

소프트웨어 요구사항 명세서

(Software Requirement Specification)

유튜브 채널 추천 어플리케이션

(Youtube Channel Recommendation Application)

2021년 3월 31일

인하대학교 컴퓨터공학과

컴퓨터공학 종합설계 - 001분반

Team. Coin-Draw

12161553 김현식

12161586 박철영

12161632 이성환

목 차

1. 서론 (Introduction)	4
1.1 목적 (Purpose)	4
1.2 범위 (Scope)	4
1.3 용어 및 약어 정리 (Definitions, Acronyms, and Abbreviations)	4
1.3.1 정의 (Definitions)	4
1.3.2 약어 (Abbreviations)	5
1.4 참고 문헌 (References)	5
1.5 개 요 (Overview)	5
 2. 종합 기술 (Overall Description)	 6
2.1 프로젝트 결과물 전망 (Product Perspective)	6
2.1.1 시스템 인터페이스(System Interface)	6
2.1.2 사용자 인터페이스(User Interface)	6
2.1.3 소프트웨어 인터페이스 (Software Interfaces)	6
2.1.4 운영 (Operations)	6
2.2 결과물의 기능 (Product Functions)	6
2.3 사용자 특성 (User Characteristics)	8
2.4 제약사항 (Constraints)	8
2.5 가정 및 의존성 (Assumptions and Dependencies)	8

3. 상세 요구사항 (Specific Requirements)	9
3.1 기능 요구사항 (Functional Requirements)	9
3.1.1 데이터베이스 요구사항 (DataBase Requirements)	9
3.1.2 정보 흐름도 (Information flow)	10
3.1.3 알고리즘 요구사항 (Algorithm Requirements)	11
3.1.3.1 성향 분석	11
3.1.3.2 카테고리 세분화	12
3.1.3.3 추천 알고리즘	12
3.2 외부 요구사항 (External Requirements)	13
3.2.1 UI 요구사항 (User Interface)	13
3.2.1.1 메인 페이지	13
3.2.1.2 추천 페이지	14
3.2.1.3 랭킹 페이지	15
3.2.1.4 상세 페이지	16
3.3 성능 요구 사항 (Performance Requirements)	17
3.4 설계 제약 조건 (Design Constraints)	17
3.5 기능별 구현 우선순위 (Functional Development Priority)	17
3.6 소프트웨어 시스템 특성 (Functional Development Priority)	18
4. 추가 이력 (Supporting Information)	18
4.1 소프트웨어 요구사항 명세서	18

1. 소개 (Introduction)

1.1 목적 (Purpose)

- 본 문서는 '컴퓨터공학 종합 설계' 001분반 Coin-Draw팀에서 유튜브 채널 추천 어플리케이션을 설계 및 구현하기 위한 것으로 이를 위한 요구사항을 정리, 분석, 기재한 내용을 바탕으로 시스템을 설계 및 구현한다.
- 본 문서는 전반적인 요구사항을 서술하여, 완성도 높은 어플리케이션을 개발하고, 발전된 기술을 만드는 것을 최우선 목적으로 한다.

1.2 범위 (Scope)

- (기능 1) 유튜브 채널 개인 선호 파라미터 설정 기능
- (기능 2) Collaborative filtering 기반 유튜브 채널 추천 정보 제공
- (기능 3) 전반적인 카테고리 별 유튜브 채널 랭킹 정보 제공
- (기능 4) 마이페이지 관리 기능
- (기능 5) 유튜브 채널 상세페이지 정보 제공
- (기능 6) 유튜브 채널 정보 검색 기능
- (기능 7) 영상 댓글 기반 감정 및 성향 태그 생성

1.3 용어 및 약어 정리 (Definitions, Acronyms, and Abbreviations)

1.3.1 정의 (Definitions)

- Collaborative Filtering⁽¹⁾
 - 특정 아이템에 대해 선호도가 유사한 고객들은 다른 아이템에 대해서도 비슷한 선호도를 보일 것이라는 기본 가정으로 사용자 혹은 아이템간 유사도를 계산하여 선호도를 예측하는 방법이다. 기존의 접근 방식과는 달리, 사용자가 평가한 정보를 사용해서 선호도를 예측하는 것이 가장 큰 차이점이자, 현재까지의 추천 시스템 중에 가장 우수한 성능을 나타내는 비결이다.

- Natural Language Processing(NLP)
 - NLP는 인간이 사용하는 언어를 컴퓨터를 이용해서 묘사할 수 있도록 하는 기술로, 한국어의 분석에는 주로, 어절의 최소 의미 단위인 형태소(단어 그 자체일 수도 있으며, 일반적으로는 단어보다 작은 단위에 해당)로 분석한다.
- OAuth 2.0
 - OAuth는 인증을 위한 오픈 스탠다드 프로토콜로, 사용자가 타 인터넷 서비스의 기능을 다른 어플리케이션에서도 사용할 수 있도록 서비스를 제공하는 것이다.

1.3.2 약어 (Abbreviations)

SRS	Software Requirement Specification, 소프트웨어 요구사항 정의서
AWS	Amazon Web Services
UI	User Interface
APP	Application
NLP	Natural Language Processing
API	Application Programming Interface

1.4 참고 문헌 (References)

- (1) 손지은, 김성범, 김현중, 조성준, 추천 시스템 기법 연구동향 분석, 대한산업공학회, 대한산업공학회지 제41권 제2호, 185-208(24 pages), 2015.04
- (2) <https://developers.google.com/youtube/v3> , Youtube, Youtube data API 공식 웹페이지

1.5 개요 (Overview)

본 문서는 2장에서 종합적인 기술사항을 서술하고, 3장에서는 기능 및 UI에 대해 상세한 요구사항에 대해서 설명한다.

2. 종합 기술사항 (Overall Description)

2.1 프로젝트 결과물 전망 (Project Result Perspective)

2.1.1 시스템 인터페이스 (System Interface)

시스템은 IOS 디바이스의 단일 소프트웨어로 구동되며, 어플리케이션의 입력은 User Interface의 터치로 이루어진다. 어플리케이션의 입력에 맞추어 Application Server에서 해당 입력에 대한 응답을 한다. 또한 추천 시스템은 Collaborative Filtering Server에서 주기적으로 Application Server의 데이터를 업데이트 하여 관리하도록 한다.

2.1.2 사용자 인터페이스 (User Interfaces)

어플리케이션의 인터페이스는 Apple의 Human Interface Guidelines를 준수하며 효율적인 채널 추천과 사용자들의 편리성을 중점으로 UI/UX 구성하도록 한다.

2.1.3 소프트웨어 인터페이스 (Software Interfaces)

- SDK : Over IOS 14.0
- 개발 환경 : XCode12, AWS EC2, AWS S3, Visual Code, Linux 16.04

2.1.4 운영 (Operations)

- 업데이트 및 배포

어플리케이션에 AppStore로의 배포 및 업데이트는 Apple의 Apple Connect를 사용하여 관리하며 각 서버의 데이터는 Amazon S3를 사용하여 버전 관리 및 최신 파일을 관리하도록 한다.

2.2 제품의 기능 (Project Functions)

본 어플리케이션은 다음과 같은 기능을 수행한다.

- (기능 1) 유튜브 채널 개인 선호 파라미터 설정 기능

사용자가 어플리케이션의 첫 사용 시 사용자의 Google 계정으로 API Key를 획득하여 사용자가 구독한 채널의 리스트를 가져와 해당 유저가 해당 채널들을 선호한다고 가정하여 입력한다. 이후 추천 페이지에서 무작위로 추천된 채널들에 대한 피드백을 통해 선호 파라미터 설정이 가능하다. 또한 사용자는 능동적으로 채널 상세페이지에

서 피드백을 주어 Collaborative Filtering에 정확도를 높일 수 있다.

- (기능 2) Collaborative filtering 기반 유튜브 채널 추천 정보 제공

기능 1에서 수집한 사용자들의 각 채널별 선호도를 데이터 셋으로 사용하여 Matrix Factorization Collaborative filtering을 사용하여 사용자에게 알맞는 유튜브 채널 추천 정보를 제공한다.

- (기능 3) 전반적인 카테고리 별 유튜브 채널 랭킹 정보 제공

YouTube Data API V3 를 사용하여 각 카테고리 별 유튜브 채널들의 정보를 가져와 서버에 저장한다. 실시간으로 랭킹을 업데이트 하기에는 제약 조건이 많으므로 주기적으로 랭킹을 업데이트 한다.

- (기능 4) 마이페이지 관리 기능

사용자는 마이페이지에서 라이선스 관련 정보와 자신이 선호하는 채널에 대한 정보들을 수정할 수 있다. 또한 사용자는 마이페이지에서 회원 탈퇴를 통하여 서버에서 자신의 정보를 삭제할 수 있다.

- (기능 5) 유튜브 채널 상세페이지 정보 제공

Youtube Data API V3를 사용하여 각 채널의 대한 정보를 받아와 사용자에게 제공한다. 사용자는 이를 통해 해당 채널의 조회수, 구독자 수, 채널 소개, 최신 영상 등 다양한 정보를 받을 수 있으며 기능 7에서 제공하는 댓글 기반 감정 및 성향 태그를 통하여 해당 채널에 대한 유저들의 반응을 알 수 있다.

- (기능 6) 유튜브 채널 정보 검색 기능

사용자는 검색 기능을 통하여 자신의 관심사 혹은 특정 단어들을 통하여 검색 할 수 있으며 기능 7에서 제공하는 댓글 기반 감정 및 성향 태그를 사용하여 검색 할 수 있다.

- (기능 7) 영상 댓글 기반 감정 및 성향 태그 생성

채널의 유튜브 댓글을 분석하여 태그를 생성하여 각 채널은 유저들의 반응에 따른 태그들을 부여받게 된다. 사용자들은 해당 태그들을 통하여 단순한 채널의 카테고리가 아닌 채널 시청자들의 반응을 알 수 있다.

2.3 사용자 특성 (User Characteristics)

유튜브를 자주 시청하는 10대 20대 대한민국 젊은층을 메인 타겟으로 한다. 일반적으로 12세 이상 대한민국 국민을 사용자로 가정하고 이들이 무리 없이 편리하게 사용할 수 있도록 설계하도록 한다.

2.4 제약사항 (Constraints)

다음의 제약 사항을 준수하여 개발하되, 이외의 사항은 개발자의 재량을 허용한다.

- 별도의 라이선스가 필요하거나 저작권료 사용이 필요한 기술 또는 소프트웨어의 사용을 피한다.
- 협업이 필요한 내용은 문서화 과정을 거치지 않고 코드를 공유하여 신속하게 해결될 수 있도록 함.
- User의 사용을 고려하여 최대한 사용자에게 친화적인 인터페이스를 개발한다.
- 다양한 기기에서 동작할 수 있도록 시스템 장비의 낭비를 줄이는 방향으로 개발한다.

2.5 가정 및 의존성 (Assumptions and Dependencies)

해당 문서의 언급되는 시스템은 ios 및 안드로이드를 가정하며 서술하였으며, 일정 버전 이상의 기기에서 작동되어, 일부 기기에서의 작동이 제한될 수 있다.

3. 상세 요구사항 (Specific Requirements)

3.1 기능 요구사항 (Functional Requirements)

3.1.1 데이터베이스 요구사항 (DataBase Requirements)

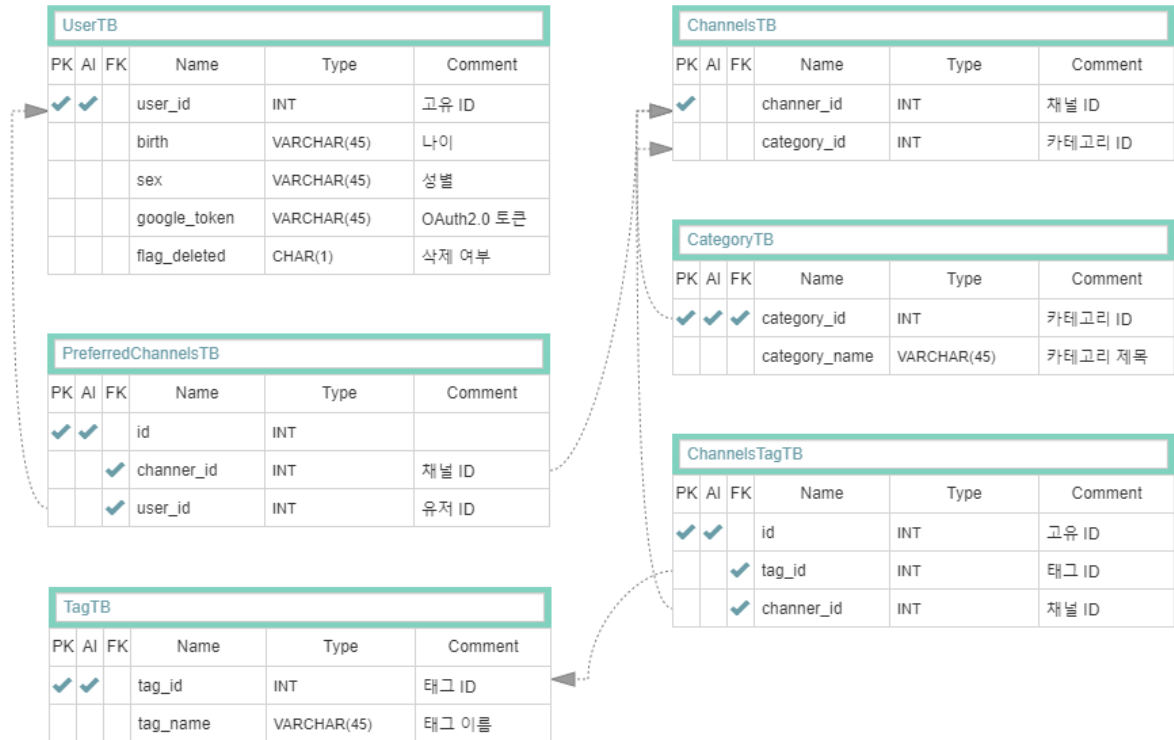


그림 1. Entity Relationship Diagram

● UserTB

- 사용자의 정보(ID, 생년월일, 성별, 구글 인증 토큰, 탈퇴여부)를 저장하여 사용자 효율적으로 관리합니다.

● ChannelsTB

- 채널 ID와 카테고리 ID를 사용하여 채널의 정보를 저장합니다.

● PreferredChannelsTB

- 사용자가 지정한 선호 채널 정보(채널 ID, 유저 ID)를 저장하는 테이블입니다.

● TagTB

- 영상이 분류될 수 있는 범위의 태그 정보를 저장하는 테이블입니다.

● CategoryTB

- 기존 유튜브가 다루는 29개 보다 많은 카테고리를 분류하기 위해서 카테고리 정보를

저장하는 테이블입니다.

- ChannelsTagTB

- 특정 채널에 해당하는 태그 정보를 저장하기 위한 테이블입니다.

3.1.2 정보 흐름(Information flow)

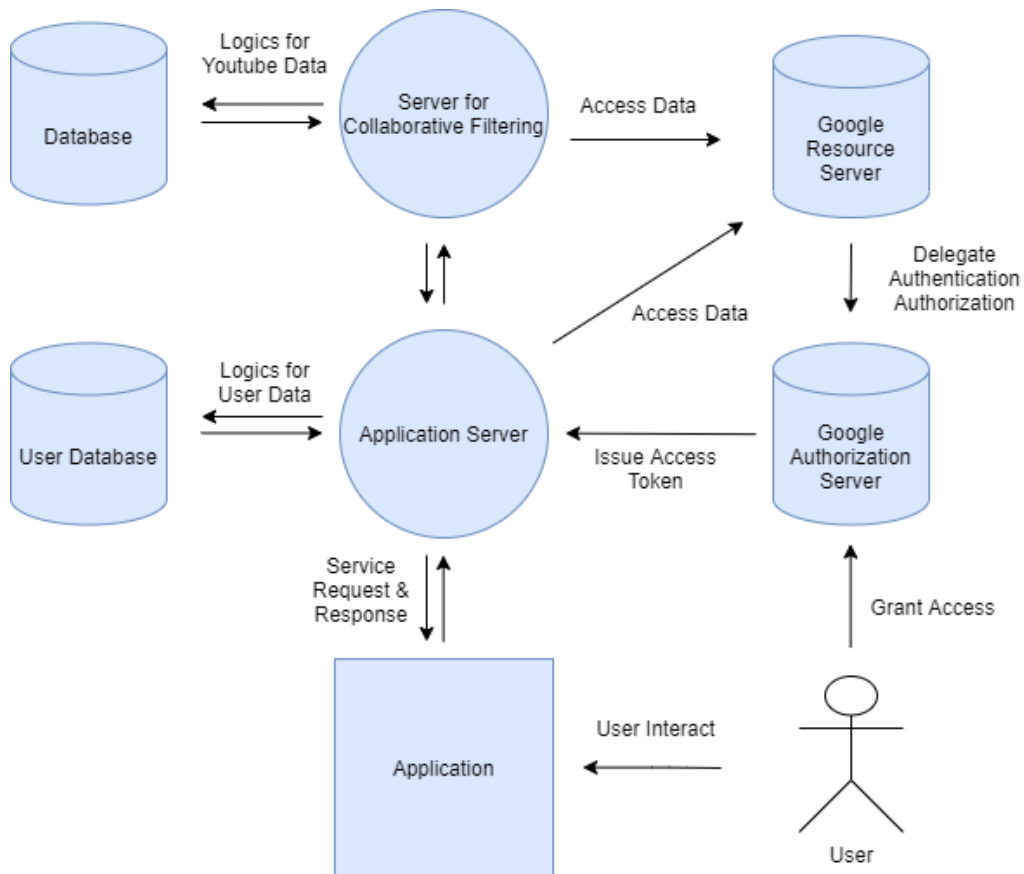


그림 2. 유튜브 채널 추천 어플리케이션의 정보 흐름도

사용자가 어플리케이션 첫 사용 시 OAuth 2.0 인증을 통하여 Google Authorization Server로 부터 Access Token을 발급받아 해당 토큰은 데이터 베이스에 저장한다.

사용자가 어플리케이션에 서비스를 요청하면 어플리케이션은 Application Server로 REST API를 사용하여 통신을 한다. Application Server에서는 데이터베이스에서 필요한 정보를 가져와 해당 요청에 대한 처리를 진행한다. 이때 Youtube Data가 필요하다면 위에서 발급 받은 해당 사용자의 Access Token을 사용하여 Google Resource Server에 데이터를 요청한다.

Collaborative Filtering을 위한 서버에서는 성향 분석과 세부 카테고리 학습에 필요한 모델을 저장하고 주기적으로 Google Resource Server에 데이터를 받아와 새로운 콘텐츠에 대한 업데이트를 진행한다. Application Server에서는 추천 서비스가 필요한 경우 해당 서버에 사용자의 정보를 보내어 Collaborative Filtering Server에서 이에 알맞는 추천 정보를 제공한다.

3.1.3 알고리즘 요구사항 (Algorithm Requirements)

3.1.3.1 성향 분석

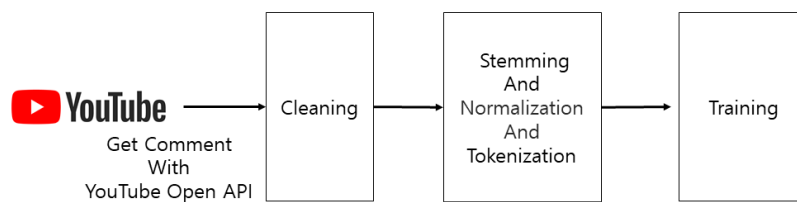


그림 3. Youtube Open API를 이용한 지도 학습도

유튜브 채널이 가질 수 있는 성향 태그를 일정 개수로 생성하여 [ex) 재미있는, 정치적인] 100명의 유튜브 채널에 생성된 태그를 매칭한다.

유튜브 채널의 댓글을 Youtube Open API를 사용하여 가져오고 정제, 분해하여 정규화 과정을 진행한다.

정규화된 단어들은 토큰화되어 기존의 매칭된 성향 태그를 활용하여 지도 학습을 진행한다.

3.1.3.2 세부 카테고리 분류 알고리즘

현재 분류된 유튜브의 29가지 카테고리는 방대한 양의 영상들을 효과적으로 분류할 수 없는 상황이다. 따라서, Youtube Open API를 통해, 유튜버들이 업로드한 동영상들의 정보를 가져오고 실제 영상의 카테고리화 연관되는 단어를 찾아 알맞은 세부 카테고리에 매칭한다. [ex) 게임 카테고리의 영상은 게임 이름이 세부 카테고리]

이런 방식으로 카테고리를 세분화하여 사용자가 원하는 정보를 쉽게 찾을 수 있도록 서비스를 제공한다.

3.1.3.3 추천 알고리즘

초기 사용자의 선호도 데이터가 없을 때, 사용자의 구독자 정보와 초기 로그인에 선택한 유튜브 채널들을 기반으로 matrix factorization collaborative filtering을 통해, 유튜브 채널과 유저간 선호도에 대한 행렬을 생성한다.

이후에는 랜덤 채널을 추천해 주는 추천 페이지에서 받아온 positive or negative feedback을 데이터에 추가하여 지속적인 업데이트를 통해 추천 정확도를 높일 수 있게 설계한다.

3.2 외부 요구사항(External interface requirements)

3.2.1 UI 요구사항 (User Interface)

3.2.1.1 메인페이지



그림 4. 메인 페이지 UI 구성 예시

- Application Server의 유저 데이터 기반 Collaborative Filtering을 사용하여 주제 별 추천 채널 리스트화.
 - ex) 요즘 뜨는 채널, 00과 비슷한 채널
- 유저가 채널 뷰를 클릭 시 해당 채널의 정보를 담은 상세페이지로 이동.
- 오른쪽 상단 검색 버튼을 통하여 검색어, 카테고리, 태그를 통하여 원하는 채널 검색 기능.
- 하단의 네비게이션 바를 사용하여 랭킹 페이지, 메인 페이지, 마이 페이지로 이동 가능.
- 상단의 추천 TextView 클릭 시 추천 페이지로 이동.

3.2.1.2 추천 페이지



그림 5. 유튜브 채널 추천 페이지의 UI 구성도

- 랜덤하게 선정된 채널들을 유저에게 추천하는 페이지. 랜덤하게 선정된 채널에 대한 정보들과 태그를 제공하여 유저는 Swipe gesture를 통해 negative or positive feedback을 줄 수 있음. 해당 정보를 사용하여 Collaborative filtering의 정확도가 향상됨.
- 상단의 메인 TextView 클릭 시 메인 페이지로 이동.
- 하단의 네비게이션 바를 사용하여 메인 페이지, 랭킹 페이지, 마이 페이지로 이동 가능.

3.2.1.3 랭킹 페이지

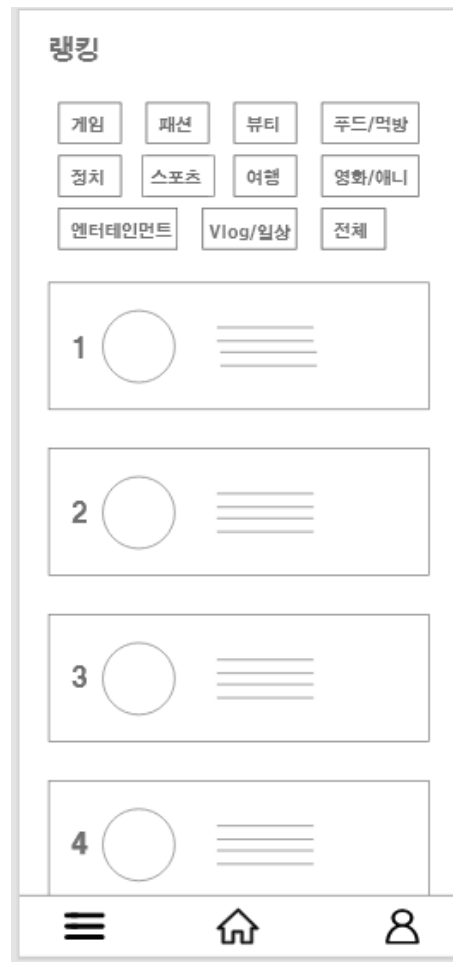


그림 6. 유튜브 채널 랭킹 페이지의 UI 구성도

- Youtube Data V3 API를 통해 하루에 한번 데이터를 분석하여 각 카테고리별 순위표를 유저에게 제공. 랭킹 카드뷰에는 유튜브 채널 이미지와 구독자 수, 채널 소개 등의 정보가 들어감.
- 랭킹 카드 뷰 클릭시 채널 상세 페이지로 이동.
- 하단의 네비게이션 바를 사용하여 메인 페이지, 랭킹 페이지, 마이 페이지로 이동 가능.

3.2.1.4 유튜브 채널 상세페이지



그림 7 . 유튜브 채널 상세 페이지 UI 구성도

- Youtube Data V3 API를 통해 구독 수, 채널 설명, 채널 배너 이미지, 카테고리 등 정보를 사용자에게 제공.
- 최신 영상과 조회수가 높았던 영상을 샘플로 제공.
- 사용자는 해당 채널을 관심있는 채널로 등록하여 Collaborative Filtering 기반 추천의 정확도를 높일 수 있음.
- 위 서술된 성향 분석 알고리즘을 통해 태그로 분류, 사용자에게 채널의 유저들의 반응을 알려줌

3.3 성능 요구 사항 (Performance Requirements)

본 어플리케이션은 IOS 14를 지원하는 iPhone 6S 부터 iPhone 12 series 디바이스들에서 안정적인 퍼포먼스를 유지해야하며 사용자의 요청으로 부터 안정적인 응답속도를 유지할 수 있도록 한다. 또한 사용자의 정보를 데이터베이스에 저장하므로 사용자의 정보를 안전하게 보관할 수 있도록 암호화 하도록 한다.

3.4 설계 제약조건 (Design Constraints)

본 어플리케이션 설계 시 짧은 개발 기간을 고려해야 한다. 따라서, 고전적인 개발방식인 폭포수 개발 방법론보다 프로토타입 개발 방법론을 선택하여 기본적인 기능의 빠른 구현을 추구한다. 또한, 팀원과 함께 협업을 진행할 수 있고, 어플리케이션에 대한 평가를 용이하게 하기 위해, 프로그래밍 소스에 주석을 포함해야하고, 변수명은 쉽게 이해할 수 있는 이름을 사용해야 한다. 기능별로 모듈화를 진행하여 객체기반 프로그래밍을 최대한 지향한다,

3.5 기능별 구현 우선순위 (Functional Development Priority)

3.5.1 최우선 구현 사항

- (기능 1) 유튜브 채널 개인 선호 파라미터 설정 기능
- (기능 2) Collaborative filtering 기반 유튜브 채널 추천 정보 제공

3.5.2 우선 구현 사항

- (기능 5) 유튜브 채널 상세페이지 정보 제공
- (기능 7) 영상 댓글 기반 감정 및 성향 태그 생성

3.5.3 선택 구현 사항

- (기능 3) 전반적인 카테고리 별 유튜브 채널 랭킹 정보 제공
- (기능 4) 마이페이지 관리 기능
- (기능 6) 유튜브 채널 정보 검색 기능

3.6 소프트웨어 시스템 특성 (Software System Property)

3.6.1 신뢰성 (Reliability)

사용자의 선호도 입력에 따라 정확한 유튜브 채널 추천 결과를 제공하여야 한다.

3.6.2 가용성 (Availability)

시간적, 공간적 제약 없이, 클라이언트, 서버, 데이터베이스 등의 구성요소들이 오류 없이 동작해야 하며, 사용가능한 수준의 속도로 처리되어야 한다.

3.6.3 유지가능성 (Maintainability)

어플리케이션의 버그 보수 및 업데이트 용이성 극대화.

3.6.4 보안성 (Security)

사용자의 정보를 안전하게 보관.

4. 추가 이력 (Supporting Information)

4.1 소프트웨어 요구사항 명세서

본 소프트웨어 요구사항 명세서 (Software Requirements Specification)는 IEEE STD 830-1998 을 준수하여 작성하였음.