Weatherlish(웨더리쉬)

Team 8

142095 강지우, 134291 문성철, 161356 정재철

Dept. Electronics and Computer Engineering

Chonnam National University

Gwangju

I.개발 동기 및 목표

1. 개발 동기

현대인들의 고민 중 하나는 "아침에 뭐 입고 나가지?" , "내일 뭐 입지?"등 이 있습니다. 그리고 막상 옷을 입고 나가면 날씨와 맞지 않아 속상할 때가 있습니다.

저희 조는 그런 일이 생기지 않도록 그날의 날씨 정보, 날씨에 맞는 일상생활 팁을 알려줄 뿐만 아니라 날씨에 어울리는 코디를 추천해 주는 프로그램 개발하고자 합니다. 사용자는 실시간으로 사용자들은 날씨 정보와 그에 맞는 생활 정보를 얻을 수 있습니다. 또한 아침에 옷 고르는데 지체되는 시간을 단축되고 다양한 패션 정보를 통해 쇼핑 트렌드를 발견할 수 있습니다. 그로 인해 소비자는 자신의 제한된 의상으로 인한 단조로움을 해결할 수 있으며, 쇼핑정보와의 연계로 패션시장에도 큰 도움이 될 것입니다.

그래서 저희는 프로그램의 이름을 '날씨'를 뜻하는 영어 웨더(Weather)와 '유행을 따른'을 뜻하는 단어인 스타일리쉬(Stylish) 두 단어를 합성하여 웨더리쉬 (Weatherlish)로 정했습니다.

2. 목표

- 분할된 어플들의 기능(날씨, 코디)을 통합화
- 사용자들이 직관적으로 이해하고 사용가능한 인터페이스 디자인.
- Open Weather Map을 통해 실시간으로 정확한 날씨 정보의 활용
- 생활정보와 옷에 관련된 페이지 이동의 간편화
- 체형과 선호 스타일에 따른 코디 자동화

II.어플리케이션 특징

A.관련 응용프로그램

Windows 10 날씨 앱 - Windows 10에 기본 탑재돼 있는 응용프로그램으로 전세계의 날씨를 쉽게 볼 수 있게 해주는 응용프로그램 중 하나입니다.

호우호우 - 정확한 날씨와 체감온도, 상황별 특보, 황사 및 미세먼지 예보를 아기자기한 캐릭터를 통해 알려주는 앱입니다.

코디북 - 사용자에게 패션, 코디의 맞춤 정보를 제공하는 앱입니다.

B.응용 프로그램의 특성

웨더리쉬는 날씨를 알고 싶은 지역의 정보를 입력하면 그 지역의 날씨와 기온, 날씨에 맞는 일상생활 팁 뿐만 아니라 사용자의 체형, 스타일과 상황에 맞는 코디 정보도 불러와줍니다. 이 때, 관련 상품을 구매할 수 있는 쇼핑몰의 링크를 소비자에게 제공해줍니다.

예를 들어 햇살이 가득한 날이면 외출하기 전에 자외선 차단제를 바르고 가는 것이 좋다거나, 하늘색 같이 땀을 흡수한 부분이 잘 보이는 옷은 피하라는 등의 생활 팁을 알려주면서 모자나 티셔츠 등을 추천해줍니다. 비가 오고 있다면 우산을 챙기거나 여분의 양말을 챙기는 것이 좋겠다는 생활 팁을 알려주면서 부츠나 레인코트, 우산 등을 추천해줍니다.

기존의 날씨 앱은 날씨 정보만을 전달하지만, 그에 비해 저희가 개발한 프로그램은 날씨와 생활 팁, 거기에 더해 코디 정보까지 알려줄 수 있기 때문에 기존의 앱과의 차별성을 두었습니다.

III.FUNCTION SPECIFICATION

<회원가입>

Void 회원정보입력(이름, 아이디, 패스워드, 체형, 취향) string
: user는 로그인 시 회원 정보를 입력한다.

boolean 아이디 중복성검사(아이디, 패스워드)
: 기존 회원들의 아이디와 중복성을 검사한다.

Void 체형, 취향확인(체형, 취향)
:회원의 체형과 취향을 확인한다.

Object_user 회원정보저장(이름, 아이디, 패스워드, 체형, 취향)
:회원의 정보를 DB에 저장한다.

<날씨보기>

Boolean 로그인(String 아이디,String 패스워드)
:아이디와 패스워드를 입력받아 등록된 회원의 경우 TRUE 값을 리턴.

String 회원위치(ObjectUser)
:회원 정보에 저장된 위치를 확인한다.

String 위치입력(String 위치)
:회원으로부터 검색할 위치를 입력받는다.

String 날씨처리(String 위치)
:새로운 지역의 날씨의 정보를 가져온다.

String 날씨파싱(String URL)
:회원이 검색한 위치에 날씨정보를 파싱한다.

Void 날씨정보표현(String 날씨정보)
:창에 날씨정보를 디스플레이한다.

<로그인>

Boolean 입력값 검사(String 아이디,String 패스워드)
: 아이디와 패스워드가 입력이 되었는지 체크

Boolean 유효성 검사(String 아이디, String 패스워드)
: 아이디와 패스워드가 입력되어야 하는 형식에 맞는지 확인

Void 아이디 요청()
: 데이터 베이스에 저장되어 있는 아이디를 가져온다.

Void 패스워드 요청()
: 데이터 베이스에 저장되어 있는 패스워드를 가져온다.

Boolean 아이디, 패스워드 검사()
: 아이디와 패스워드가 데이터 베이스에 있는지 검사한다.

<날씨에 따른 코디>

void initialFile()

:파일로부터 옷객체를 가져온다.

void parameterTranslate(String styleSt, String sizeSt, String body)
:db로부터 가져오는 데이터들을 정리한다.

void weightAssignment(int temp)
:온도에 따른 가중치를 할당한다. (온도가 매우 낮거나 높은 날은 옷의 두께감에 가중치를 높게, 그렇지 않은 날에는 스타일에 가중치를 높게 둠.)

void produceClothMatrix()
:옷 객체들이 갖는 변수들의 점수와 가중치를 계산한 값을 갖는 Matrix를 생성한다.

void producePriorityMatrix(int n)
:ClothMatrix의 값들을 우선순위에 따라서 정렬한 PriorityMatrix를 생성한다.

matchCloth(int temp, String styleSt, String sizeSt, String body)
:날씨의 온도와 회원의 정보를 통해 ClothMatrix와 PriorityMatrix를 생성한다.

codiObject(matchCloth match)
:matchCloth가 갖는 PriorityMatrix를 통해서 최상위 순위의 옷들로 코디를 매치해 완성된 codi객체를 생성한다.

void refreshCodi(int temp, String stylest,String sizest,String body)
:코디를 새로 고침 해준다.

codiDisplay(String sex, String id, String temp, String stylest, String sizest, String body)
:메인에서 직접적으로 생성될 class로 코디 Panel에 코디를 그려준다.

<생활 정보 보이기>

String 생활정보요청()
:날씨 정보에 맞는 생활 정보를 불러온다.

String 날씨처리(String 위치)
:지역의 날씨 정보를 가져온다.

Void 쇼핑몰링크요청(string link)
:생활정보에 맞는 링크를 요청한다.

Void 메인화면()
:메인화면을 보여준다.

<관리자 - 생활정보, 코디 추가>

void 업로드코디(String clothpath,String genderpath,String filename)
:옷에 관련된 파일을 DB에 업로드한다.

void 업로드생활정보(String tip, String link)
:관리자는 생활정보를 DB에 업로드 한다.

<날씨 파싱하기>

String 검색한 지역의 날씨(String location, String url)
: OpenWeatherMap의 url값에 지역을 검색하여 정보 얻기

Void Json내에 접근(String data)
: 웹 상에서 얻어온 날씨 값의 형태인 Json을 저장

Weather_Object Data선정 (String Json)
: Json형태로 되어 있는 값들을 선별해서 저장.

String 날씨표현(Weather_Object 날씨정보)
: 선별된 날씨를 화면 상에 표현

<회원 정보 수정>

UserObject 디비에서 유저정보 얻기()
: 디비에서 유저의 아이디와 매칭되는 정보를 얻어온다.

Void 유저정보 표현(UserObject us)
: 디비에서 얻어온 유저 정보를 화면상에 표출

UserObject 유저정보 수정 ()
: 화면상의 수정된 유저정보를 디비에 보낸다

Void 데이터베이스 Update(UserObject us)
: 데이터베이스 내의 사용자 정보를 수정시킨다.

IV.REQUIREMENT LISTS

Fig. 1. 사용자

- 사용자는 등록 시 ID(중복X), 이름, 패스워드, 성별, 체형, 스타일, 지역 등을 입력하여야 한다.
- 등록된 사용자는 ID로 로그인이 가능하다.
- 사용자는 언제든지 자신의 정보를 바꿀 수 있다.

Fig. 2. 날씨

- 사용자의 즐겨 찾는 지역의 날씨가 먼저 디스플레이 된다.
- 날씨, 온도, 습도, 최고온도, 최저온도 확인이 가능하다.
- 하루에 3시간 단위 날씨를 업데이트한다.
- 5일의 날씨 확인이 가능하다.
- 지역 검색이 가능하다.

Fig. 3. 코디

- 성별, 체형, 선호스타일에 맞는 코디를 추천해준다.
- 사용자의 취향에 따라 다른 코디 추천도 받을 수 있다.
- 추천해준 코디의 쇼핑몰 링크를 걸어준다.

Fig. 4. 생활 정보

- 날씨에 맞는 생활정보를 알려준다.
- 필요한 생활용품은 판매처 링크를 걸어준다.
- 관리자 페이지를 통해 새로운 생활 정보를 추가할 수 있다

Fig. 5. 관리자

- 새로운 생활정보를 업데이트가 가능하다.
- 새로운 옷들을 업데이트가 가능하다.

V.USECASES

1. 사용자 등록 및 로그인

1.1 사용자 등록이 되어있지 않는 경우, 사용자는 이름, 성별, 페스워드, 성별, 체형, 스타일 지역등을 입력하여 회원 정보를 저장한다.

1.2 사용자 등록이 되어있는 경우, 자신의 이름을 작성하여 다음 화면으로 넘어간다.

1.3 사용자는 자신의 정보를 수정할 수 있다.

2. 자신의 원하는 지역을 검색한다.

3. 사용자가 원하는 지역의 날씨를 불러온다.

4. 날씨에 맞는 생활 정보를 제공한다.

4.1 사용자가 구매하고 싶은 생활용품 판매처링크를 걸어둔다.

5. 사용자의 정보와 날씨를 매칭하여 그에 맞는 코디를 불러온다.

5.1 불러 온 코디를 사용자가 클릭하면 판매처의 링크를 연결해준다.

5.2 사용자의 취향에 따라 다른 코디를 추천 받는다.

6. 관리자 로그인

6.1 관리자는 생활정보를 업데이트한다.

6.2 관리자는 새로운 옷을 업데이트한다.

VI.usecase diagram

사용자는 로그인 후 날씨 검색, 코디 정보, 생활정보를 얻을 수 있다.

OpenWeatherMap에서 날씨 데이터를 전송 받아 날씨에 대한 정보를 확인할 수 있다.

관리자는 정기적으로 코디정보와 생활 정보를 업데이트한다.

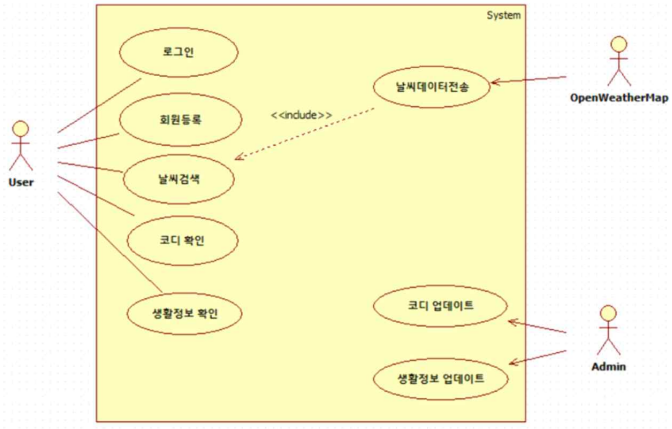

Fig. 1. Usecase Diagram

VII.sequence diagram

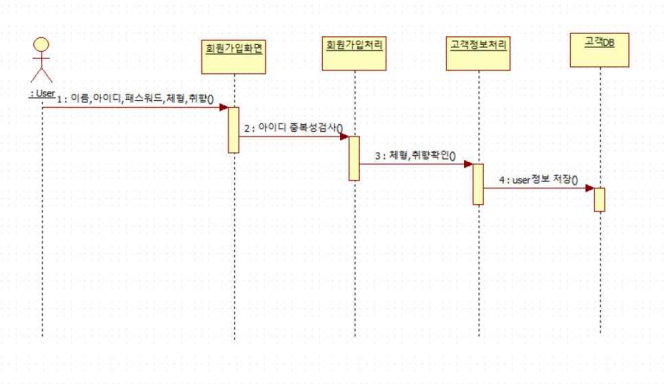

Fig. 1. 회원가입 시퀀스 다이어그램

User는 회원 가입 화면을 통해 자신의 정보를 입력하고, 입력한 정보는 회원 DB에 저장된다.

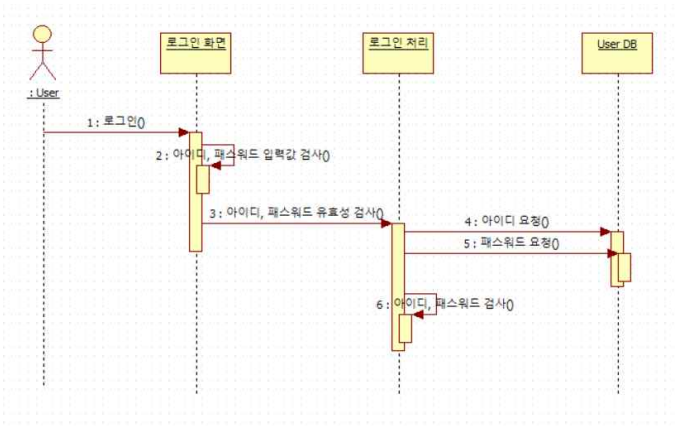

Fig. 2. 로그인 시퀀스 다이어그램

User는 로그인 화면에서 ID와 PASSWORD를 입력하고 User 데이터베이스에 저장된 정보와 검사해 로그인을 한다.

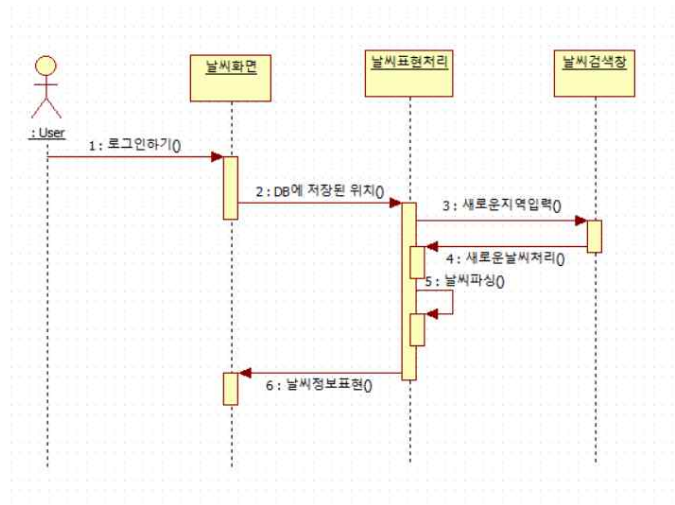

Fig. 3. 날씨 검색

User가 로그인을 하면 데이터베이스로부터 User의 위치 정보를 불러와 날씨를 보여준다. User는 검색창을 이용해 새로운 지역의 날씨 정보를 얻는다.

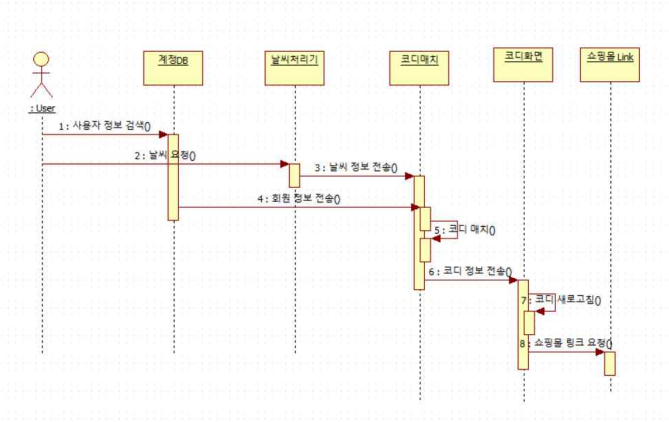

Fig. 4. 코디 추천

User는 회원가입 시 입력한 정보를 토대로 코디를 요청한다. 그 외의 추가의 코디를 요청할 수 있으며, 쇼핑몰 Link정보도 요청할 수 있다.

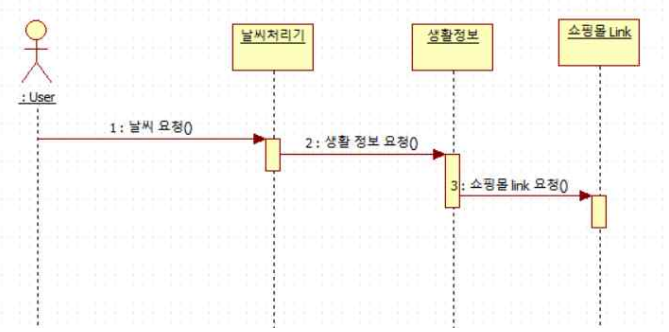

Fig. 5. 생활정보 추천

User는 날씨에 맞는 생활정보를 얻는다. 생활정보 역시 쇼핑몰Link를 통해 접근이 가능하다.

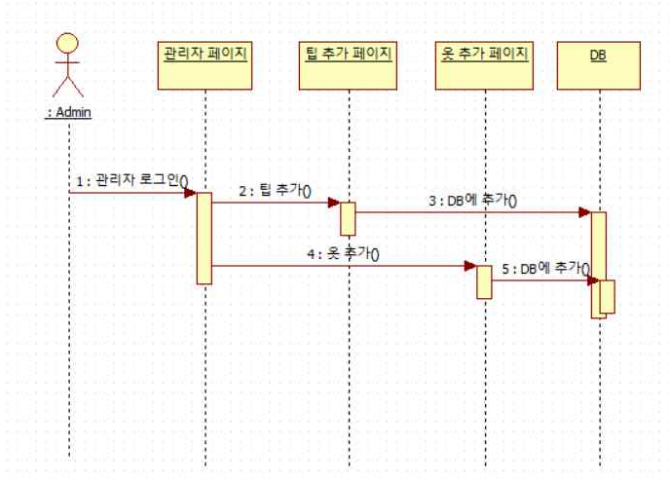

Fig. 6. 관리자 DATA 추가

Admin은 팁, 옷을 업데이트 한다.

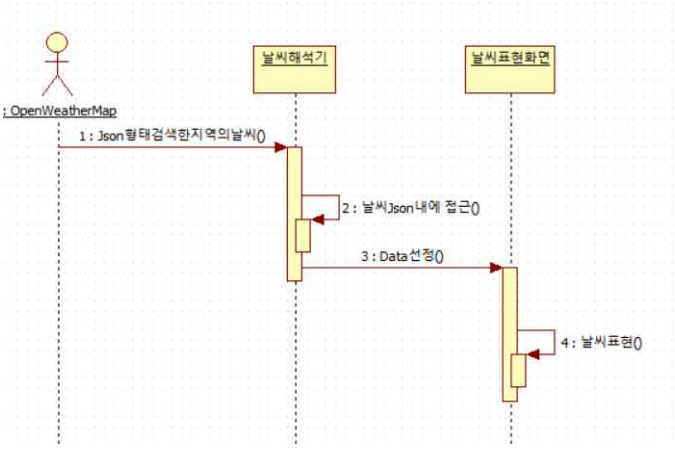

Fig. 7. 날씨 불러오기

날씨 Json내에 접근해 data를 선정하고 날씨 화면에 날씨를 표현한다.

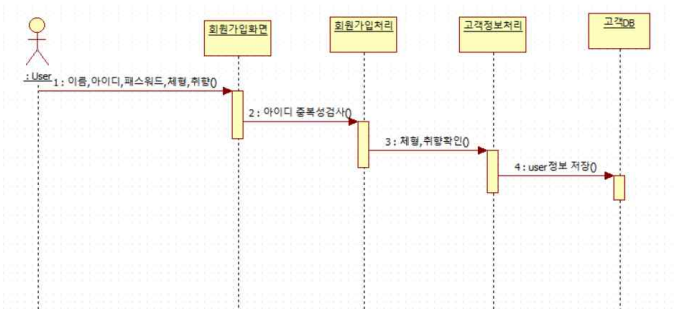

Fig. 8. 회원가입 시퀀스 다이어그램

User는 회원 가입 화면을 통해 자신의 정보를 입력하고, 입력한 정보는 회원 DB에 저장된다.

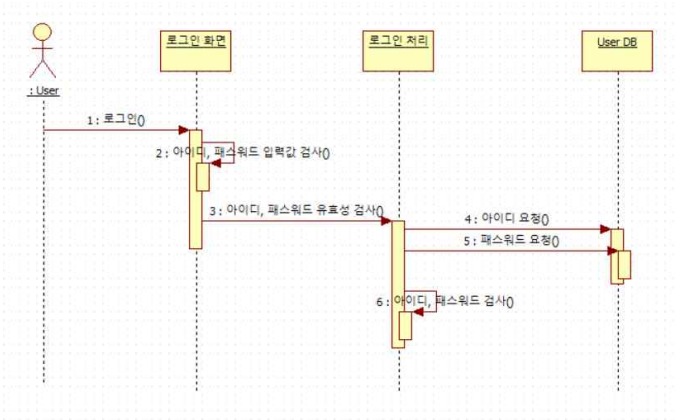

Fig. 9. 로그인 시퀀스 다이어그램

User는 로그인 화면에서 ID와 PASSWORD를 입력하고 User 데이터베이스에 저장된 정보와 검사해 로그인을 한다.

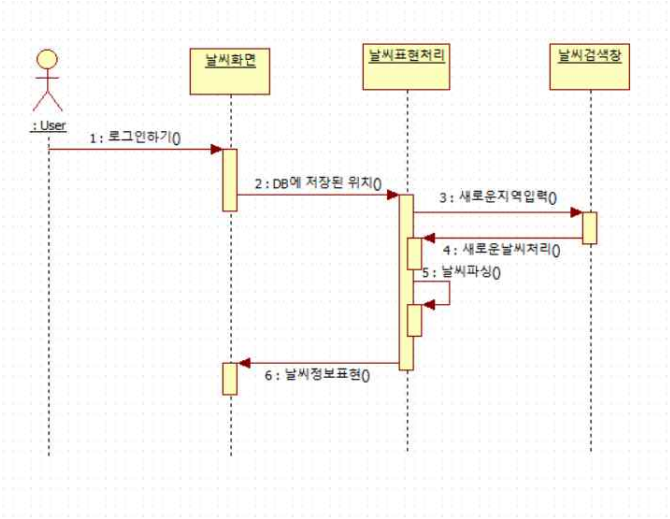

Fig. 10. 날씨 검색

User가 로그인을 하면 데이터베이스로부터 User의 위치 정보를 불러와 날씨를 보여준다. User는 검색창을 이용해 새로운 지역의 날씨 정보를 얻는다.

VIII.User Interface Designs

Fig. 1. 로그인 화면

- 프로그램 실행 시, 로그인 창이 뜨게 됩니다. 회원가입이 된 회원은 아이디와 패스워드를 입력 하여 로그인하여 메인화면으로 이동하고, 비회원의 경우, 회원가입하는 창으로 이동할 수 있습니다.

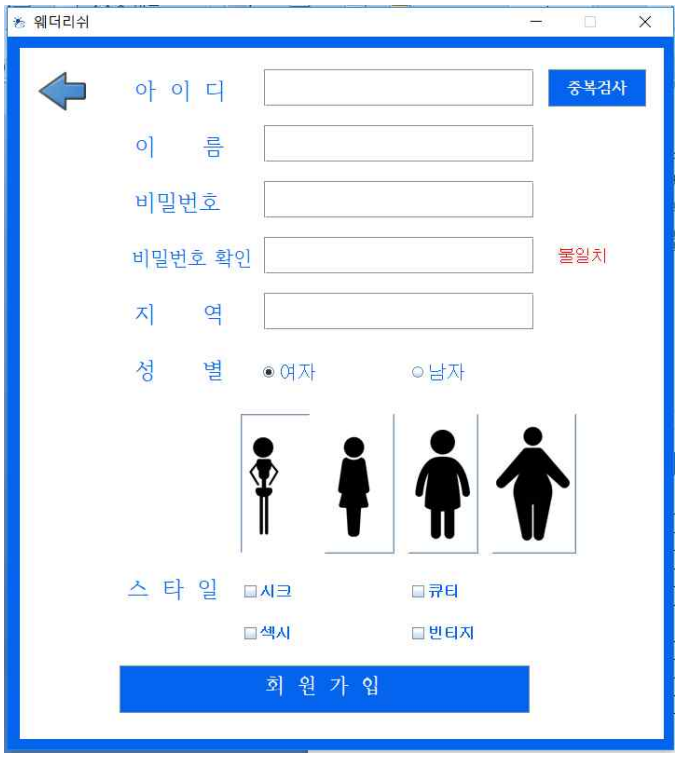

Fig. 2. 회원가입 화면

- 회원 계정을 생성하기 위해, 비회원의 경우는 회원가입 화면으로 이동하게 됩니다. 이 때, 데이터가 전체적으로 입력이 다 되었는지 확인하고, DB에 다른 회원 데이터 간의 중복성 검사를 합니다. 체형는 4가지 중 하나 선택이 가능하고, 스타일 같은 경우는 원하는 스타일은 중복 선택이 가능합니다.

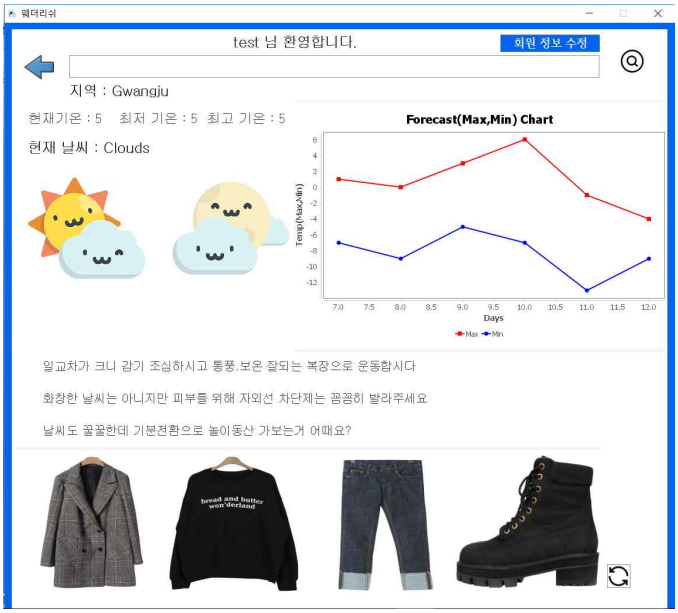

Fig. 3. 메인 화면

- 기존 회원이 아이디와 패스워드를 입력하고 로그인을 하게 되면, 다음과 같은 메인 화면이 나타납니다. 사용자가 회원가입 시 입력한 지역을 바탕으로 메인 화면에 날씨가 Display됩니다. 그 외의 지역을 검색하고 싶을 경우, 화면 상단에 있는 검색창에 지역을 검색하게 되면 날씨가 수정 되어 나타납니다.

- 중간 패널 부분엔 날씨에 맞는 생활정보 팁을 보여줍니다. 쇼핑몰과 연동이 필요한 팁이 추천이 될 경우 링크를 통해 웹페이지로 이동할 수 있도록 하였습니다.

- 메인 화면 하단에 보시면 해당 날씨에 알맞은 코디가 추천되어 집니다. 사용자는 우측 하단에 새로고침 버튼을 통해 새로운 코디를 추천 받을 수 있습니다. 추천된 의상의 구매를 원할 시 옷마다에 링크가 부여되어 있기 때문에 클릭 한번으로 해당 쇼핑몰로 이동이 가능하게끔 하였습니다.

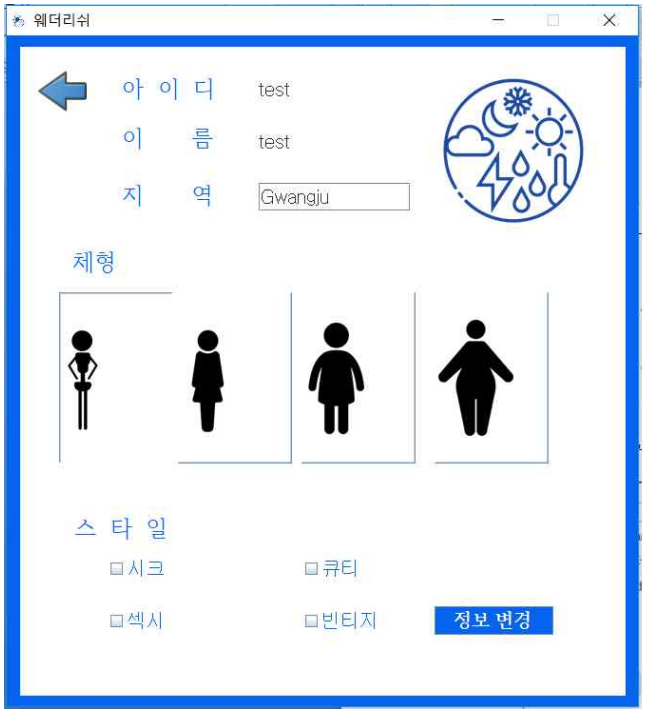

Fig. 4. 회원 정보 수정 화면

- 메인 화면 상단 위에 '회원 정보 수정' 버튼을 누르면 자신의 정보를 수정할 수 있게 됩니다. 아이디와 이름을 제외하고, 지역, 체형, 스타일을 변경하면 DB에 저장되어 로그인 시 변경된 정보가 반영됩니다.

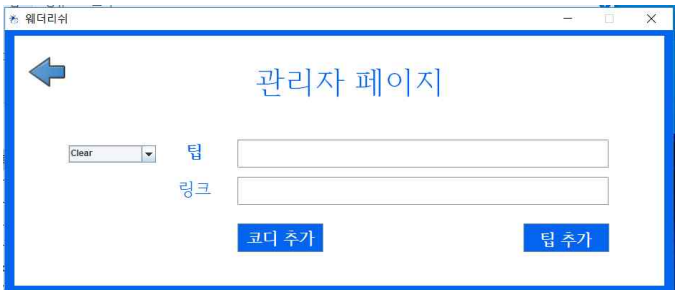

Fig. 5. 관리자 화면 – 팁 추가

- 관리자가 로그인 시, 관리자 페이지로 이동하여 팁을 추가 할 수 있습니다. 좌측에서 날씨를 선택한 후 팁과 링크를 추가합니다.

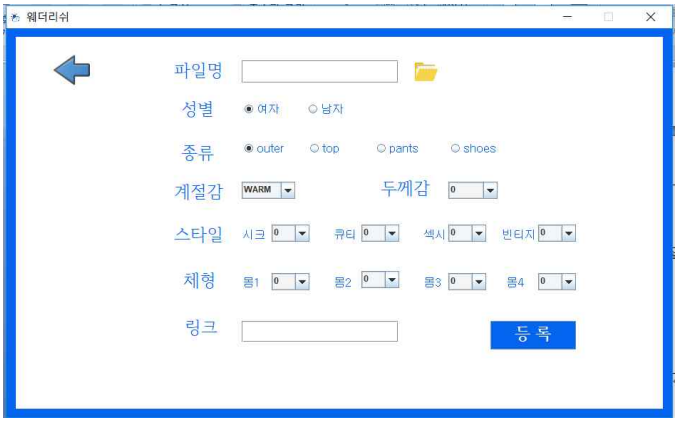

Fig. 6. 관리자 화면 – 옷 추가

- 관리자가 로그인 시, 관리자는 새로운 옷들을 추가 할 수 있습니다. Directory 창을 통해 추가할 파일을 선택하고 해당 옷에 알맞은 정보를 입력한 후 등록 합니다.

IX.Class Diagram

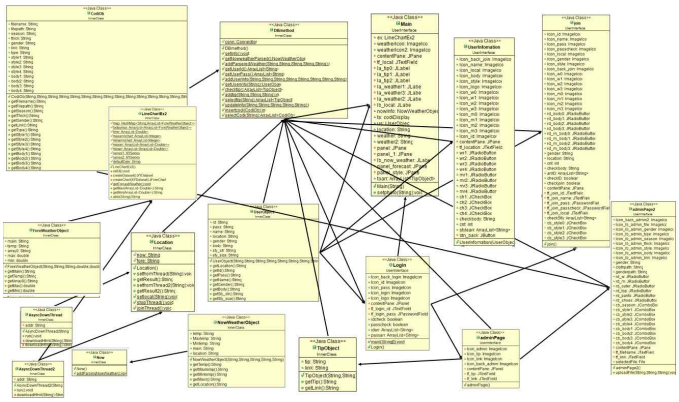

Fig. 1. Class Diagram(날씨, 팁)

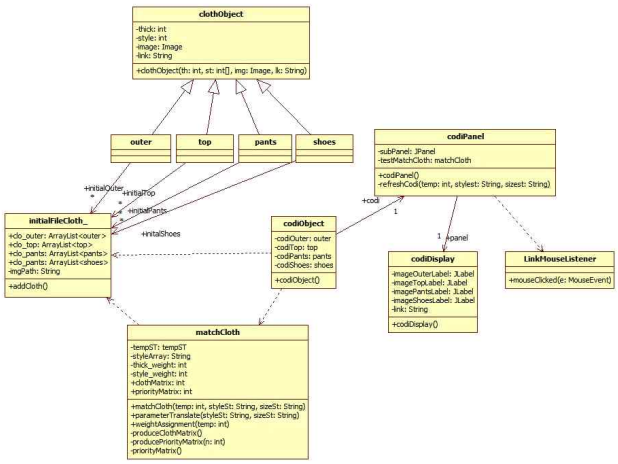

Fig. 2. Class Diagram(코디 관련)

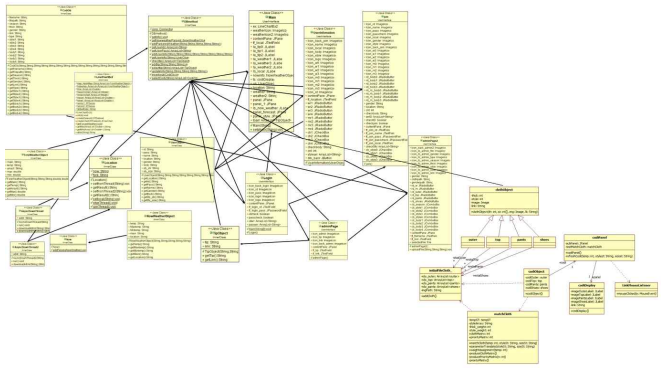

Fig. 3. Class Diagram(전체)

X.Screenshot of API Manual

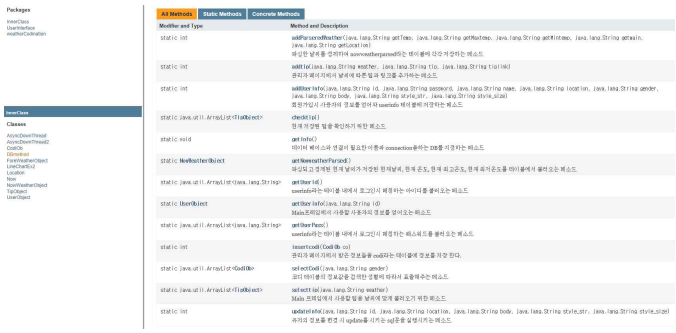

Fig. 1. API(1)

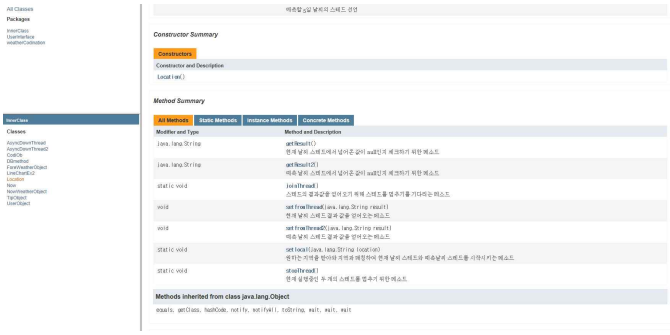

Fig. 2. API

XI.Screenshot of Application demo

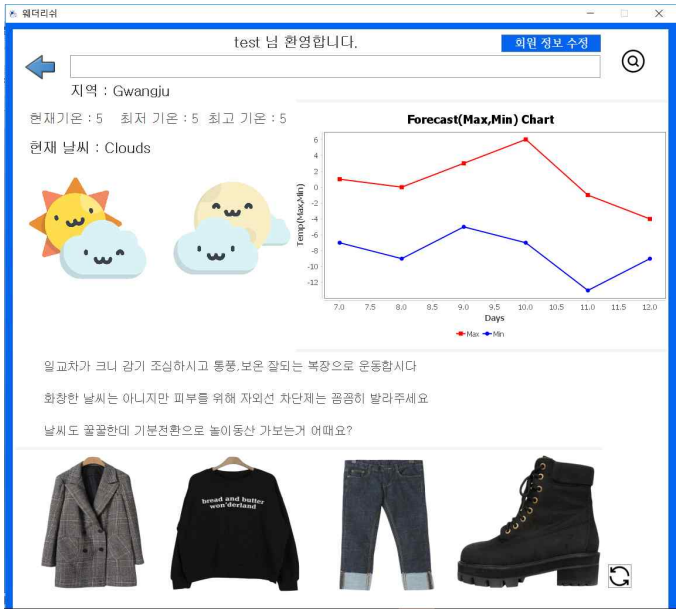

Fig. 1. 메인화면

- 해당 프로그램은 test 회원으로 여자, 광주 지역에 거주하고 선호스타일로 큐티, 마른 체형의 회원이 실행한 프로그램입니다. 현재 날씨가 매우 춥기 때문에 스타일보다는 두꺼운 옷들을 더 고려한 코디가 나온 화면입니다.

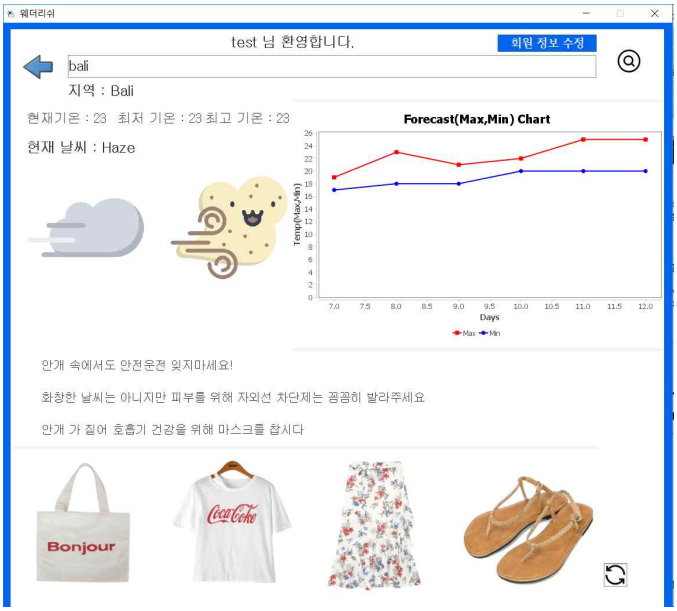

Fig. 2. 더운 지역(Bali)을 검색했을 경우

- 더운 지역에 해당하는 큐티하게 코디된 옷이 나왔음을 볼 수 있습니다.

XII.Screenshot of Github Pages

>>GitHub 주소 : https://github.com/SungChulMoon/Project2

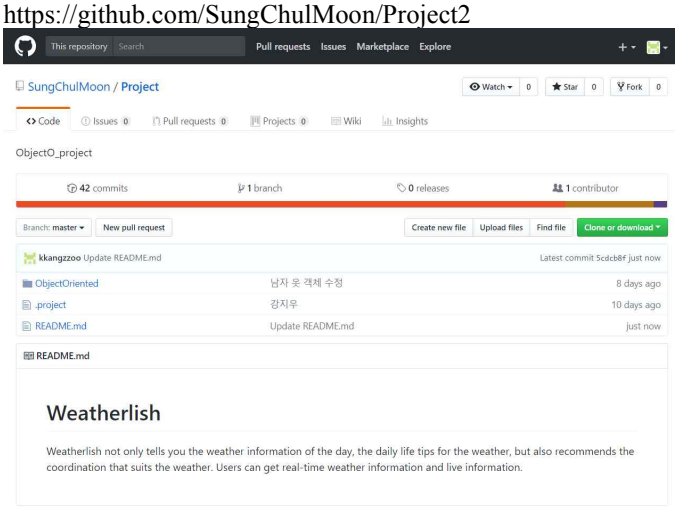

Fig. 1. Github Main Page

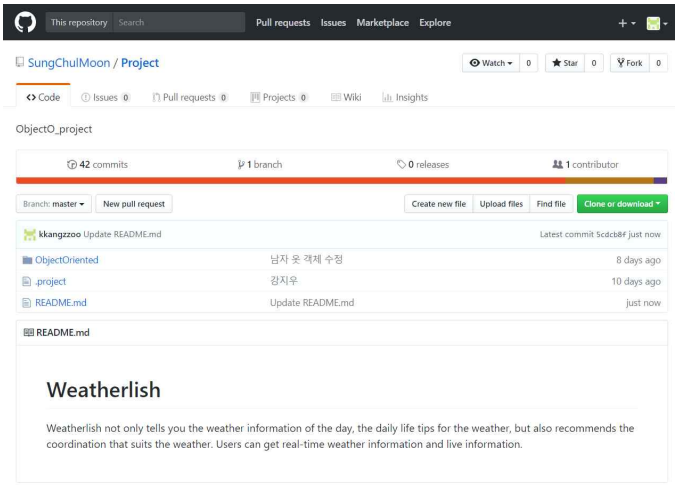

Fig. 2. History

XIII.Work distribution & 진행 결과

1. 역할 분담

이름	역할
강지우	**<코디 정보 보이기>** -코디 인터페이스 구현 -사용자에 맞는 코디 정보 불러오기 -새로운 코디 불러오기 -쇼핑몰 링크 연동 **<옷 업데이트>** -옷 파일 업데이트 **<관리자 옷 추가 페이지>** -인터페이스 구현 - 파일 처리 관리 **<DataBase>** - 옷 개체 생성 - 옷 DB 정규화 - 옷 DB 접근 시 정보 추출
문성철	**<날씨 정보 파싱>** -웹에서 날씨 정보 얻기 -웹에서 얻은 날씨 정보 저장 -날씨 값 선별 **<날씨 보이기>** -회원 정보 위치에 맞는 날씨 보이기 -새로운 지역 날씨 불러오기 -날씨 인터페이스 구현 **<회원가입>** -회원 정보 입력 받기 -회원 정보 DB로 접근 **<로그인>** -ID, PASSWORD 입력 받기 -회원 DB에 ID, PASSWORD 검사 **<DataBase>** - 회원 개체 생성 - 회원 DB 정규화 - 회원 DB 접근 시 정보 추출 **<유저정보 업데이트>** - 기존 유저의 정보 변환
정재철	**<생활 정보 보이기>** -생활 정보 인터페이스 구현 -생활 정보 보이기 -쇼핑몰 링크 연동 **<생활 정보 업데이트>** -생활 정보 업데이트 -쇼핑몰 링크 업데이트 **<GUI 디자인>** -전반적 인터페이스 디자인 **<DataBase>** - 생활정보 개체 생성 - 생활정보 DB 정규화 - 생활정보 DB 접근 시 정보 추출

2. 진행 결과

날씨는 OpenWeatherMap을 통해 실시간으로 날씨 정보를 파싱해와 사용자들이 쉽게 보고 파악할 수 있는 GUI를 구현하였습니다. 그러나 그래프 객체에 대한 이해 부족으로 예상 날씨 정보 표출이 부족합니다.

생활 정보 추천에선, DB에 해당 날씨와 관련된 여러 팁들과 링크를 저장하여 날씨의 변화에 따라 알맞게 생활정보를 추천해주는 부분은 구현 되었습니다.

하지만 생활 정보가 매우 한정적이기 때문에 다양한 추천이 어렵습니다

코디 부분에선, 날씨에 적합한 코디를 추천하기 위해서 고려해야할 사항이 매우 많았기 때문에, 소프트웨어적으로 구현할 수 있는 알고리즘을 짜기가 매우 까다로웠습니다. 그렇기 때문에 널리 통용되있는 생각들을 바탕으로 알고리즘을 구현하려 하였으며, 그에 알맞는 코디는 추천이 가능합니다. 하지만 유행에 따라 추천되는 부분에 있어서 함수적인 구현이 어려웠으며, 기온에 따른 추천이기 때문에 나라별 계절감에 대한 고려는 하지 못하였습니다.

이 어플을 통해서 사용자는 실시간으로 날씨 정보와 그에 맞는 생활정보와 코디를 추천 받을 수 있기 때문에 세부적인 부분은 부족함이 있을 수 있으나 현재 구현이 가능한 부분에선 진행 결과가 완전히 이루어 졌음을 알려드립니다.

<라인 수>

문성철 : 1904 lines

강지우 : 1154 lines

정재철 : 774 lines