

주니어 디자인 프로젝트 최종 프로젝트 보고서

The Online Shopping Mall Simulator

2014112068 김가림

2014112037 윤재문

2015년 12월 19일

차 례

1. 프로젝트 개요
 - 1.1. 프로젝트 주제
 - 1.1.1 주제 선정 이유
 - 1.1.2 주제 분석
 - 1.2. 프로젝트 목표
 - 1.3. 프로젝트 진행 일정
 - 1.4. 역할 분담
2. 프로젝트 내용
 - 2.1. 프로그램 구성
 - 2.1.1. UML 모델링
 - 2.1.1.1. Use Case Diagram
 - 2.1.1.2. Class Diagram
 - 2.1.2. 자료구조
 - 2.2. 클래스 분석
 - 2.2.1. START 클래스
 - 2.2.2. USER 클래스
 - 2.2.3. ADMIN 클래스
 - 2.2.4. PRODUCT 클래스
 - 2.3. 프로그램 기능
 - 2.3.1. 사용자
 - 2.3.1.1. 로그인 / 회원가입

- 2.3.1.2. 상품목록 확인
 - 2.3.1.3. 상품 검색
 - 2.3.1.4. 내 정보 확인
 - 2.3.1.5. 장바구니
 - 2.3.1.6. 상품 주문 / 결제
 - 2.3.1.7. 코인 충전
 - 2.3.2. 관리자
 - 2.3.2.1. 로그인
 - 2.3.2.2. 주문서 확인 / 처리
 - 2.3.2.3. 상품 목록 확인
 - 2.3.2.4. 상품 검색
 - 2.3.2.5. 상품 수정
 - 2.3.2.6. 상품 추가
 - 2.3.2.7. 재고 위험 상품 정렬
 - 2.3.2.8. 히트 상품
- 3. 프로그램 장점
 - 3.1. 상품 추천 기능
 - 3.1.1. 추천 기능 설명
 - 3.1.2. 알고리즘
- 4. 프로그램 테스트 과정
 - 4.1. 사용자
 - 4.2. 관리자
- 5. 사용자 매뉴얼
- 6. 프로젝트 마무리
 - 6.1. 평가 및 느낀 점

- 7. 전체 소스 코드
- 8. 설문조사 내용

1. 프로젝트 개요

1.1. 프로젝트 주제

1.1.1. 주제 선정 이유

우리 조의 프로젝트 이름은 'The Online Shopping Mall Simulator'이고, 콘솔 창에서 온라인 의류 쇼핑몰을 구현하는 프로그램이다. 많은 사람들이 쇼핑몰을 이용해서 옷을 구매하고, 우리나라뿐 아니라 전세계적으로 인터넷 쇼핑몰의 수는 매우 많고, 또 매년 증가하고 있다. 평소에 인터넷을 이용해서 많은 쇼핑을 하기 때문에 그 동안 쇼핑몰을 이용하면서 특별한 기능이 있었으면 좋겠다는 생각을 많이 했고, 그 것을 이번 학기 주니어 디자인 프로젝트 수업을 이용해서 구체화하고 구현해보고 싶다는 생각을 해서 해당 주제로 프로젝트를 진행하게 되었다.

1.1.2. 주제 분석

온라인 쇼핑몰은 기본적으로 고객이 인터넷 브라우저를 통해 해당 쇼핑몰 서버에 접속하여 물건의 목록을 확인하고, 주문을 할 수 있고, 마찬가지로 쇼핑몰 관리자는 고객의 주문 목록을 확인하고 제품을 배송하는 시스템이다. 관리자나 고객의 공통점은 등록된 제품목록을 확인할 수 있어야 한다는 점이다. 또한 대부분의 쇼핑몰은 회원 전용으로 이루어진다. 회원 전용 쇼핑몰을 구현하기 위해서 고객과 관리자의 입장에서 처리할 요구사항을 정리하여 보았다.

◆ 고객

- 회원가입이 된 사용자만 이용 가능하다.

- 로그인 후 제품 목록을 확인할 수 있다.
- 제품의 정보를 확인 후 고객은 장바구니에 특정 제품을 등록할 수 있다.
- 로그아웃 후 프로그램 재접속시 장바구니에 등록된 제품정보는 삭제된다.
- 회원은 주문한 제품의 배송정보 등 주문정보를 확인할 수 있다.
 - ◆ 관리자
- 관리자는 로그인 후 제품정보를 등록할 수 있다.
- 등록된 제품 정보를 확인할 수 있다.
- 고객이 주문한 주문 목록을 확인할 수 있다.
- 고객이 주문한 정보를 바탕으로 배송상태를 처리할 수 있다.

1.1.3. 프로젝트 목표

위에서 언급한 내용들은 대부분의 인터넷 쇼핑몰에서 제공하는 기능이다. 우리는 기본 개념을 포함해서 특별한 기능을 추가하고자 하였다. 여러 가지 기능보다 우리는 상품 추천 기능에 목표를 맞추기로 하였다. 다음은 상품 추천 기능의 세부 내용이다.

- 고객과 제품은 각각 [cuty, sexy, formal, casual]의 4가지 특성을 갖고 각 특성은 최대 100점의 점수를 갖는다.
- 고객이 회원가입 할 때 설문조사를 통해서 위의 특성의 값을 초기화 할 수 있다.
- 제품을 등록할 때 관리자는 제품의 특성 값을 입력해야 한다.
- 고객은 상세정보 페이지에서 자신의 특성 정보를 확인할 수 있다.
- 고객과 특성 일치율이 80% 이상인 상품들은 추천 상품 목록 페이지에서 확인할 수 있다.
- 고객이 상품을 구매했을 때 특정 알고리즘에 의해 상품의 특성이 고객의 특성에 누적해서 반영된다.

이러한 기능들을 이용해서 사용자는 자신이 평소에 어떤 옷을 입는지, 어떤 성향 인지 쉽게 파악할 수 있고, 그래서 새로운 옷을 구매할 때 자신에게 더 잘

어울리고 자신과 비슷한지 알 수 있어서 인터넷을 이용해서 옷을 사더라도 실패할 확률이 적어질 것이다. 혹은 자신이 입고 싶은 옷의 성향이 있을 때 상품 구매 시 그 성향이 높은 옷을 구매함으로써 고객이 원하는 스타일의 옷을 구매하기 쉬워진다. 또한 이 쇼핑몰에서 옷을 구매하면 할수록 구매하는 상품이 자신의 성향에 반영되기 때문에 많은 옷을 구매할수록 자신에게 더 잘 맞는 옷을 구매할 수 있다.

1.1.2절에서 얘기한 기본 기능들과 해당 절에서의 추가적인 기능의 내용들을 콘솔 어플리케이션으로 만들기 위해 계획하고 구현하는 것이 프로젝트의 최종 목표이다.

1.1.4. 프로젝트 진행 일정

초반에는 프로그램의 개념을 정확히 하고, 세부 기능을 다진 후 추가기능 구현 시 필요한 알고리즘을 정확히 명시한 후에 코드를 작성하기로 하였다. 아래 표는 실제 진행된 프로젝트의 일정이다.

구 분		일 정								비 고
		1	2	3	4	5	6	7	8	
	프로젝트									

개 념 수 립	주제 확립									
	기본 기능									
	특징적인 아이디어									
	ADT 작성									윤 재 문
	파일입출력									윤 재 문
코 드 작 성	로그인 /회원가입									김 가 림
	상품 검색 /상품 목록									윤 재 문
	상품 추천 분석(일치 도)									김 가 림
	결제/주문 서									
	테스팅									
	보고서 작 성									

1.1.5. 프로젝트 역할 분담

김가림: 조장으로서 전체적인 프로젝트의 아이디어를 수립하였다. 매 회 발표마다 PPT 제작을 도맡아서 했고 상품 추천 알고리즘에 대한 부분을 구상하였다. 코드상에서는 주로 User클래스의 메소드를 작성하였고, 상품 추천과 분석, 그리고 결제와 주문서, 장바구니 등의 코드를 작성하였다.

윤재문: 프로그램에서 자료의 저장을 텍스트파일을 이용한 파일입출력으로 했는데 해당 부분을 도맡아서 구현했다. 또한 프로그램에서 제일 핵심적인 자료구조인 벡터를 사용한 자료저장방법을 구상했다. 주로 Admin클래스의 메소드를 작성하였고 검색과 메뉴출력을 담당하였으며 콘솔창에서 문자열 입력을 이용한 선택이 아닌 키보드와 스페이스바를 이용하는 조작으로 함수를 구현하여서 더욱 편리하게 사용할 수 있게 하였다.

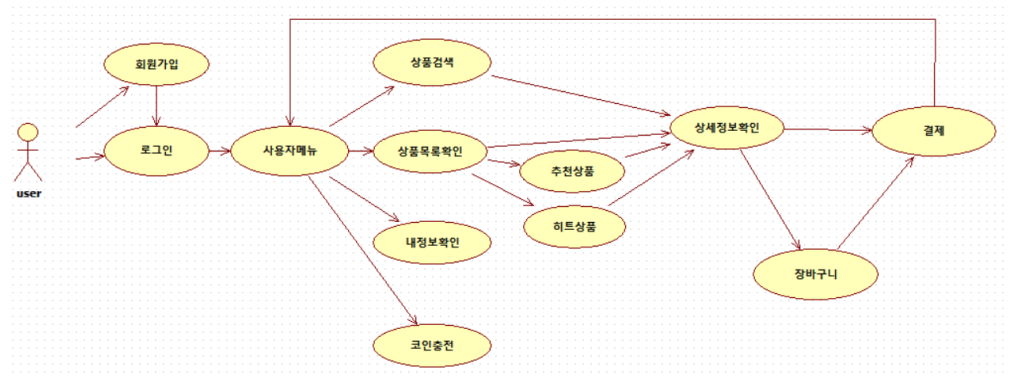
2. 프로젝트 내용

2.1. 프로그램 구성

2.1.1. UML 모델링

2.1.1.1. Use Case Diagram

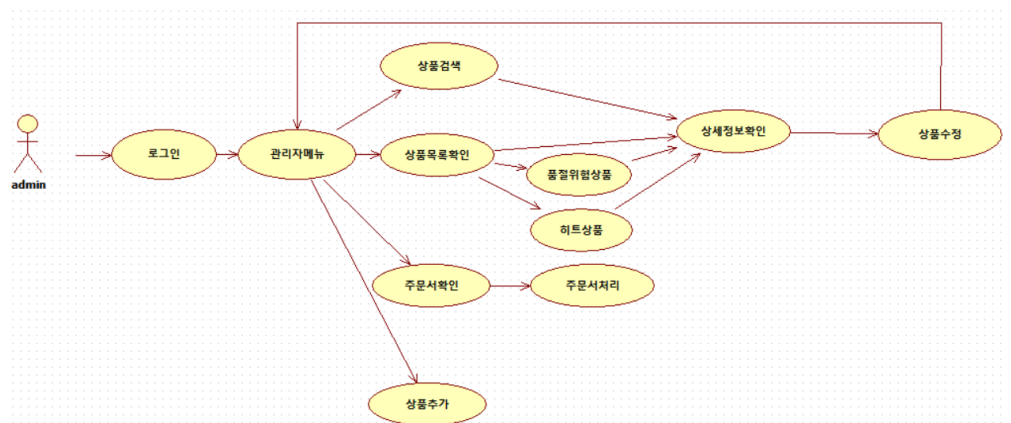
- 사용자



사용자는 위의 그림과 같이 로그인, 회원가입 후 사용자 메뉴 출력, 상품 검색과 목록확인을 통해서 상세정보 확인이 가능하다. 그 후 장바구니에 담거나 바로 결제를 한 다음 다시 사용자 메뉴로 돌아올 수 있다.

나의 정보 확인이나 코인 충전을 통해 사용자 편의성을 제공하였다.

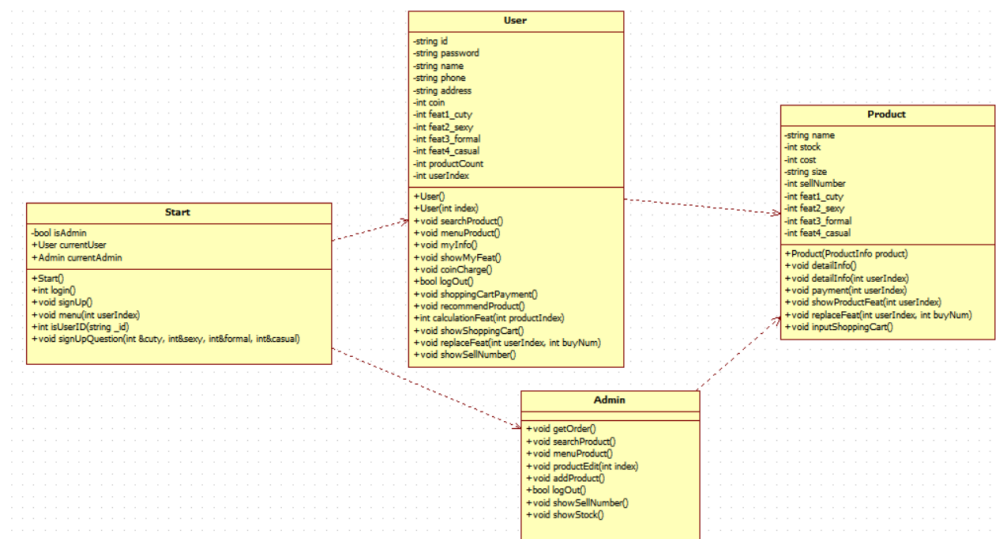
- 관리자



관리자는 로그인 후 관리자 메뉴를 보게 된다. 상품을 검색하고

목록을 확인한 후 상세정보를 확인할 수 있는데, 그 후 상품의 수정이 가능하다. 상품 목록에서는 품질위험 상품, 히트상품을 확인할 수 있다. 또한 사용자들이 주문한 내역을 확인하고 배송정보를 처리할 수 있다.

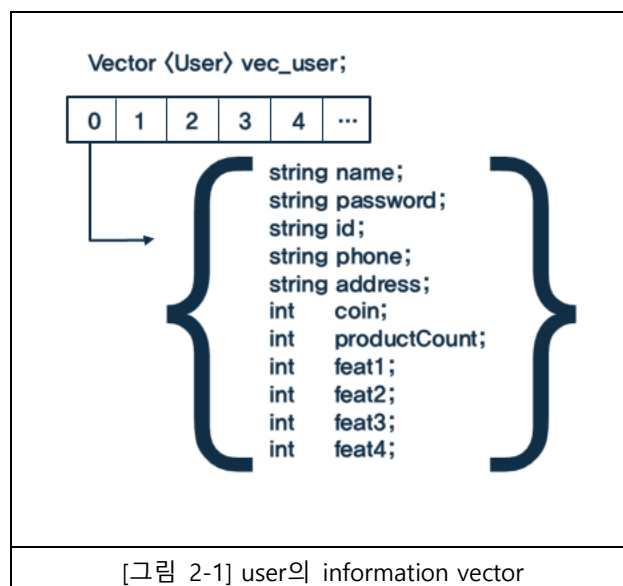
2.1.1.2. Class Diagram



클래스들은 위와 같이 구성되어있다. Start에서 로그인이 완료되면 관리자인지 사용자인지 여부를 판단하여 Admin과 User중 한 클래스를 사용한다. Admin과 User클래스는 각각 관리자와 사용자의 정보와 기능을 담고 있으며 Product클래스를 사용한다. Product클래스는 상품에 대한 기능을 담고있다.

2.1.2. 자료 구조

프로그램이 시작되면 텍스트파일에서 정보들을 차례대로 읽어와서 구조체에 저장하고, 구조체는 그 구조체를 자료형으로 하는 벡터에 저장된다. 프로그램에서 사용되는 구조체와 벡터를 그림으로 나타내면 다음과 같다.



위의 그림은 User라는 구조체를 자료형으로 하는 vector의 모습을 도식화한 것이다. 그림에서 알 수 있듯이 하나의 사용자의 특성들이 User구조체에 담긴 후 `vec_user`라는 벡터에 쌓여서 저장된다. 위와 같은 벡터는 User를 포함해서 Admin, Product 이렇게 총 3가지가 존재한다.

2.2. 클래스 분석

2.2.1. Start 클래스

```

class Start
{
private:
    bool isAdmin;
public:
    Start();
    int login();
    void signUp();
    void menu(int userIndex);
    int isUserID(string _id);
    User currentUser;
    Admin currentAdmin;
    void signUpQuestion(int &cuty, int&sexy, int&formal, int&casual);
};

```

클래스들중 가장 먼저 생성되어 사용되는 클래스이다. 프로그램의 시작부분에 생성되며, 텍스트에서 데이터를 가져와 벡터에 저장하고 로그인, 회원가입기능을 지원한다. 필드의 bool isAdmin을 통해 사용자를 관리자/사용자로 나누고 각각에 맞는 메뉴를 선택해 사용할 수 있도록 설계했다.

2.2.2. User 클래스

```

class User
{
private:
    string id;
    string password;
    string name;
    string phone;
    string address;
    int coin;
    int feat1_cuty;
    int feat2_sexy;
    int feat3_formal;
    int feat4_casual;
    int productCount;
    int userIndex;
public:
    User(){};
    User(int index);
    void searchProduct();
    void menuProduct();
    void myInfo();
    void showMyFeat();
    void coinCharge();
    bool logOut();
    void shoppingCartPayment();
    void recommendProduct();
    int calculationFeat(int productIndex);
    void showShoppingCart();
    void replaceFeat(int userIndex, int buyNum);
    void showSellNumber();
};

```

일반 사용자의 기능을 담당하는 클래스이다. 필드에 사용자들의 정보가 저장할 수 있도록 설계했고, 메소드에는 사용자가 필요로 하는 내정보확인, 코인충전 및 상품검색, 상품목록, 장바구니 기능을 지원하도록 설계했다.

2.2.3. Admin 클래스

```

class Admin
{
public:
    void getOrder();
    void searchProduct();
    void menuProduct();
    void productEdit(int index);
    void addProduct();
    bool logOut();
    void showSellNumber();
    void showStock();
};

```

관리자의 기능을 지원하도록 설계하였다. 관리자가 주문서를 확인 및 수정하고, 상품을 추가하는 기능을 지원하며, 상품검색, 상품목록을 보여주어 원하는 상품을 선택하면 해당 상품의 정보를 수정 할 수 있도록 설계하였다.

2.2.4. Product 클래스

```

class Product
{
private:
    string name;
    int stock;
    int cost;
    string size;
    int sellNumber;
    int feat1_cuty;
    int feat2_sexy;
    int feat3_formal;
    int feat4_casual;
public:
    Product(ProductInfo product);
    void detailInfo();
    void detailInfo(int userIndex);
    void payment(int userIndex);
    void showProductFeat(int userIndex);
    void replaceFeat(int userIndex, int buyNum);
    void inputShoppingCart();
};

```

필드에 상품의 정보를 저장하도록 설계했으며, 상품의 정보를 볼 수 있도록 하였고, 사용자일 경우 해당상품 결제, 관리자일 경우 해당상품 수정 및 삭제의 기능을 할 수 있도록 설계 하였다.

2.3. 프로젝트 기능

2.3.1. 사용자

2.3.1.1. 로그인 / 회원가입

프로그램의 시작에서 사용자는 로그인 혹은 회원가입을 선택할 수 있다. 회원가입 시 입력하는 아이디는 "admin"일 수 없고(관리자의 아이디이다) 이미 존재하는 아이디일 수 없다. 이는 isUserID라는 메소드로 확인한다. 아이디를 입력 후 비밀번호와 이름, 전화번호, 주소를 입력받는다. 그 후 사용자의 성향 초기화 설정을 위해서 설문조사 페이지로 넘어간다.

설문조사는 총 20문항으로, 평소 사용자의 스타일이 어떤지를 주로 물어보며 사용자가 원하는 스타일을 파악할 수 있게 한다. 설문조사 내용은 8장에 첨부하였다. 설문조사를 끝내게 되면 총 4개의 사용자 특성 정보의 초기화가 완료된다. 회원가입이 완료 되면 가입 축하 코인 10만원을 적립해준다.

회원가입을 완료하게 되면 로그인 페이지로 넘어간다. 로그인시 아이디를 입력하면 isUserID메소드에서 user 벡터에 저장된 아이디를 확인 후 벡터의 index값을 반환하고 아이디가 없을 시에는 -1을 반환한다. 반환받은 벡터의 index값에 있는 구조체의 비밀번호가 사용자가 입력한 비밀번호와 같을 때에 로그인에 성공하게 된다.

2.3.1.2. 상품 목록 확인

사용자는 등록된 모든 상품을 볼 수 있다. 기본적으로 상품이 등록된 순서대로 상품이 나열되지만, 버튼을 누르면 일치율이 나와 높은 상품을 보여주는 상품 추천 페이지, 현재까지 누적 판매개수가 높은 상품들을 보여주는 히트상품 페이지로 넘어간다. 여기에서 상품을 선택하면 상세상품 페이지로 넘어간다.

2.3.1.3. 상품 검색

원하는 상품의 이름을 통해서 상품을 검색할 수 있는 기능이다. 완벽한 단어가 아니더라도 상품이 모두 출력된다. 상품 목록과 마찬가지로 스페이스바를 누르면 상품을 주문할 수 있는 상세정보 페이지로 넘어간다.

2.3.1.4. 내 정보 확인

사용자는 내 정보 확인 페이지에서 자신의 회원 정보를 확인할 수 있다. 회원가입 시 입력했던 이름, 핸드폰번호, 주소를 확인할 수 있고 현재까지 구매한 물건의 수를 파악할 수 있다. 또한 그 상품들을 주문하면서 누적해서 변화된 자신의 성향을 그래프로 보여주기 때문에 자신의 특성을 파악하기 쉽다.

2.3.1.5. 장바구니

사용자는 상품의 상세정보 페이지에서 바로 주문을 할 수 있을 뿐더러, 장바구니에 담을 수 있다. 이 장바구니는 사용자가 로그인 했을 때에만 유지되고, 사용자가 로그아웃 하고 난 후에는 남아있지 않게 되어있다. 장바구니에 담은 상품은 장바구니 페이지에서 확인할 수 있는데, 장바구니 페이지에서는 주문 페이지로 바로 넘어갈 수 있다. 또한 주문이 완료 되었을 때에는 장바구니가 비워진다.

2.3.1.6. 상품 주문 / 결제

사용자는 상품의 상세페이지 혹은 장바구니에서 상품의 주문이 가능하다. 상품 주문시 첫번째로 주문하는 상품을 확인한다. 상품의 이름, 사이즈를 확인한 후 구매 수량을 입력받는다. 그 후 배송비를 포함한 가격을 보여준다.

두번째로는 주문자의 정보를 확인한다. 사용자가 회원가입 시 입력했던 기존 주소를 보여주고 배송지와 같은지 확인한다. 사용자가 구매하는 것이 아닌 선물등의 상황을 고려했다. 배송지와 주문지가 같지 않다면 새로운 주소를 입력받는다. 같은 이유로 주문자의 전화번호를 물어보고 다를 시에 입력받는다.

마지막으로 결제 정보를 보여주는데, 이 부분에서는 사용자의 코인과 결제할 금액을 비교한다. 만약에 사용자가 금액이 부족하면 구매는 종료되고, 부족하지 않을 시에는 사용자의 금액에서 구매하는 상품의 금액이 차감된다.

2.3.1.7. 코인 충전

코인 충전 페이지에서는 사용자가 원하는 만큼 코인을 충전할 수 있다. 회원의

전화번호를 입력 받은 후 회원가입 시 입력했던 번호와 같을 때에만 코인 충전이 가능하다. 충전한 코인은 현재 코인에 누적되어서 저장된다.

2.3.2. 관리자

2.3.2.1. 주문서 확인 / 처리

사용자가 상품을 구매하여 결제가 완료되면 해당 내역에 대한 주문서가 생성된다. 생성된 주문서들은 order벡터에 저장된 후 최종적으로 order.txt에 저장되어 프로그램 종료 이후에도 주문서에 대한 정보 손실이 없도록 설계하였다.

텍스트파일에서 불러온 주문서들은 벡터에 저장되며 관리자는 주문서 확인 메뉴를 통해 어떤 주문이 들어왔는지 바로 확인 할 수 있다. 주문서에는 주문자에 대한 정보와 상품에 대한 정보가 담겨있어 관리자가 물품 배송시에 참고할 수 있도록 하였고 관리자는 주문서를 확인한 후 배송여부를 수정할 수 있도록 하였다.

2.3.2.2. 상품 목록 확인

관리자가 등록된 모든 상품들의 목록을 볼 수 있도록 하는 기능이다. 관리자는 나열된 상품들 중 하나를 선택하여 상품정보 수정이나 상품삭제를 할 수 있게 설계했다. 처음 프로그램 설계에서는 아래 그림과 같이 상품의 번호를 입력받아 해당상품을 선택하는 방법으로 설계했었다.

상품 번호를 입력하세요:

1. 상품1
2. 상품2
3. 상품3
.
.
.
.
.
.
.
.

이때 상품목록이 많아지면 한정된 콘솔창안에 상품들을 모두 나열 할 수 없었고 상품이 창안에 너무 많이 나열되어 글씨 가독성이 좋지 않다고 생각

하였다.

7주차 회의를 통해 처음계획에 약간의 수정을 하였는데, 상품을 10개씩 한 페이지로 나누어 한 페이지당 10개씩의 상품만 나열되도록 하였고 상하방향키를 통해 상품변경을 하고 현재 가리키고 있는 상품은 글씨를 붉게 표시해 현재 어떤 상품을 가리키는가를 알기 쉽도록 구현하였다. 좌우방향키를 통해서는 페이지를 넘길 수 있도록 하여 이후 목록을 볼 수 있도록 하였고 스페이스 바를 통해 해당 상품을 선택할 수 있도록 계획하고 구현하였다.

1. 상품1	
2. 상품2	←
3. 상품3	
.	
.	
.	
.	
.	
10.상품10	

1페이지

상품목록 기능을 완료한 뒤 추가기능을 구현할 때 관리자라면 상품을 어떻게 정렬하면 좋을까 생각해보았다. 재고량이 얼마 남지않은 상품의 재고를 조정해주기 위해 재고량이 10이하인 상품을 목록으로 뽑아주는 기능과 판매자가 어떤 상품이 인기가 많은지 정렬해볼 수 있도록 판매량 순으로 정렬하는 기능이 있으면 관리자가 편리하게 이용할 것이라 생각하여 상품목록에서 각각 s와 h를 입력하면 해당 목록을 볼 수 있도록 코드를 추가하였다. 재고량보기와 히트상품(판매량순 정렬) 메뉴 모두 일반메뉴와 마찬가지로 상품을 선택하면 해당 상품의 상세보기 페이지로 넘어가게 된다.

2.3.2.3. 상품 검색

원하는 상품의 이름을 통해서 검색 할 수 있는 기능이다. 검색어가 들어가 있는 상품은 모두 출력된다. 예를 들어 "플레이어스커트"의 경우 "스커트"를 검색하면 출력되게 된다. 상품 목록과 마찬가지로 상품을 선택하면 해당 상품 상세보기 페이지로 넘어가서 관리자는 상품정보를 수정할 수 있다. 상품 목록과 마찬가지로 7주차 회의를 통해 스위치 문을 이용하는 방식에서 키보드로 조작하는 방식으로 수정하고 검색된 상품이10개가 넘는 경우 10개씩 여러

페이지로 나누어 출력되도록 구현하였다.

2.3.2.4. 상품 수정

사용자의 경우 상품을 선택하면 결제로 진행되지만 관리자의 경우 상품을 선택하면 상품수정기능을 사용할 수 있게 된다. 상품 수정기능은 상품을 삭제하는 상품삭제, 상품의 재고를 조정하는 상품입고, 가격조정, 사이즈조정, 상품 성향조정등 상품의 정보를 수정하는 기능을 지원한다. 상품삭제의 경우 벡터에서 상품을 삭제하고 텍스트 파일의 내용을 수정하며, 상품 입고의 경우 입력한 값으로 재고량이 바뀌는 것이 아닌 +=연산을 하여 관리자가 추가된 상품량 만 입력해도 현재 상품재고와 합해지도록 설계하였다,

2.3.2.5. 상품 추가

상품을 상품목록에 추가하는 기능이다. 상품명, 재고량, 가격, 사이즈, 4가지성향을 입력받아 상품을 벡터에 추가하고 수정된 벡터의 내용을 텍스트 파일에 저장한다

2.3.2.6. 재고 위험 상품 정렬

관리자가 재고량이 적은 상품을 확인 할 수 있도록 재고량이 10이하인 상품만을 목록으로 출력한다. 해당 목록에서 상품을 선택하면 해당 상품의 상세정보페이지로 바로 이동이 가능하다. 재고량이 얼마 남지 않은 상품을 확인 할 수 있을 뿐만 아니라 해당 상품들의 입고를 편리하게 할 수 있다.

2.3.2.7. 히트 상품

일반적인 쇼핑몰의 히트 상품기능과 동일하다, 판매량 순으로 상위에 출력되어 사용자와 관리자는 어떤 상품이 많이 팔렸는지 파악 할 수 있다. 이를 통해서 유행하는 상품의 유행을 확인 할 수 있고, 관리자는 어떤 상품의 재고를 더 준비해야 할지, 어떤 성향의 상품을 준비하는 것이 더 좋을지 파악 할 수 있다.

3. 프로그램 장점

3.1. 상품 추천 기능

3.1.1. 추천 기능 설명

사용자와 상품은 각각 고유의 4가지의 특성[sexy, cuty, formal, casual]을 갖고 있고, 이를 비교해서 사용자에게 더 잘 어울리고 더 잘 맞는 상품을 추천해주는 것이 이번 프로젝트의 핵심 기능이다.

우선 사용자는 상품 목록 페이지에서 자신과 특성 일치율이 높은 상품들의 목록만 따로 볼 수 있다. 기존 쇼핑몰에서는 단순히 상품을 나열했다. 혹은 판매량이 높은 상품들이 추천상품으로 목록 페이지 상단에 올려져 있던 것을 종종 볼 수 있었는데, 많은 사람들이 구매했다는 이유로 사용자가 이 상품에 잘 어울리지는 알 수 없었다. 상품과 사용자의 특성을 비교해서 일치율이 높은 상품만 보여주기 때문에 사용자 맞춤 서비스라고 할 수 있다.

또한 제품별 상세 페이지에서는 사용자와 상품의 정보를 비교해서 그래프로 나타내므로 사용자가 이 상품이 자신에게 얼마나 어울리는지를 시각적으로 한눈에 알아보기 쉽다.

사용자가 상품을 구매하였을 시에는 상품의 성향이 사용자에게 누적해서 반영이 된다. 따라서 사용자의 특성은 사용자가 이 쇼핑몰에서 옷을 많이 사면 살수록 특성이 누적되기 때문에 사용자가 이 쇼핑몰에서 구매를 많이 할수록 실제 사용자와 더 비슷해지게 된다.

3.1.2. 사용자 추천 알고리즘

사용자가 상품을 구매했을 때 바뀌는 성향의 알고리즘은 아래와 같다.

$$\frac{(\text{사용자 특성} \times \text{사용자가 현재까지 구매했던 상품 수}) + (\text{상품특성} \times \text{사용자가 현재 구매하는 상품 수})}{\text{총 구매 상품 수}}$$

현재까지 구매했던 상품 수를 사용자 특성에 곱함으로써, 현재까지의 특성의 평균을 구할 수 있고, 새로 구매하는 상품의 수만큼 상품 특성을 곱하고 두가지의 수를 더해서 총 구매 상품수로 나눠서, 현재까지 구매 특성의 평균을 구한다고 생각할 수 있다.

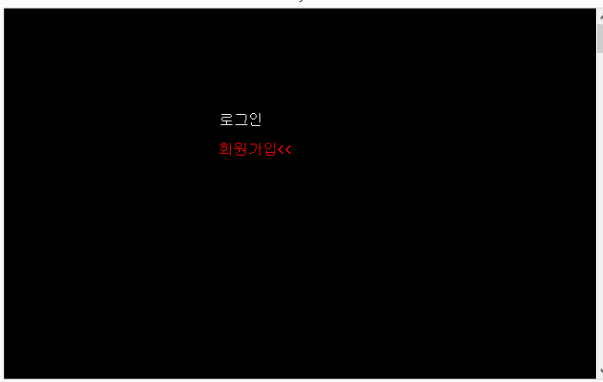
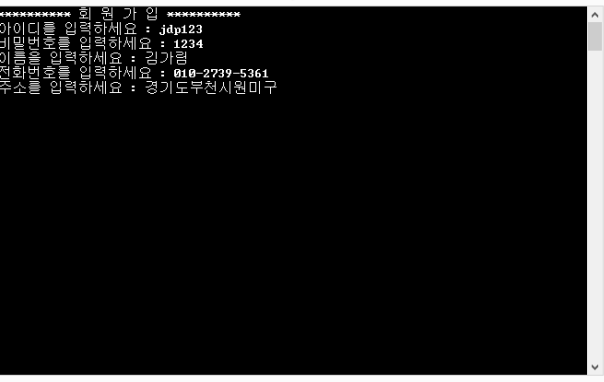
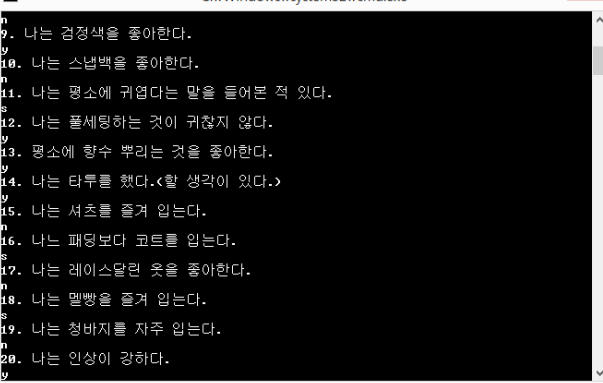

위와 같은 공식을 네가지의 특성에 모두 적용해서 물건을 한번 구매할 때마다

사용자의 네가지의 특성이 다 변하게 된다.

4. 프로그램 테스트

4.1. 사용자

4.1.1. 회원가입 / 로그인

	
<p>[그림 4.1.1-1] 로그인 / 회원가입 결정 페이지</p>	<p>[그림 4.1.1-2] 회원가입 페이지</p>
	
<p>[그림 4.1.1-3] 설문조사 페이지</p>	<p>[그림 4.1.1-4] 로그인 페이지</p>

- 시작화면에서 사용자는 로그인과 회원가입 중 하나를 선택해야 한다.

- 회원가입시 기본 정보를 입력받고, 사용자의 성향 분석을 위한 설문조사 페이지로 넘어간다.
- 회원가입 후 시작 화면으로 돌아가고, 사용자는 로그인을 할 수 있다.

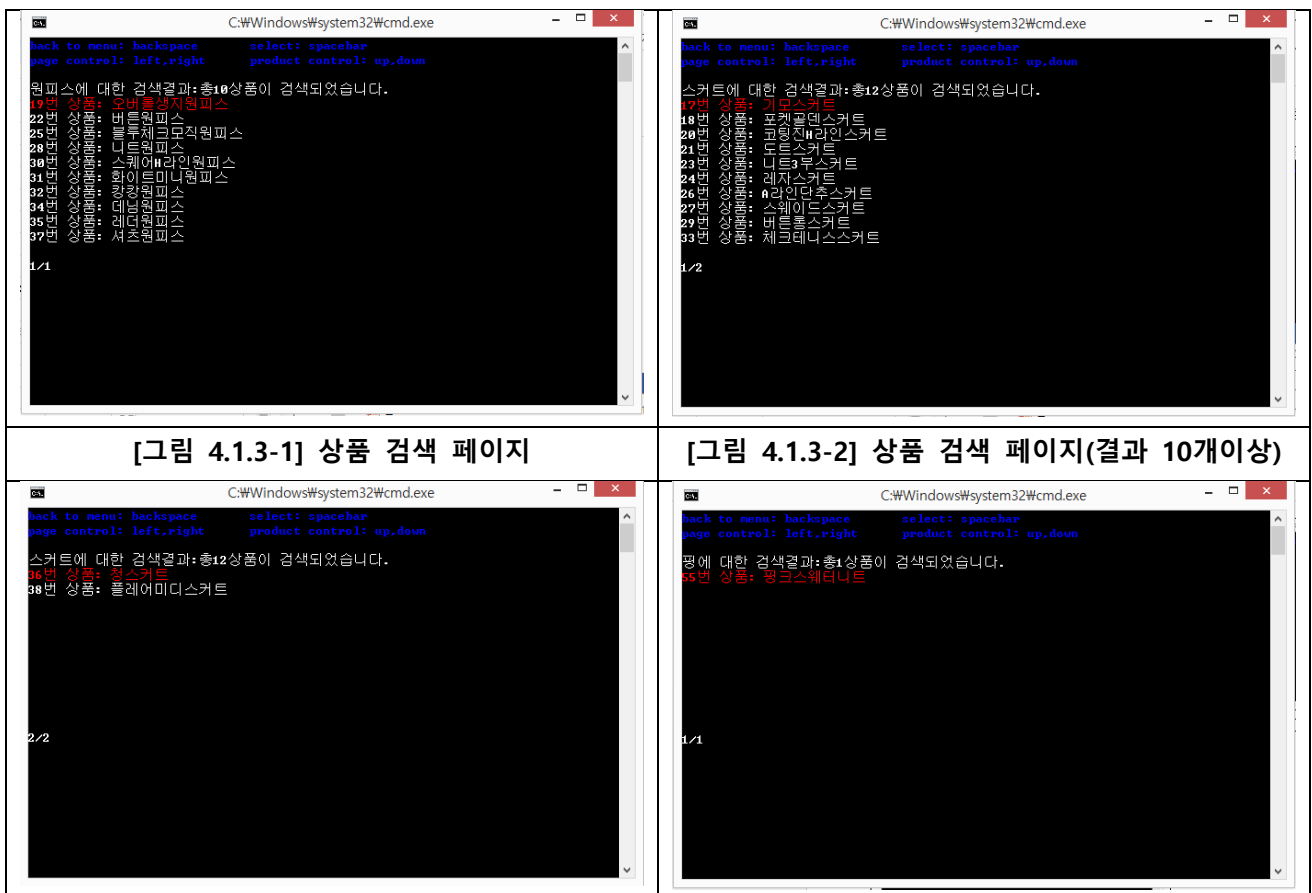
4.1.2. 상품 목록

<p>[그림 4.1.2-1] 사용자 메뉴</p>	<p>[그림 4.1.2-2] 상품 목록 확인(기본)</p>
<p>[그림 4.1.2-3] 추천 상품 목록</p>	<p>[그림 4.1.2-4] 히트 상품 목록</p>

- 사용자 메뉴에서 키보드의 방향키로 메뉴를 이동할 수 있다.

- 기본적으로 10개의 상품을 보여주고, 그것보다 많을 경우에 왼쪽, 오른쪽의 방향키로 페이지를 이동할 수 있다.
- 상품 목록을 입력하면 기본적으로 저장된 순서에 따른 상품 목록이 나온다.
- R을 누르면 추천상품페이지로 넘어간다. 특성 일치율이 85%가 넘는 상품들만 보여준다.
- H를 누르면 히트상품페이지로 넘어간다. 누적 판매량수가 높은 상품부터 보여진다.
- 모든 목록 페이지에서 스페이스바를 누르면 상세 정보 페이지로 넘어간다.
- 모든 페이지에서 뒤로가기를 누르면 이전의 페이지로 돌아간다.

4.1.3. 상품 검색



[그림 4.1.3-3] 상품 검색 페이지(결과 10개 이상)	[그림 4.1.3-4] 단어 검색이 아닌 경우
-----------------------------------	---------------------------

- 검색하고자 하는 상품이 있을 때에는 메뉴에서 상품 검색 페이지를 이용한다.
- 상품을 검색했을 때 페이지에 보여지는 상품의 수는 최대 10개이다. 10개 이상인 경우 방향키를 이용해서 다음 페이지로 넘어갈 수 있다.
- 전체 상품명이나 올바른 단어를 검색하지 않았을 때에도 그림 4와 같이 검색이 가능하다.

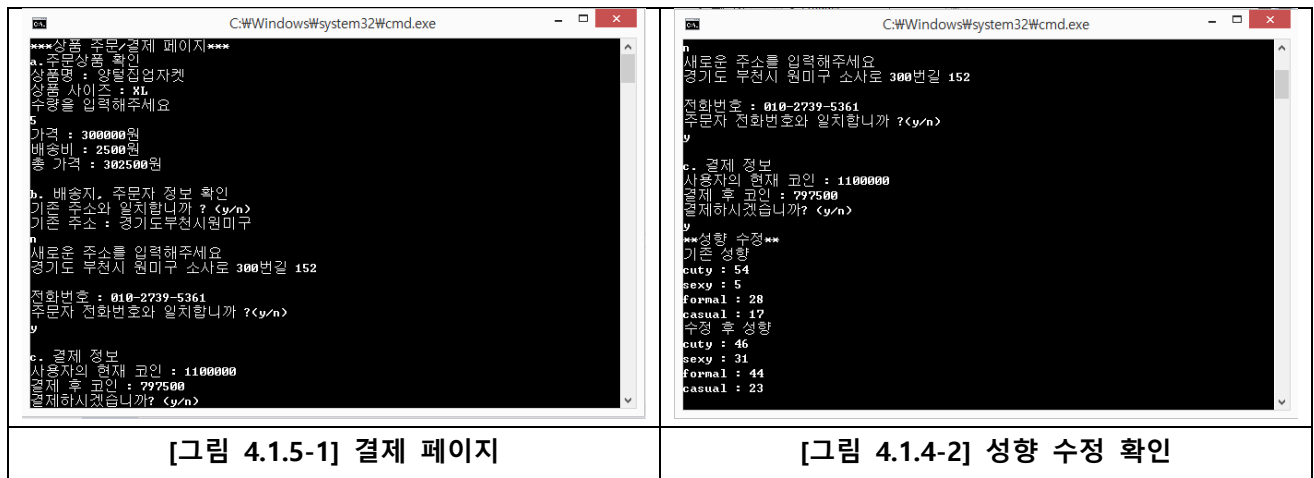
4.1.4. 상세 정보 / 장바구니

[그림 4.1.4-1] 상품 상세정보 페이지	[그림 4.1.4-2] 상품 상세정보 페이지(재고 0)

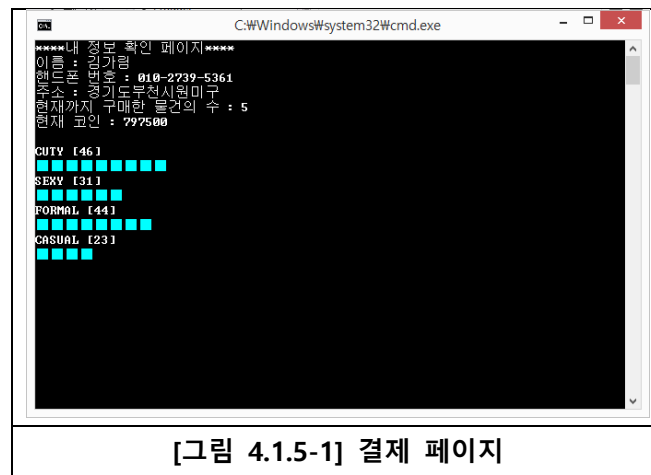
[그림 4.1.4-3] 장바구니 페이지	[그림 4.1.4-4] 장바구니 페이지

- 상품의 상세정보 페이지에서는 고객 특성 정보와 상품의 정보가 한눈에 보기 쉽게 그래프로 표현된다. (하늘색이 제품, 노란색이 고객 정보)
- 재고가 없을 시에는 결제가 불가능하다.
- 장바구니에 담은 상품은 이름과 가격, 사이즈를 확인할 수 있다. 일괄 주문이 가능하다.

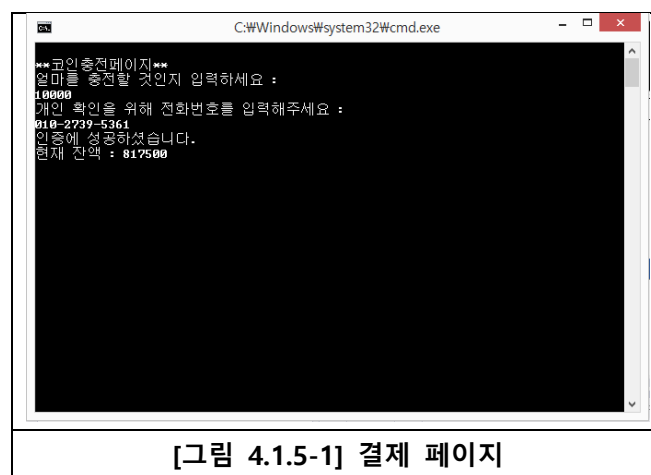
4.1.5. 주문/결제



4.1.6. 내 정보 확인

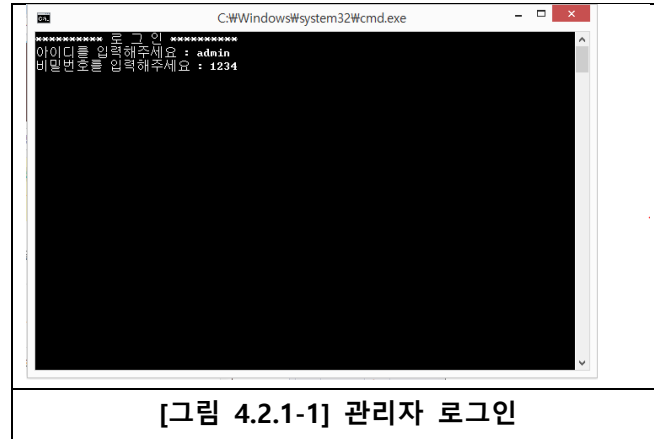


4.1.7. 코인 충전



4.2. 관리자

4.2.1. 로그인


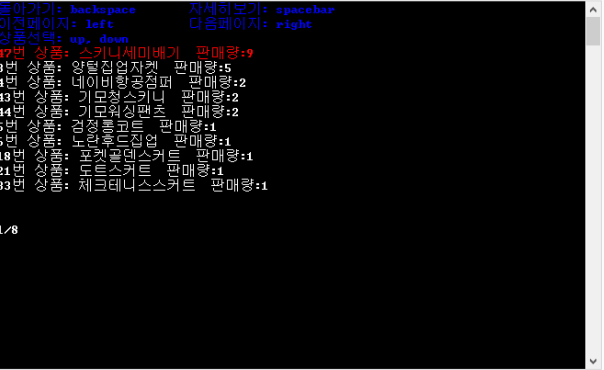
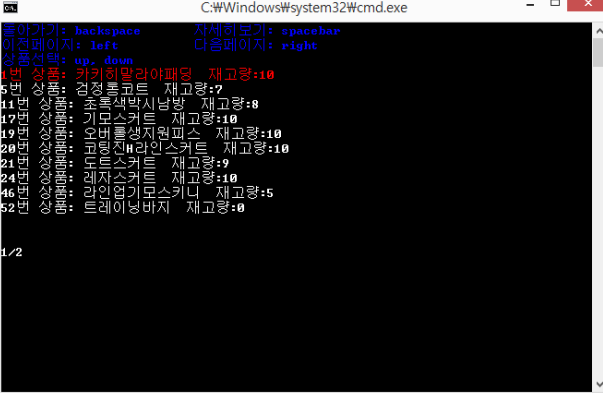
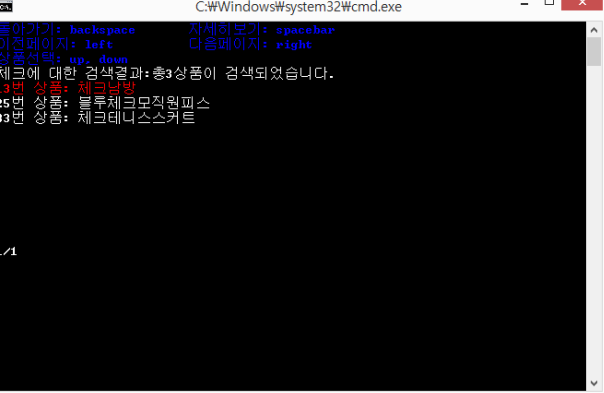


4.2.2. 주문서 확인



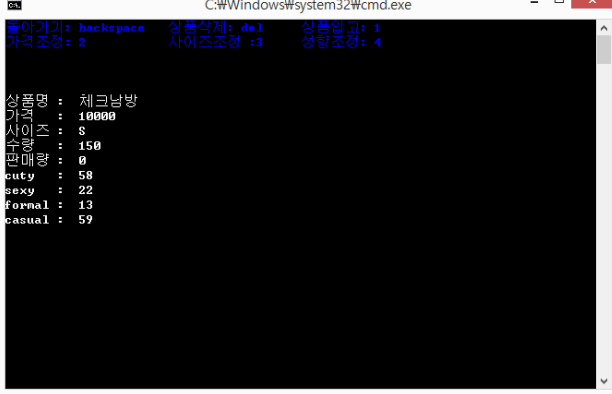
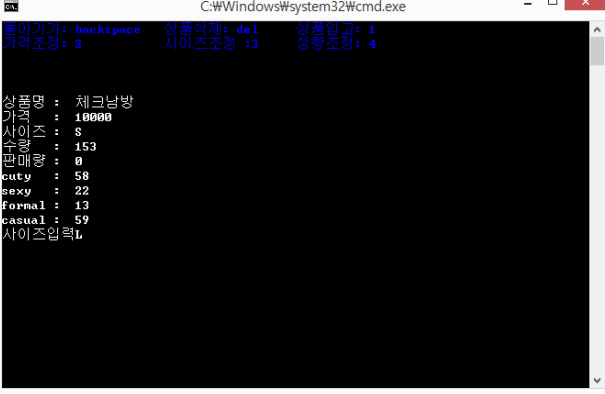
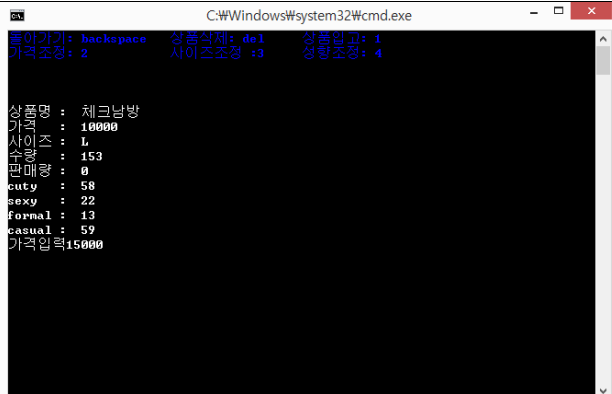
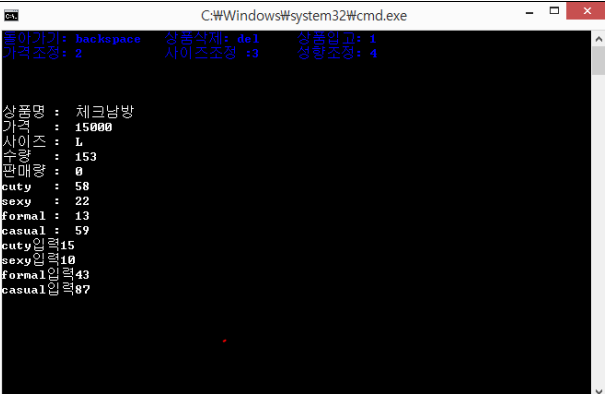
- 관리자는 주문서 확인 페이지에서 모든 주문서를 확인할 수 있다. 그 후 배송 여부를 수정할 수 있다.
- 그림 2에서 마지막 주문서의 배송 여부가 바뀐 것을 볼 수 있다.

4.2.3. 상품 목록 확인 / 상품 검색

	
<p>[그림 4.2.3-1] 상품 목록 페이지</p>	<p>[그림 4.2.3-2] 상품 목록 - 판매량 순으로 정렬</p>
	
<p>[그림 4.2.3-3] 재고량 부족 페이지</p>	<p>[그림 4.2.3-4] 상품 검색 페이지</p>

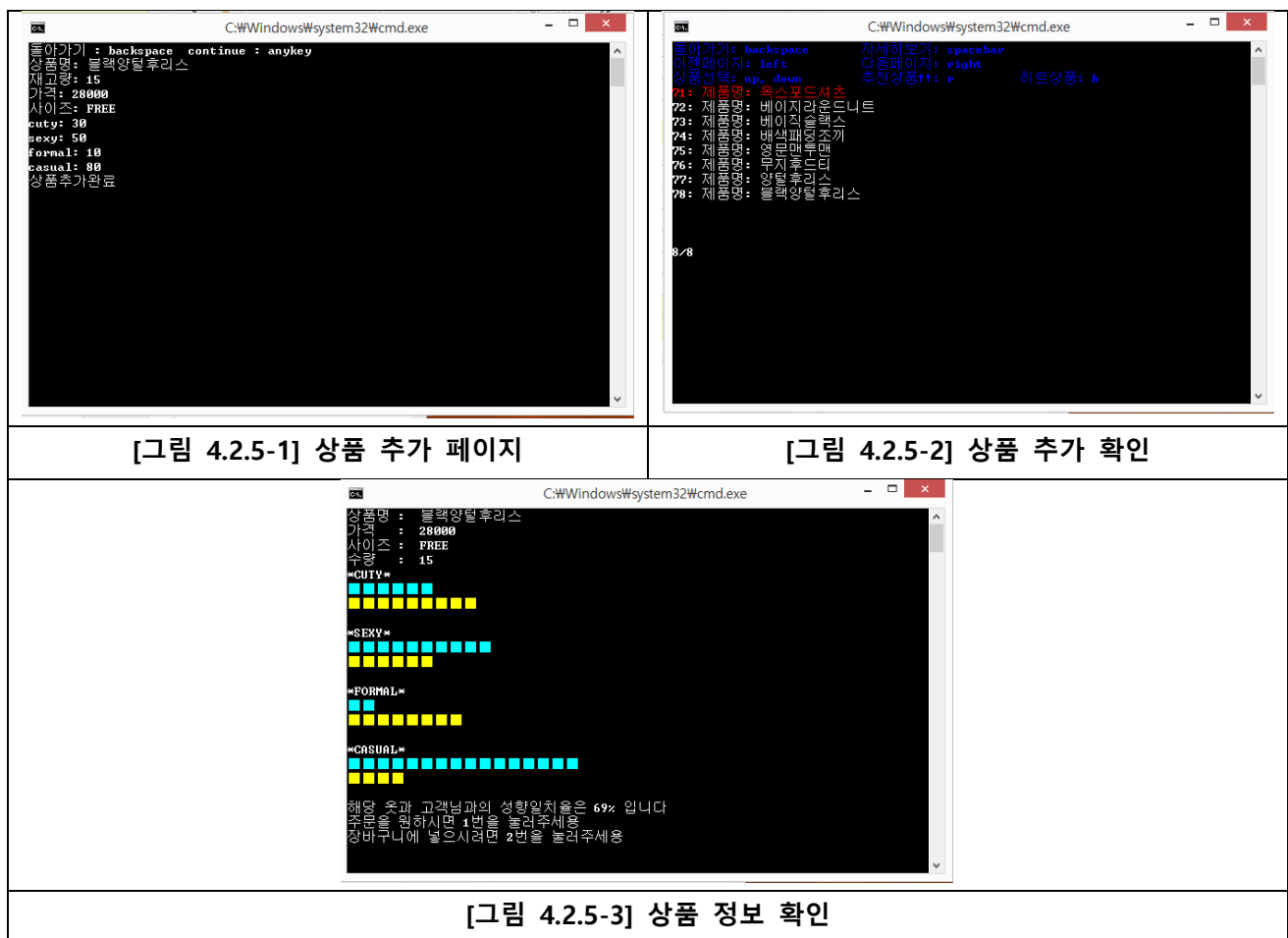
- 관리자도 사용자와 마찬가지로 상품의 목록을 확인할 수 있고, 상품 검색으로 상품을 찾을 수 있다.
- 사용자는 잘 팔리는 물건을 판매량 순으로 정렬해서 어떤 상품이 인기가 있는지 확인할 수 있다.
- 재고량이 10 이하인 상품들은 따로 페이지에 보여준다.

4.2.4. 상품 수정

	
<p>[그림 4.2.4-1] 상품 수정 페이지</p>	<p>[그림 4.2.4-2] 상품 수정 - 사이즈 수정</p>
	
<p>[그림 4.2.4-3] 상품 수정 - 가격 수정</p>	<p>[그림 4.2.4-4] 상품 수정 - 성향 수정</p>

- 관리자는 상세 페이지에서 상품의 정보를 수정할 수 있다.
- 수정 가능한 정보로는 가격, 사이즈, 재고량, 성향이 있다.
- 상품 삭제도 가능하다.

4.2.5. 상품 추가



- 관리자는 상품을 추가할 수 있는데, 추가시 상품 명, 재고량, 가격, 사이즈, 그리고 상품의 특성들을 입력받는다.

- 로그아웃 후 사용자로 로그인 했을 때 상품 목록의 마지막에 등록했던 상품이 추가되었음을 볼 수 있다.
- 그림 3에서처럼 사용자의 성향과 비교도 가능하다.

5. 사용자 매뉴얼

5.1. 텍스트 파일

프로그램과 같은 위치에 order.txt, product.txt, user.txt가 존재해야 한다. 또한 텍스트파일은 각각 다음과 같은 양식을 따른다. 각각의 구분은 한칸의 공백문자로 한다.

5.1.1. user.txt

id / 비밀번호 / 이름 / 전화번호 / 보유 코인 / 구매상품수 / feat 4개

5.1.2. product.txt

상품명 / 재고 / 가격 / 사이즈 / 판매량 / feat5 4개

5.1.3. order.txt

주문자 이름 / 전화번호 / 주소 / 상품명 / 사이즈 / 주문개수 / 가격 / 배송여부

5.2. 기본 매뉴얼

코드 작성 언어는 C++,

프로그램 실행시 로고가 출력된다. 아무 키나 누르면 다음으로 진행된다. 그 후 로그인, 회원가입 메뉴가 출력되고 키보드 상, 하키로 메뉴를 바꾸고 스페이스바로 진입한다.

5.2.1. 회원가입

아이디, 비밀번호, 이름, 전화번호, 주소를 입력하면 초기값 설정을 위한 설문조사가 실행된다. 맞으면 Y 보통은 S 아니면 N를 입력한다. 20문항에 대한 설문조사가 완료되면 사용자의 초기값이 출력되고 코인 10만원이 적립된다.

5.2.2. 로그인

아이디를 입력하고 비밀번호를 입력한다. 아이디가 admin(비밀번호 1234)일 경우에 관리자 메뉴로 진입하고 일반 사용자인 경우에는 회원가입시 저장한 아이디와 비밀번호를 입력한다.

5.2.3. 사용자

5.2.3.1. 상품 검색

원하는 상품명을 입력하면 해당 키워드가 들어간 상품이 모두 출력된다. 좌우 방향으로 페이지조작, 상하 방향으로 상품을 고를 수 있으며 스페이스바를 누르면 해당 상품의 상세정보를 볼 수 있다.

5.2.3.2. 상품 목록

모든 상품을 볼 수 있다. 키보드로 조작하며 스페이스바를 눌러 해당 상품의 상세 정보를 볼 수 있다. R을 입력하면 사용자 성향에 맞춰 추천된 상품들을 볼 수 있으며 h를 누르면 판매량이 많은 순으로 정렬된 상품을 볼 수 있다.

5.2.3.3. 내 정보 확인

내 정보를 볼 수 있다. 백스페이스를 누르면 메뉴로 돌아간다.

5.2.3.4. 코인 충전

충전할 금액과 전화번호를 입력하면 코인이 충전된다.

5.2.3.5. 장바구니

장바구니에 저장된 상품들을 볼 수 있다. 스페이스로 장바구니에 저장된 상품들을 주문할 수 있다.

5.2.3.6. 로그아웃

로그아웃 후 로그인 화면으로 돌아간다.

5.2.3.7. 상품 상세 정보 확인

상품을 선택하여 진입할 수 있고 사용자와 해당 상품의 성향을 비교하는 그래프를 출력하고 성향 일치율을 보여준다. 상품 주문 또는 장바구니에 넣는 것이 가능하다.

5.2.3.8. 상품 결제

상품 상세정보 또는 장바구니를 통해 진입할 수 있다. 현재 코인이 가격보다 적으면 구매할 수 없다. 재고가 구매할 양보다 적은 경우에도 구매할 수 없다. 주소, 연락처 등을 확인한 후 모두 완료되면 결제가 완료되어 주문서가 작성되고 코인이 차감된다.

5.2.4. 관리자

5.2.4.1. 주문서 확인

주문서들을 불러온다. y/n을 입력하여 주문서 배송여부를 수정할 수 있다.

5.2.4.2. 상품 목록

사용자와 마찬가지로 상품목록을 출력해준다. 키보드 조작이 가능하며 h를 입력하면 판매량순으로 보기, s를 입력하여 재고량이 10 이하인 상품 보기를 지원한다.

5.2.4.3. 상품 검색

원하는 상품명을 입력하면 해당 키워드가 들어간 상품이 모두 출력된다. 좌우 방향으로 페이지조작, 상하 방향으로 상품을 고를 수 있으며 스페이스바를 누르면 해당 상품의 상세정보를 볼 수 있다.

5.2.4.4. 상품 추가

상품명, 수량, 가격, 사이즈, 상품 성향을 입력하면 해당 상품이

추가된다.

5.2.4.5. 상품 상세정보

상품을 선택하여 진입할 수 있다. 현재 상품의 정보를 보여주고 상품 삭제, 상품 입고, 가격 조정, 사이즈 조정, 성향 조정이 가능하다.

6. 프로젝트 마무리

6.1. 평가 및 느낀 점

김가림: 처음 팀을 꾸리고 주제를 생각했을 때부터 현재 프로젝트를 마무리 할 때까지 처음 계획에서 일정이나 방향 등의 부분에서 크게 달라진 것이 없었다. 한 학기동안 하나의 주제를 가지고 프로젝트를 이끌어 나가면서 이렇게 순조롭게 일정이 진행된 것은

처음이었던 것 같다. 기존에는 프로그래밍 과제이던, 다른 과제이든 일단 주제를 정하면 어떻게든 결과를 내려고부터 했었다.

하지만 주니어 디자인 프로젝트 과목에서는 프로젝트를 하나하나 설계하는 법을 배웠던 것 같다. 초반에는 예전과 비슷하게 주제를 정하고 바로 코드를 짜려고 했었지만, 더 설계가 필요하다는 교수님의 말씀을 듣고 다시 처음부터 기능 하나하나 완벽하게 설계 한 후 코드를 구현하였다. 코드를 작성하는 상황에서 일어날 수 있는 예외나 돌발상황은 미리 설계시 생각을 하고 예외처리를 하는 방법을 만들어 놔있기 때문에 어려움 없이 빠르게 작성할 수 있었다.

팀원과의 호흡도 좋았던 것 같다. 나는 주로 프로그램의 전체적인 주제와 아이디어들을 생각해 냈고, 그것을 자세히 구현하는 코드적 아이디어는 다른 팀원이 잘 생각해내는 등의 서로의 강점을 보완할 수 있는 팀이었다.

윤재문: 처음에 프로그램의 전체구성을 설계할 때 프로그램에서 객체와 객체와의 관계를 우선적으로 고려해서 설계하였다. 실제로도 Start에서 User와 Admin중 한 클래스를 선택하여 사용하고 이후 두 클래스들이 Product를 사용하도록 설계하였는데 헤더파일을 먼저 작성한 뒤 코드를 구체화 시키는 방법으로 프로젝트를 진행해서 처음 설계와 크게 벗어나지 않게 프로젝트를 완료한 것 같다.

처음의 설계에서는 기능을 구현하는 것을 목표로 하였는데 생각보다 프로젝트의 기본기능구현이 빠르게 진행되어서 기존설계의 문제점을 분석해보고 이를 보완하는 새로운 설계를 해볼 시간적 여유가 있었다, 이를 통해서 기존의 설계를 키보드로 구현 및 사용자, 관리자에 맞는 메뉴제공, 장바구니기능 등 추가기능을 구현하여서 더 프로그램이 다양한 기능을 포함할 수 있게 된 것 같다.

이번 프로젝트를 되돌아보면 프로젝트 내용 외적으로도 역할분담 및 팀워크 부분에서도 매우 만족스러웠던 것 같다. 각자의 강점을 살려서 프로그램 설계부분, ppt제작, 보고서작성 등을 나누어 담당해서 효율적이었고, 전체 코드에서 사용자와 관리자로 나누어 각자 구현하기로 했던 부분도 서로 더 좋은 점이 있으면 보완해주고 새로운 아이디어를 제공하는 등 서로에게 도움이 많이 되었던 것 같다. 이외에 아이디어 회의 시에도 팀원이 두명이라 다른 팀들 보다 더 자주, 쉽게 만날 수가 있었고, 회의에서도 편하게 서로의 생각을 말해보고 좋은 아이디어를 프로그램설계에 반영시키는 부분에서도 많은 시간이 걸리지 않고 바로 결정을 내릴 수 있어 편했던 것 같다.

7. 전체 소스 코드

7.1. 헤더파일

```
#include <iostream>
#include <conio.h>
#include <string>
#include <vector>
#include <fstream>
#include <Windows.h>
#include <sstream>
#include <time.h>

using namespace std;
typedef struct UserInfo
{
    string id;
    string password;
    string name;
    string phone;
    string address;
    int coin;
    int productCount;
    int feat1_cuty;
    int feat2_sexy;
    int feat3_formal;
    int feat4_casual;
}UserInfo;
typedef struct ProductInfo
{
    string name;
    int stock; //재고
    int cost;
    string size;
    int sellNumber; //팔린양
    int feat1_cuty;
    int feat2_sexy;
    int feat3_formal;
    int feat4_casual;
}ProductInfo;
typedef struct Order
{
    string orderDate;
    string userName;
    string userPhone;
    string userAddress;
    /**받는사람의 이름, 전화번호 ,주소**/
    string productName;
    string productSize;
```

```

    int productNum;        //상품 몇개 주문하는지
    /** 상품 정보**/
    int totalPrice;        //배송비 포함 얼마인지
    /**현재 물건**/
    bool isDelivered;      //배송 여부
}Order;

vector <UserInfo> vec_user;
vector <ProductInfo> vec_product;
vector <Order> vec_order;
//장바구니기능(15.12.03)
vector <ProductInfo> vec_shoppingCart;

class User
{
private:
    string id;
    string password;
    string name;
    string phone;
    string address;
    int coin;
    int feat1_cuty;
    int feat2_sexy;
    int feat3_formal;
    int feat4_casual;
    int productCount; //지금까지 물품을 구입한 수량
    int userIndex;

public:
    User(){};
    User(int index);
    void searchProduct();
    void menuProduct();
    void myInfo();
    void showMyFeat();
    void coinCharge();
    bool logOut();
    void shoppingCartPayment();
    void recommendProduct();
    int calculationFeat(int productIndex);
    //장바구니기능(15.12.03)
    void showShoppingCart();
    void replaceFeat(int userIndex, int buyNum);
    void showSellNumber();
};

class Admin
{
public:
    void getOrder();

```

```

        void searchProduct();
        void menuProduct(); ///관리자도 상품 검색/목록 확인 가능하게
        void productEdit(int index);
        void addProduct();
        bool logOut();
        void showSellNumber();
        void showStock();
};

class Product
{
private:
    string name;
    int stock; ///재고
    int cost;
    string size;
    int sellNumber; ///팔린양
    int feat1_cuty;
    int feat2_sexy;
    int feat3_formal;
    int feat4_casual;
public:
    Product(ProductInfo product);
    void detailInfo(); ///관리자
    void detailInfo(int userIndex); ///이용자
    void payment(int userIndex);
    void showProductFeat(int userIndex);
    void replaceFeat(int userIndex, int buyNum);
    ///((15.12.03)장바구니기능추가
    void inputShoppingCart();
};

class Start
{
private:
    bool isAdmin;
public:
    Start();
    int login();
    void signUp();
    void menu(int userIndex);
    int isUserID(string _id); ///아이디가 존재하는지 확인 있다면 index를 return
    User currentUser;
    Admin currentAdmin;
    void signUpQuestion(int &cuty, int&sexy, int&formal, int&casual);
};

void setColor(string color);
void gotoxy(int x, int y);
string itos(int num);

```

7.2. C++파일

```
/*
주니어 디자인 프로젝트
2015.10.13~
the Online Shopping Mall Simulator
김가림2014112068
윤재문2014112037
*/
#include "header.h"
string itos(int num)
{
    //int->string
    ostringstream oss;
    oss << num;
    return oss.str();
}
void setColor(string color)
{
    // 윤재문(15.11.29) - 글씨색을 조정가능하게 하는 함수
    if (color == "red")
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 12);
    else if (color == "blue")
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 9);
    else
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
}
void gotoxy(int x, int y) //function that set position of console window
{
    COORD Pos = { x - 1, y - 1 };
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
}
void showStartingScreen()
{
    /*****
    - 김가림 (15.11.26)
    - Function : showStartingScreen
    - Input : None
    - Output : None
    - Procedure : 프로그램 시작시 실행
    - 수정 날짜 :
    *****/
    cout <<
    " | _____ | " << endl;
    cout << " | THE
    | " << endl;
    cout << " | ■■■ ■ ■ ■■■ ■■■ ■■■ ■■■ ■■■ ■■■
    | " << endl;
```



```

        out1 << s;

        else

            out1 << s << endl;

    }
    out1.close();
    ofstream out2("product.txt");
    for (int i = 0; i<vec_product.size(); i++)
    {
        string s;
        s = vec_product.at(i).name + " " + itos(vec_product.at(i).stock) + " " +
        itos(vec_product.at(i).cost) + " " + vec_product.at(i).size + " " +
        itos(vec_product.at(i).sellNumber);
        s += " " + itos(vec_product.at(i).feat1_cuty) + " " +
        itos(vec_product.at(i).feat2_sexy) + " " + itos(vec_product.at(i).feat3_formal) + " " +
        itos(vec_product.at(i).feat4_casual);
        if (i == vec_product.size() - 1)
            out2 << s;

        else

            out2 << s << endl;

    }
    out2.close();
    ofstream out3("order.txt");
    for (int i = 0; i<vec_order.size(); i++)
    {
        string s;
        s = vec_order.at(i).userName + " " + vec_order.at(i).userPhone + " " +
        vec_order.at(i).userAddress + " " + vec_order.at(i).productName + " " +
        vec_order.at(i).productSize + " " + itos(vec_order.at(i).productNum) + " " +
        itos(vec_order.at(i).totalPrice) + " " + itos(vec_order.at(i).isDelivered);
        if (i == vec_order.size() - 1)
            out3 << s;

        else

            out3 << s << endl;

    }
    out3.close();
    Sleep(150);
}
Start::Start()
{
    /*****
    - 윤재문(15.11.14)
    - Function   : Start()
    - Input      : None
    - Output     : None
    - Procedure  : Start구조체의 기본생성자
    - 수정 날짜  : 윤재문(15.11.22)
    - productInfo 텍스트 벡터에 저장하는부분 추가
    - login() 리턴타입 void->int , menu()->menu(int )수정
    윤재문(15.11.28)
    *****/
}

```



```

-productCount 추가
윤재문(15.11.30)
-order 불러오는 부분 추가
윤재문(15.12.05)
-키보드 조작가능하도록 수정
-로그아웃 가능하도록 변경
*****/
ifstream in1("user.txt");
while (!in1.eof()){
    string s;
    getline(in1, s);
    struct UserInfo temp;
    temp.id = s.substr(0, s.find(" "));
    s = s.substr(s.find(" ") + 1);
    temp.password = s.substr(0, s.find(" "));
    s = s.substr(s.find(" ") + 1);
    temp.name = s.substr(0, s.find(" "));
    s = s.substr(s.find(" ") + 1);
    temp.phone = s.substr(0, s.find(" "));
    s = s.substr(s.find(" ") + 1);
    temp.coin = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.productCount = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.feat1_cutty = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.feat2_sexy = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.feat3_formal = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.feat4_casual = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.address = s;
    vec_user.push_back(temp);
}
in1.close();

ifstream in2("product.txt");
while (!in2.eof()){
    string s;
    getline(in2, s);
    struct ProductInfo temp;
    temp.name = s.substr(0, s.find(" "));
    s = s.substr(s.find(" ") + 1);
    temp.stock = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.cost = stoi(s.substr(0, s.find(" ")));
    s = s.substr(s.find(" ") + 1);
    temp.size = s.substr(0, s.find(" "));
}

```

```

        s = s.substr(s.find(" ") + 1);
        temp.sellNumber = stoi(s.substr(0, s.find(" ")));
        s = s.substr(s.find(" ") + 1);
        temp.feat1_cuty = stoi(s.substr(0, s.find(" ")));
        s = s.substr(s.find(" ") + 1);
        temp.feat2_sexy = stoi(s.substr(0, s.find(" ")));
        s = s.substr(s.find(" ") + 1);
        temp.feat3_formal = stoi(s.substr(0, s.find(" ")));
        s = s.substr(s.find(" ") + 1);
        temp.feat4_casual = stoi(s);
        vec_product.push_back(temp);
    }
    in2.close();

    ifstream in3("order.txt");
    while (!in3.eof()){
        string s;
        getline(in3, s);
        struct Order temp;
        temp.userName = s.substr(0, s.find(" "));
        s = s.substr(s.find(" ") + 1);
        temp.userPhone = s.substr(0, s.find(" "));
        s = s.substr(s.find(" ") + 1);
        temp.userAddress = s.substr(0, s.find(" "));
        s = s.substr(s.find(" ") + 1);
        temp.productName = s.substr(0, s.find(" "));
        s = s.substr(s.find(" ") + 1);
        temp.productSize = s.substr(0, s.find(" "));
        s = s.substr(s.find(" ") + 1);
        temp.productNum = stoi(s.substr(0, s.find(" ")));
        s = s.substr(s.find(" ") + 1);
        temp.totalPrice = stoi(s.substr(0, s.find(" ")));
        s = s.substr(s.find(" ") + 1);
        temp.isDelivered = stoi(s);
        vec_order.push_back(temp);
    }
    in3.close();
    int point = 0;
    while (1)
    {
        setColor("white");
        gotoxy(30, 8);
        char key;
        if (point == 0)
        {
            setColor("red");
            cout << "로그인 <<" << endl;
            setColor("white");
            gotoxy(30, 10);

```

```

        cout << "회원가입" << endl;
    }
    else if (point == 1)
    {
        setColor("white");
        cout << "로그인" << endl;
        setColor("red");
        gotoxy(30, 10);
        cout << "회원가입<<" << endl;
        setColor("white");
    }
    key = getch();
    if (key == 72 && point != 0)
    {
        point--;
    }
    else if (key == 80 && point != 1)
    {
        point++;
    }
    else if (key == 32)
    {
        if (point == 0)
        {
            system("cls");
            int userIndex;
            userIndex = login();
            menu(userIndex);
        }
        else if (point == 1)
        {
            system("cls");
            signUp();
        }
    }
}
//saveData();
}
int Start::login()
{
    /*****
    - 김가림(15.11.15)
    - Function : login()
    - Input : None
    - Output : None
    - Procedure : 벡터의 처음부터 끝까지
    - 수정 날짜 : 윤재문(15.11.22) - 리턴타입 void-> int로 변경하여 userIndex를
반환하도록 수정
    김가림(15.11.28) - 아이디가 admin일 때 관리자 로그인
    *****/

```

```

*****/

string tempID;
string tempPW;
int userIndex;
cout << "***** 로 그 인 *****" << endl;
cout << "아이디를 입력해주세요 : ";
cin >> tempID;
if (tempID == "admin") //관리자 아이디
{
    cout << "비밀번호를 입력해주세요 : ";
    cin >> tempPW;
    if (tempPW == "1234")
        //관리자 비밀번호
        isAdmin = true;
    return -1;
}
else{
    while ((userIndex = isUserID(tempID)) == -1)
    {
        cout << "아이디가 정확하지 않습니다. 다시 입력해주세요 : ";
        cin >> tempID;
    }
    cout << "비밀번호를 입력해주세요 : ";
    cin >> tempPW;
    while (vec_user[userIndex].password != tempPW)
    {
        cout << "비밀번호가 정확하지 않습니다. 다시 입력해주세요 : ";
        cin >> tempPW;
    }
    cout << "***로그인 성공***" << endl;
    Sleep(1500);
    isAdmin = false;
    return userIndex;
}
}

void Start::signUp()
{
    /*****
    - 김가림(15.11.15)
    - Function : signUp()
    - Input : None
    - Output : None
    - Procedure : 새로운 회원의 정보를 받아와서 vector에 저장
    - 수정 날짜 : (15.12.01) 중복된 아이디 수정
    *****/

    UserInfo newUser;
    string tempID;

```

```

cout << "***** 회 원 가 입 *****" << endl;
cout << "아이디를 입력하세요 : ";
cin >> tempID;
while (tempID == "admin")
{
    cout << "이 아이디는 사용할 수 없습니다." << endl;
    cout << "아이디를 입력하세요 : ";
    cin >> tempID;
}
while (isUserID(tempID) != -1)    //아이디가 있을때 다시 입력받는다
{
    cout << "중복된 아이디입니다. 새로운 아이디를 입력하세요 : ";
    cin >> tempID;
}
newUser.id = tempID;
cout << "비밀번호를 입력하세요 : ";
cin >> newUser.password;

cout << "이름을 입력하세요 : ";
cin >> newUser.name;

cout << "전화번호를 입력하세요 : ";
cin >> newUser.phone;

cout << "주소를 입력하세요 : ";
cin >> newUser.address;

signUpQuestion(newUser.feats1_cut, newUser.feats2_sexy, newUser.feats3_formal,
newUser.feats4_casual);

cout << "cut : " << newUser.feats1_cut << endl;
cout << "sexy:" << newUser.feats2_sexy << endl;
cout << "formal : " << newUser.feats3_formal << endl;
cout << "casual: " << newUser.feats4_casual << endl;

cout << "**회원가입 축하 코인 10만원을 적립해드립니다**" << endl;
newUser.coin = 100000;    //회원가입시 coin 10만원 적립
newUser.productCount = 0;    //productCount를 0으로 초기화
vec_user.push_back(newUser);    //새로운 user를 벡터에 저장
}
void Start::menu(int userIndex)
{
    /*****
    - 윤재문,김가림(15.11.15)
    - Function : menu()
    - Input : int userIndex
    - Output : None
    *****/
}

```

- Procedure : 메뉴창을 통해 원하는 기능으로 진입
 - 수정 날짜 : 윤재문(15.11.22) - searchProduct(), menuProduct() 구현
 김가림(15.12.05) - 장바구니 기능 구현
 윤재문(15.12.05) - 로그아웃 기능 구현, switch->키보드 조작하도록 수정
 *****/

```
int inputKey;
bool logOut = false;
int point = 0;
system("cls");
while (!isAdmin && logOut == false) // 관리자가 아닐 때
{
    gotoxy(1, 1);
    currentUser = User(userIndex);
    char key;
    if (point == 0)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 상품 검색    <<" << endl;
        cout << "2. 상품 목록    " << endl;
        cout << "3. 내정보 확인  " << endl;
        cout << "4. 코인 충전    " << endl;
        cout << "5. 장바구니     " << endl;
        cout << "6. 로그아웃     " << endl;
    }
    else if (point == 1)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 상품 검색    " << endl;
        cout << "2. 상품 목록    <<" << endl;
        cout << "3. 내정보 확인  " << endl;
        cout << "4. 코인 충전    " << endl;
        cout << "5. 장바구니     " << endl;
        cout << "6. 로그아웃     " << endl;
    }
    else if (point == 2)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 상품 검색    " << endl;
        cout << "2. 상품 목록    " << endl;
        cout << "3. 내정보 확인  <<" << endl;
        cout << "4. 코인 충전    " << endl;
        cout << "5. 장바구니     " << endl;
        cout << "6. 로그아웃     " << endl;
    }
    else if (point == 3)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 상품 검색    " << endl;
    }
}
```

```

        cout << "2. 상품 목록      " << endl;
        cout << "3. 내정보 확인      " << endl;
        cout << "4. 코인 충전      <<" << endl;
        cout << "5. 장바구니      " << endl;
        cout << "6. 로그아웃      " << endl;
    }
    else if (point == 4)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 상품 검색      " << endl;
        cout << "2. 상품 목록      " << endl;
        cout << "3. 내정보 확인      " << endl;
        cout << "4. 코인 충전      " << endl;
        cout << "5. 장바구니      <<" << endl;
        cout << "6. 로그아웃      " << endl;
    }
    else if (point == 5)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 상품 검색      " << endl;
        cout << "2. 상품 목록      " << endl;
        cout << "3. 내정보 확인      " << endl;
        cout << "4. 코인 충전      " << endl;
        cout << "5. 장바구니      " << endl;
        cout << "6. 로그아웃      <<" << endl;
    }
    key = getch();
    if (key == 72 && point != 0)
        point--;
    else if (key == 80 && point != 5)
    {
        point++;
    }
    else if (key == 32)
    {
        system("cls");
        if (point == 0)
        {
            currentUser.searchProduct();
        }
        else if (point == 1)
        {
            currentUser.menuProduct();
        }
        else if (point == 2)
        {
            currentUser.myInfo();
        }
        else if (point == 3)

```

```

        {
            currentUser.coinCharge();
        }
        else if (point == 4)
        {
            currentUser.showShoppingCart();
        }
        else if (point == 5)
        {
            logOut = currentUser.logOut();
        }
    }
}

while (isAdmin && logOut == false)
{
    system("cls");
    gotoxy(1, 1);
    char key;
    if (point == 0)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 주문서확인    <<" << endl;
        cout << "2. 상품 목록      " << endl;
        cout << "3. 상품 검색      " << endl;
        cout << "4. 상품 추가      " << endl;
        cout << "5. 로그 아웃      " << endl;
    }
    else if (point == 1)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 주문서확인    " << endl;
        cout << "2. 상품 목록    <<" << endl;
        cout << "3. 상품 검색    " << endl;
        cout << "4. 상품 추가    " << endl;
        cout << "5. 로그 아웃    " << endl;
    }
    else if (point == 2)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 주문서확인    " << endl;
        cout << "2. 상품 목록    " << endl;
        cout << "3. 상품 검색    <<" << endl;
        cout << "4. 상품 추가    " << endl;
        cout << "5. 로그 아웃    " << endl;
    }
    else if (point == 3)
    {
        cout << "***** 메뉴 *****" << endl;
    }
}

```



```

        cout << "1. 주문서확인" << endl;
        cout << "2. 상품 목록" << endl;
        cout << "3. 상품 검색" << endl;
        cout << "4. 상품 추가" << endl;
        cout << "5. 로그 아웃" << endl;
    }
    else if (point == 4)
    {
        cout << "***** 메뉴 *****" << endl;
        cout << "1. 주문서확인" << endl;
        cout << "2. 상품 목록" << endl;
        cout << "3. 상품 검색" << endl;
        cout << "4. 상품 추가" << endl;
        cout << "5. 로그 아웃" << endl;
    }
    key = getch();
    if (key == 72 && point != 0)
        point--;
    else if (key == 80 && point != 4)
    {
        point++;
    }
    else if (key == 32)
    {
        system("cls");
        if (point == 0)
        {
            currentAdmin.getOrder();
        }
        else if (point == 1)
        {
            currentAdmin.menuProduct();
        }
        else if (point == 2)
        {
            currentAdmin.searchProduct();
        }
        else if (point == 3)
        {
            currentAdmin.addProduct();
        }
        else if (point == 4)
        {
            logOut = currentAdmin.logOut();
        }
    }
}

}

int Start::isUserID(string _id)

```

```

{
    /*****
    - 김가림(15.11.15)
    - Function : isUserID(string _id)
    - Input : 확인할 id
    - Output : 아이디가 없다면 0, 있다면 벡터 내의 인덱스 return
    - Procedure : 벡터에 회원의 아이디가 존재하는지 확인
    - 수정 날짜 : 아이디가 없을때 0이 아니라 -1을 리턴하도록 수정
    *****/

    int flag = -1;
    for (int i = 0; i < vec_user.size(); i++)
    {
        string tempString = vec_user.at(i).id;
        if (_id.compare(tempString) == 0) //같은 값이면 if문 실행
        {
            flag = i;
        }
    }
    return flag; //없으면 -1 리턴
}

void Start::signUpQuestion(int &sexy, int &cuty, int &formal, int &casual)
{
    /*****
    - 김가림(15.11.25)
    - Function : signUpQuestion()
    - Input : none
    - Output : None
    - Procedure : 설문조사를 통해 회원 정보의 초기값을 받는다.
    - 수정 날짜 : 김가림(15.12.13) 입력받을 때에 예외처리
    *****/

    sexy = 0;
    cuty = 0;
    formal = 0;
    casual = 0;
    system("cls");
    char inputChar;
    cin.sync();

    cout << "**회원 정보 초기값 입력을 위한 설문조사**" << endl;
    cout << "맞으면 Y, 보통은 S, 아니면 N를 입력하세요." << endl;
    cout << "1. 나는 밝은 옷을 좋아한다." << endl;
    cin >> inputChar;
    while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
        (inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
    {
        cout << "올바른 값을 입력해주세요." << endl;
        cin >> inputChar;
    }
}

```

```

    }
    switch (inputChar){
    case('y') :
    case('Y') :
        cuty += 3;
        casual += 1;
        break;
    case('s') :
    case('S') :
        cuty += 1;
        break;
    case('n') :
    case('N') :
        sexy += 3; formal += 1;
        break;
    default:
        break;
    }

    cout << "2. 나는 노출에 대해 거부감이 없다." << endl;
    cin >> inputChar;
    while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
    {
        cout << "올바른 값을 입력해주세요." << endl;
        cin >> inputChar;
    }
    switch (inputChar){
    case('y') :
    case('Y') :
        sexy += 3;
        break;
    case('s') :
    case('S') :
        sexy += 1; casual += 1;
        break;
    case('n') :
    case('N') :
        formal += 5; sexy += 5;
        break;
    default:
        break;
    }

    cout << "3. 나는 캐릭터가 그려진 옷을 좋아한다." << endl;
    cin >> inputChar;
    while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
    {

```

```

        cout << "올바른 값을 입력해주세요." << endl;
        cin >> inputChar;
    }
    switch (inputChar){
    case('y') :
    case('Y') :
        cuty += 5;
        break;
    case('s') :
    case('S') :
        cuty += 3; casual += 3;
        break;
    case('n') :
    case('N') :
        formal += 5; sexy += 3;
        break;
    default:
        break;
    }

    cout << "4. 나는 격식을 갖춰서 옷을 입는다." << endl;
    cin >> inputChar;
    while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
    {
        cout << "올바른 값을 입력해주세요." << endl;
        cin >> inputChar;
    }
    switch (inputChar){
    case('y') :
    case('Y') :
        formal += 5;
        break;
    case('s') :
    case('S') :
        formal += 3;
        break;
    case('n') :
    case('N') :
        casual += 5; sexy += 3;
        break;
    default:
        break;
    }

    cout << "5. 나는 피어싱을 했다.(할 생각이 있다)" << endl;
    cin >> inputChar;
    while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))

```

```

{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    sexy += 3; casual += 5;
    break;
case('s') :
case('S') :
    break;

case('n') :
case('N') :
    formal += 5;
    break;
default:
    break;
}

cout << "6. 나는 핑크색을 좋아한다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    cuty += 5;
    break;
case('s') :
case('S') :
    cuty += 3;
    break;
case('n') :
case('N') :
    sexy += 5;
    break;
default:
    break;
}

cout << "7. 나는 스니커즈를 좋아한다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))

```

```

{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    casual += 5;
    break;
case('s') :
case('S') :
    casual += 3;
    break;
case('n') :
case('N') :
    formal += 5; sexy += 3;
    break;
default:
    break;
}

cout << "8. 나는 스트라이프 티를 좋아한다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    casual += 5; formal += 3;
    break;
case('s') :
case('S') :
    casual += 3;
    break;
case('n') :
case('N') :
    sexy += 5;
    break;
default:
    break;
}

cout << "9. 나는 검정색을 좋아한다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&

```

```

(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    sexy += 3;
    break;
case('s') :
case('S') :
    sexy += 1;
    break;
case('n') :
case('N') :
    cuty += 5;
    break;
default:
    break;
}

cout << "10. 나는 스냅백을 좋아한다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    casual += 5;
    break;
case('s') :
case('S') :
    casual += 3;
    break;
case('n') :
case('N') :
    sexy += 3; formal += 3;
    break;
default:
    break;
}

cout << "11. 나는 평소에 귀엽다는 말을 들어본 적 있다." << endl;
cin >> inputChar;

```

```

while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    cuty += 7;
    break;
case('s') :
case('S') :
    cuty += 5;
    break;
case('n') :
case('N') :
    sexy += 5; formal += 5;
    break;
default:
    break;
}

cout << "12. 나는 풀세팅하는 것이 귀찮지 않다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    sexy += 5; formal += 5;
    break;
case('s') :
case('S') :
    formal += 3;
    break;
case('n') :
case('N') :
    casual += 5;
    break;
default:
    break;
}

cout << "13. 평소에 향수 뿌리는 것을 좋아한다." << endl;

```



```

cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    sexy += 3;
    break;
case('s') :
case('S') :
    sexy += 1;
    break;
case('n') :
case('N') :
    casual += 3;
    break;
default:
    break;
}

cout << "14. 나는 타투를 했다.(할 생각이 있다.)" << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    sexy += 5;
    break;
case('s') :
case('S') :
    sexy += 3; casual += 3;
    break;
case('n') :
case('N') :
    cuty += 5; formal += 3;
    break;
default:
    break;
}

```

```

cout << "15. 나는 셔츠를 즐겨 입는다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    formal += 5;
    break;
case('s') :
case('S') :
    formal += 3;
    break;
case('n') :
case('N') :
    casual += 5;
    break;
default:
    break;
}

cout << "16. 나는 패딩보다 코트를 입는다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    formal += 3; sexy += 1;
    break;
case('s') :
case('S') :
    formal += 1;
    break;
case('n') :
case('N') :
    casual += 5;
    break;
default:
    break;
}

```

```

cout << "17. 나는 레이스달린 옷을 좋아한다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    cuty += 5;
    break;
case('s') :
case('S') :
    cuty += 3;
    break;
case('n') :
case('N') :
    sexy += 3; casual += 3;
    break;
default:
    break;
}

cout << "18. 나는 멜빵을 즐겨 입는다." << endl;
cin >> inputChar;
while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
{
    cout << "올바른 값을 입력해주세요." << endl;
    cin >> inputChar;
}
switch (inputChar){
case('y') :
case('Y') :
    cuty += 5; casual += 3;
    break;
case('s') :
case('S') :
    casual += 1;
    break;
case('n') :
case('N') :
    sexy += 5;
    break;
default:
    break;
}

```

```

    }

    cout << "19. 나는 청바지를 자주 입는다." << endl;
    cin >> inputChar;
    while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
    {
        cout << "올바른 값을 입력해주세요." << endl;
        cin >> inputChar;
    }
    switch (inputChar){
    case('y') :
    case('Y') :
        casual += 5;
        break;
    case('s') :
    case('S') :
        casual += 3;
        break;
    case('n') :
    case('N') :
        sexy += 5; formal += 5;
        break;
    default:
        break;
    }

    cout << "20. 나는 인상이 강하다." << endl;
    cin >> inputChar;
    while ((inputChar != 'y') && (inputChar != 'Y') && (inputChar != 's') &&
(inputChar != 'S') && (inputChar != 'n') && (inputChar != 'N'))
    {
        cout << "올바른 값을 입력해주세요." << endl;
        cin >> inputChar;
    }
    switch (inputChar){
    case('y') :
    case('Y') :
        sexy += 5; formal += 5;
        break;
    case('s') :
    case('S') :
        sexy += 3;
        break;
    case('n') :
    case('N') :
        cuty += 5;
        break;
    default:

```

```

        break;
    }

    system("cls");
}
User::User(int index)
{
    /*****
    - 김가림
    - Function   : User
    - Input      : int index
    - Output     : None
    - Procedure  : User 클래스의 생성자
    - 수정 날짜   :
    *****/

    userIndex = index;
    id = vec_user.at(index).id;
    password = vec_user.at(index).password;
    name = vec_user.at(index).name;
    phone = vec_user.at(index).phone;
    coin = vec_user.at(index).coin;
    address = vec_user.at(index).address;
    feat1_cuty = vec_user.at(index).feat1_cuty;
    feat2_sexy = vec_user.at(index).feat2_sexy;
    feat3_formal = vec_user.at(index).feat3_formal;
    feat4_casual = vec_user.at(index).feat4_casual;
    productCount = vec_user.at(index).productCount;
}
void User::searchProduct()
{
    /*****
    - 윤재문(15.11.22)
    - Function   : searchProduct
    - Input      : vector <ProductInfo> vec
    - Output     : None
    - Procedure  : 상품명으로 검색
    - 수정 날짜   :
    윤재문(15.11.28)
    - 상품명중 일부만 포함해도 검색가능하도록 구현
    윤재문(15.12.06)
    - 키보드로 조작가능하게 수정, 추천상품 처리가능하도록 구현
    *****/

    string search;
    vector <int> temp; //검색된 결과를 담는 벡터
    int productIndex = -1;
    int page = 1;     //페이지표시를 위한 변수

```

```

int point = 0; //상품 선택을 위한 변수
int check;
system("cls");
cout << "상품명을 입력해 주세요 >>";
cin >> search;
/*****검색된 결과 저장*****/
for (int i = 0; i < vec_product.size(); i++)
{
    if (vec_product.at(i).name.find(search) != vec_product.at(i).name.npos)
    {
        temp.push_back(productIndex = i);
    }
}
if (temp.size() != 0) //검색된 상품이 있으면
{
    while (1)
    {
        /*****검색결과 출력부분*****/
        gotoxy(1, 1);
        char key;
        setColor("blue");
        cout << "back to menu: backspace          select: spacebar\n";
        control: left,right    product control: up,down \n\n";
        setColor("white");
        cout << " " << search << " " << "에 대한 검색결과:" << "총" <<
temp.size() << "상품이 검색되었습니다." << endl;
        for (int i = 10 * (page - 1); i < page * 10; i++)
        {
            if (i < temp.size())
            {
                if (i % 10 == point)
                {
                    setColor("red");
                    cout << temp.at(i) + 1 << "번 상품: " <<
vec_product.at(temp.at(i)).name << endl;

                    check = temp.at(i);
                    setColor("white");
                }
                else
                    cout << temp.at(i) + 1 << "번 상품: " <<
vec_product.at(temp.at(i)).name << endl;
            }
        }
        gotoxy(1, 16);
        cout << page << "/" << (temp.size() - 1) / 10 + 1; //페이지 표시
        /*****키보드조작부분*****/
        key = getch();
        if (key == 8)
        {

```

```

        system("cls");
        return;
    }
    else if (key == 75 && page != 1)    //left arrow
    {
        page--;
        point = 0;
        system("cls");
    }
    else if (key == 72 && point != 0)    //down arrow
        point--;
    else if (key == 77 && page != ((temp.size() - 1) / 10) + 1)
//right arrow
    {
        page++;
        point = 0;
        system("cls");
    }
    else if (key == 80 && point != 9)    //up arrow
    {
        if (page == (temp.size() / 10) + 1 && point ==
(temp.size() % 10) - 1)    //상품이 10개미만인경우 마지막 상품까지만 가리킴
        {
            //do noting
        }
        else
            point++;
    }
    else if (key == 32)
    {
        Product temp(vec_product.at(check));
        temp.detailInfo(userIndex);
    }
    }
}

void User::menuProduct()
{
    /*****
    - 윤재문(15.11.28)
    - Function    : menuProduct
    - Input      :
    - Output     : None
    - Procedure  : 모든 상품들을 나열하여 보여줌
    - 수정 날짜   : 김가림(15.11.28)
    *****/
    int page = 1;
    int point = 0;
    int check;

```

```

system("cls");
while (1)
{
    /*****목록 출력*****/
    gotoxy(1, 1);
    char key;
    setColor("blue");
    cout << "돌아가기: backspace      자세히보기: spacebar" << endl;
    cout << "이전페이지: left      다음페이지: right" << endl;
    cout << "상품선택: up, down      추천상품!: r      히트상품: h" <<
endl;

    setColor("white");
    for (int i = 10 * (page - 1); i < page * 10; i++)
    {
        if (i < vec_product.size())
        {
            if (i % 10 == point)
            {
                setColor("red");
                cout << i + 1 << ": 제품명: " <<
vec_product.at(i).name << endl;

                check = i;
                setColor("white");
            }
            else
                cout << i + 1 << ": 제품명: " <<
vec_product.at(i).name << endl;
        }
    }
    gotoxy(1, 15);
    cout << page << "/" << (vec_product.size() - 1) / 10 + 1; //페이지 표시
    /*****키보드조작 부분*****/
    key = getch();
    if (key == 8)
    {
        system("cls");
        return;
    }
    else if (key == 75 && page != 1) //left arrow
    {
        page--;
        point = 0;
        system("cls");
    }
    else if (key == 72 && point != 0) //down arrow
        point--;
    else if (key == 77 && page != ((vec_product.size() - 1) / 10) + 1) //right
arrow
    {

```



```

        page++;
        point = 0;
        system("cls");
    }
    else if (key == 80 && point != 9)    //up arrow
    {
        if (page == (vec_product.size() / 10) + 1 && point ==
(vec_product.size() % 10) - 1)    //상품이 10개미만인경우 마지막 상품까지만 가리킴
        {
            //do noting
        }
        else
            point++;
    }
    else if (key == 32)    //spacebar
    {
        Product temp(vec_product.at(check));
        temp.detailInfo(userIndex);
    }
    else if (key == 'r' || key == 'R')
    {
        recommendProduct();
    }
    else if (key == 'h' || key == 'H')
    {
        showSelINumber();
    }
    Sleep(30);
}
}
void User::myInfo()
{
    /*****
    - 김가림(15.11.28)
    - Function    : myInfo()
    - Input       : None
    - Output      : None
    - Procedure   : 내 정보 확인 페이지
    - 수정 날짜   :
    *****/
    system("cls");
    gotoxy(1, 1);
    cout << "****내 정보 확인 페이지****" << endl;
    cout << "이름 : " << name << endl;
    cout << "핸드폰 번호 : " << phone << endl;
    cout << "주소 : " << address << endl;
    cout << "현재까지 구매한 물건의 수 : " << productCount << endl;
    cout << "현재 코인 : " << coin << endl;
    showMyFeat();
}

```

```

        system("cls");
        return;
    }
    void User::showMyFeat()
    {
        /*****
        - 김가림
        - Function   : showMyFeat
        - Input      : None
        - Output     : None
        - Procedure  : 사용자의 정보를 보여준다
        - 수정 날짜   : (15.12.01) 색 추가
        *****/
        cout << endl;
        cout << "CUTY [" << feat1_cut << "]" << endl;
        for (int i = 0; i < feat1_cut / 5; i++)
        {
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
            cout << "■";
            Sleep(25);
        }
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
        cout << endl;

        cout << "SEXY [" << feat2_sexy << "]" << endl;
        for (int i = 0; i < feat2_sexy / 5; i++)
        {
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
            cout << "■";
            Sleep(25);
        }
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
        cout << endl;

        cout << "FORMAL [" << feat3_formal << "]" << endl;
        for (int i = 0; i < feat3_formal / 5; i++)
        {
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
            cout << "■";
            Sleep(25);
        }
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
        cout << endl;

        cout << "CASUAL [" << feat4_casual << "]" << endl;
        for (int i = 0; i < feat4_casual / 5; i++)
        {
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
            cout << "■";

```

```

        Sleep(25);
    }
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
    cout << endl;
    getch();
}
void User::coinCharge()
{
    /*****
    - 김가림(15.11.26)
    - Function   : coinCharge
    - Input      : None
    - Output     : None
    - Procedure  : 사용자의 전화번호와 같은 번호를 입력하면 코인 충전 성공
    - 수정 날짜   :
    *****/
    cin.sync();
    int tempCoin = 0;
    string tempPhone;
    cout << endl;
    cout << "***코인충전페이지**" << endl;
    cin.sync();
    cout << "얼마를 충전할 것인지 입력하세요 : " << endl;
    cin >> tempCoin;
    cin.sync();
    cout << "개인 확인을 위해 전화번호를 입력해주세요 : " << endl;
    cin >> tempPhone;
    while (tempPhone != phone)
    {
        cout << "잘못된 전화번호를 입력하셨습니다." << endl;
        cout << "정확한 번호를 입력해주세요 : ";
        cin >> tempPhone;
    }
    cin.sync();
    cout << "인증에 성공하셨습니다." << endl;
    coin += tempCoin;
    vec_user.at(userIndex).coin += tempCoin;
    cout << "현재 잔액 : " << coin << endl;
    saveData();
    getch();
    system("cls");
}
bool User::logout()
{
    system("cls");
    saveData();
    return true;
}

```

```

void User::shoppingCartPayment()
{
    /*****
    - 김가림(2015.12.01)
    - Function : shoppingCartPayment
    - Input : None
    - Output : None
    - Procedure : 장바구니에 들어있는 상품을 주문, 결제한다.
    - 수정 날짜 :
    윤재문(2015.12.05)
    -주문서 양식에 맞게 출력하도록 수정
    -결제완료시 쇼핑카트 비우도록 구현
    김가림(2015. 12.13)
    -주문 후 상품의 해당 상품의 누적 구매 수 올라가도록 수정
    *****/

    cout << "장바구니에 있는 상품 결제" << endl;
    int totalCost = 0; //배송비를 포함한 총 가격

    char isEqualAddress;
    char isEqualPhone;
    char isPayment;

    string userAddress;
    string userPhone;

    Order tempOrder;

    ProductInfo tempProduct;
    cout << "a.주문상품 확인" << endl;
    for (int i = 0; i < vec_shoppingCart.size(); i++)
    {
        tempProduct = vec_shoppingCart.at(i);
        cout << i + 1 << "번 째 상품" << endl;
        cout << "상품명 : " << tempProduct.name << endl;
        cout << "가격 : " << tempProduct.cost << endl;
        cout << "사이즈 : " << tempProduct.size << endl;
        totalCost += tempProduct.cost;
        cout << endl;
    }

    cout << "상품 가격 : " << totalCost << "원" << endl;
    totalCost += 2500;
    cout << "배송비를 포함한 전체 가격 : " << totalCost << "원" << endl;

    cout << "b. 배송지, 주문자 정보 확인" << endl;
    cout << "기존 주소와 일치합니까 ? (y/n)" << endl;
    cout << "기존 주소 : ";
    cout << address << endl;

```

```

cin >> isEqualAddress;
if (isEqualAddress == 'n')
{
    cout << "새로운 주소를 입력해주세요" << endl;
    getline(cin, userAddress);
}
else
    userAddress = address;
cout << endl;

cout << "전화번호 : " << phone << endl;
cout << "주문자 전화번호와 일치합니까?(y/n)" << endl;
cin >> isEqualPhone;
if (isEqualPhone == 'n')
{
    cout << "새로운 주문자 전화번호를 입력해주세요" << endl;
    cin >> userPhone;
}
else
    userPhone = phone;

cout << endl;

cout << "c. 결제 정보" << endl;
cout << "사용자의 현재 코인 : " << coin << endl;
if (coin < totalCost)
{
    cout << "coin이 부족합니다. 충전을 해주세요" << endl;
    Sleep(3000);
    return;
}
cout << "결제 후 코인 : " << coin - totalCost << endl;
cout << "결제하시겠습니까? (y/n)" << endl;
cin >> isPayment;

if (isPayment == 'n')
{
    cout << "결제를 취소합니다." << endl;
    Sleep(3000);
    return;
}
vec_user.at(userIndex).coin -= totalCost;    //사용자 코인 차감

for (int i = 0; i < vec_shoppingCart.size(); i++)    //주문서 생성
{
    tempProduct = vec_shoppingCart.at(i);
    if (i==0)
        tempOrder.productName =tempProduct.name;
    else

```

```

        tempOrder.productName += "," + tempProduct.name;
    }
    tempOrder.isDelivered = false;
    tempOrder.productNum = vec_shoppingCart.size();
    if (vec_shoppingCart.size()==1)
        tempOrder.productSize = vec_shoppingCart.at(0).size;
    else
        tempOrder.productSize =
vec_shoppingCart.at(0).size+"외"+itos(vec_shoppingCart.size()-1)+"개";
    tempOrder.totalPrice = totalCost;
    tempOrder.userAddress = userAddress;
    tempOrder.userName = name;
    tempOrder.userPhone = userPhone;
    vec_order.push_back(tempOrder);

    for (int i = 0; i < vec_shoppingCart.size(); i++)    //사용자 정보 수정
    {
        tempProduct = vec_shoppingCart.at(i);
        int tempNum;

        tempNum =
(vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat1_cuty) +
tempProduct.feat1_cuty;
        vec_user.at(userIndex).feat1_cuty = tempNum /
(vec_user.at(userIndex).productCount + 1);

        ///feat2
        tempNum =
(vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat2_sexy) +
tempProduct.feat2_sexy;
        vec_user.at(userIndex).feat2_sexy = tempNum /
(vec_user.at(userIndex).productCount + 1);

        ///feat3
        tempNum =
(vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat3_formal) +
tempProduct.feat3_formal;
        vec_user.at(userIndex).feat3_formal = tempNum /
(vec_user.at(userIndex).productCount + 1);

        ///feat4
        tempNum =
(vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat4_casual) +
tempProduct.feat4_casual;
        vec_user.at(userIndex).feat4_casual = tempNum /
(vec_user.at(userIndex).productCount + 1);

        vec_user.at(userIndex).productCount++;
    }

```

```

        for (int i = 0; i < vec_shoppingCart.size(); i++)                //주문 후 상품의
재고, 판매량 수정
        {
            tempProduct = vec_shoppingCart.at(i);
            for (int j = 0; j < vec_product.size(); j++)
            {
                if (tempProduct.name == vec_product.at(j).name)
                {
                    vec_product.at(j).stock--;
                    vec_product.at(j).sellNumber++;
                }
            }
        }

        feat1_cuty = vec_user.at(userIndex).feat1_cuty;
        feat2_sexy = vec_user.at(userIndex).feat2_sexy;
        feat3_formal = vec_user.at(userIndex).feat3_formal;
        feat4_casual = vec_user.at(userIndex).feat4_casual;
        productCount = vec_user.at(userIndex).productCount;
        coin = vec_user.at(userIndex).coin;
        vec_shoppingCart.clear();
        saveData();
    }
    void User::recommendProduct()
    {
        /*****
        - 김가림(15.11.30)
        - Function    : recommendProduct
        - Input       : None
        - Output      : None
        - Procedure   : 일치율이 높은 상품은 menuProduct페이지 상단에 보여준다
        - 수정 날짜   :
        윤재문(15.12.06)
        -코드 전체를 admin의 search와 비슷하게 수정
        -temp벡터에 추천결과를 담고 키보드를 이용하여 페이지,상품선택이 가능하도록
        수정
        *****/

        string search;
        vector<int> temp; //검색된 결과를 담는 벡터
        int productIndex = -1;
        int page = 1;    //페이지표시를 위한 변수
        int point = 0;   //상품 선택을 위한 변수
        int check;
        system("cls");
        for (int i = 0; i < vec_product.size(); i++)
        {
            if (calculationFeat(i) >= 85)

```

```

        {
            temp.push_back(i);
        }
    }
    if (temp.size() != 0)    //검색된 상품이 있으면
    {
        while (1)
        {
            /*****검색결과 출력부분*****/
            gotoxy(1, 1);
            char key;
            setColor("blue");
            cout << "돌아가기: backspace      자세히보기: spacebar " << endl;
            cout << "이전페이지: left      다음페이지: right  " << endl;
            cout << "상품선택: up, down      Wn" << endl;
            setColor("white");
            cout << "총" << temp.size() << "상품이 추천되었습니다!!" << endl;
            for (int i = 10 * (page - 1); i < page * 10; i++)
            {
                if (i < temp.size())
                {
                    if (i % 10 == point)
                    {
                        setColor("red");
                        cout << temp.at(i) + 1 << "번 상품: " <<
vec_product.at(temp.at(i)).name << endl;

                        check = temp.at(i);
                        setColor("white");
                    }
                    else
                        cout << temp.at(i) + 1 << "번 상품: " <<
vec_product.at(temp.at(i)).name << endl;
                }
            }
            gotoxy(1, 17);
            cout << page << "/" << (temp.size() - 1) / 10 + 1; //페이지 표시
            /*****키보드조작부분*****/
            key = getch();
            if (key == 8)
            {
                system("cls");
                return;
            }
            else if (key == 75 && page != 1)    //left arrow
            {
                page--;
                point = 0;
                system("cls");
            }
        }
    }
}

```



```

        else if (key == 72 && point != 0)    //down arrow
            point--;
        else if (key == 77 && page != ((temp.size() - 1) / 10) + 1)
//right arrow
        {
            page++;
            point = 0;
            system("cls");
        }
        else if (key == 80 && point != 9)    //up arrow
        {
            if (page == (temp.size() / 10) + 1 && point ==
(temp.size() % 10) - 1)    //상품이 10개미만인경우 마지막 상품까지만 가리킴
            {
                //do noting
            }
            else
                point++;
        }
        else if (key == 32)
        {
            Product temp(vec_product.at(check));
            temp.detailInfo(userIndex);
        }
    }
}

void User::showShoppingCart()
{
    /*****
    - 김가림(15.12.05)
    - Function   : showShoppingCart()
    - Input      : proudctIndex
    - Output     : concordanceRate
    - Procedure  : product의 Index를 받아와서 user와의 일치율을 계산해서 return
    - 수정 날짜  : 김가림(15.12.13)장바구니가 비었을 때 명시
    *****/

    ProductInfo tempProduct;
    cout << "***쇼핑카트***" << endl << endl;
    if (vec_shoppingCart.empty() == true)
        cout << "장바구니가 비었습니다" << endl;
    for (int i = 0; i < vec_shoppingCart.size(); i++)
    {
        tempProduct = vec_shoppingCart.at(i);
        cout << i + 1 << "번 째 상품" << endl;
        cout << "상품명: " << tempProduct.name << endl;
        cout << "가격 : " << tempProduct.cost << endl;
        cout << "사이즈 : " << tempProduct.size << endl;
    }
}

```

```

        cout << endl;
    }
    cout << "주문을 원하시면 스페이스를 눌러주세요" << endl;
    cout << "뒤로가기는 백스페이스" << endl;
    while (1)
    {
        char key = getch();
        if (key == 32)
        {
            if (vec_shoppingCart.size() == 0)
            {
                cout << "장바구니에 상품이 없습니다\n";
                Sleep(1000);
                system("cls");
                return;
            }
            else
                shoppingCartPayment();
        }
        else if (key == 8)
        {
            system("cls");
            return;
        }
    }
}

int User::calculationFeat(int productIndex)
{
    /*****
    - 김가림(15.11.30)
    - Function   : calculationFeat
    - Input      : productIndex
    - Output     : concordanceRate
    - Procedure  : product의 Index를 받아와서 user와의 일치율을 계산해서 return
    - 수정 날짜  :
    *****/

    int error[4];
    int sumError, concordanceRate;
    error[0] = feat1_cuty - vec_product.at(productIndex).feat1_cuty;
    error[1] = feat2_sexy - vec_product.at(productIndex).feat2_sexy;
    error[2] = feat3_formal - vec_product.at(productIndex).feat3_formal;
    error[3] = feat4_casual - vec_product.at(productIndex).feat4_casual;

    for (int i = 0; i < 4; i++)
    {
        if (error[i] < 0)
            error[i] = -error[i];
    }
}

```

```

sumError = error[0] + error[1] + error[2] + error[3];
sumError /= 4;
concordanceRate = 100 - sumError;
return concordanceRate;
}
void User::showSelNumber()
{
    /*****
    - 윤재문(15.12.08)
    - Function : showSelNumber
    - Input : None
    - Output : None
    - Procedure : 판매량이 높은 순으로 상품을 나열해줌
    - 수정 날짜 :
    *****/
    system("cls");
    int flag=0;
    vector<ProductInfo> temp = vec_product;
    vector<int> result;
    for (int i = 0; i < temp.size(); i++)
    {
        for (int j = 0; j<temp.size(); j++)
        {
            if (temp.at(j).selNumber>temp.at(flag).selNumber)
            {
                flag = j;
            }
        }
        result.push_back(flag);
        temp.at(flag).selNumber = -1;
    }
    int page = 1;
    int point = 0;
    int check;
    system("cls");
    while (1)
    {
        /*****검색결과 출력부분*****/
        gotoxy(1, 1);
        char key;
        setColor("blue");
        setColor("blue");
        cout << "돌아가기: backspace" << endl;
        cout << "이전페이지: left" << endl;
        cout << "상품선택: up, down" << endl;
        setColor("white");
        setColor("white");
        for (int i = 10 * (page - 1); i < page * 10; i++)
        {
            자세히보기: spacebar " << endl;
            다음페이지: right " << endl;
        }
    }
}

```

```

        if (i < result.size())
        {
            if (i % 10 == point)
            {
                setColor("red");
                cout << result.at(i) + 1 << "번 상품: " <<
vec_product.at(result.at(i)).name << " 판매량:" << vec_product.at(result.at(i)).sellNumber<<
endl;

                check = result.at(i);
                setColor("white");
            }
            else
                cout << result.at(i) + 1 << "번 상품: " <<
vec_product.at(result.at(i)).name << " 판매량:" << vec_product.at(result.at(i)).sellNumber<<
endl;
        }
    }
    gotoxy(1, 16);
    cout << page << "/" << (result.size() - 1) / 10 + 1;           //페이지 표시
    /*****키보드조작부분*****/
    key = getch();
    if (key == 8)
    {
        system("cls");
        return;
    }
    else if (key == 75 && page != 1)    //left arrow
    {
        page--;
        point = 0;
        system("cls");
    }
    else if (key == 72 && point != 0)    //down arrow
        point--;
    else if (key == 77 && page != ((temp.size() - 1) / 10) + 1)    //right arrow
    {
        page++;
        point = 0;
        system("cls");
    }
    else if (key == 80 && point != 9)    //up arrow
    {
        if (page == (temp.size() / 10) + 1 && point == (temp.size() % 10) -
1)    //상품이 10개미만인경우 마지막 상품까지만 가리킴
        {
            //do noting
        }
        else
            point++;
    }

```

```

    }
    else if (key == 32)        //spacebar
    {
        Product temp(vec_product.at(check));
        temp.detailInfo(userIndex);
    }
    Sleep(30);
}
}
void Admin::getOrder()
{
    /*****
    - 김가림(15.11.30)
    - Function   : getOrder
    - Input      : none
    - Output     : none
    - Procedure  : 관리자가 현재까지의 주문서를 모두 볼 수 있는 페이지
    - 수정 날짜   :
    *****/

    cout << "**주문서 확인 페이지**";
    for (int i = 0; i < vec_order.size(); i++)
    {
        cout << "[주문서 " << i + 1 << "]" << endl;
        cout << "주문자 : " << vec_order.at(i).userName << endl;
        cout << "배송지 : " << vec_order.at(i).userAddress << endl;
        cout << "주문자 번호 : " << vec_order.at(i).userPhone << endl;
        cout << "상품명 : " << vec_order.at(i).productName << endl;
        cout << "주문 수량 : " << vec_order.at(i).productNum << endl;
        cout << "상품 사이즈 " << vec_order.at(i).productSize << endl;
        if (vec_order.at(i).isDelivered == true)
            cout << "배송 여부 : 배송 완료" << endl;
        else
            cout << "배송 여부 : 배송중" << endl;
        cout << endl << endl;
    }
    char inputChar;
    cout << "일괄 배송여부를 수정하시겠습니까?(y/n) " << endl;
    cin >> inputChar;
    if (inputChar == 'y')
    {
        for (int i = 0; i < vec_order.size(); i++)
        {
            vec_order.at(i).isDelivered = true;
        }
        cout << "모든 배송을 완료하였습니다." << endl;
    }
    saveData();
}
}

```

```

void Admin::menuProduct()
{
    /*****
    - 윤재문(15.11.22)
    - Function    : menuProduct
    - Input       : vector <ProductInfo> vec
    - Output      : None
    - Procedure   : 모든 상품들을 나열하여 보여줌
    - 수정 날짜   :
    윤재문(15.11.28)- 10개씩 1페이지로 나누어 보여주도록 수정
    윤재문(15.11.29)- 숫자입력에서 키보드로 조작하도록 수정
    윤재문(15.12.08)- 판매량이 높은순으로 볼수있는 메뉴추가
                        - 재고량이 10이하인 상품을 볼수있는 메뉴추가
    *****/

    int page = 1;
    int point = 0;
    int check;
    system("cls");
    while (1)
    {
        /*****목록 출력*****/
        gotoxy(1, 1);
        char key;
        setColor("blue");
        cout << "돌아가기: backspace      자세히보기: spacebar " << endl;
        cout << "이전페이지: left      다음페이지: right " << endl;
        cout << "상품선택: up, down      판매량보기: h      재고량보기: s"
        << endl;

        setColor("white");
        for (int i = 10 * (page - 1); i < page * 10; i++)
        {
            if (i < vec_product.size())
            {
                if (i % 10 == point)
                {
                    setColor("red");
                    cout << i + 1 << ": 제품명: " <<

                    check = i;
                    setColor("white");

                }
                else
                    cout << i + 1 << ": 제품명: " <<

                    vec_product.at(i).name << endl;

            }
        }
        gotoxy(1, 15);
        cout << page << "/" << (vec_product.size()-1) / 10 + 1;    //페이지 표시
    }
}

```

```

/*****키보드조작 부분*****/
key = getch();
if (key == 8)
{
    system("cls");
    return;
}
else if (key == 75 && page != 1)    //left arrow
{
    page--;
    point = 0;
    system("cls");
}
else if (key == 72 && point != 0)    //down arrow
    point--;
else if (key == 77 && page != ((vec_product.size()-1) / 10) + 1)    //right
arrow
{
    page++;
    point = 0;
    system("cls");
}
else if (key == 80 && point != 9)    //up arrow
{
    if (page == (vec_product.size() / 10) + 1 && point ==
(vec_product.size() % 10) - 1)    //상품이 10개미만인경우 마지막 상품까지만 가리킴
    {
        //do noting
    }
    else
        point++;
}
else if (key == 32)    //spacebar
{
    Product temp(vec_product.at(check));
    temp.detailInfo();
    productEdit(check);
}
else if (key == 'h' || key == 'H')
{
    showSelNumber();
}
else if (key == 's' || key == 'S')
{
    showStock();
}
Sleep(30);
}

```

```

}
void Admin::searchProduct()
{
    /*****
    - 윤재문(15.11.22)
    - Function : searchProduct
    - Input : vector <ProductInfo> vec
    - Output : None
    - Procedure : 상품명으로 검색
    - 수정 날짜 :
    윤재문(15.11.28)
    - 상품명중 일부만 포함해도 검색가능하도록 수정
    - 총 검색된 상품수 출력추가
    - 키보드로 조작하도록 구현
    윤재문(15.12.05)
    -검색한 상품정보 수정가능하도록 구현
    *****/
    string search;
    vector<int> temp; //검색된 결과를 담는 벡터
    int productIndex = -1;
    int page = 1; //페이지표시를 위한 변수
    int point = 0; //상품 선택을 위한 변수
    int check;
    system("cls");
    cout << "상품명을 입력해 주세요 >>";
    cin >> search;
    /*****검색된 결과 저장*****/
    for (int i = 0; i < vec_product.size(); i++)
    {
        if (vec_product.at(i).name.find(search) != vec_product.at(i).name.npos)
        {
            temp.push_back(productIndex = i);
        }
    }
    if (temp.size() != 0) //검색된 상품이 있으면
    {
        while (1)
        {
            /*****검색결과 출력부분*****/
            gotoxy(1, 1);
            char key;
            setColor("blue");
            cout << "돌아가기: backspace" << "자세히보기: spacebar" << endl;
            cout << "이전페이지: left" << "다음페이지: right" << endl;
            cout << "상품선택: up, down" << endl;
            setColor("white");
            cout << " " << search << "에 대한 검색결과:" << "총" <<
temp.size() << "상품이 검색되었습니다." << endl;
            for (int i = 10 * (page - 1); i < page * 10; i++)

```



```

        {
            if (i < temp.size())
            {
                if (i % 10 == point)
                {
                    setColor("red");
                    cout << temp.at(i) + 1 << "번 상품: " <<
vec_product.at(temp.at(i)).name << endl;

                    check = temp.at(i);
                    setColor("white");
                }
                else
                    cout << temp.at(i) + 1 << "번 상품: " <<
vec_product.at(temp.at(i)).name << endl;
            }
        }
        gotoxy(1, 16);
        cout << page << "/" << (temp.size()-1) / 10 + 1; //페이지 표시
        /*****키보드조작부분*****/
        key = getch();
        if (key == 8)
        {
            system("cls");
            return;
        }
        else if (key == 75 && page != 1) //left arrow
        {
            page--;
            point = 0;
            system("cls");
        }
        else if (key == 72 && point != 0) //down arrow
            point--;
        else if (key == 77 && page != ((temp.size()-1) / 10) + 1) //right
arrow
        {
            page++;
            point = 0;
            system("cls");
        }
        else if (key == 80 && point != 9) //up arrow
        {
            if (page == (temp.size() / 10) + 1 && point ==
(temp.size() % 10) - 1) //상품이 10개미만인경우 마지막 상품까지만 가리킴
            {
                //do noting
            }
            else
                point++;
        }
    }
}

```

```

    }
    else if (key == 32)
    {
        Product temp(vec_product.at(check));
        temp.detailInfo();
        productEdit(check);
    }
}

}

}

}

}

void Admin::productEdit(int index)
{
    /*****
    - 윤재문(15.12.05)
    - Function   : productEdit
    - Input      : int index
    - Output     : None
    - Procedure  : 상품의 벡터에서의 위치를 받아 해당상품 정보수정
    - 수정 날짜   :
    *****/
    while (1)
    {
        setColor("blue");
        gotoxy(1, 1);
        cout << "돌아가기: backspace   상품삭제: del       상품입고: 1" << endl;
        cout << "가격조정: 2           사이즈조정 :3       성향조정: 4"<<endl;
        setColor("white");
        gotoxy(1, 15);
        cout << "                                Wn";
        cout << "                                Wn";
        cout << "                                Wn";
        gotoxy(1, 15);
        char key;
        key = getch();
        if (key == 16)
        {
            vec_product.erase(vec_product.begin() + index);
        }
        else if (key == 49)
        {
            cout << "입고량입력";
            int a;
            cin >> a;
            vec_product.at(index).stock += a;
            break;
        }
        else if (key == 50)
        {
            cout << "가격입력";

```

```

        int a;
        cin >> a;
        vec_product.at(index).cost = a;
        break;
    }
    else if (key == 51)
    {
        cout << "사이즈입력";
        string a;
        cin >> a;
        vec_product.at(index).size = a;
        break;
    }
    else if (key == 52)
    {
        int a;
        cout << "cuty입력";
        cin >> a;
        vec_product.at(index).feat1_cuty = a;
        cout << "sexy입력";
        cin >> a;
        vec_product.at(index).feat2_sexy = a;
        cout << "formal입력";
        cin >> a;
        vec_product.at(index).feat3_formal = a;
        cout << "casual입력";
        cin >> a;
        vec_product.at(index).feat4_casual = a;
        break;
    }
    else if (key == 8)
    {
        saveData();
        break;
    }
    Sleep(100);
}

system("cls");
}

void Admin::addProduct()
{
    /*****
    - 윤재문(15.11.25)
    - Function   : addProduct
    - Input      : None
    - Output     : None
    - Procedure  : 상품추가 기능
    - 수정 날짜   :
    *****/
}

```

```

string s;
ProductInfo temp;
cout << "돌아가기 : backspace  continue : anykey\\n";
char key;
key = getch();
if (key == 8)
    return;
fflush(stdin);
cout << "상품명: ";
getline(cin, s);
temp.name = s;
cout << "재고량: ";
getline(cin, s);
temp.stock = stoi(s);
cout << "가격: ";
getline(cin, s);
temp.cost = stoi(s);
cout << "사이즈: ";
getline(cin, s);
temp.size = s;
temp.sellNumber = 0;
cout << "cuty: ";
getline(cin, s);
temp.feat1_cuty = stoi(s);
cout << "sexy: ";
getline(cin, s);
temp.feat2_sexy = stoi(s);
cout << "formal: ";
getline(cin, s);
temp.feat3_formal = stoi(s);
cout << "casual: ";
getline(cin, s);
temp.feat4_casual = stoi(s);
vec_product.push_back(temp);
cout << "상품추가완료";
saveData();
Sleep(2000);
}
bool Admin::logOut()
{
    /*****
    - 윤재문(15.12.05)
    - Function   : logOut
    - Input      : None
    - Output     : bool
    - Procedure  : 로그아웃 기능
    - 수정 날짜  :
    *****/
    system("cls");

```

```

        saveData();
        return true;
    }
    void Admin::showSelINumber()
    {
        /*****
        - 윤재문(15.12.08)
        - Function    : showSelINumber
        - Input       : None
        - Output      : None
        - Procedure   : 판매량이 높은순으로 상품을 나열해줌
        - 수정 날짜   :
        *****/
        system("cls");
        int flag = 0;
        vector <ProductInfo> temp = vec_product;
        vector <int> result;
        for (int i = 0; i < temp.size(); i++)
        {
            for (int j = 0; j < temp.size(); j++)
            {
                if (temp.at(j).selINumber > temp.at(flag).selINumber)
                {
                    flag = j;
                }
            }
            result.push_back(flag);
            temp.at(flag).selINumber = -1;
        }
        int page = 1;
        int point = 0;
        int check;
        system("cls");
        while (1)
        {
            /*****검색결과 출력부분*****/
            gotoxy(1, 1);
            char key;
            setColor("blue");
            setColor("blue");
            cout << "돌아가기: backspace"      자세히보기: spacebar " << endl;
            cout << "이전페이지: left"        다음페이지: right " << endl;
            cout << "상품선택: up, down"      " << endl;
            setColor("white");
            setColor("white");
            for (int i = 10 * (page - 1); i < page * 10; i++)
            {
                if (i < result.size())
                {

```

```

        if (i % 10 == point)
        {
            setColor("red");
            cout << result.at(i) + 1 << "번 상품: " <<
vec_product.at(result.at(i)).name << " 판매량:" << vec_product.at(result.at(i)).sellNumber
<< endl;

            check = result.at(i);
            setColor("white");
        }
        else
            cout << result.at(i) + 1 << "번 상품: " <<
vec_product.at(result.at(i)).name << " 판매량:" << vec_product.at(result.at(i)).sellNumber
<< endl;
    }
}
gotoxy(1, 16);
cout << page << "/" << (result.size() - 1) / 10 + 1;           //페이지 표시
/*****키보드조작부분*****/
key = getch();
if (key == 8)
{
    system("cls");
    return;
}
else if (key == 75 && page != 1)    //left arrow
{
    page--;
    point = 0;
    system("cls");
}
else if (key == 72 && point != 0)    //down arrow
    point--;
else if (key == 77 && page != ((temp.size() - 1) / 10) + 1)    //right arrow
{
    page++;
    point = 0;
    system("cls");
}
else if (key == 80 && point != 9)    //up arrow
{
    if (page == (temp.size() / 10) + 1 && point == (temp.size() % 10) -
1)    //상품이 10개미만인경우 마지막 상품까지만 가리킴
    {
        //do noting
    }
    else
        point++;
}
else if (key == 32)

```

```

        {
            Product temp(vec_product.at(check));
            temp.detailInfo();
            productEdit(check);
        }
        Sleep(30);
    }
}

void Admin::showStock()
{
    /*****
    - 윤재문(15.12.08)
    - Function   : showSellNumber
    - Input      : None
    - Output     : None
    - Procedure  : 재고량이 10이하인 상품들을 보여줌
    - 수정 날짜   :
    *****/
    system("cls");
    int flag = 0;
    vector<int> result;
    for (int i = 0; i < vec_product.size(); i++)
    {
        if (vec_product.at(i).stock <= 10)
        {
            result.push_back(i);
        }
    }
    int page = 1;
    int point = 0;
    int check;
    system("cls");
    while (1)
    {
        /*****검색결과 출력부분*****/
        gotoxy(1, 1);
        char key;
        setColor("blue");
        cout << "돌아가기: backspace      자세히보기: spacebar" << endl;
        cout << "이전페이지: left      다음페이지: right  " << endl;
        cout << "상품선택: up, down      " << endl;
        setColor("white");
        for (int i = 10 * (page - 1); i < page * 10; i++)
        {
            if (i < result.size())
            {
                if (i % 10 == point)
                {
                    setColor("red");

```

```

        cout << result.at(i) + 1 << "번 상품: " <<
vec_product.at(result.at(i)).name << " 재고량:" << vec_product.at(result.at(i)).stock <<
endl;

        check = result.at(i);
        setColor("white");
    }
    else
        cout << result.at(i) + 1 << "번 상품: " <<
vec_product.at(result.at(i)).name << " 재고량:" << vec_product.at(result.at(i)).stock <<
endl;
    }
}
gotoxy(1, 16);
cout << page << "/" << (result.size() - 1) / 10 + 1; //페이지 표시
/*****키보드조작부분*****/
key = getch();
if (key == 8)
{
    system("cls");
    return;
}
else if (key == 75 && page != 1) //left arrow
{
    page--;
    point = 0;
    system("cls");
}
else if (key == 72 && point != 0) //down arrow
    point--;
else if (key == 77 && page != ((result.size() - 1) / 10) + 1) //right
arrow
{
    page++;
    point = 0;
    system("cls");
}
else if (key == 80 && point != 9) //up arrow
{
    if (page == (result.size() / 10) + 1 && point == (result.size() %
10) - 1) //상품이 10개미만인경우 마지막 상품까지만 가리킴
    {
        //do noting
    }
    else
        point++;
}
else if (key == 32)
{
    Product temp(vec_product.at(check));

```



```

        temp.detailInfo();
        productEdit(check);
    }
    Sleep(30);
}
}
Product::Product(ProductInfo product)
{
    /*****
    - 김가림(15.12.05)
    - Function   : Product
    - Input      : productInfo
    - Output     :
    - Procedure  : Product구조체의 생성자
    - 수정 날짜   :
    *****/

    this->name = product.name;
    this->stock = product.stock;
    this->cost = product.cost;
    this->size = product.size;
    this->sellNumber = product.sellNumber;
    this->feat1_cuty = product.feat1_cuty;
    this->feat2_sexy = product.feat2_sexy;
    this->feat3_formal = product.feat3_formal;
    this->feat4_casual = product.feat4_casual;
}
void Product::detailInfo(int userIndex)
{
    /*****
    - 김가림(15.11.25)
    - Function   : detailInfo
    - Input      : userIndex
    - Output     : None
    - Procedure  : userIndex를 받아와서 해당 상품을 구매할지 아니면 장바구니에 넣을지를
결정
    - 수정 날짜   : 김가림(15.12.01) 장바구니에 넣는 기능 추가
                    김가림(15.12.13) 재고가 없을때는 재고가 없다는 화면 출력
    *****/
    char inputKey;
    while (1)
    {
        system("cls");
        cout << "상품명 : " << name << endl;
        cout << "가격   : " << cost << endl;
        cout << "사이즈 : " << size << endl;
        cout << "수량   : " << stock << endl;
        showProductFeat(userIndex);
    }
}

```

```

        if (stock < 1)
        {
            cout << "재고가 없습니다." << endl;
        }
        else
        {
            cout << "주문을 원하시면 1번을 눌러주세요" << endl;
            cout << "장바구니에 넣으시려면 2번을 눌러주세요" << endl;
            inputKey = getch();
            if (inputKey == '1')
                payment(userIndex);
            else if (inputKey == '2')
                inputShoppingCart();
        }
        inputKey = getch();
        if (inputKey == 8)
        {
            system("cls");
            return;
        }
    }
}

void Product::inputShoppingCart()
{
    /*****
    - 김가림(15.12.05)
    - Function   : inputShoppingCart
    - Input      :
    - Output     :
    - Procedure  : 해당 product를 벡터에 저장
    - 수정 날짜  :
    *****/

    ProductInfo tempProduct;
    tempProduct.cost = cost;
    tempProduct.name = name;
    tempProduct.sellNumber = sellNumber;
    tempProduct.size = size;
    tempProduct.stock = stock;
    tempProduct.feat1_cuty = feat1_cuty;
    tempProduct.feat2_sexy = feat2_sexy;
    tempProduct.feat3_formal = feat3_formal;
    tempProduct.feat4_casual = feat4_casual;

    vec_shoppingCart.push_back(tempProduct);
    //saveData();
}

void Product::detailInfo()

```

```

{
    /*****
    - 김가림(15.12.05)
    - Function : detailInfo
    - Input :
    - Output :
    - Procedure : 상품의 정보를 보여준다(관리자용)
    - 수정 날짜 :
    *****/

    system("cls");
    gotoxy(1, 6);
    cout << "상품명 : " << name << endl;
    cout << "가격 : " << cost << endl;
    cout << "사이즈 : " << size << endl;
    cout << "수량 : " << stock << endl;
    cout << "판매량 : " << sellNumber << endl;
    cout << "cuty : " << feat1_cuty << endl;
    cout << "sexy : " << feat2_sexy << endl;
    cout << "formal : " << feat3_formal << endl;
    cout << "casual : " << feat4_casual << endl;
}

void Product::showProductFeat( int userIndex)
{
    /*****
    - 김가림(15.12.05)
    - Function : showProductFeat
    - Input : userIndex
    - Output : None
    - Procedure : 해당 상품과 사용자의 일치율을 계산해서 그래프로 보여줌
    - 수정 날짜 : (15.12.10) 색 추가
    *****/

    int error[4];
    int sumError;
    int concordanceRate;

    cout << "*CUTY*" << endl;
    //상품 정보
    for (int i = 0; i < feat1_cuty / 5; i++)
    {
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
        cout << "■";
        Sleep(5);
    }
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
    cout << endl;
    //사용자 정보
    for (int i = 0; i < vec_user.at(userIndex).feat1_cuty / 5; i++)

```

```

{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
    cout << "■";
    Sleep(5);
}
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
cout << endl << endl;

cout << "*SEXY*" << endl;
for (int i = 0; i < feat2_sexy / 5; i++)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
    cout << "■";
    Sleep(5);
}
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
cout << endl;
for (int i = 0; i < vec_user.at(userIndex).feat2_sexy / 5; i++)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
    cout << "■";
    Sleep(5);
}
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
cout << endl << endl;

cout << "*FORMAL*" << endl;
for (int i = 0; i < feat3_formal / 5; i++)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
    cout << "■";
    Sleep(5);
}
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
cout << endl;

for (int i = 0; i < vec_user.at(userIndex).feat3_formal / 5; i++)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
    cout << "■";
    Sleep(5);
}
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
cout << endl << endl;
cout << "*CASUAL*" << endl;
for (int i = 0; i < feat4_casual / 5; i++)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 11);
    cout << "■";

```

```

        Sleep(5);
    }
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
    cout << endl;

    for (int i = 0; i < vec_user.at(userIndex).feat4_casual / 5; i++)
    {
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 14);
        cout << "■ ";
        Sleep(5);
    }
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
    cout << endl << endl;

    error[0] = feat1_cuty - vec_user.at(userIndex).feat1_cuty;
    error[1] = feat2_sexy - vec_user.at(userIndex).feat2_sexy;
    error[2] = feat3_formal - vec_user.at(userIndex).feat3_formal;
    error[3] = feat4_casual - vec_user.at(userIndex).feat4_casual;

    for (int i = 0; i < 4; i++)
    {
        if (error[i] < 0)
            error[i] = -error[i];
    }
    sumError = error[0] + error[1] + error[2] + error[3];
    sumError /= 4;
    concordanceRate = 100 - sumError;
    cout << "해당 옷과 고객님의 성향";
    cout << "일치율은 " << concordanceRate << "% 입니다" << endl;
    // Sleep(3000);
}

void Product::payment(int userIndex)
{
    system("cls");
    /*****
    - 김가림(15.11.25)
    - Function : payment
    - Input : userIndex
    - Output : None
    - Procedure : 해당 상품 구매 / 결제 페이지
    - 수정 날짜 : 구매한 상품의 재고와 판매량수 조절
    *****/

    int buyNum; //몇개 사는지 정보
    int totalCost; //배송비를 포함한 총 가격

    char isEqualAddress;
    char isEqualPhone;
    char isPayment;

```

```

string userAddress;
string userPhone;

Order tempOrder;

cout << "***상품 주문/결제 페이지***" << endl;
cout << "a. 주문상품 확인" << endl;
cout << "상품명 : " << name << endl;
cout << "상품 사이즈 : " << size << endl;
while (1)
{
    cout << "수량을 입력해주세요" << endl;
    cin >> buyNum;
    if (buyNum <= stock)
    {
        break;
    }
    else
    {
        cout << "재고량보다 많습니다." << endl;
    }
}
cout << "가격 : " << cost*buyNum << "원" << endl;
cout << "배송비 : 2500원" << endl;
totalCost = cost*buyNum + 2500;
cout << "총 가격 : " << totalCost << "원" << endl << endl;
cout << "b. 배송지, 주문자 정보 확인" << endl;
cout << "기존 주소와 일치합니까 ? (y/n)" << endl;
cout << "기존 주소 : ";
cout << vec_user.at(userIndex).address << endl;
cin.sync();
cin >> isEqualAddress;
if (isEqualAddress == 'n')
{
    cout << "새로운 주소를 입력해주세요" << endl;
    cin.sync();
    getline(cin, userAddress);
}
else
    userAddress = vec_user.at(userIndex).address;
cout << endl;

cout << "전화번호 : " << vec_user.at(userIndex).phone << endl;
cout << "주문자 전화번호와 일치합니까 ?(y/n)" << endl;
cin >> isEqualPhone;
if (isEqualPhone == 'n')
{
    cout << "새로운 주문자 전화번호를 입력해주세요" << endl;

```

```

        cin >> userPhone;
    }
    else
        userPhone = vec_user.at(userIndex).phone;

    cout << endl;
    cout << "c. 결제 정보" << endl;
    cout << "사용자의 현재 코인 : " << vec_user.at(userIndex).coin << endl;
    if (vec_user.at(userIndex).coin < totalCost)
    {
        cout << "coin이 부족합니다. 충전을 해주세요" << endl;
        Sleep(3000);
        return;
    }
    cout << "결제 후 코인 : " << vec_user.at(userIndex).coin - totalCost << endl;
    cout << "결제하시겠습니까? (y/n)" << endl;
    cin >> isPayment;

    if (isPayment == 'n')
    {
        cout << "결제를 취소합니다." << endl;
        Sleep(3000);
        return;
    }

    for (int i = 0; i < vec_product.size(); i++)
    {
        if (name == vec_product.at(i).name)
        {
            vec_product.at(i).stock -= buyNum;
            vec_product.at(i).sellNumber += buyNum;
        }
    }

    vec_user.at(userIndex).coin -= totalCost; //사용자 코인 차감
    vec_user.at(userIndex).productCount += buyNum; //사용자 총 구매 상품수 증가
    replaceFeat(userIndex, buyNum); //사용자 성향

    stock -= buyNum;
    sellNumber += buyNum;
    ///order구조체 생성
    tempOrder.productName = name;
    tempOrder.productNum = buyNum;
    tempOrder.productSize = size;
    tempOrder.totalPrice = totalCost;
    tempOrder.userAddress = userAddress;
    tempOrder.userName = vec_user.at(userIndex).name;
    tempOrder.userPhone = userPhone;
    tempOrder.isDelivered = false;

```

조절

```

        vec_order.push_back(tempOrder);
        //vector에 push_back함
        saveData();
    }
    void Product::replaceFeat(int userIndex, int buyNum)
    {
        /*****
        - 김가림(15.11.25)
        - Function : replaceFeat
        - Input : userIndex, buyNum
        - Output : None
        - Procedure : 상품을 결제한 후 상품의 정보를 내 정보에 누적 반영
        - 수정 날짜 :
        *****/

        int tempNum;
        cout << "**성향 수정**" << endl;
        cout << "기존 성향" << endl;
        cout << "cuty : " << vec_user.at(userIndex).feat1_cuty << endl;
        cout << "sexy : " << vec_user.at(userIndex).feat2_sexy << endl;
        cout << "formal : " << vec_user.at(userIndex).feat3_formal << endl;
        cout << "casual : " << vec_user.at(userIndex).feat4_casual << endl;

        ///feat1
        tempNum = (vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat1_cuty)
+ (buyNum*feat1_cuty);
        vec_user.at(userIndex).feat1_cuty = tempNum / (vec_user.at(userIndex).productCount +
buyNum);

        ///feat2
        tempNum = (vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat2_sexy)
+ (buyNum*feat2_sexy);
        vec_user.at(userIndex).feat2_sexy = tempNum / (vec_user.at(userIndex).productCount +
buyNum);

        ///feat3
        tempNum =
(vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat3_formal) +
(buyNum*feat3_formal);
        vec_user.at(userIndex).feat3_formal = tempNum / (vec_user.at(userIndex).productCount
+ buyNum);

        ///feat4
        tempNum =
(vec_user.at(userIndex).productCount)*(vec_user.at(userIndex).feat4_casual) +
(buyNum*feat4_casual);
        vec_user.at(userIndex).feat4_casual = tempNum / (vec_user.at(userIndex).productCount
+ buyNum);
    }

```



```
    cout << "수정 후 성향" << endl;
    cout << "cuty : " << vec_user.at(userIndex).feat1_cuty << endl;
    cout << "sexy : " << vec_user.at(userIndex).feat2_sexy << endl;
    cout << "formal : " << vec_user.at(userIndex).feat3_formal << endl;
    cout << "casual : " << vec_user.at(userIndex).feat4_casual << endl;
}
int main()
{
    showStartingScreen();
    getch();
    system("cls");
    Start a;
    return 0;
}
```

8. 설문조사 내용

1. 나는 밝은 옷을 좋아한다

예) 큐티+3 캐주얼+1

보통) 큐티+1

아니오) 섹시+3 포멀+1

2. 나는 노출에 대해 거부감이 없다

예) 섹시+3

보통) 섹시+1 캐주얼+1

아니오) 큐티+3

3. 나는 캐릭터가 그려진 옷을 좋아한다

예) 큐티+5

보통) 큐티+3 캐주얼+3

아니오) 포멀+5 섹시+3

4. 나는 격식을 갖추서 옷을 입는다.

예) 포멀+5

보통) 포멀+3

아니오) 캐주얼+5 섹시+3

5. 피어싱을 하셨습니까?

예) 섹시+3 캐주얼+5

아니오) 포멀+5

6. 나는 핑크색을 좋아한다

예) 큐티+5

보통) 큐티+3

아니오) 섹시+5

7. 나는 스니커즈를 좋아한다

예) 캐주얼+5

보통) 캐주얼+3

아니오) 포멀+5 섹시+3

8. 나는 스트라이프 티를 좋아한다

예) 캐주얼+5 포멀+3

보통) 캐주얼+3

아니오) 섹시+5

9. 나는 검정색을 좋아한다

예) 섹시+3

보통) 섹시+1

아니오) 큐티+5

10. 나는 스냅백을 좋아한다.

예) 캐주얼+5

보통) 캐주얼+3

아니오) 섹시+3 포멀+3

11. 나는 평소에 귀엽다는 들어본적 있다.

예) 큐티+7

보통) 큐티+5

아니오) 섹시+5 포멀+5

12. 나는 풀세팅하는것이 귀찮지 않다.

예) 섹시+5 포멀+5

보통) 포멀+3

아니오) 캐주얼+5

13. 평소에 향수 뿌리는것을 좋아한다.

예) 섹시+3

보통) 섹시+1

아니오) 캐주얼+3

14. 나는 타투를 했다.(할 생각이 있다.)

예) 섹시+5

보통) 섹시 +3 캐주얼+3

아니오) 큐티+5 포멀+3

15. 나는 셔츠를 즐겨 입는다.

예) 포멀+5

보통) 포멀+3

아니오) 캐주얼+5

16. 나는 패딩보다 코트를 입는다.

예) 포멀+3 섹시+1

보통) 포멀+1

아니오) 캐주얼+5

17. 나는 레이스달린 옷을 좋아한다.

예) 큐티+5

보통) 큐티+3

아니오) 섹시+3 캐주얼+3

18. 나는 멜빵을 즐겨 입는다.

예) 큐티+5 캐주얼+3

보통) 캐주얼+1

아니오) 섹시+5

19. 나는 청바지를 자주 입는다.

예) 캐주얼+5

보통) 캐주얼+3

아니오) 섹시+5 포멀+5

20. 나는 인상이 강하다.

예) 섹시+5 포멀+5

보통) 섹시+3

아니오) 큐티+5