# **[2021-2 Software Engineering]**

시스템 설계서(SDD)

제목:

제출일: 2021 년 11 월 19 일

팀 명: USB

팀 원: 심효섭, 오승민, 최유림, 전상우

1. 개발 시스템 개요

인터넷 검색은 21세기에 빠질 수 없는 필수 기능이다. 많은 사람들이 여러 번의 검색을 시도한다. 하지만 검색 결과는 항상 자신이 원하는 내용이 위에 있지 않는다. 수많은 검색 결과 중 자신이 원하는 결과를 찾기 위해 사람들은 많은 시간을 보내고 있다. 우리 팀은 이를 해결하기 위해 사용자의 평소 취향을 분석하고 이에 맞게 검색 결과를 추천해주는 시스템을 개발하여 사용자의 검색 효율을 향상시키고자 한다.

1.1. 개발 목표

사용자의 요구 사항은 지나치게 많은 검색 결과에서 자신이 원하는 결과만 보고싶다는 내용이다. 이를 위해서는 사용자가 평소에 어떤 장르에 흥미를 가지고 있는지 알고 이를 통해서 검색 결과를 해당 장르로 다시 정렬하는 과정을 진행해야 한다. 본 프로젝트에서는 사용자의 취향을 파악하고 검색을 했을 때 검색 결과를 다시 취향에 맞게 재정렬 해주는 시스템을 개발하고자 한다. 우리 프로젝트의 차별성은 포털 사이트와 다르게 검색 결과를 사용자 맞춤형으로 제공해준다는 점에 있다.

1.2. 주요 기능

1. 회원가입 및 로그인
2. 검색 결과 추천
3. 취향 분석
4. 취향 업데이트
5. 취향 시각화

우선 로그인을 통해 회원정보를 DB에서 선택한다. 만일 회원 가입이 이루어 지지 않은 상태면 회원가입을 통해 아이디, 비밀번호, 생년월일, 성별, 이메일을 등록해, DB에 저장한다. 저장 후, 사용자는 개인 선호 카테고리 설정을 통해 관심사를 등록하고 이에 기반한 검색이 이루어진다. 검색이 많을 수록 데이터가 축적되고 이를 머신러닝을 통해 추후에는 처음에 선정한 데이터가 아닌 학습된 데이터를 기반으로한 검색이 이루어 지게 할 계획이다. 다만 선호사항 설정을 그대로 두어 이를 기반한 검색이 여전히 가능하도록 유지할 계획이다. /\* 여기는 잘 모르겠습니다 \*/

1.3. 설계상 제약 사항 (및 계획서 대비 변경된 점)

해당 프로젝트를 시작하면서 생긴 큰 문제로는 기술적인 어색함 및 미숙함이 였다. 어떤 팀원은 플라스크등의 라이브러리를 통해 프로그래밍 언어가 html과 연동하는 법을 몰라 이를 새로 배우는데에 많은 시간을 투자했고 이에 따른 시간 지연이 있었다. 또한 DB가 코드랑 연동이 쉽게 이루어지지 않아 이에 따른 불이익도 경험하였다. 특히 플라스크를 통해 파이썬 환경에서 동적으로 웹페이지에서 데이터를 불러와 이를 파이썬 환경에서 입력받은 데이터를 통해 sql문을 작성 및 실행하였다. sql문을 작성하는데에 사용된 파이썬 라이브러리는 sqlAlchemy로 라이브러리를 통해 DB에서 조건에 만족하는 sql문을 코드 단위에서 보다 쉽게 검색할 수 있도록 설정했다. 해당 프로젝트의 핵심은 성향에 따른 검색정렬로 이를 위해 적용되는 기술로 개발에는 큰 문제가 없어보인다. 다만 초기에는 사용자의 검색 결과에따라 머신러닝을 통한 선호도 조정 까지 구현하려 했지만 아직은 시기상조라 판단해, 선호도 설정은 수동으로 대체 되었다

다만 아직까지는 프로젝트 전체적인 관점에서 바라보았을 때, 계획이 크게 수정되어야 하는지를 판단하기는 어렵다.

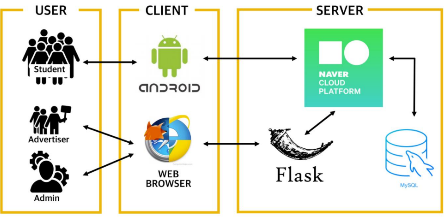
1.4. 용어정리

선호사항: 총 32개의 카테고리로 그 중 3개를 선택해, 사용자의 취향을 db에 저장해, 검색시, 이를 통한 정렬이 이루어진다.

취향분석: 검색 알고리즘 내의 프로세스를 포괄적으로 나타내는 표현으로 db에 연동해 선호사항을 기반으로 검색 정렬 프로세스

2. 시스템 구조

2.1. 전체 시스템 구성도

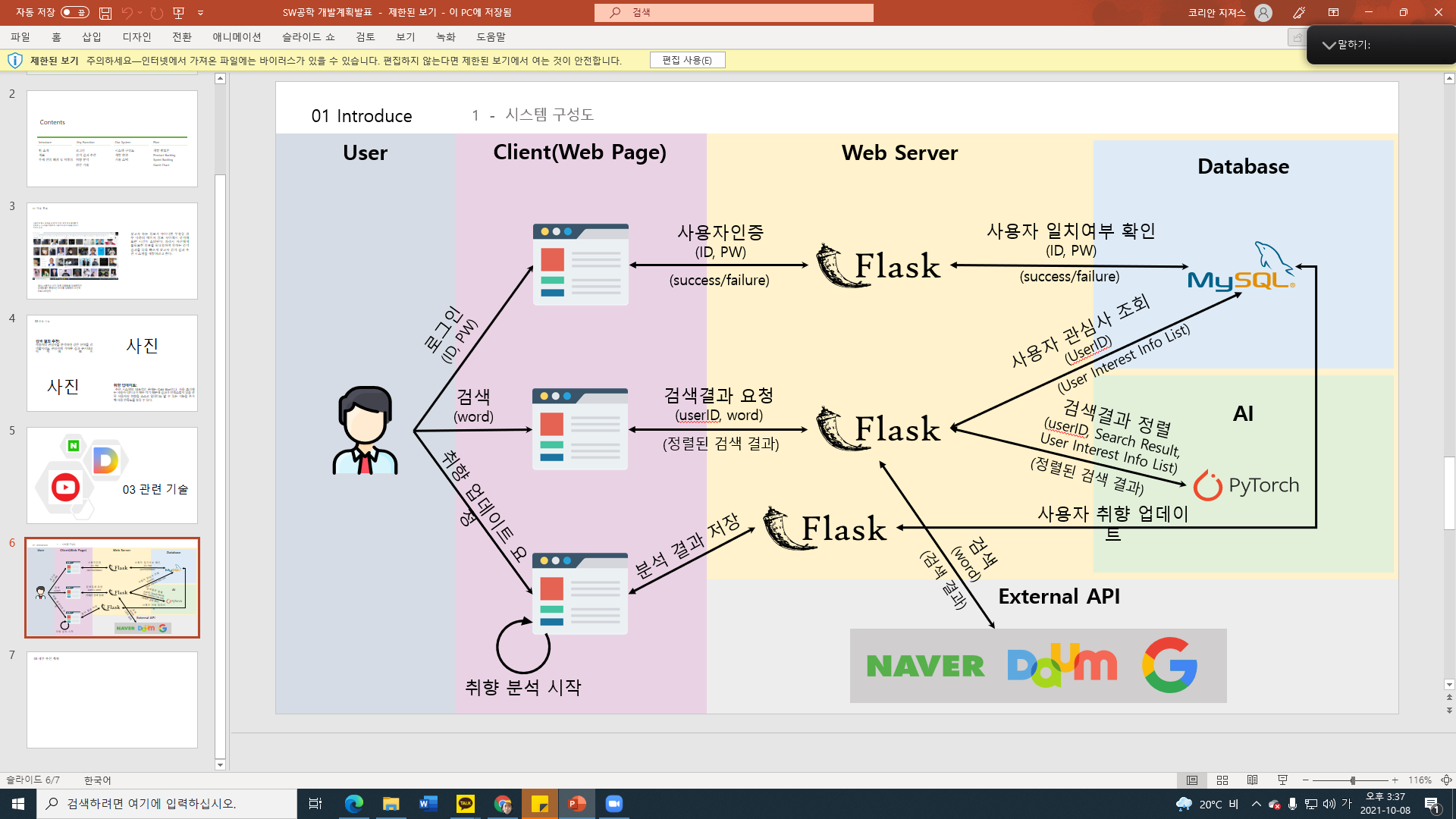


프로그램의 서버는 파이썬 환경에서 제작 되었으며, 파이썬 라이브러리 플라스크를 활용해 html5 웹 페이지와 연동해 서비스를 제공할 예정이다. DB는 MYSQL환경에서 제작되었고, 유저는 웹브라우저를 통해, 서비스를 제공 받을 수 있다.

2.2. 주요 기능별 구조와 데이터 흐름

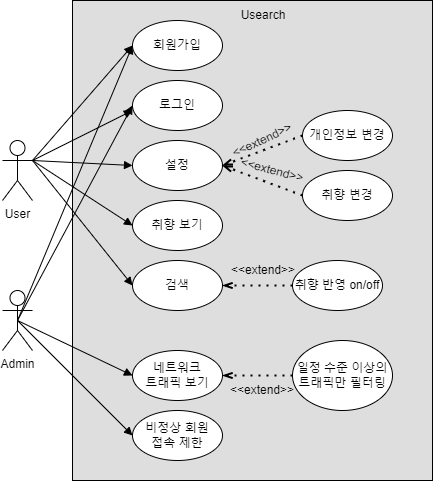
현재 개발하는 프로그램은 웹 프로그램으로 웹페이지를 구현하는 프론트엔드 영역은 html 과 자바스크립트로 제작되었다. 백엔드 부분은 파이썬의 플라스크를 활용해 작성되었으며, 계획서의 있는 모든 기능들을 구현하기에는 다소 무리가 있다 판단되어, 제품 백로그에 작성된 계획대로 따르되 여유가 있다면 추가적인 기능을 구현할 여지는 있다. 다만 현 상태로 파악하기에는 계획에 따르는것이 목표이다.

프로그램이 동작한 후, 유저가 마주할 첫 페이지는 검색화면이다. 해당 페이지는 검색창 및, 로그인 버튼, 회원가입 버튼을 가지고 있다. 우리는 해당 프로그램을 무조건적인 로그인을 필요로하지 않는다. 다만 로그인이 이루어지지 않는다면 해당 프로그램의 대부분인 개인 성향별 검색설정이 이루어지지 않기에, 많은 기능들이 비회워에게는 제한된다. 해당 프로그램의 괄목할 점은, 회원가입이 이루어지면, 유저는 자신의 검색성향을 설정해야한다. 가입 초기에는 회원의 성향을 찾기 힘들뿐더러, 회원이 검색한 성향의 데이터가 충분히 축적되어있지 않는 상황이기에 내린 선택이다. 회원가입이 완료되면, 해당 정보를 유저 DB에 저장하고 유저는 자신이 설정한 아이디와 비밀번호를 활용해 로그인을 할 수 있다. 로그인이 완료 되면 남은 액티비티는 검색이다. 검색 하기 전, 유저는 자신이 선호하는 웹사이트를 미리 선정 할 수 있다(네이버, 다음, 구글).



3. 세부 설계

3.1. Usecase Diagram



3.2. Usecase 명세서

<표 1> Usecase 명세서 – 회원가입

| 유스케이스 이름 | 회원가입 |
| --- | --- |
| 액터 | 사용자 |
| 목표 | 사용자가 자신의 계정으로 회원가입을 성공 |
| 시작조건 | 로그인이 되어있지 않은 상태임 |
| 정상적 사건 흐름 | 1. 사용자가 회원가입을 누른다.  2. 이름, 나이, 선호 카테고리를 입력한다.  3. 회원가입 버튼을 누른다.  4. 성공 여부를 디스플레이한다. |
| 대안 흐름 | A1. 비정상데이터  1. 3단계에서 사용자가 회원 정보를 모두 입력하지 않으면 시작된다.  2. 입력되지 않은 곳을 사용자에게 표시하고 입력하도록 한다.  3. 3단계를 다시 진행한다.  A2. 다른 비밀번호  1. 3단계에서 ‘비밀번호 다시 입력’시 비밀번호가 서로 다르다면 경고 메시지와 함께 비밀번호 수정을 권장한다. |

<표 2> Usecase 명세서 – 로그인

| 유스케이스 이름 | 로그인 |
| --- | --- |
| 액터 | 회원, 관리자 |
| 목표 | 회원은 적절한 아이디와 비밀번호로 로그인해서 검색 페이지로 진입 |
| 시작조건 | 로그인이 되어있지 않은 상태임 |
| 정상적 사건 흐름 | 1. 홈페이지에 접속했을 때, 로그인 되어 있지 않다면 로그인 페이지로 리다이렉션된다.  2. 아이디와 비밀번호를 입력한다.  3. 회원정보가 일치하면 검색페이지로 전환된다. |
| 대안 흐름 | A1. 비정상데이터  1. 3단계에서 회원정보가 일치하지 않는다면 error메시지로 아이디나 패스워드가 일치하지 않음을 알린다. |

<표 3> Usecase 명세서 – 취향 수정

| 유스케이스 이름 | 취향 수정 |
| --- | --- |
| 액터 | 회원 |
| 목표 | 회원의 취향을 본인이 원하는 대로 수정 |
| 시작조건 | 로그인이 되어있는 상태 |
| 정상적 사건 흐름 | 1. 취향 수정 버튼을 누른다.  2. 3가지의 새로운 취향을 선택한다.  3. 수정하기 버튼을 통해 취향을 수정한다. |
| 대안 흐름 | A1. 3개 이상의 취향 선택  1. 3단계에서 취향이 너무 많다는 error를 보내고 3개로 맞출 수 있도록 권유한다.  2. 다시 수정하기 버튼을 통해 취향을 수정한다.  A2. 취소하기  1. 선호사항 수정을 취소하여, 선호사항의 변경 내용을 삭제한다. |

<표 4> Usecase 명세서 – 취향 보기

| 유스케이스 이름 | 취향 보기 |
| --- | --- |
| 액터 | 회원 |
| 목표 | 회원이 자신의 top3취향의 정보를 봄 |
| 시작조건 | 로그인이 되어있고, 취향 정보가 업데이트 된 상태 |
| 정상적 사건 흐름 | 1. 홈페이지에서 취향 보기를 선택한다.  2. 1,2,3등의 취향을 볼 수 있다. |
| 대안 흐름 | A1. 선택한 취향이 없는 상태  1. 1단계에서 선택된 취향이 없다면 취향을 선택하는 페이지로 리다이렉션 해준다. |

<표 5> Usecase 명세서 – 검색

| 유스케이스 이름 | 검색 |
| --- | --- |
| 액터 | 회원 |
| 목표 | 회원이 검색한 키워드로 취향에 맞게 정렬을 하여 보여줌 |
| 시작조건 | 로그인이 되어 있고, 취향이 선택되어 있는 상태 |
| 정상적 사건 흐름 | 1. 검색창에 검색어를 입력한다.  2. 로딩을 기다린다.  3. 추천된 검색 결과를 본다. |
| 대안 흐름 | A1. 취향이 선택되지 않은 상태  1. 취향이 없을 경우 정렬하지 않고 그대로 검색 결과를 제공한다.  2. 취향이 선택되지 않은 상황임을 회원에게 알린다. |

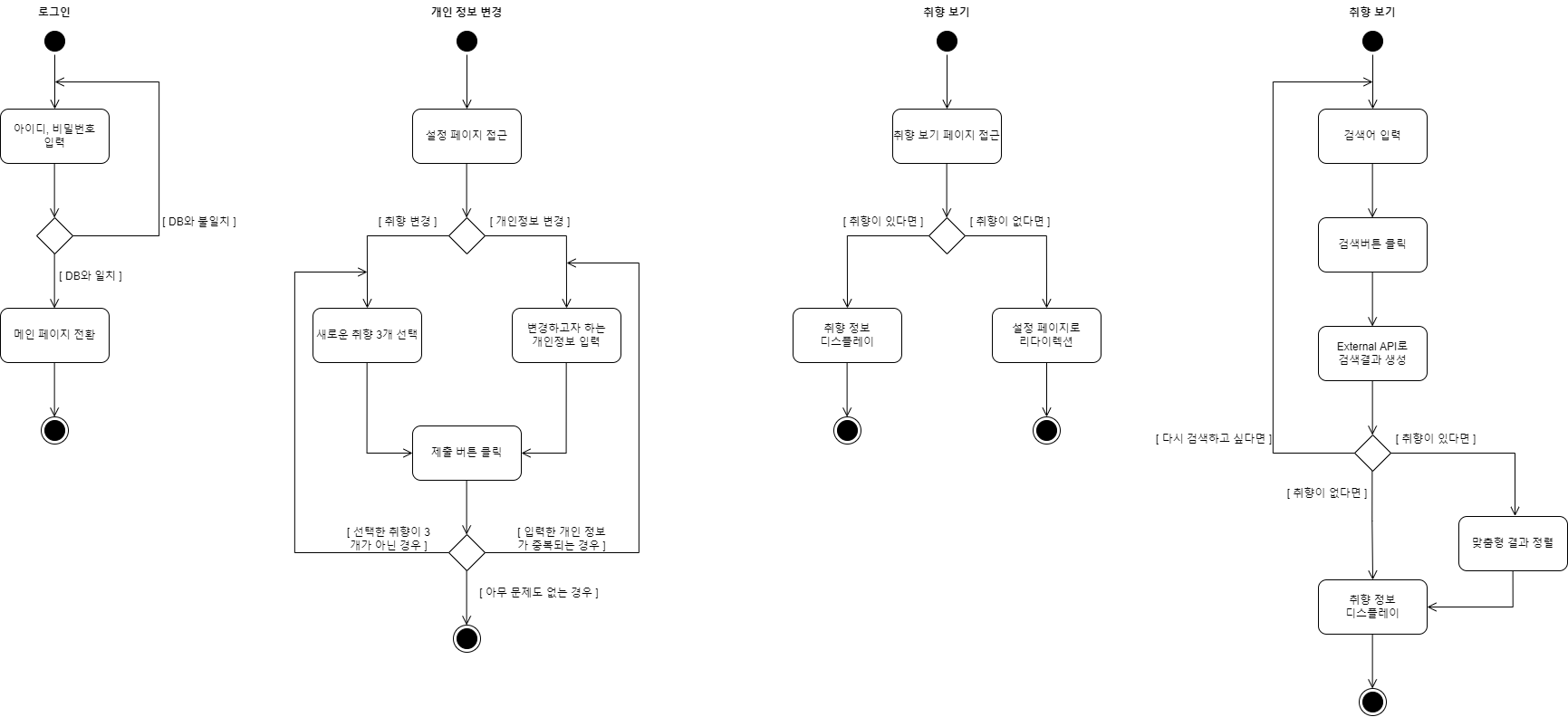
<표 6> Usecase 명세서 – 네트워크 트래픽 보기

| 유스케이스 이름 | 네트워크 트래픽 보기 |
| --- | --- |
| 액터 | 관리자 |
| 목표 | 실시간으로 네트워크 트래픽을 보여줌 |
| 시작조건 | 관리자로 로그인이 되어 있는 상태 |
| 정상적 사건 흐름 | 1. 네트워크 트래픽 메뉴로 들어간다  2. 네트워크 트래픽을 눈으로 확인할 수 있다.  3. 필터를 이용해 일정 수준 이상을 트래픽만 확인할 수도 있다. |
| 대안 흐름 | A1. 네트워크 통신이 좋지 못한 상태  1. 통신이 좋지 않음을 사용자에게 알린다.  2. 페이지를 새로고침할 수 있는 버튼을 제안한다. |

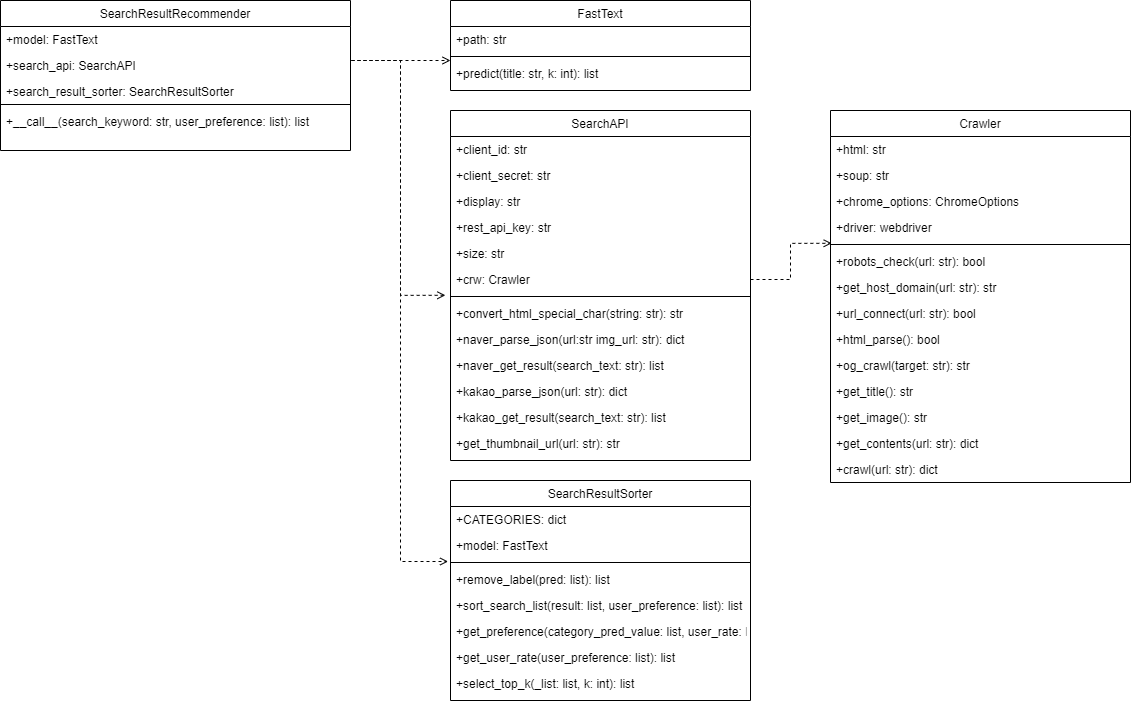
<표 7> Usecase 명세서 – 비정상 회원 접속 제한

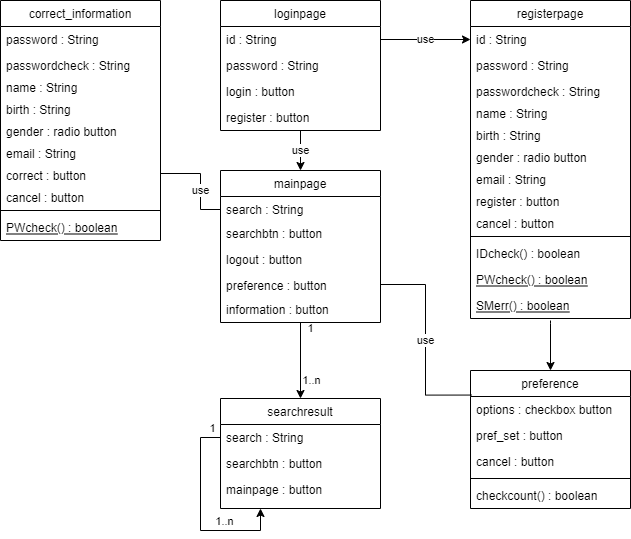
| 유스케이스 이름 | 비정상 회원 접속 제한 |
| --- | --- |
| 액터 | 관리자 |
| 목표 | 비정상으로 이용하고 있는 회원의 이용을 금지할 수 있다. |
| 시작조건 | 관리자로 로그인이 되어 있는 상태 |
| 정상적 사건 흐름 | 1. 비정상 회원 관리 페이지로 들어간다.  2. 비정상적인 이용이라고 판단되는 회원들의 네트워크 트래픽의 정도와 분당-초당 검색 수, 검색어 등을 확인한다.  3. 비정상 회원이라고 여겨지면 관리자는 회원 상태를 금지한다. |
| 대안 흐름 | - |

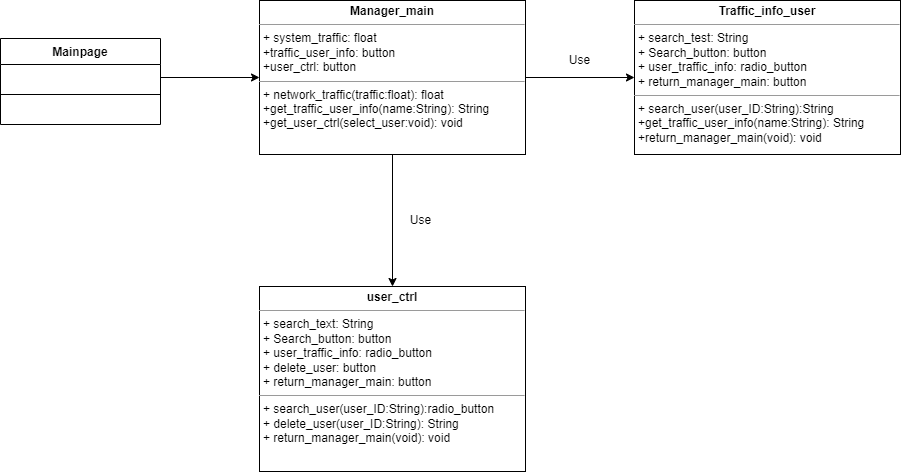
3.3. Activity Diagram



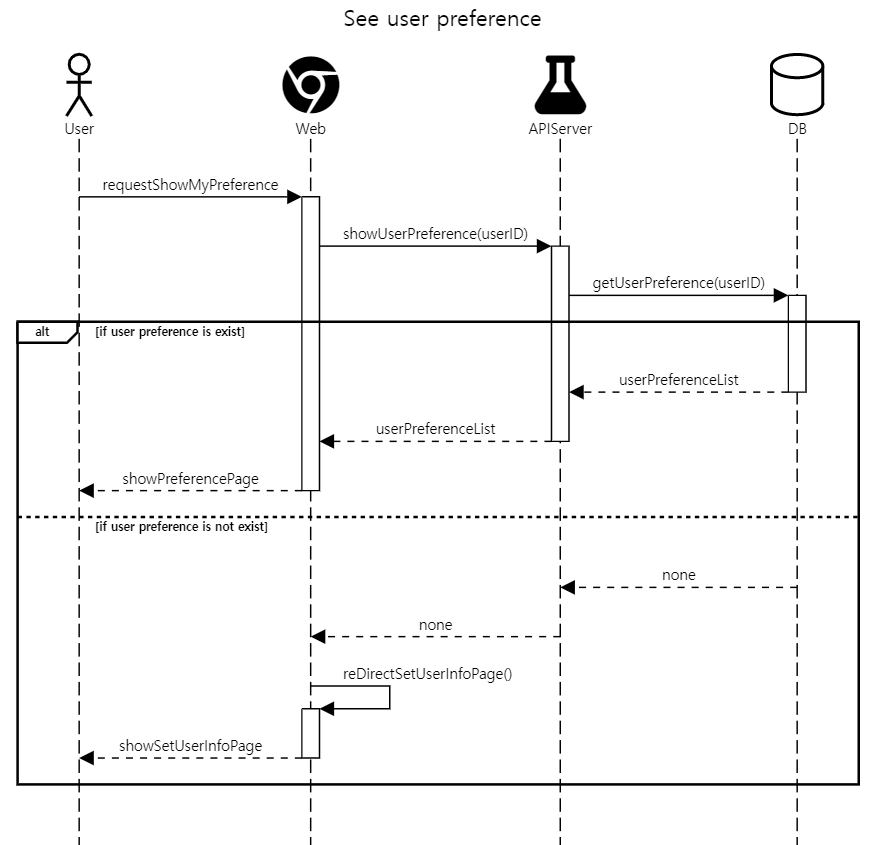
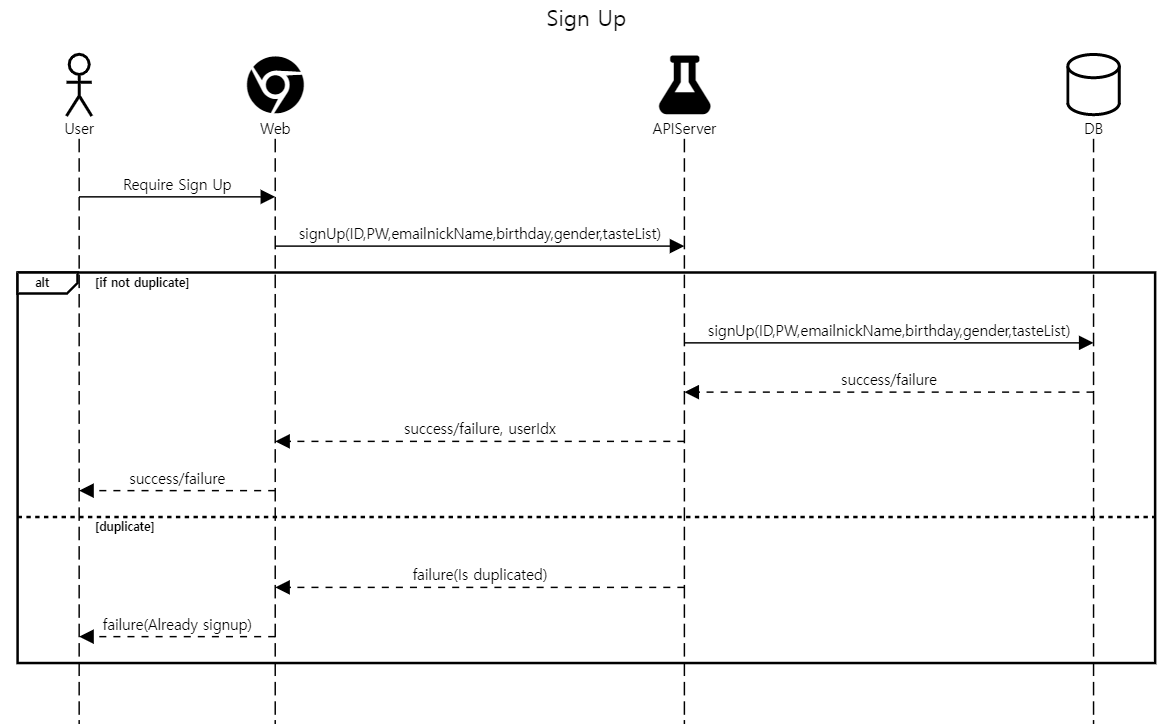
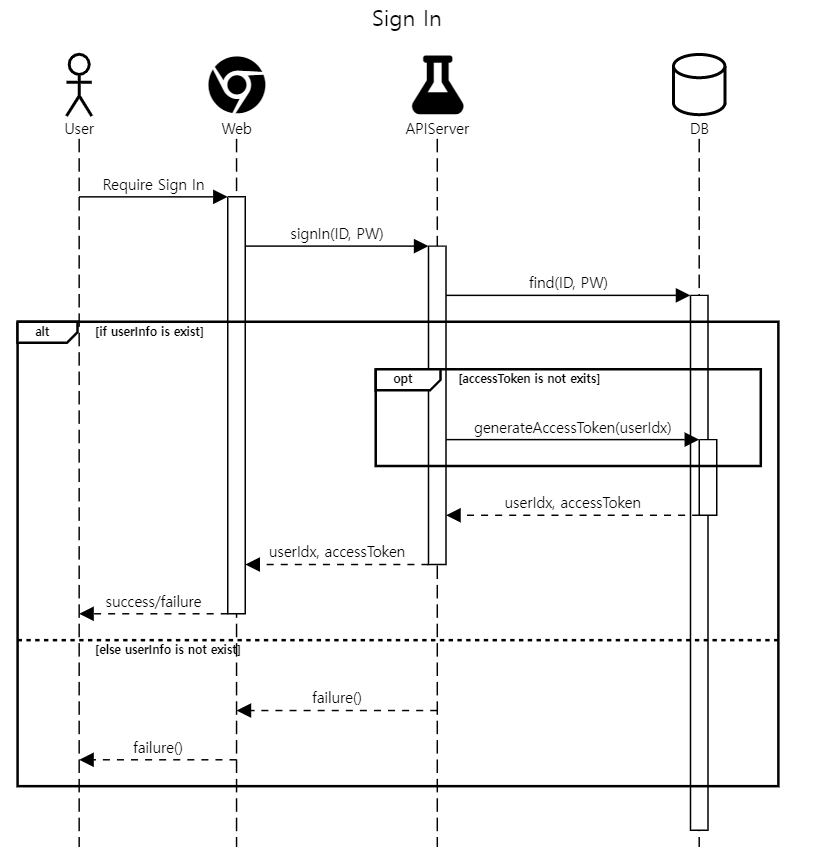
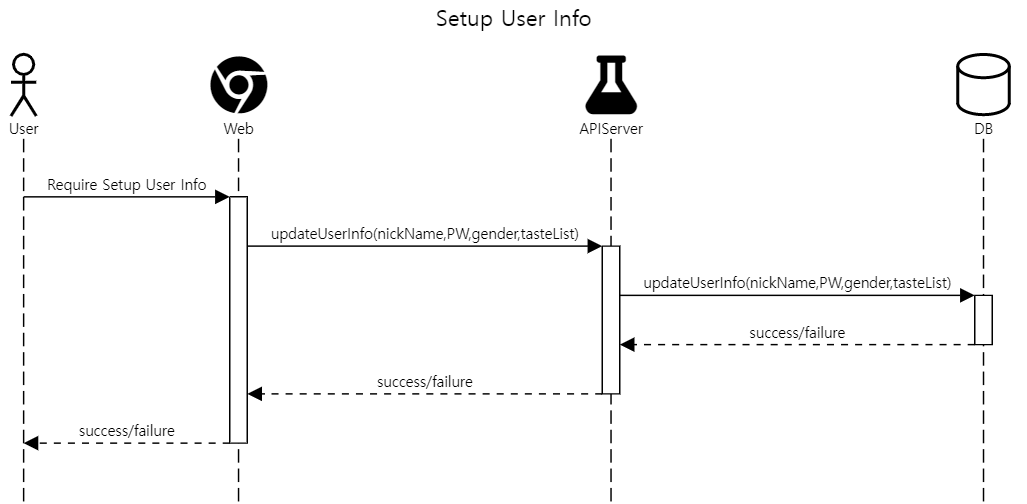
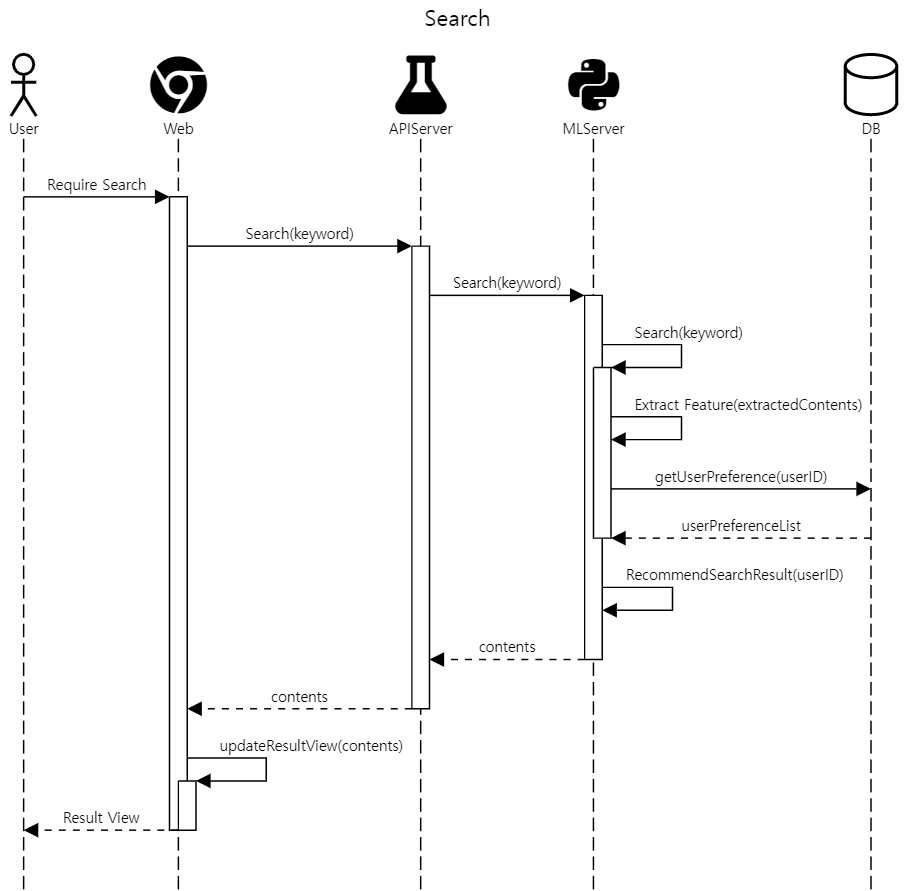
3.4. Class Diagram

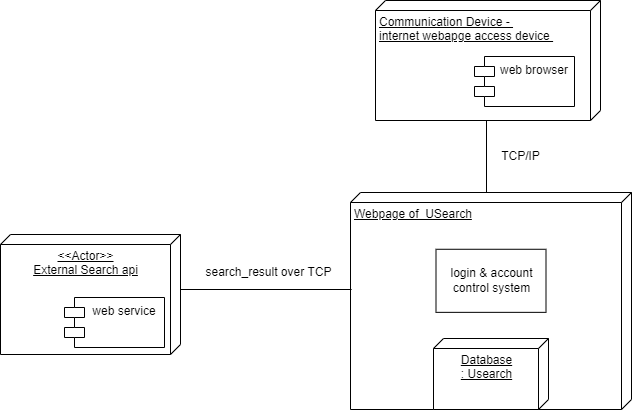






3.5. Sequence Diagram



3.6. Deployment Diagram

4. 개발 진행상황 및 역할분담

4.1. 개발 진행 상황

해당 프로젝트를 진행하면서 그 동안 여러 백엔드 및 프론트 엔드 언어들의 기능들과 해당 언어가 지원하는 여러 가지 라이브러리들을 새로 숙달하는데에 상당한 시간이 들었고, 이는 예상치 못한 변수들을 불러일으켰다. 또한 여러 과목에서 배운 기술들을 하나의 프로젝트에 적용하는 점에서 우리 팀에게는 새로운 시도였다. 현 상황에서는 새로운 기능들과 새로운 기술들을 최대한 융화시켜서 그동안의 뒤쳐진 부분을 최대한 복구하는 것이 목표이다.

4.2. 역할 분담

심효섭: 웹 페이지 제작(정)

최유림: 서버, DB 제작

오승민: 검색 알고리즘 제작(정)

전상우: 프론트, 백-엔드 제작(부)

5. 참고자료