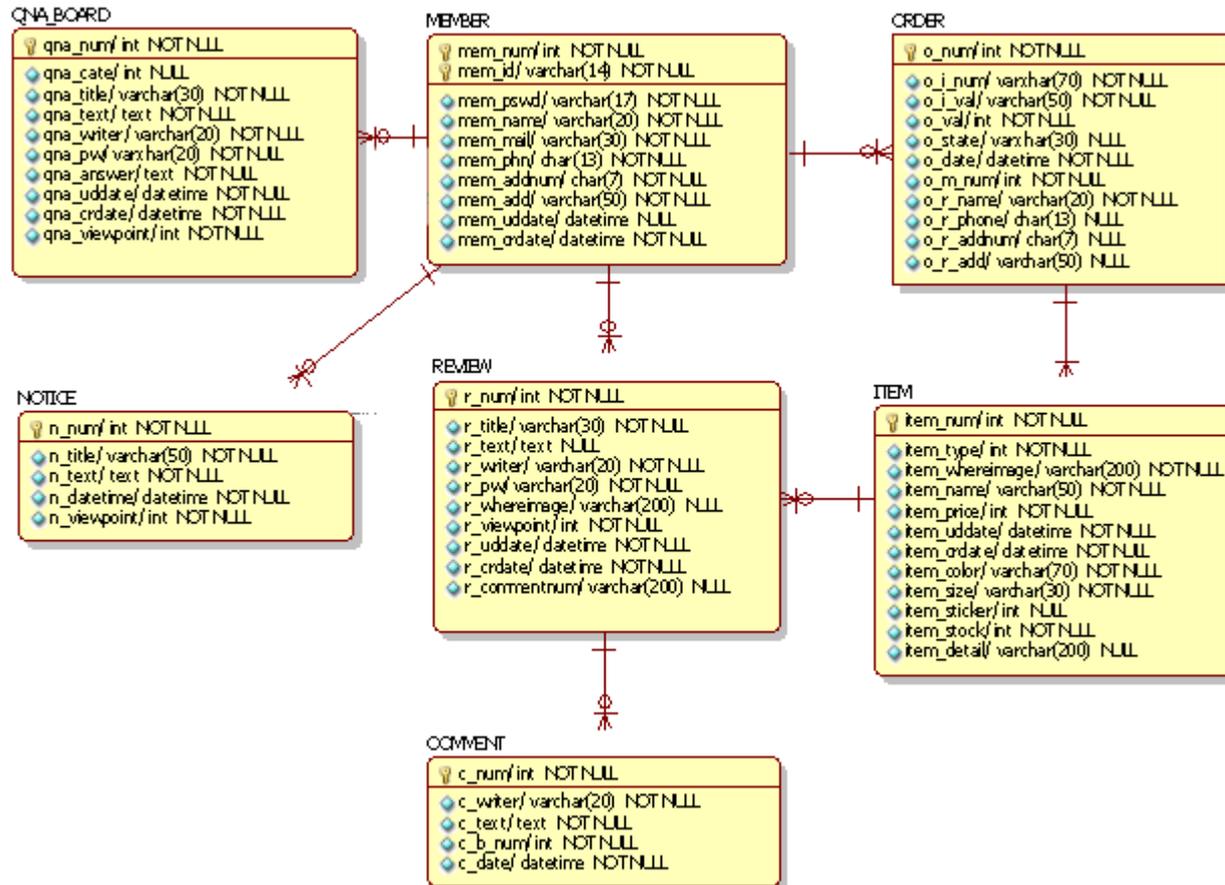
3. UC

다. UC3. 상품주문

사용사례	UC3. 상품주문
액터	회원, 비회원
시작조건	회원은 로그인 후 비회원은 로그인 하지 않은 상태로 상품을 구매
종료조건	회원과 비회원이 상품을 주문함.
주요흐름	<ol style="list-style-type: none"> 1. 회원과 비회원은 해당 의류 쇼핑몰에 올라와 있는 상품을 구경한다. 2. 회원과 비회원은 원하는 상품에 대한 정보를 본다. 3. 회원과 비회원은 마음에는 들지만 다른 상품을 좀 더 구경하고자 할 때 상품을 장바구니에 담아놓고, 구매를 원할 때<<UC6. 장바구니>>를 실행한다.(장바구니 기능은 회원만 가능) 4. 회원과 비회원은 마음에 드는 상품을 바로 구매하고자 할 때 상품에 대한 양식을 선택한다.<<E1: 상품을 선택했는데, 상품에 대한 재고가 없을 때>> 5. 시스템은 결제페이지를 회원 또는 비회원에게 보여준다. 6. 회원 또는 비회원은 결제를 위한 양식을 입력한다. 7. 선택한 상품을 주문한다.
대안흐름	<p>E1 상품에 대한 재고가 없을 때</p> <ol style="list-style-type: none"> 1. 시스템은 회원, 비회원에게 재고가 없음을 화면에 보여주고 다른 상품을 선택하도록 한다.
비.기.능 요구사항	<ol style="list-style-type: none"> 1. 상품구매 실적은 영구적으로 보관해야 한다. 2. 상품선택은 회원만 가능하다.

4. DB설계서

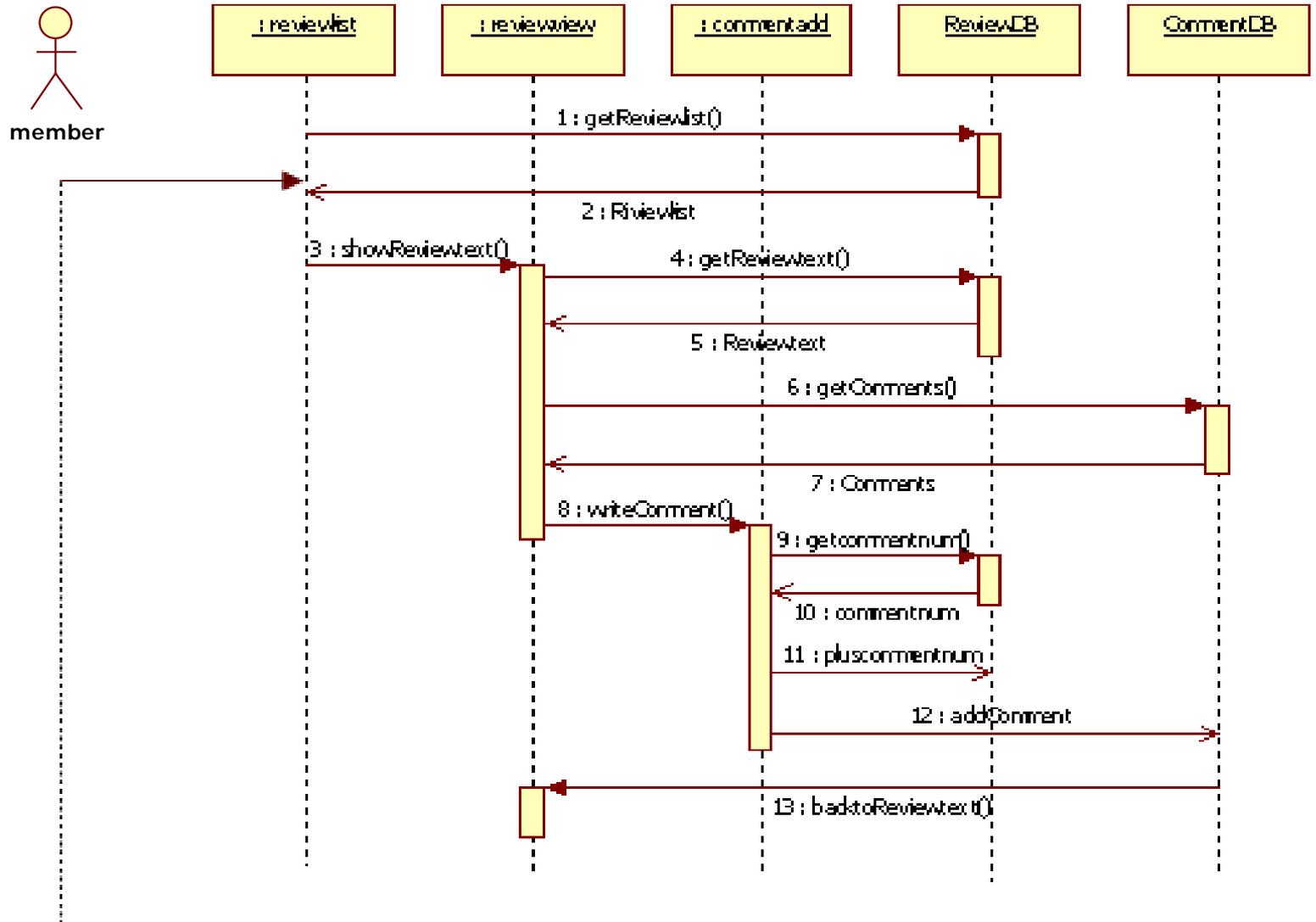
<그림3> E-R다이어그램

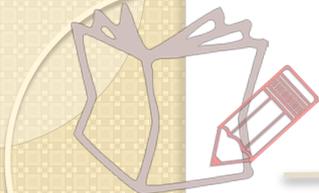


<표5> 테이블 정의서

테이블ID	ORDER						시스템	주문내역
테이블명	주문 정보						업무영역	주문관리
작성자	정지은 ,이화준						사용빈도	고
생성일	2011-12-11 오전 10:33						예상항목수	5000 개
번호	컬럼ID	한글명	데이터 타입	널허용	키	FK	기본값	비고
1	o_num	주문번호	int	NOT	PK1		IDENTITY(1,1)	
3	o_i_num	주문상품번호	varchar(70)	NOT				
4	o_i_val	상품별주문수량	varchar(50)	NOT				
5	o_val	총 가격	int	NOT				
6	o_state	주문 상태	varchar(30)	YES				
7	o_date	주문일	datetime	NOT				
8	o_m_name	주문회원 ID	int	NOT				
9	o_r_name	수취인이름	varchar(20)	NOT				
10	o_r_phone	수취인연락처	char(13)	YES				(형식:010-222-3333)
11	o_r_addnum	수취인 우편번호	char(7)	YES				(형식:111-222)
13	o_r_add	수취인 주소	varchar(50)	YES				
인덱스 정의 (필요할 때만 정의)								
번호	인덱스 명	컬럼명	Ordering	용도				
1								
2								
테이블 및 인덱스 생성 스크립트								
<pre> /***** * 주문정보 *****/ CREATE TABLE ORDER (o_num INT IDENTITY(1,1) NOT NULL, /* 주문테이블 고유키 */ o_i_num VARCHAR(70) NOT NULL, /* 주문 상품번호 */ o_i_val VARCHAR(50) NOT NULL, /* 상품별 주문수량 */ o_val INT NOT NULL, /* 총 가격 */ o_state VARCHAR(30) NOT NULL, /* 주문 상태 */ o_date DATETIME NOT NULL, /* 주문일 */ o_m_name INT NOT NULL, /* 주문자 회원번호 */ o_r_name VARCHAR(20) NOT NULL, /* 수취인 이름 */ o_r_phone CHAR(13) NOT NULL, /* 수취인 연락처 */ o_r_addnum CHAR(7) NOT NULL, /* 수취인 우편번호 */ o_r_add VARCHAR(50) NOT NULL, /* 수취인 주소 */) </pre>								

5. SSD

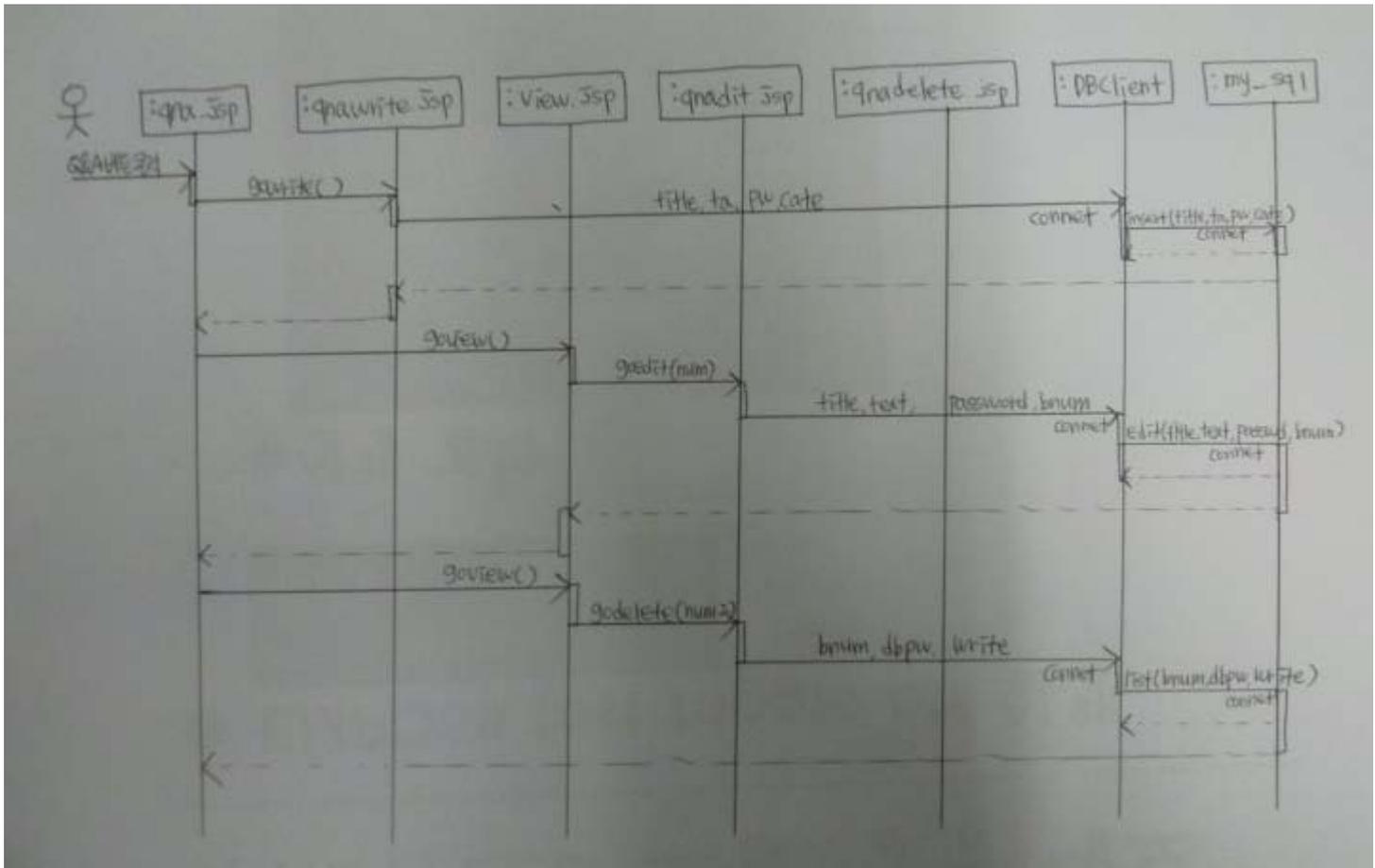


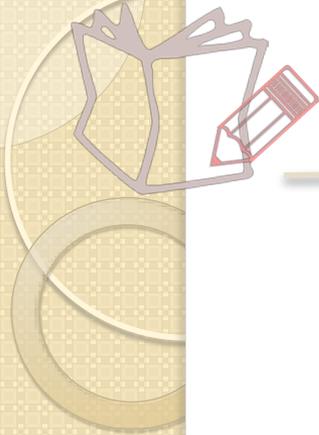


6. S.O 약정

Operation:	회원가입
Cross References:	Use Cases: 회원가입
Preconditions:	회원가입이 진행 중 이다.
Postconditions:	- LoginID가 생성되었다. - ID가 현재의 장바구니 상품과 연관되었다. - ID가 현재의 Q&A와 연관되었다. - ID가 장바구니에서 찾아진 상품과 결제가 연관되었다.

7. S.D





8. 전체클래스도

OrderMgr	ProductMgr	MemberMgr
-pool: DBConnectionMgr = null	-pool: DBConnectionMgr = null	-pool: DBConnectionMgr = null
<<create>>+OrderMgr() +insertOrder(order: OrderBean): void +getOrder(id: String): Vector +getOrderList(): Vector +getOrderDetail(no: String): OrderBean +updateOrder(no: String, state: String): boolean +deleteOrder(no: String): boolean	+ProductMgr() +insertProduct(req: HttpServletRequest): boolean +getProduct(no: String): ProductBean +updateProduct(req: HttpServletRequest): boolean +reduceProduct(order: OrderBean): void +getProductList(): Vector +deleteProduct(no: String): boolean	<<create>>+MemberMgr() +checkId(id: String): boolean +zipcodeRead(area3: String): Vector +getMember(mem_id: String): RegisterBean +getMemberList(): Vector +memberUpdate(rehBean: RegisterBean): boolean +loginCheck(id: String, passwd: String): boolean +adminCheck(admin_id: String, admin_passwd: String): boolean

9. 코드 (장바구니)

```
1 <? page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
>
<? page import="java.util.*, java.sql.*"
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"
>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
<title>Insert title here</title>
<style type="text/css"
<!--
table{
padding:0;
border-spacing:10px;
font-family:NanumGothic;
color:gray;
font-size:15px;
font-weight:bold;
border-width:0.5pt;

border-color:#000000;
line-height:1.2em;
text-align:center;
}
table.a{border-style:solid;}
a{ text-decoration:none; color:gray; text-style:border; text-align:center; font-family:NanumGothic;}
a:link{ text-decoration:none; color:gray; text-style:border; text-align:center; font-family:NanumGothic;}
a:hover{text-decoration:none; color:red; text-style:border; text-align:center; font-family:NanumGothic;}
-->
</style>
</head>
<body>
<?
if(session.getAttribute("id")==null){out.println("<script>alert('로그인 후에 장바구니 보기가 가능합니다');location.href='../beans/login.jsp';</script>");
String realFolder="";

//파일 업로드 폴더 지정
String saveFolder="pj1_item";
String encType="UTF-8";
int maxSize=5*1024*1024;

ServletContext context=getServletContext();
//현재jsp페이지의 웹 애플리케이션상 절대 경로
realFolder=context.getRealPath(saveFolder);

request.setCharacterEncoding("UTF-8");
Connection con=null;
PreparedStatement pstmt=null;
ResultSet rs=null;
int npage=0;

int cknum[]=new int[30];
int ckspre[]=new int[30];
String cknumex[]=new String[30];
String cknumn[]=new String[30];
String ckcr[]=new String[30];
```

```
2 String cksz[]=new String[30];
Cookie[] cookies=request.getCookies();
if(cookies!=null){
for(int i=0;i<cookies.length;i++){
if(cookies[i].getName().equals("wishnum")){
cknumex[i]=cookies[i].getValue();
String ctk[]=new String[20];
int c=0;
StringTokenizer stk = new StringTokenizer(cknumex[i], "/");
while(stk.hasMoreTokens()){
// st.nextToken()은 다음 값을 읽어온다.
ctk[c]=stk.nextToken();
c++;
}
cknum[i]=Integer.parseInt(ctk[0]);
cknumex[i]=ctk[0];
ckspre[i]=Integer.parseInt(ctk[1]);
ckcr[i]=ctk[2];
cksz[i]=ctk[3];
ctk[0]=null;ctk[1]=null;ctk[2]=null;ctk[3]=null;
}
}
/*if(cookies[i].getName().equals("wishselnum")){
cksnum[i]=cookies[i].getValue();
}
if(cookies[i].getName().equals("wishselcor")){
ckcr[i]=cookies[i].getValue();
}
if(cookies[i].getName().equals("wishselsize")){
cksz[i]=cookies[i].getValue();
}*/
}
}
try{
//디비관련
String jdbcUrl="jdbc:mysql://localhost/project1";
String dbId="root";
String dbPass="docs88";

Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection(jdbcUrl,dbId,dbPass);

String image[]=new String[30];
String name[]=new String[30];
int price[]=new int[30];
for(int j=0;j<cookies.length;j++){
String sql="select * from item where item_num="+cknum[j];
pstmt=con.prepareStatement(sql);
rs=pstmt.executeQuery();
while(rs.next()){
image[j]=rs.getString("item_image");
price[j]=rs.getInt("item_price");
```

3

```

        name[j]=rs.getString("iten_name");
    }
    rs=null;
}

```

```
int selectedrow=cookies.length;
```

```

//손질
int garo=selectedrow%4;
int sero=selectedrow/4;
int tt=0,n=0, bk=0;

```

```

if(selectedrow!=0){
    %<table align="left" border="0" class="a">
        <tr><td></td>
        <td>상품 이름</td>
        <td>색</td>
        <td>사이즈</td>
        <td>상품 선택 갯수</td>
        <td>가격</td>
    </tr><%

```

```

for(int k=0;k<cookies.length;k++){
    %<tr><%

```

```

        String where=realFolder+'\\'+inage[tt];
        int oo=price[tt]*ckspre[tt];
        if(name[tt]==null){bk++;break;}
        %<td >
        <a href="..hello/viewproduct.jsp?num=<%=cknun[tt]%>"
        </a></td><td width="400">
        <a href="..hello/viewproduct.jsp?num=<%=cknun[tt]%>"
        <%=name[tt] %></a></td><td>
        <%=ckcr[tt] %></td><td>
        <%=cksz[tt] %></td><td>
        <%=ckspre[tt] %></td><td>
        <font style="text-align:center"><%=oo %> 원

```

```

        <%=tt++; %></font>
        </td></tr><%

```

```

}
%></table>
<br><br><br><br><%

```

```

if(name[0]==null)
{<%등록된 상품이 없습니다.<br><%}

```

4

```
%>
```

```

<%
}catch(Exception e){
    e.printStackTrace();
}finally{
    if(pstmt!=null)try{pstmt.close();}catch(SQLException sqle){}
    if(con!=null)try{con.close();}catch(SQLException sqle){}
} %>

```

```

</body>
</html>

```

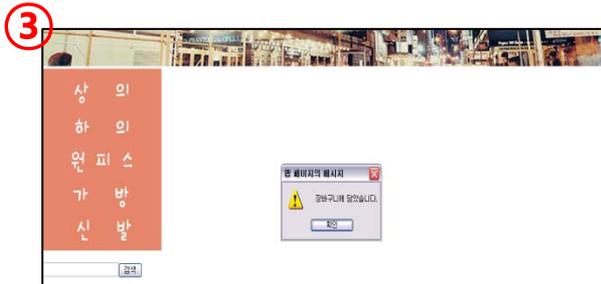
10. 실행결과UI



마음에 드는 신발을 선택한다.



상품의 수량과 색상 사이즈를 선택한다.



장바구니버튼을 눌러 장바구니에 담아놓는다.



담아놓은 상품을 보고자 할 때, 장바구니 페이지로 한다.



11. 결론

배운 점 - 프로젝트를 하면서 **기본**이 얼마나 **중요**한 것인지, **설계**가 프로젝트를 만드는데 있어 얼마만큼 큰 역할을 하는지 느꼈습니다. 둘 다 실력이 미흡하다 보니 프로젝트를 하면서 막히는 부분도 적지 않아 많이 있었고, 어떻게 조금이라도 우리 힘으로 완성해보자 완성해보자 했던 것이 많이 부족하지만 지금의 프로젝트인데, 막상 해보고 나니 **뿌듯함**도 들고 **언어에 대한 자신감**도 붙은 것 같아 무언가 기분이 색다릅니다.

향후 계획 - 앞으로도 남의 힘을 빌리지 않고 혼자 힘으로 무엇에 있어서든 열심히 할 것이고, **부족한 부분**을 **공부**함으로써 남은 프로젝트를 좀 더 완성도 높은 프로그램을 만들기 위해 열심히 잘 하겠습니다.

JSP를 이용한 인터넷 의류쇼핑몰 제작

목 차

1. 개 요
2. U C D i a g r a m
3. U C (O r d e r)
4. S S D
5. 시 스템 오 퍼 레 이 션 약 정
6. 시 스템 오 퍼 레 이 션
7. D B 설 계 서

목 차

8. UI 설계도

9. 전체 클래스도

10. 코드 (Order)

11. 실행 결과

12. 결론

1. 개요

① 필요성

이번 학기에 웹 프로그래밍에 대해 배우게 되었기 때문에 프로젝트 주제로 인터넷 의류 쇼핑몰로 해볼 것을 생각하게 되었고, 컴퓨터 공학과라면 한번쯤 시도해야 할 주제라고 생각되어 구상하게 되었습니다.

② 목적

고객에게 상품을 잘 어필 할 수 있는 쇼핑몰 제작

③ 범위

개발 범위: 회원 - 로그인, 회원가입, 장바구니, 주문조회
관리자 - 회원관리, 상품관리, 주문관리

개발하지 않을 범위: 결제기능

④ 사용자 및 주요 관심사항

사용자: 쇼핑몰 방문자(회원, 비회원), 쇼핑몰 관리자

사용자	주요 관심사	사용할 주요 기능
쇼핑몰 방문자	상품 정보(가격, 사이즈 등)	회원가입, 로그인, 상품보기, 상품주문
쇼핑몰 관리자	상품 어필, 상품 관리, 주문관리	상품관리, 주문관리

1. 개 요

⑤ 주요 기능

주요 사용자	기능	기능 설명	사용 빈도
쇼핑물 방문자	회원가입	비회원이 회원가입 절차를 거쳐 회원이 됨 (DB에 회원 정보 추가)	고
	로그인	방문자가 ID와 비밀번호를 입력하고 DB 조회 후, 회원임을 확인함	중
	상품주문	선택한 상품을 주문함.	저
사이트 관리자	상품관리	상품을 등록하거나 상품정보를 수정 할 수 있도록 함.	중
	주문관리	주문정보를 확인할 수 있도록 하고, 주문상태(ex> 상품준비중, 배송중)를 수정할 수 있도록 함.	중

⑥ 제약 사항

- 웹 프로그래밍이므로 location.href 오류로 인한 무한 루프에 빠지거나 하지 않도록 합니다.
- 로그인 / 비 로그인 사용자, 관리자의 구분을 두어 페이지 접근을 제한하도록 합니다.
- 되도록 여러 상황이 발생할 것을 생각하며 프로그램을 만들도록 합니다.

1. 개요

⑦ 타당성 분석(기술적, 경제적, 조직적)

가. 기술적 타당성

팀원 모두 응용 분야에 익숙하지 않고, JSP와 JavaScript 등의 언어를 배워야 하므로, 기술적으로도 미숙합니다.

프로젝트 크기는 약 두 달 정도의 시간이 걸리며, 크게 네 가지 기능으로 분류가 가능한 정도입니다.

나. 경제적 타당성

	2011년	2012년	2013년	총합
수익				
매출증가		84,000	94,000	178,000
합계		84,000	94,000	178,000
비용				
개발 비용	30,080			30,080
- 프린터	30			30
- 소프트웨어	50			50
- 개발 인건비	30,000			30,080
운영 비용		60,099	60,099	120,198
- 호스팅 비용	11000	6000	6000	23000
- 운영인건비		60,000	60,000	120,000
- 도메인 비용		22	22	44
합계	30,080	60,099	60,099	120,198
수익-비용		23,901	33,901	57,802

1. 개요

다. 조직적 타당성

프로그램 개발 이후, 소스를 관리자가 직접 관리해야 하기 때문에 소스를 누구나 알아보기 쉬운 방향으로 작성하며, 사이트 이용자 관점에서 편한 인터페이스를 구성하도록 합니다.

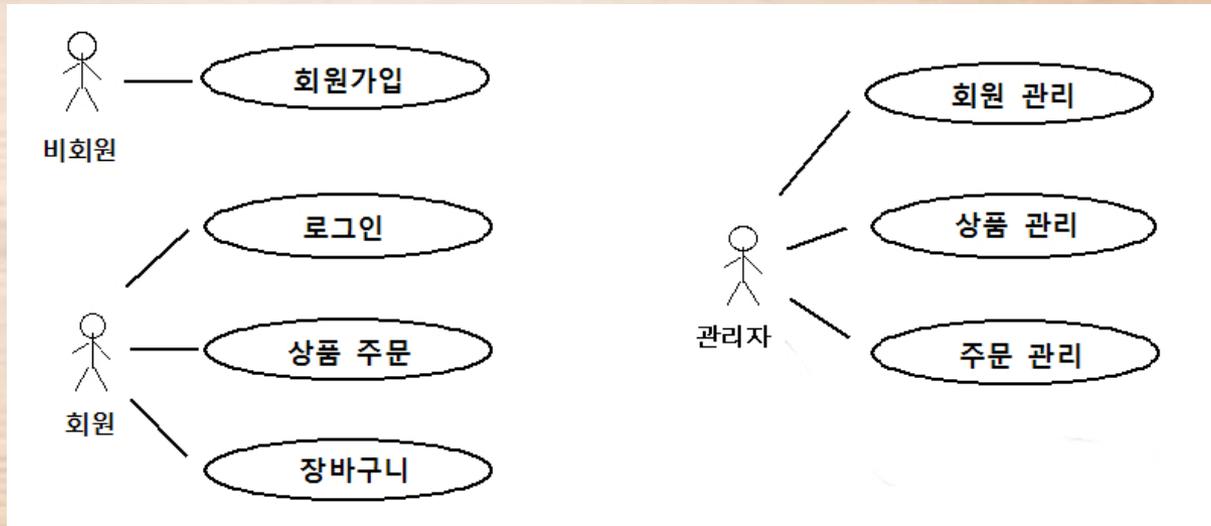
⑧ SWOT 또는 벤치마크 분석

강점 (S)	약점 (W)
검색 기능을 지원해 원하는 상품, 컬러 등으로 검색해 결과를 얻을 수 있습니다.	미적 관점에서 보았을 때, 다른 쇼핑몰에 뒤처지고, 세세한 기능들이 많이 부족합니다.
기회 (O)	위기 (T)
다른 소규모 의류 쇼핑몰들은 검색기능을 두는 경우가 적으므로, 다양하고 특이한 상품 위주로 운영한다면 성공할 가능성이 있습니다.	이미 많은 의류 쇼핑몰들이 있고, 인터넷 쇼핑몰의 상품 신뢰성이 낮습니다.

⑨ 기대효과

팀원 각자의 웹 프로그래밍 기술을 높일 수 있고, 프로젝트에 대한 경험도 또한 높일 수 있다.

2. U C Diagram

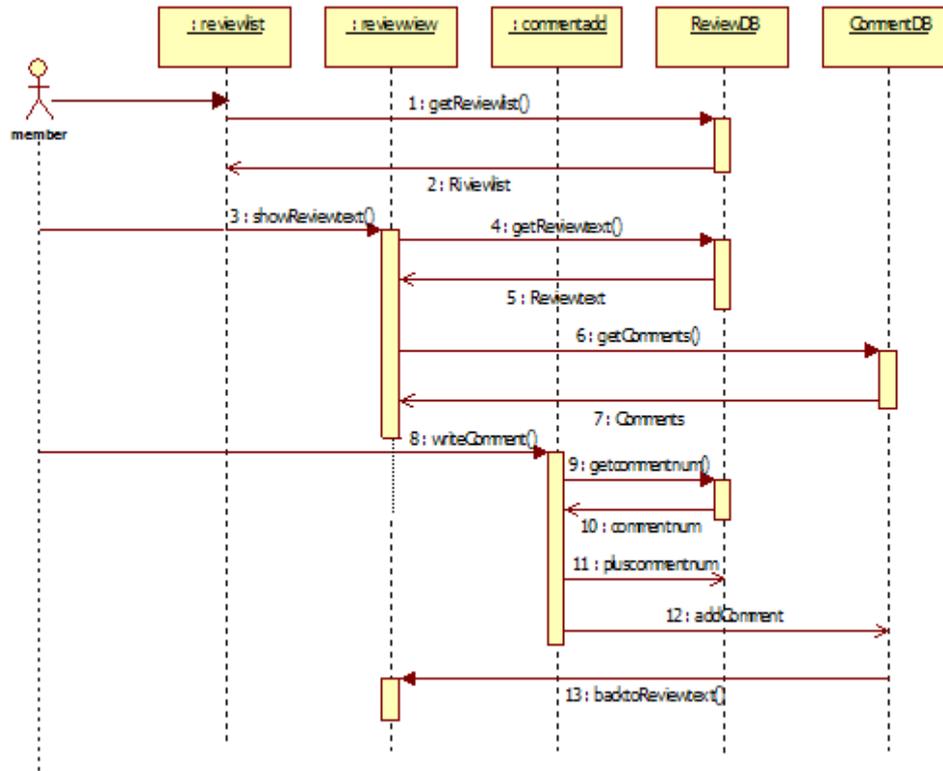


3. UC (주문 관리)

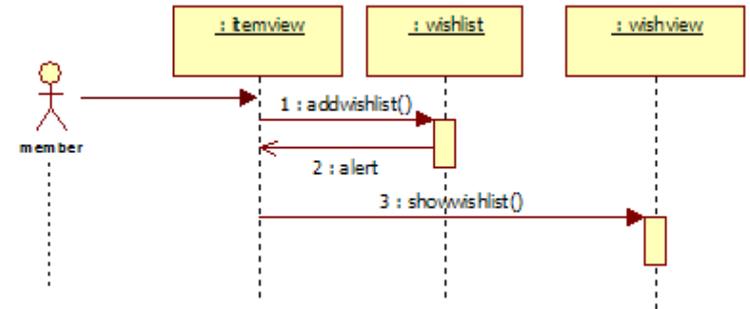
사용사례	주문 관리
액터	관리자
시작조건	관리자 로그인이 되어 있어야 하며, 메인 페이지에서 주문관리를 클릭한다.
종료조건	주문 목록 수정 또는 뒤로가기를 클릭한다.
주요흐름	<ul style="list-style-type: none">① 관리자가 메인 페이지를 누른다.② 주문 관리 탭을 선택한다.③ 주문 목록에서 주문들을 선택한 후 수정 버튼을 누른다.④ 주문 정보들 중 배송 상태를 수정해 확인을 누른다.⑤ 선택했던 주문 목록의 DB 정보가 수정된다.
대안흐름	<p>A1: 뒤로가기</p> <ul style="list-style-type: none">1. 관리자가 정보 수정 중 뒤로가기를 클릭할 경우2. 주문 목록 수정이 취소되고 관리자페이지의 주문 관리 탭으로 돌아간다. <p>A2: 관리자 로그아웃</p> <ul style="list-style-type: none">1. 관리자가 로그아웃할 경우2. 주문 목록 수정이 취소되고 메인페이지로 돌아간다.

3. S S D

① REVIEW 보기 & 코멘트 작성



② 장바구니 담기



4. 시스템 오퍼레이션 약정

① 약정 CO1: checkDuplicateld

오퍼레이션 : checkDuplicateld(

상호참조 : Use Cases: Join

사전조건 : Join의 입력이 진행중이다.

사후조건 :

- AddJoin 인스턴스 aj 가 생성되었다.
- aj 가 join 인스턴스 j와 연관되었다.
- aj 의 num이 j 의 입력값 num을 받아 변경된다.
- aj 가 DB를 체크해 num 값이 옳은지 판단해 j에 전달한다.
- aj 가 소멸한다.

② 약정 CO2: plusViewpoint

오퍼레이션 : plusViewpoint(

상호참조 : Use Cases: ViewReview

사전조건 : Review 게시판 목록을 확인하고 내용확인페이지를 누른다.

사후조건 :

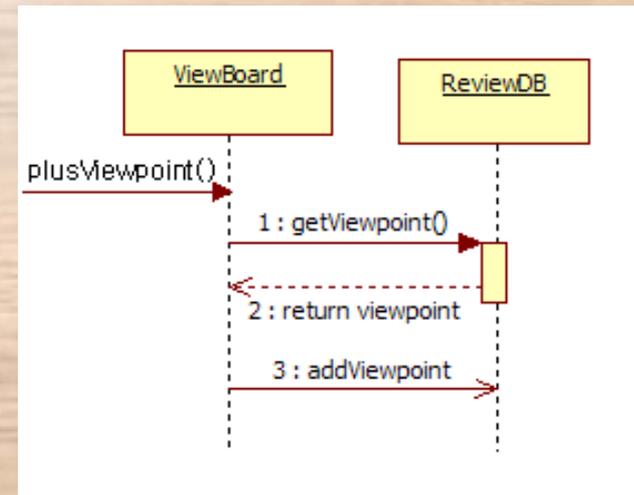
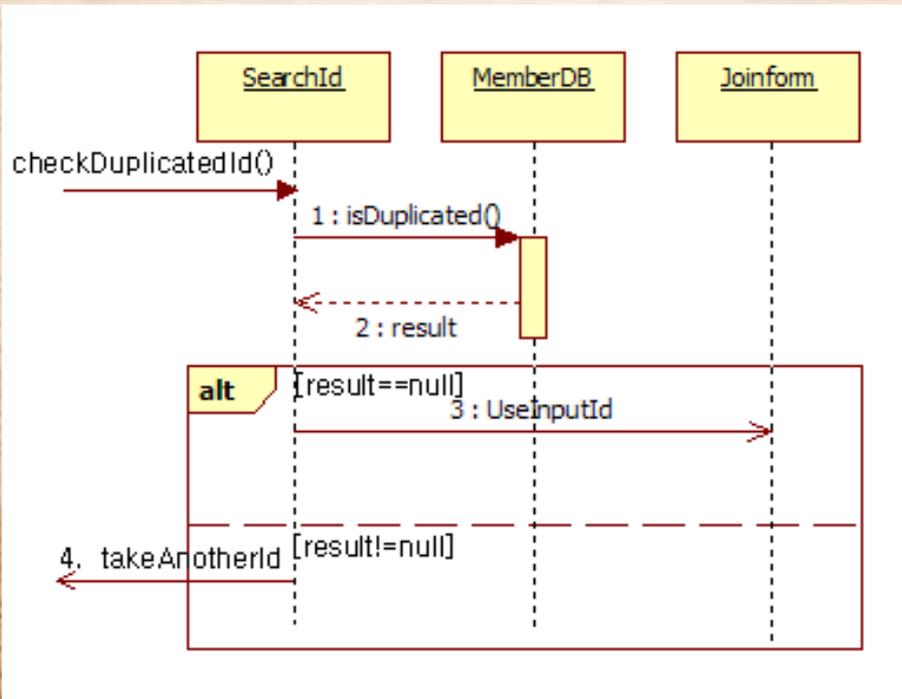
- detailViewReview 인스턴스 vr 이 생성되었다.
- vr 이 Reviewlist 인스턴스 r과 연관되었다.
- vr 에서 r의 num 을 받아 변경된다.
- vr 이 DB를 체크해 같은 num 값을 가지는 레코드를 담아온다.
- vr 이 레코드의 viewpoint 값을 1 증가시킨다.
- ViewReview 인스턴트 nvr이 생성된다.
- vr 이 viewpoint 값을 nvr에 전달한다.
- vr 이 소멸한다.

5. S . D / C . O

① S. D

- 약정 CO1 : checkDuplicatedId()

- 약정 CO2 : plusViewpoint()

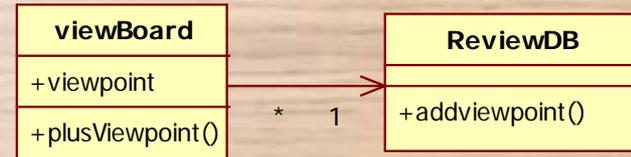
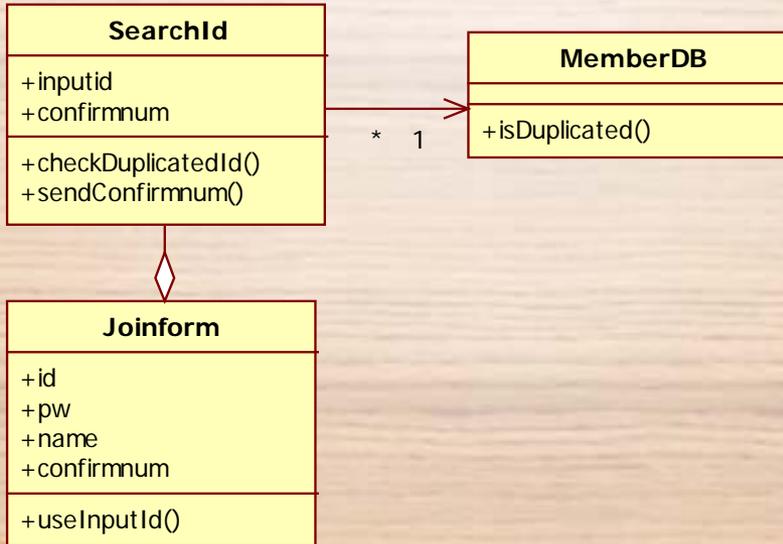


3. S . D / C . D

② C . D

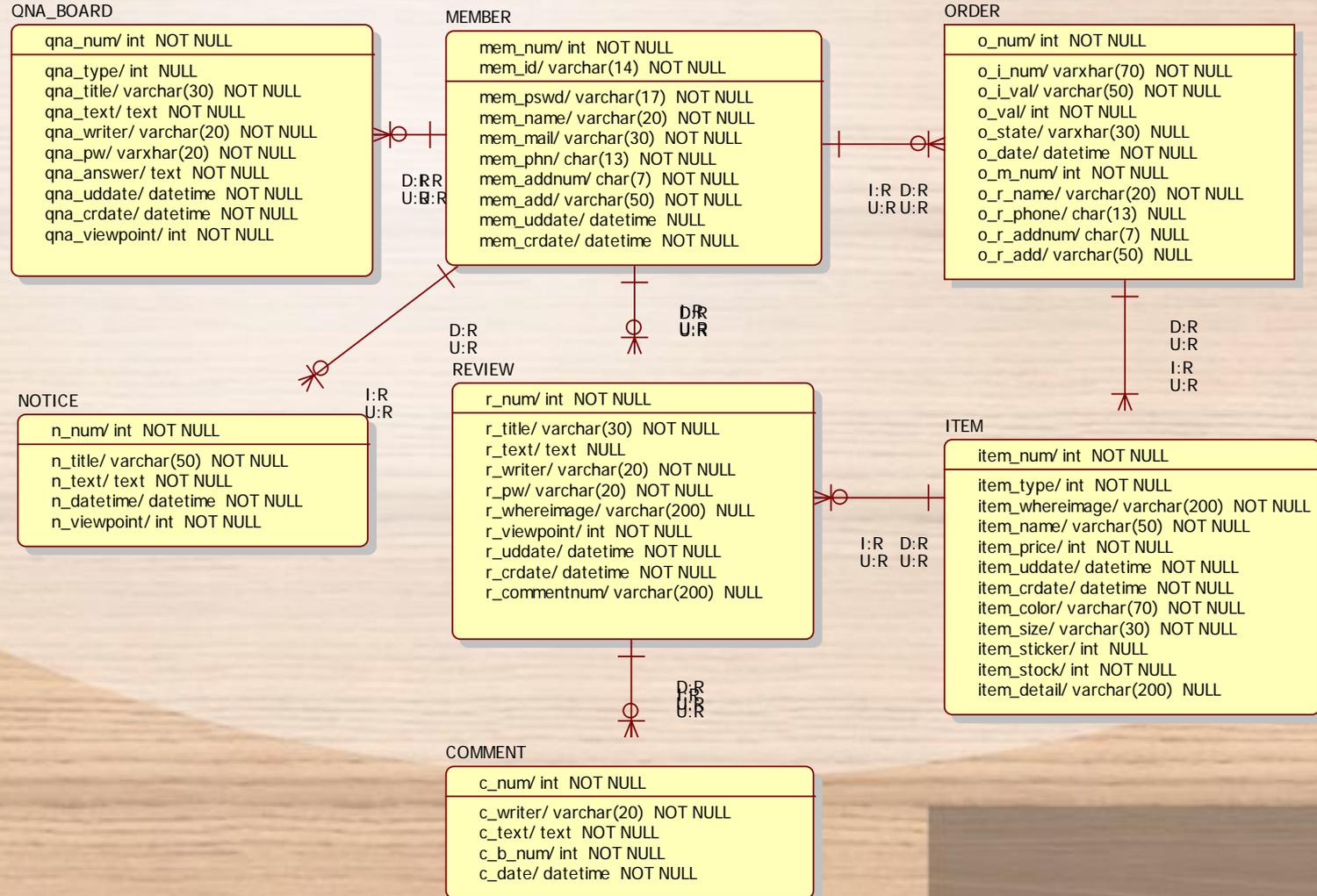
- 약정 CO1 : checkDuplicatedId()

- 약정 CO2 : plusViewpoint()



7. DB 설계서

① E-R 다이어그램



7. DB 설계서

MEMBER		ITEM		ORDER	
mem_num	회원 번호	item_num	상품 번호	o_num	주문 번호
mem_id	회원 아이디	item_type	상품 종류	o_i_num	주문상품번호
mem_pswd	비밀번호	item_whereimage	상품 이미지 저장위치	o_i_val	상품별 주문수량
mem_name	이름	item_name	상품 이름	o_val	총 주문량
mem_mail	메일	item_price	상품 가격	o_state	주문상태
mem_phn	연락처	item_uddate	상품수정일자	o_date	주문일
mem_addnum	우편번호	item_crdate	상품등록일	o_m_num	주문회원번호
mem_add	주소	item_color	상품 색상	o_r_name	수취인이름
mem_uddate	정보수정일자	item_size	크기	o_r_phone	수취인연락처
mem_crdate	가입일자	item_sticker	hit/new/sale 표시여부	o_r_addnum	수취인 우편번호
		item_stock	재고 수량	o_r_add	수취인 주소
		item_detail	상품상세정보		

QNA_BOARD		NOTICE		REVIEW		COMMENT	
qna_num	질문번호	n_num	공지번호	r_num	리뷰번호	c_num	댓글번호
qna_type	질문종류	n_title	제목	r_title	리뷰제목	c_writer	작성자
qna_title	질문제목	n_text	내용	r_text	내용	c_text	내용
qna_text	질문내용	n_datetime	작성일	r_writer	작성자	c_b_num	댓글이 달린 글 번호
qna_writer	작성자	n_viewpoint	조회수	r_pw	비밀번호	c_date	작성일
qna_pw	질문글 비밀번호			r_whereimage	이미지 저장위치		
qna_answer	답변			r_viewpoint	조회수		
qna_uddate	수정일			r_uddate	수정일		
qna_crdate	작성일			r_crdate	작성일		
qna_viewpoint	조회수			r_commentnum	댓글 수		

7. DB 설계서

② 테이블 정의 (ORDER)

테이블ID	ORDER					시스템	
테이블명	주문 정보					업무영역	주문관리
작성자	정지은					사용빈도	중
생성일						예상행목수	5000 개
번호	컬럼ID	컬럼명	데이터 타입	널허용	키	FK	기본값
1	o_num	주문번호	int	NOT	PK1		IDENTITY(1,1)
3	o_i_num	주문상품번호	varchar(70)	NOT			
4	o_i_val	상품별 주문수량	varchar(50)	NOT			
5	o_val	총 가격	int	NOT			
6	o_state	주문 상태	varchar(30)	YES			
7	o_date	주문일	datetime	NOT			
8	o_m_name	주문회원 ID	int	NOT			
9	o_r_name	수취인이름	varchar(20)	NOT			
10	o_r_phone	수취인연락처	char(13)	YES			(형식: 010- 2222- 3333)
11	o_r_addnum	수취인 우편번호	char(7)	YES			(형식: 111- 222)
13	o_r_add	수취인 주소	varchar(50)	YES			
인덱스 정의 (필요할 때만 정의)							
번호	인덱스 명	컬럼명	Ordering	용도			
1							
테이블 및 인덱스 생성 스크립트							
<pre> /***** * 주문정보 *****/ CREATE TABLE ORDER (o_num INT IDENTITY(1,1) NOT NULL, /* 주문테이블 고유키 */ o_i_num VARCHAR(70) NOT NULL, /* 주문 상품번호 */ o_i_val VARCHAR(50) NOT NULL, /* 상품별 주문수량 */ o_val INT NOT NULL, /* 총 가격 */ o_state VARCHAR(30) NOT NULL, /* 주문 상태 */ o_date DATETIME NOT NULL, /* 주문일 */ o_m_num INT NOT NULL, /* 주문자 회원번호 */ o_r_name VARCHAR(20) NOT NULL, /* 수취인 이름 */ o_r_phone CHAR(13) NOT NULL, /* 수취인 연락처 */ o_r_addnum CHAR(7) NOT NULL, /* 수취인 우편번호 */ o_r_add VARCHAR(50) NOT NULL, /* 수취인 주소 */); </pre>							

8. UI 설계도

사용자 UI설계서					
시스템명	의류 인터넷 쇼핑몰	서비스시스템명	.	화면ID	main01
파일명	index.jsp	화면명	메인 화면	작성 자	정지은
화면설명	index.jsp 의 화면 구성입니다.			작성 일	2011-12-11

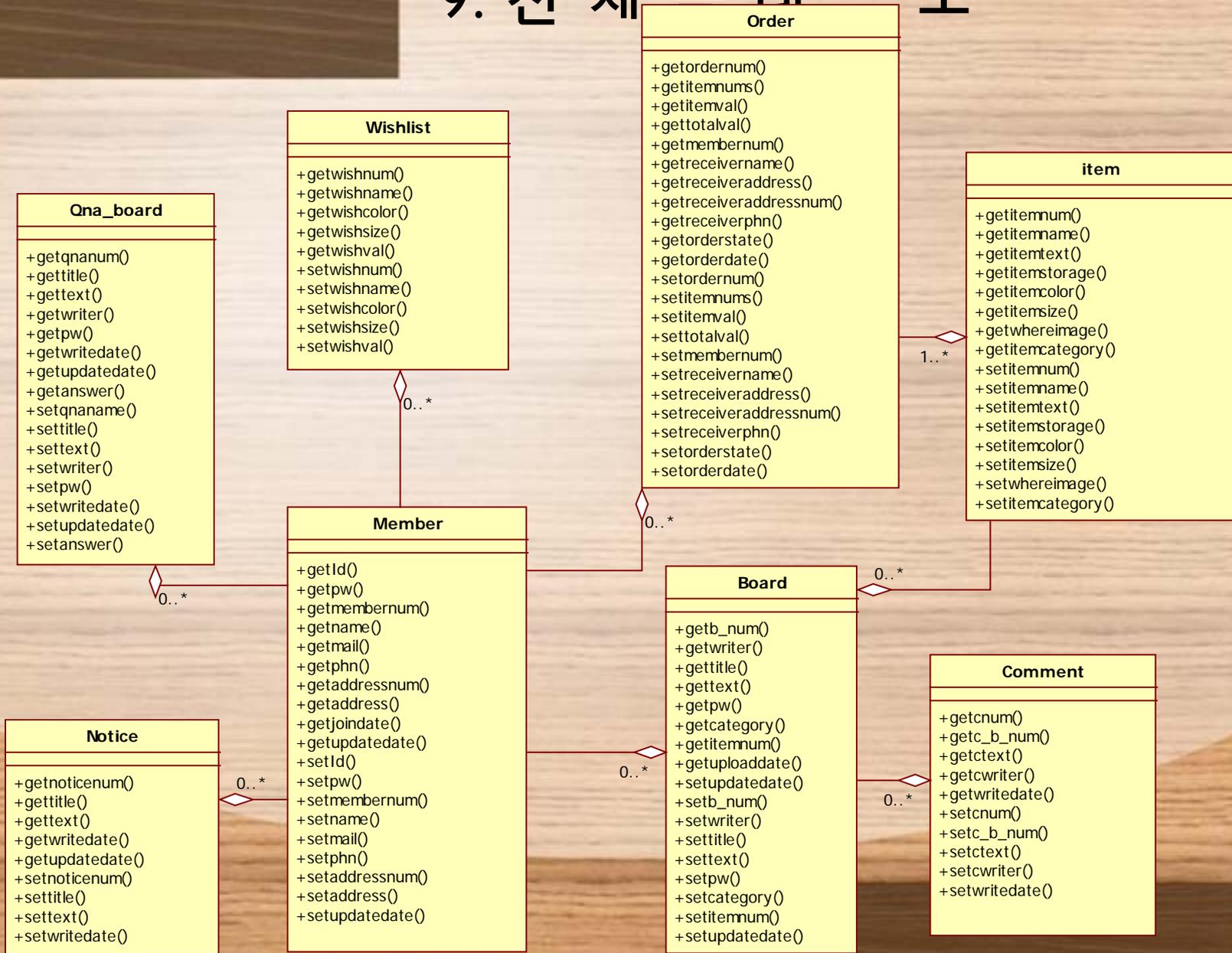


쇼핑몰을 맨 처음 클릭했을 때 보여지는 페이지입니다.

상단 메뉴 및 좌측 메뉴, 검색창 등을 누르면 내부 프레임과 연결되는 페이지가 변경됩니다.

main 페이지의 내부프레임은 처음에는 hello.jsp와 연결 되어 있으며, 상품 목록 위에는 베스트 상품이 흘러가면서 보이도록 해 방문자의 눈길을 끌도록 구성했습니다.

9. 전체 클래스도



10. 코드 (Wishlist)

① wishlist.jsp (장바구니 담기)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.util.*, java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script language="javascript">
<!--
function go(){
location.href="../hello/view.jsp";
}
-->
</script>
<title></title>
</head>
<body>
<%
String realFolder="";

//파일 업로드 폴더 지정
String saveFolder="pj1_item";
String encType="UTF-8";
int maxSize=5*1024*1024;

ServletContext context=getServletContext();
//현재jsp페이지의 웹 애플리케이션상 절대 경로
realFolder=context.getRealPath(saveFolder);
request.setCharacterEncoding("UTF-8");
Connection con=null;
PreparedStatement pstmt=null;
ResultSet rs=null;
int npage=0;
String num=request.getParameter("num");
String selnum=request.getParameter("selnum");
String selcor=request.getParameter("selcol");
String selsize=request.getParameter("selsize");
String where=realFolder+'WW'+num;
```

```
num+='/' +selnum;
num+='/' +selcor;
num+='/' +selsize;
Cookie cookie2[]=request.getCookies();
```

```
int cn=cookie2.length-1;
out.println(cn);
String wishnum="wishnum"+cn;
Cookie cookie=new Cookie(wishnum, num);
cookie.setMaxAge(3600*36);
```

```
response.addCookie(cookie);
out.println("<script>alert('장바구니에 담았습니다.');
```

② wishview.jsp (장바구니 보기)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"
%>
<%@ page import="java.util.*, java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
if(session.getAttribute("id")==null){out.println("<script>alert('로그인 후에
장바구니 보기가 가능합니다.');
```

10. 코드 (Wishlist)

```
ServletContext context=getServletContext();
realFolder=context.getRealPath(saveFolder);
request.setCharacterEncoding("UTF-8");
Connection con=null;
PreparedStatement pstmt=null;
ResultSet rs=null;
int npage=0;
int cknum[]=new int[30];
int ckspre[]=new int[30];
String cknumex[]=new String[30];
String cksnum[]=new String[30];
String ckcr[]=new String[30];
String cksz[]=new String[30];
Cookie[] cookies=request.getCookies();
if(cookies!=null){
for(int i=0;i<cookies.length;i++){
String hi="wishnum"+i;
if(cookies[i].getName().equals(hi)){
cknumex[i]=cookies[i].getValue();
String ctk[]=new String[20];
int c=0;
StringTokenizer stk = new StringTokenizer(cknumex[i], "/");
while(stk.hasMoreTokens()){
    ctk[c]=stk.nextToken();
    c++; }
cknum[i]=Integer.parseInt(ctk[0]);
cknumex[i]=ctk[0];
ckspre[i]=Integer.parseInt(ctk[1]);
ckcr[i]=ctk[2];
cksz[i]=ctk[3];
ctk[0]=null;ctk[1]=null;ctk[2]=null;ctk[3]=null;
}}
try{
String jdbcUrl="jdbc:mysql://localhost/project1";
String dbld="root";
String dbPass="docs88";
Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection(jdbcUrl,dbld,dbPass);
String image[]=new String[30];
String name[]=new String[30];
int price[]=new int[30];
for(int j=0;j<cookies.length;j++){
String sql="select * from item where item_num="+cknum[j];
```

```
pstmt=con.prepareStatement(sql);
rs=pstmt.executeQuery();
while(rs.next()){
image[j]=rs.getString("item_image");
price[j]=rs.getInt("item_price");
name[j]=rs.getString("item_name");
}rs=null;}
int selectedrow=cookies.length;
int garo=selectedrow/4;
int sero=selectedrow/4;
int tt=0,m=0, bk=0;
if(selectedrow!=0){
%><table align="left" border="0" class="a">
<tr><td></td><td>상품 이름</td><td>색</td><td>사이즈</td>
<td>상품 선택 갯수</td><td>가격</td></tr><%
for(int k=0;k<cookies.length;k++){
%><tr><%
String where=realFolder+'WW'+image[tt];
int oo=price[tt]*ckspre[tt];
if(name[tt]==null){bk++;break;}
%><td >
<a href=" ../hello/viewproduct.jsp?num=%=cknum[tt]%">
</a></td><td width="400">
<a href=" ../hello/viewproduct.jsp?num=%=cknum[tt]%">
%=name[tt] %></a></td><td>
%=ckcr[tt] %></td><td>
%=cksz[tt] %></td><td>
%=ckspre[tt] %></td><td>
<font style="text-align:center">%=oo %> 원
<%tt++; %></font>
</td></tr><%
}}
%></table><br><br><br><br><br><br>
if(name[0]==null)
{%>등록된 상품이 없습니다.<br><%}><%
}catch(Exception e){
e.printStackTrace();
}finally{
if(pstmt!=null)try{pstmt.close();}catch(SQLException sqle){}
if(con!=null)try{con.close();}catch(SQLException sqle){}
} %>
</body>
</html>
```

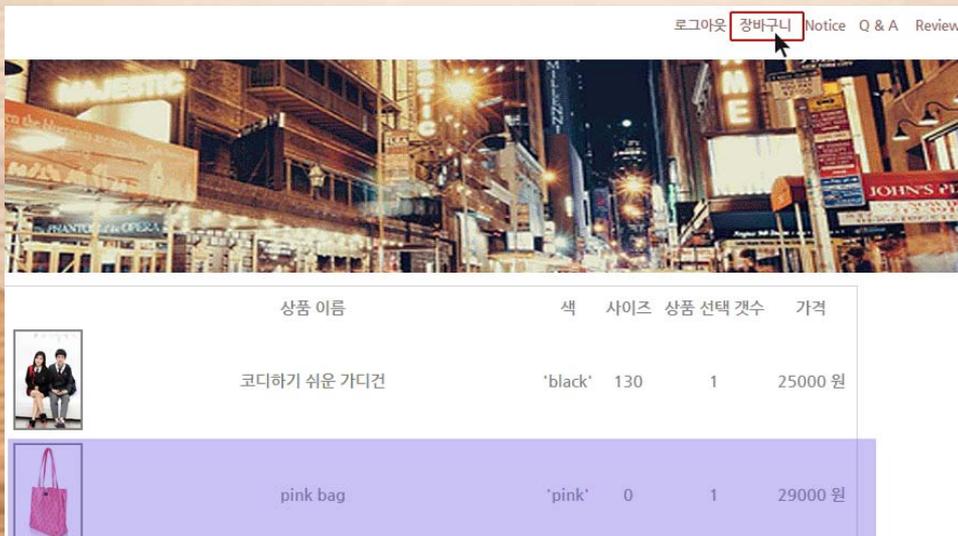
11. 실행 결과 (wish list)



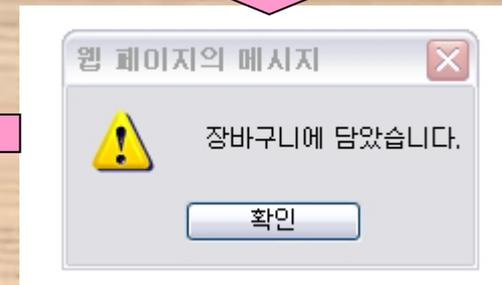
① 상품 목록에서 장바구니에 담은 상품을 선택합니다.



② 상품 상세 페이지에서 수량, 색상, 사이즈를 체크하고 장바구니 버튼을 클릭합니다.



④ 상단의 장바구니를 클릭해 목록에서 담은 상품을 확인합니다.



③ 장바구니에 담았다는 메시지를 확인합니다.

12. 결 론

① 배운 점

프로그램을 만들 때 무작정 달려드는 것보다, 프로그램의 목적과 성격에 맞게 계획하여 하나 하나 단계를 밟아 가는 것이 훨씬 효율적이라는 것을 배웠습니다. 또 실전에서 클라이언트의 요구가 변할 수 있다는 점도 알게 되었고 가독성 높고, 유연한 프로그램을 짜도록 해야 한다는 것을 알게 되었습니다.

과제로 작성했었던 요구분석서나, DB 설계서 등을 실제 코딩 할 때 참고 해 보고 나서, UC나 E-R 다이어그램, 테이블 정의서 등이 정말 유용하다는 것을 알 수 있었습니다.

② 향후 계획

강의를 듣기 전에는 아무 계획 없이 달려들었다가 골치 아픈 상황에 빠진 적이 많았는데, 앞으로는 프로그램을 만들 때 꼭 UC와 SSD, E-R 다이어그램을 만들어 보는 습관을 가질 것입니다.

다음 프로젝트를 준비할 때는 배운 것들을 바탕으로 프로그램 설계를 좀 더 꼼꼼히 한 후에 할 예정입니다.