**중간 보고서**

**(데이터베이스 설계 2019년 1학기)**

**어푸어푸 - 날씨에 따른 옷 추천 프로그램**

제출일: 2019년 5월 28일

|  |  |
| --- | --- |
| 담당 교수 | 이영구 |
| 조명 | 주호식당 |
| 조원 | **배상현(조장)**, 이수인 , 김대철, 박진영 |
| 연락처 | EMAIL: bsh01111@naver.com  C.P:010-4205-1027 |

1. 설계 배경
2. 프로젝트 중요성/필요성



기온별 옷차림은 따로 표가 나와있을 정도로 사람들이 자주 검색하는 항목이다. 기온별로 어떤 옷을 입어야할지 매번 검색하는 번거로운 일을 반복하고 있는 것이다. 이에 따라서 그날 날씨에 맞는 옷을 추천해주는 시스템이 있다면 사람들은 날씨 어플에서 날씨를 확인하고 다시 인터넷에서 이에 따른 옷차림을 검색하는 번거로운 절차를 거치지 않고 편하게 그 날 입을 옷을 정할 수 있게 된다.

뿐만 아니라 쇼핑몰 앱의 일부 기능을 융합함으로써, 실제로 다른 브랜드에서는 추천된 옷들을 어떤 상품으로 판매하고 있는지, 맘에 든다면 어떤 쇼핑몰의 어떤 브랜드인지에 대한 정보를 알려주어 직접 구매가 가능하도록 도울 수 있다.

1. 설계 구현 범위

설계구현할 범위는 다음과 같다. 먼저 기상청에서 기본적으로 알려주는 날씨 정보를 기반으로 날씨에 관한 데이터베이스를 작성한다. 이를 기반으로 날짜에 따른 날씨 정보를 만든 후, 날씨에 따라서 어떤 옷이 적당할 지에 대한 정보를 가지고 있는 데이터베이스를 작성한다. 위의 정보들을 통해서 사용자에게 당일의 날씨와 날씨에 맞는 옷차림을 추천해준다.

쇼핑몰에서 차용할 기능은 성별에 따른 브랜드 목록 기능, 브랜드에 따른 실제 상품들 정보 출력 또한 해당 쇼핑몰의 위치 출력 기능이다. 이를 통해서 사용자는 실제로 어떠한 옷들이 해당 시즌에 판매되고 있는지 역시 한 눈에 볼 수 있고 그 중 마음에 드는 상품이 있다면 해당 위치의 쇼핑몰로 찾아가 상품을 구매할 수 있다.

1. 기대 효과

* 사용자가 입력한 날씨 정보에 대한 옷을 결과로 나타내어 스타일에 맞는 옷을 고를 수 있으므로 어떤 옷을 입어야 할 지에 대한 고민을 줄여 줄 수 있을 것이다.
* 선택한 옷의 구매정보도 보여주어 사용자가 원하는 옷을 구매할 수 있어 사용자에게는 만족도를 높여주고, 쇼핑몰은 이로 인한 홍보효과를 볼 수 있을 것 이다.

1. 내용 요약

기상청의 데이터를 기반으로 계절, 기온, 날씨에 따라 그날 입을 옷을 여러 브랜드에서 지원하는 옷을 남성, 여성, 옷 종류 등 으로 범주를 나누어 사람들에게 추천해 사람들이 좀 더 쉽고 편리하게 옷에 대한 정보를 제공하며 또한 사용자는 옷들이 어떤 브랜드인지 어떤 곳에서 어떤 시즌에 판매되고 있는지 한 눈에 알아보기 쉽게 상품을 접하는 방법을 제공해주는 프로젝트를 설계한다.

1. 설계 목표
2. 요구 사항

* 고객 정보 관리

→ 회원가입 : 새로운 사용자를 DB에 추가

→ 로그인 / 로그아웃 : DB에서 회원의 존재 유무와 아이디, 비밀번호가 일치하는지 확인하여 로그인

* 사용자 옷장 관리

→ 옷장 관리 : 해당 사용자가 가지고 있는 옷들에 대한 정보 조회 후 열람 가능

이때, 옷은 조회(SELECT), 추가(INSERT), 수정(UPDATE), 삭제(DELETE)가 모두 가능

* 날씨 정보 관리

→ 기상청 DB에서 필요한 정보를 골라(SELECT) 날씨 DB에 추가

→ 날씨에 따른 옷 유형 값을 계산하여 새로운 애트리뷰트 값으로 저장

* 쇼핑몰 관리

→ 쇼핑몰에 관한 정보를 DB에 저장

→ 관리자가 쇼핑몰을 추가(INSERT), 수정(UPDATE), 삭제(DELETE) 할 수 있음

→ 추천된 옷들 중 DB에 존재하는 쇼핑몰의 옷은 조회 가능(SELECT)

→ 해당 쇼핑몰의 주소를 통해 사이트로 연결 가능

* 옷 추천

→ 날씨에 따라 값을 지정했던 유형 애트리뷰트에 따라 사용자에게 옷 추천

→ 사용자 옷 DB와 쇼핑몰 옷 DB를 조회(SELECT)하여 옷 추천

1. 고려해야 할 사항

* 실제 상황과 걸맞은가?

우리가 제작할 프로그램은 최대한 실생활에 접목할 수 있을법한 데이터베이스를 사용하는데 초점을 둘 것이다. 실제 날씨와 온도, 계절은 사용자로 하여금 옷을 고를 때 가장 중요한 요소로 작용한다. 이러한 필수적인 요소는 물론 현실의 백화점의 정보를 테이블에 넣어 가장 접근성이 좋은 구매처를 간추려 낼 수 있는 프로그램을 만드는 것이 목표이다.

* 데이터베이스의 중복 최소화

관계데이터베이스를 사용하는 주요 원인중 하나가 바로 중복을 최소화 하는 것이다. 이를 고려하여 테이블간의 join시 중복이 가장 없는 방향으로 구현하는 것을 목표로 할 것이다.

* 효율적인 SQL문 작성

같은 SQL문을 작성하더라도 결과를 도출할 때 더욱 더 효율적인 연산을 할 수 있는 SQL문을 사용하도록 노력한다.

1. 시스템 환경

|  |  |
| --- | --- |
| OS 환경 | Android |
| Front-end | Android Studio |
| Database | SQLite |
| 사용언어 | SQL, Java |

1. 평가 항목

* 결과가 사용할 만 한가?

우리가 설정한 조건으로 하여금 올바른 결과를 도출하는지의 확인 또한 시스템을 평가할 수 있는 항목이라고 할 수 있다. 예를 들어 여름의 옷을 설정하는데 패딩을 골라주거나 하는 일은 일어나서는 안 될 것이다.

* 테이블에 올바른 값이 들어갔는가?

위의 잘못된 결과가 도출되는 것을 방지하기 위해 우리는 적절한 데이터를 테이블에 추가하거나 올바른 테이블의 외부키를 이용하는 등의 방법을 사용할 것이다. 이를 확인 하는 것 또한 평가할 수 있는 항목중 하나라고 할 수 있겠다.

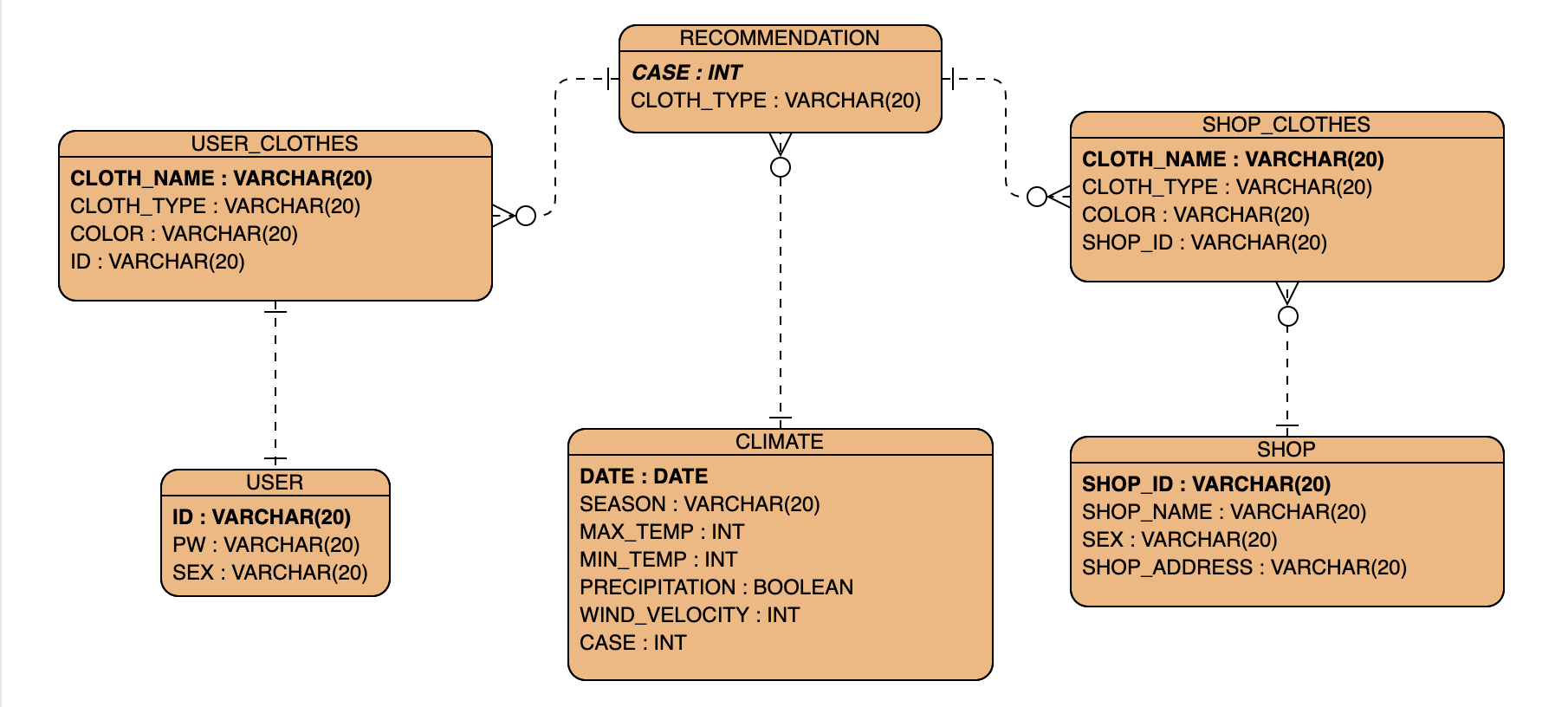
* 사용자가 사용하기에 편리한가?

전체적인 사용자 인터페이스가 어렵지 않게 구성되어 있고, 오랜 시간을 들이지 않아도 어플의 사용방법에 쉽게 익숙해 질 수 있어야 한다.

사용자가 사용하면서 불편함을 최소한으로 느낄 수 있도록 제작해야 한다.

* 전체적인 기능이 잘 구현되었는가?

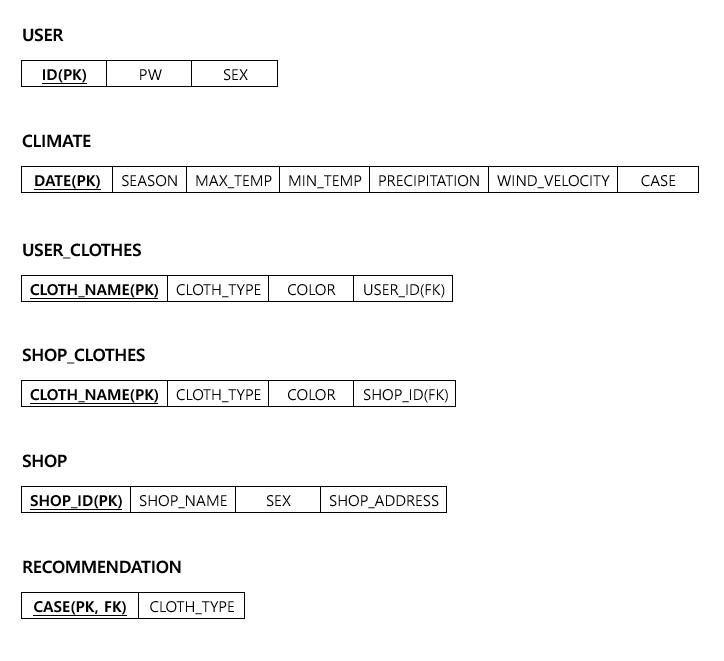
처음에 구현하고자 한 성별, 온도, 날씨,등에 맞는 의상이 추천되었는지를 최우선적으로 확인하고 부가적인 기능이었던 실제 쇼핑몰 사이트와의 연동이 잘 되었는 지를 확인한다.

1. 설계 내용
2. ER 다이어그램
3. 데이터베이스 요약

각 유저 마다 고유한 USER\_CLOTHES 테이블을 가지며(1:1), CLIMATE 테이블 에서 추천 알고리즘을 통해 CASE를 산출해 RECOMMENDATION 테이블에 추천 옷의 데이터를 넣어준다 (1:N).

추천해준 옷을 SHOP에서 찾을 수 있도록 RECOMMENDATION 과 SHOP\_CLOTHES를 연결하고(1:N) SHOP\_CLOTHES와 SHOP 은 (1:N)의 관계를 가진다.

1. ER 스키마와 응용프로그램 설계



* 고객 정보 관리

→ USER 테이블을 통해 사용자 ID와 PW를 확인하여 일치하는 경우 로그인 가능

→ 회원가입의 경우, 빈 애트리뷰트가 존재하지 않도록 존재하는 모든 애트리뷰트(아이디, 비밀번호, 성별)에 관한 정보를 전부 입력해야 가입 가능

* 사용자 옷장 관리

→ USER 테이블의 기본키 ID를 USER\_ID라는 외래키로 갖는 USER\_CLOTHES 테이블에서 사용자 옷장 관리

→ 사용자가 해당 테이블의 데이터를 조회, 수정, 삽입, 삭제 가능

→ CLOTH\_NAME은 기본키임으로 중복을 허용하지 않음

→ USER\_ID를 통해 각 사용자에 따른 자신의 옷장을 매칭할 수 있음

* 날씨 정보 관리

→ 제공되는 api 에서 필요한 정보만을 추출해 CLIMATE 테이블 생성

→ 일자 별로 날씨가 저장되므로 DATE를 기본키로 지정

→ 나머지 애트리뷰트(계절, 최고 기온, 최저 기온, 강수량, 풍속)을 모두 통해 추천해야할 옷의 유형을 지정. 이는 CASE라는 새로운 애트리뷰트 값으로 저장됨

* 쇼핑몰 관리

→ 효율적인 테이블 관리를 위해 임의로 인덱스를 생성함. 이를 SHOP\_ID라는 애트리뷰트로 두고 기본 키로 지정

* 옷 추천

→ 모든 옷 추천은 CLOTH\_TYPE 애트리뷰트를 고려해 상의, 하의 한 벌씩을 추천

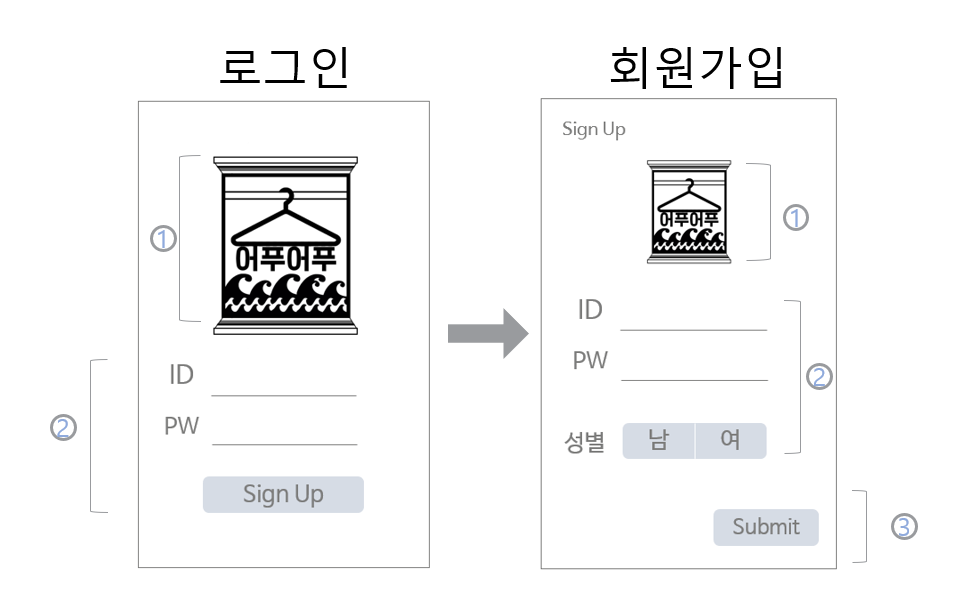
→ 개인 옷장에서 옷을 추천 하는 경우 : CLIMATE 테이블의 오늘 일자 데이터 중 CASE를 SELECT하여 USER\_CLOTHES의 데이터 중 같은 CASE 값을 가지는 데이터만을 SELECT

→ 쇼핑몰 옷을 추천하는 경우 : CLIMATE 테이블의 오늘 일자 데이터 중 CASE를 SELECT하여 SHOP\_CLOTHES의 데이터 중 같은 CASE 값을 가지는 데이터만을 SELECT

→ 쇼핑몰 옷을 추천할 때는 옷 마다 외래키인 SHOP\_ID를 통해 해당 옷을 판매하는 쇼핑몰로 링크를 통해 이동이 가능하도록 함

1. UI 설계

* 로그인 및 회원가입



(1) : 각 화면 상단 중앙에 로고를 배치

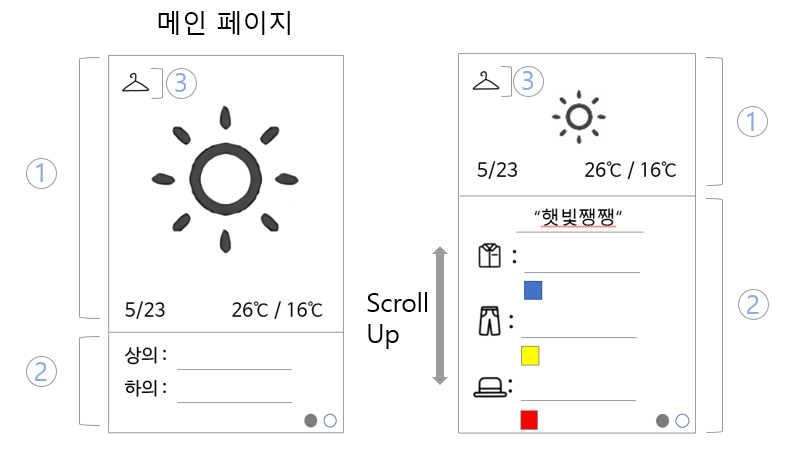
(2) : 로그인 : User의 ID, PW를 입력받아 Sign Up을 통해 로그인을 실시.

Sign Up 버튼 클릭시 메인페이지로 이동

회원가입 : User의 ID, PW, 성별을 입력받아 User 의 정보를 기록

(3) : Submit 버튼을 통해 User생성

* 메인페이지

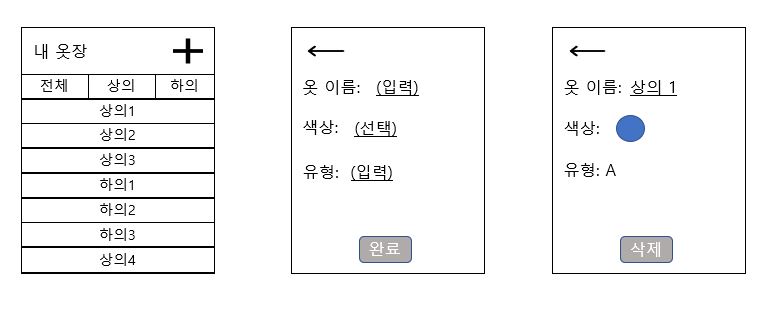


(1) : 화면 상단에 날씨정보 및 아이콘 배치

(2) : 화면 하단에 회원에 대한 옷추천 뷰를 배치

뷰 안에는 User\_clothes, Shop\_clothes 의 정보를 출력

(3) : 아이콘 클릭을 통해 “내 옷장” 으로 뷰 전환



(1) : “내 옷장” 전환 후 현재 사용자가 가지고 있는 옷들의

리스트를 표시하고 이는 전체, 상의, 하의 크게 3개의 탭으로

구성

(2) : 현재 가지고 있는 리스트를 클릭하면 옷에 대한 정보가 표시

(3) : + 아이콘 클릭을 통해 사용자가 구매한 옷에 대한 정보를

추가

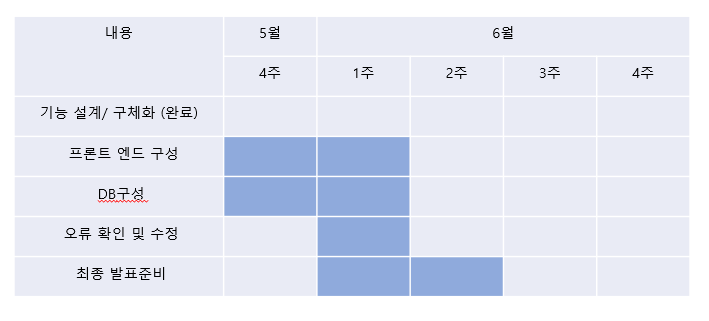
(4) : 사용자가 현재 가지고 있는 옷을 길게 클릭하면 삭제 여부

메시지가 표시

1. 추진 일정
2. 현재까지의 진행 내용

* 앱 기능 설계 및 구체화 실시
* DB구조 설계
* 추천 알고리즘 설계
* UI 설계 및 개발 중

1. 최종 구현까지의 일정



1. 앞으로의 보완점

* 중복이 적은 알고리즘을 사용함으로써 DB 속도 향상
* 효율적인 날씨추천 알고리즘을 위해 세부적인 추천유형 고안
* 전반적인 기능을 사용자가 이용할 수 있는 APP 제작

1. 설계 과정에서의 중점 사항

* 각 릴레이션의 키 및 애트리뷰트 중복을 최소화
* 데이터베이스를 구성하는 릴레이션들 간의 관계 명확성
* 주요 기능과 부가 기능의 시뮬레이션을 통한 프로그램의 전반적인 흐름 파악

1. 구현 과정에서의 중점 사항

* 데이터베이스에 접근하여 데이터를 불러올 때, 실행시간이 가장 빠른 효율적인 SQL문 사용
* 사용자가 이용하기 편리한 인터페이스 구현