

Final Report



캡스톤 디자인

이유철 교수님

정보통신공학과 3팀

2017112211 김수호

2018112134 송창현

2018112167 안준수

2018112158 이승규

1. 서론	4
1.1. 프로젝트 목적 및 Key Value.....	4
1.2. 기존 설계와 달라진 점.....	6
2. 구현 방법	7
2.1. 플러터.....	7
2.1.1 크로스 플랫폼 개발	7
2.1.2 디자인 패턴 : BLoC 및 프로바이더.....	8
2.1.3 디자인 패턴 : 싱글톤.....	9
2.2. Firebase, Firestore	9
2.3 NAS, 백엔드 도커 배포.....	10
3. 구현된 기능.....	11
3.1 나의 옷장	11
3.2 옷 추천.....	13
3.3 나의 코디	17
3.4 옷 업로드	18
3.5 코디 업로드.....	20
3.6 코디 평가.....	21
3.7 유저정보 설정.....	24
4. UI/UX 완성도	25
4.1. 기존 설계와 달라진 UI/UX.....	25
4.2. 사용자 편의성 평가.....	26
4.3. UI/UX 반영 결과.....	27
5. 기능완성도 및 성능 평가.....	29
5.1. 옷추천 시스템 정교화.....	29

5.1.1 GPT-3.5 vs GPT-4	29
5.1.2 파라미터 조절	33
5.1.3 프롬프트 엔지니어링	34
5.1.4. 파인튜닝	36
5.2. API 안정성 테스트	40
5.3. DB 안정성 테스트	41
6. 기술 내재화 및 독창성.....	44
6.1. 독창적 아이디어	44
6.2. 현실세계를 잇는 차세대 인터페이스 설계	45
7. 팀워크 및 역할 분배	46
7.1. 팀원별 역할 분배	46
7.2. 협력 및 의사소통 과정	46
8. 결론 및 향후계획.....	51
8.1. 프로젝트 총평	51
8.2. 향후 개선사항 및 앞으로의 계획	52
부록.....	53
9.1. 참고문헌	53
9.2. 소스코드	53

1. 서론

1.1. 프로젝트 목적 및 Key Value

현대 사회에서 패션은 개인의 스타일과 취향을 표현하는 수단이고 중요한 요소로 여겨진다. 하지만 여전히 많은 사람들이 일상생활에서 자신만의 독특한 스타일을 찾는데 어려움을 느끼고, 옷장에 수많은 옷이 있음에도 불구하고 옷을 조합하는 데에 많은 시간과 고민을 들인다.

그래서 이를 해결하는 서비스를 제공하고자 한다. Crush는 사용자 중심의 서비스를 제공하기 위해 AI 코디 추천 기능, 나의 옷장 기능, 코디 평가 기능 제공한다. 이를 통해 사용자들은 자신만의 스타일을 찾고, 서로의 감각을 교류하며, 보유 옷 기반의 추천정보를 받을 수 있다.



프로젝트의 주요 목적, Key Value는 다음과 같다:

- 효율적인 옷장 관리 및 코디 도움

사용자들이 옷을 저장하고 관리할 수 있는 플랫폼을 이용할 수 있다. 이를 통해

일상 생활에서 코드를 선택하고 조합하는 데 도움이 되도록 한다.
이를 위해 옷 추천 알고리즘을 정교화 할 필요가 있다.

- **유저들의 코디 센스 향상**

서로의 코드를 공유하고 평가할 수 있는 커뮤니티 기능을 통해, 사용자들이 자신의 코디센스를 평가 받고 기를 수 있다.

1.2. 기존 설계와 달라진 점

실제 구현을 진행하며 기존 설계에서 누락되거나 미흡하다고 판단된 부분을 보완하였다. 이를 통해 완성도를 높게 만들었다.

- **옷 추천 알고리즘 강화**

기존 설계에는 단순히 gpt-3.5만을 활용해 서비스를 구축하고자 했다. 하지만, 이는 다른 서비스와의 차별점을 강화할 수 없고 옷추천 정확도가 높지 않다는 문제점이 있었다. 그래서 모델을 gpt-4로 업그레이드 하고, 프롬프트 엔지니어링 및 파인튜닝으로 옷추천 정확도를 높이는 작업을 진행했다.

- **UI/UX 개선**

기존 설계보다 더 깔끔하고 직관적인 UI로 변경하였고, 사용감과 완성도를 좋게 하기 위해 앱 내에 역동적인 애니메이션을 추가했다. 또한 유저가 옷 추천, 코디 평가 등을 진행하면서 서버와 통신하는 동안 지루하지 않도록 대기 페이지들을 추가했다. 그리고 앱 내의 핵심 기능과 숨겨진 제스처 동작을 알려주기 위해 도움말 기능을 추가해서 유저 편의성을 증가시켰다.

- **옷 추천에 다양한 파라미터 요소 활용**

기존의 설계에서 나이, 성별 등 더 다양한 파라미터를 고려하게 하였다.

- **프롬프트 엔지니어링 강화**

기존 설계때보다 더 정확한 결과가 나오도록 프롬프트 엔지니어링 프롬프트를 더 강화했다.

- **DB 문서 객체에 일부 필드 추가**

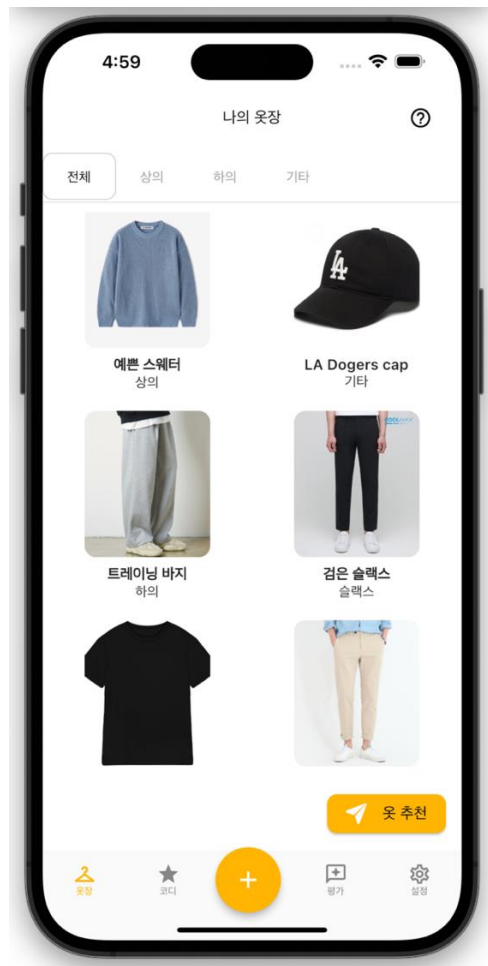
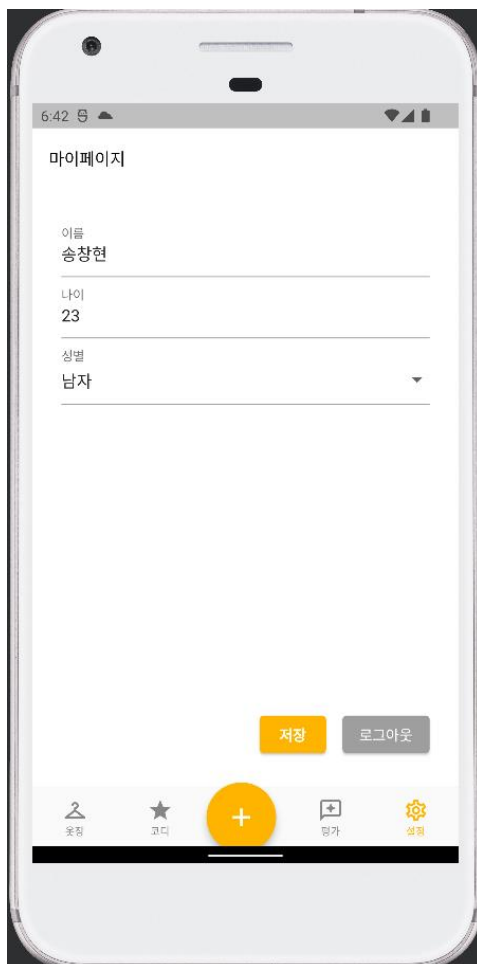
코디 객체에서 description 필드 등 일부 앱 구현에 있어서 필수적인 필드가 누락되어 있어서 추가하였다.

2. 구현 방법

Crush는 플러터로 개발되었고, 개발 과정에서 디자인 패턴을 적용하여 성능상의 이점과 클린코딩을 유지했다.

2.1. 플러터

2.1.1 크로스 플랫폼 개발



왼쪽: 안드로이드 오른쪽 IOS



플러터의 가장 큰 장점은 멀티플랫폼을 지원한다는 점이다.

즉, Crush는 iOS와 Android를 모두 지원한다.

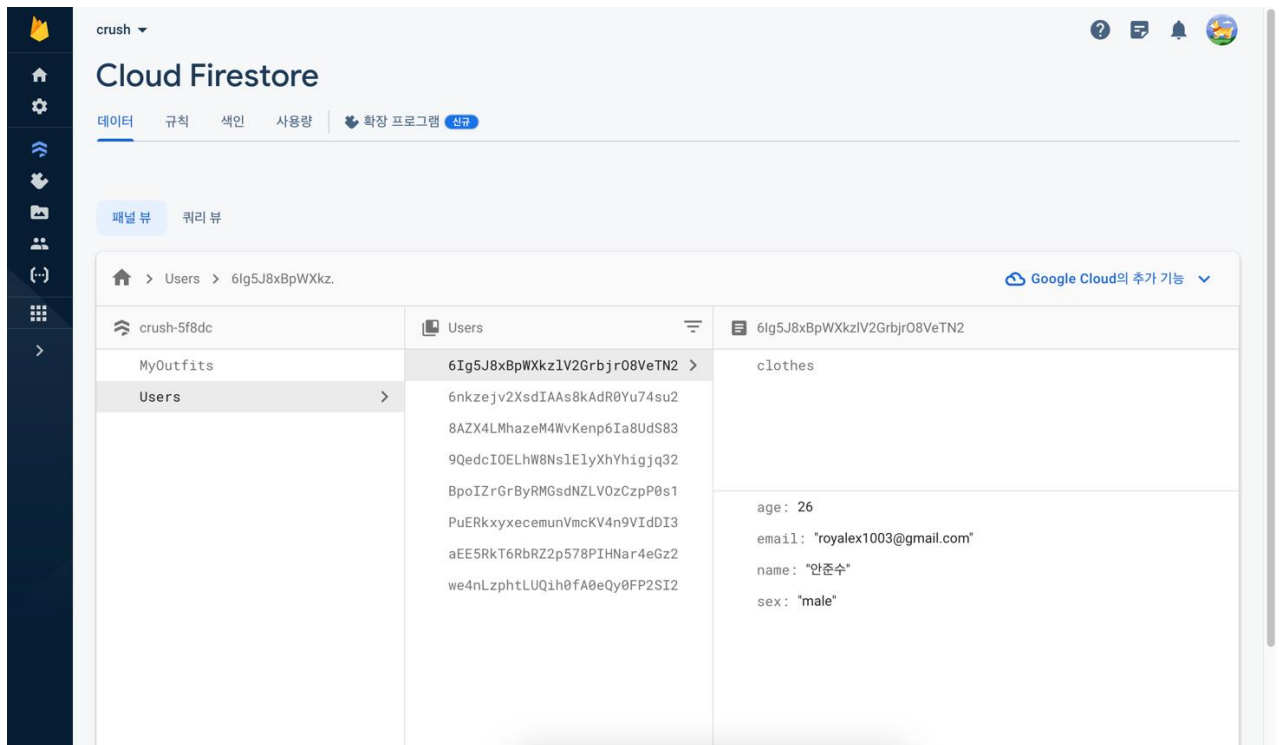
2.1.2 디자인 패턴 : BLoC 및 프로바이더

Crush 구현에 있어서 BLoC 및 프로바이더 패턴을 적용했다. 이를 통해 객체 간의 결합도를 낮추고, 클린코드를 유지할 수 있었다.

2.1.3 디자인 패턴 : 싱글톤

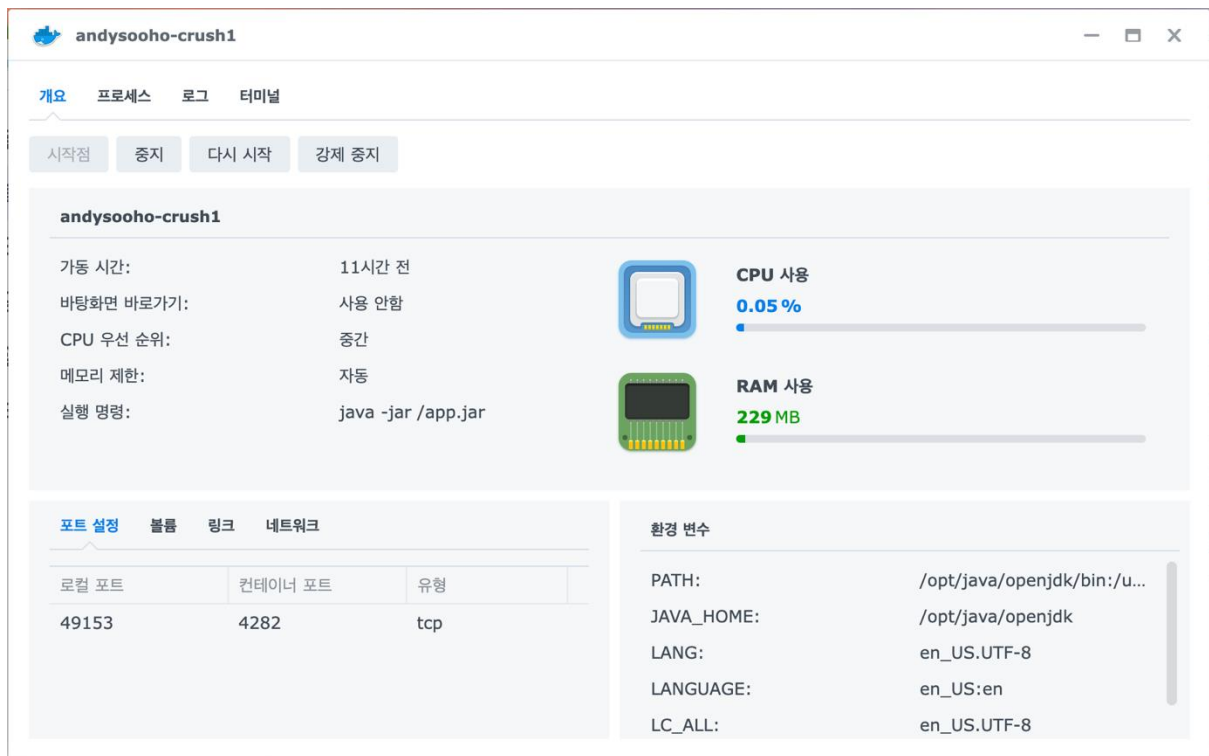
Crush 구현에 있어서 싱글톤 패턴도 적용했다. 인스턴스를 전역적으로 공유하여 객체 생성을 제한하고, 어디서든 동일한 인스턴스에 접근할 수 있다. 이를 통해, 성능상의 이점을 챙겼다.

2.2. Firebase, Firestore



구글의 서버리스인 Firebase를 이용해 유저와 옷, 코디 정보를 저장한다.
사용자가 업로드 하는 사진 파일은 Firestore로 관리된다.

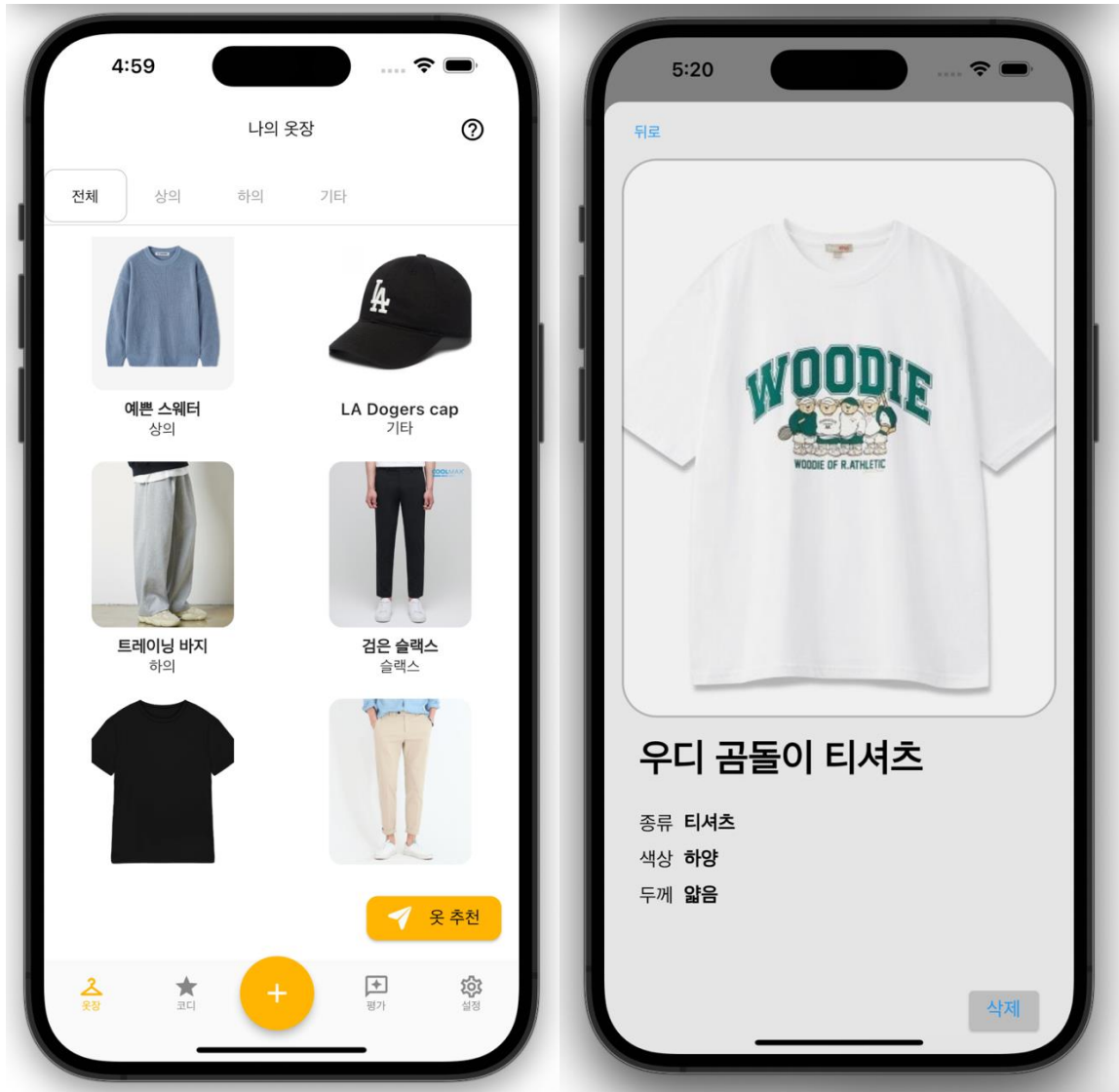
2.3 NAS, 백엔드 도커 배포



옷 추천 API 를 제공하는 스프링 서버를 직접 운영하는 클라우드 서버에 배포하였다. 서버에는 DDNS주소가 있고 네트워크 라우터에서 포트포워딩 및 로컬포트 매핑 설정을 통해 실제로 사용할 수 있게 배포하였다.

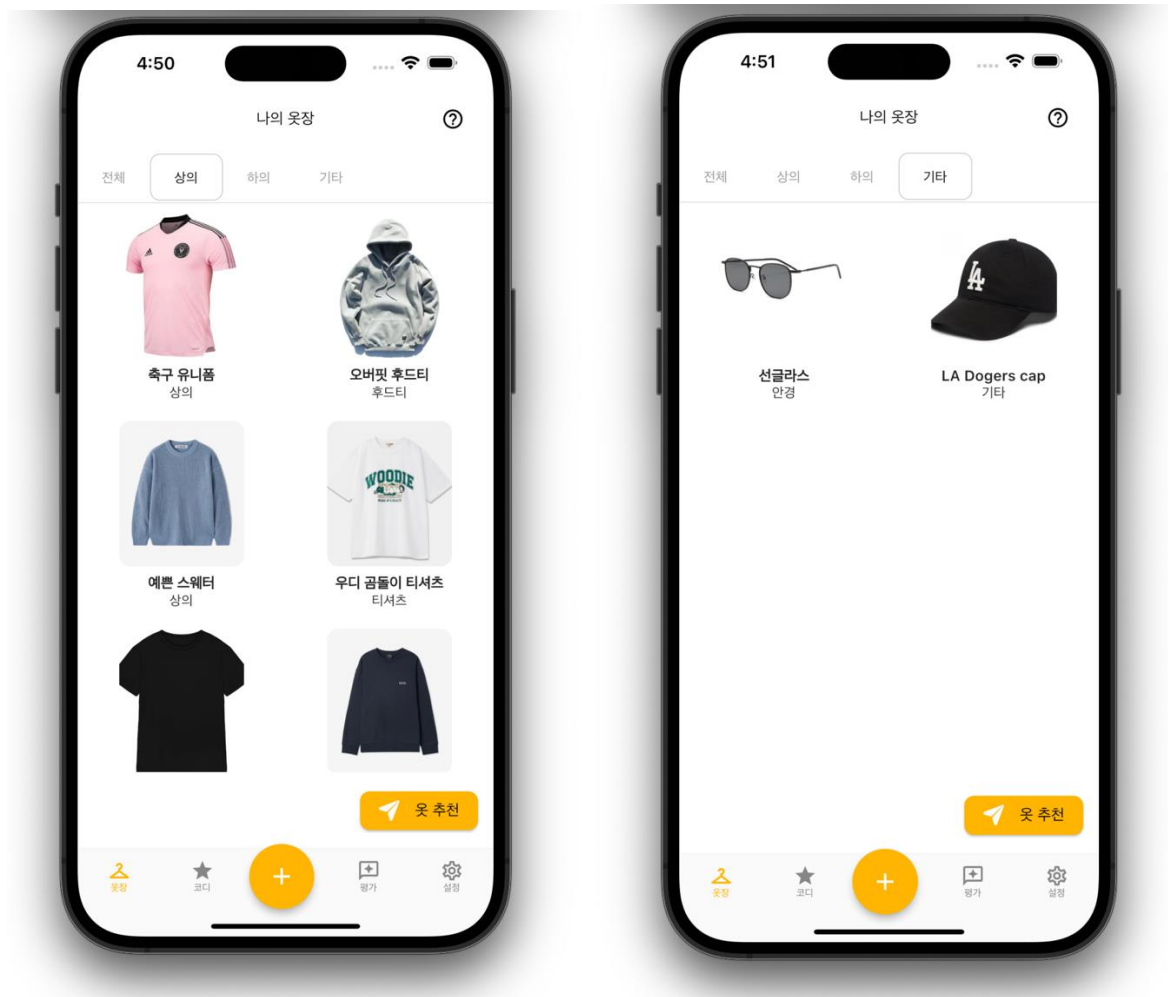
3. 구현된 기능

3.1 나의 옷장



- 나의 옷 관리

보유한 옷을 관리 할 수 있다. Firestore에서 저장된 옷 정보를 받아오고, 리스트로 항목을 출력한다. 항목을 누르면 세부 정보를 볼 수 있다.

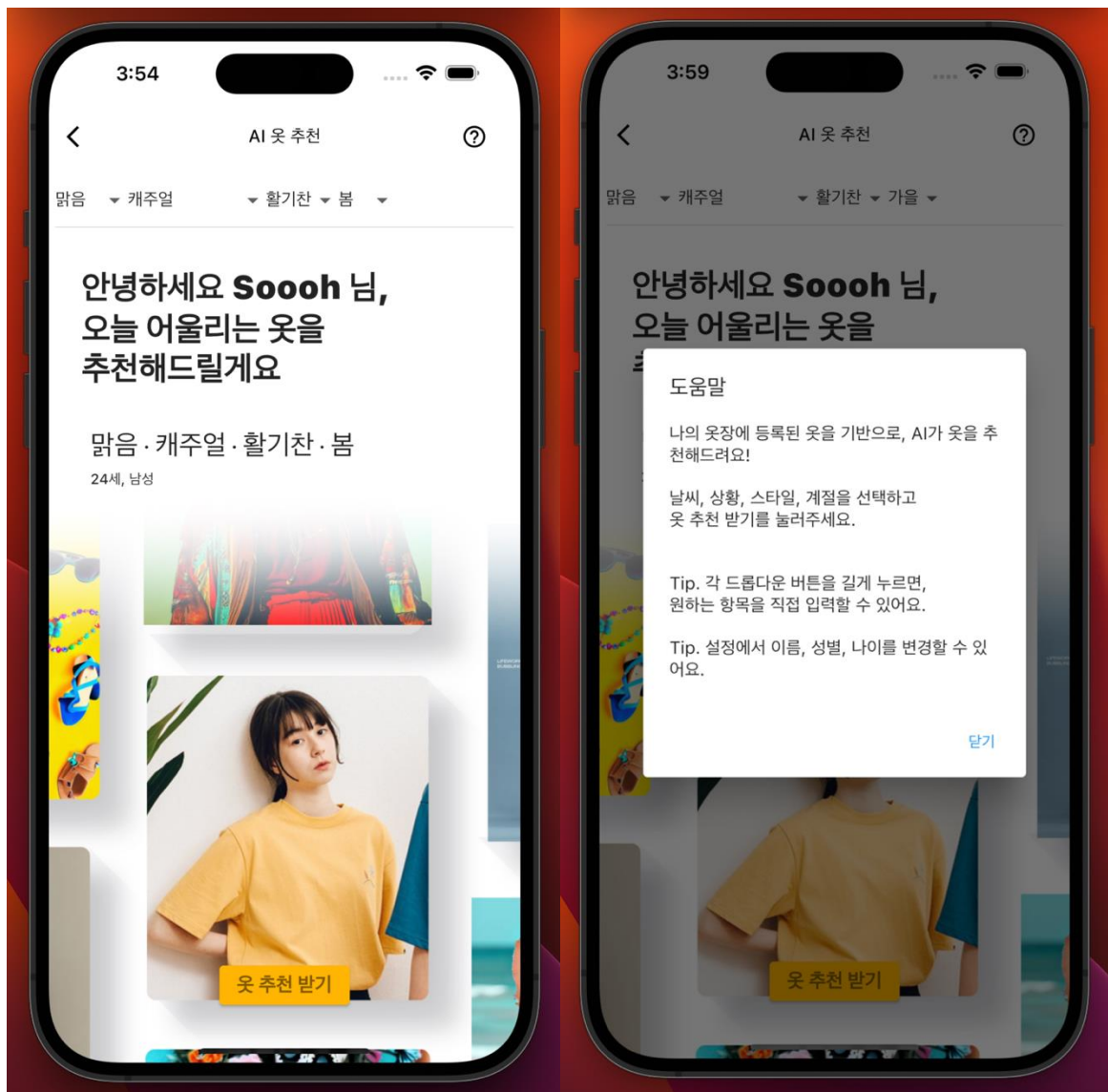


- **탭으로 분류**

옷장은 전체, 상의, 하의, 기타의 4개의 탭으로 나누어져 있다.

각 탭에서 리스트를 불러올 때 각 항목에 해당하는 옷만 필터링해서 출력하도록 구현되었다.

3.2 옷 추천



- 보유 옷 기반 옷 조합 추천

메인 화면에서 옷 추천 버튼을 클릭하면 AI 옷 추천 페이지로 이동한다.

불러온 보유한 옷 기반으로 옷 추천 조언을 받을 수 있다.

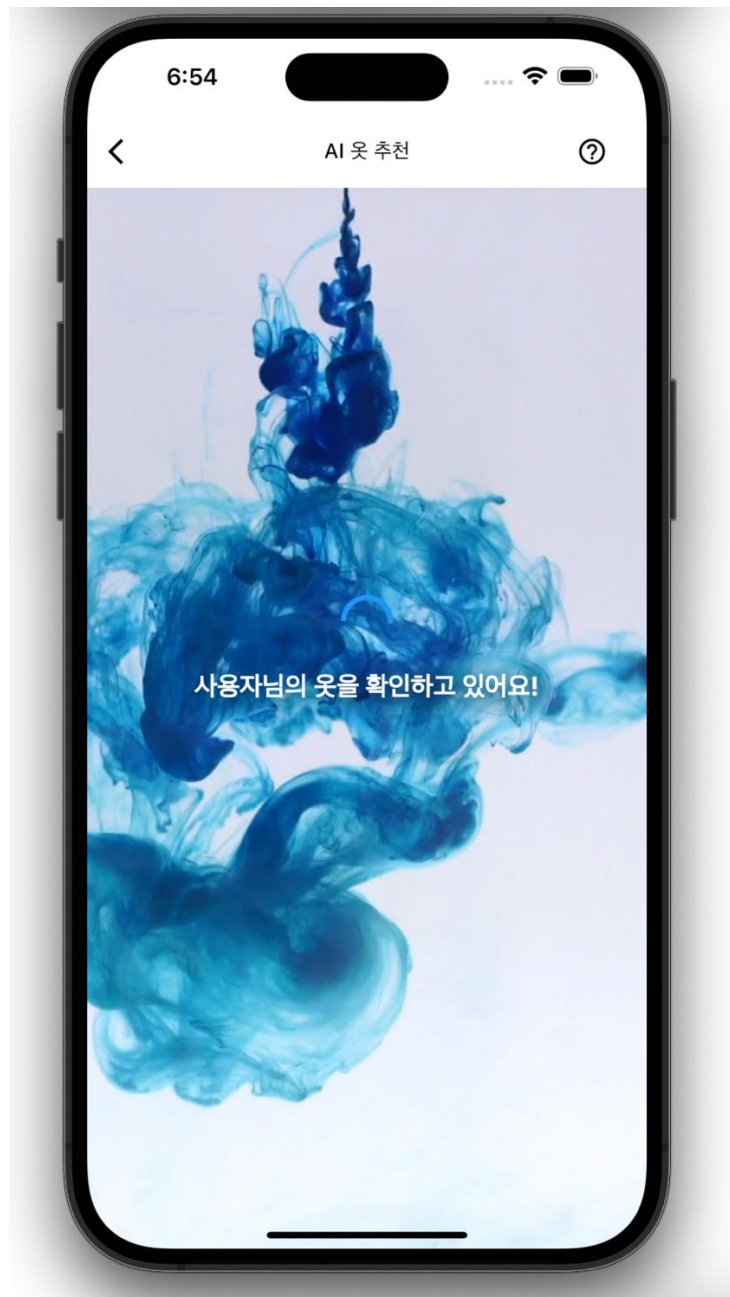
상단의 물음표 버튼을 통해 AI 옷 추천 기능의 이용 방법에 대한 도움말을 볼 수 있다.

- **상황 선택**

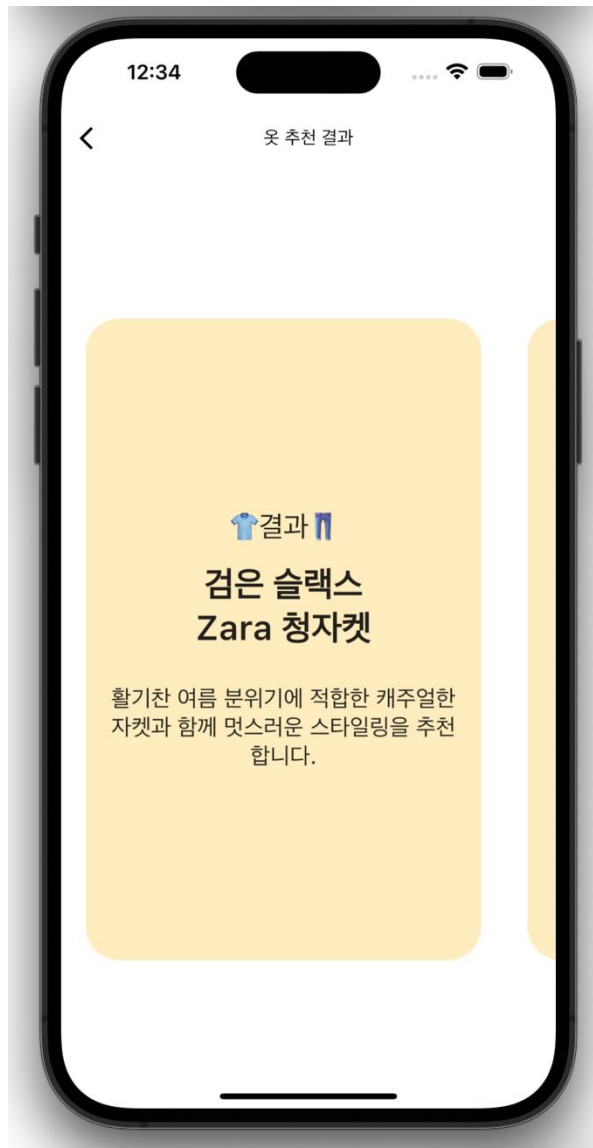
상단의 드롭다운 바를 통해 날씨, 상황, 스타일, 계절을 선택하고 조언을 받을 수 있다. 나이, 성별 또한 추천 요소에 포함이 되는데 이는 설정에서 수정할 수 있다. 이 내용은 도움말에서 볼 수 있다.

- **로딩화면**

옷 추천에 5~10초정도가 소요되는데 이를 사용자가 기다려야 한다. 그래서 사용자가 이를 기다릴 수 있도록 유도해야 한다. 로딩 화면을 통해 해당 작업이 일어나고 있음을 알린다.

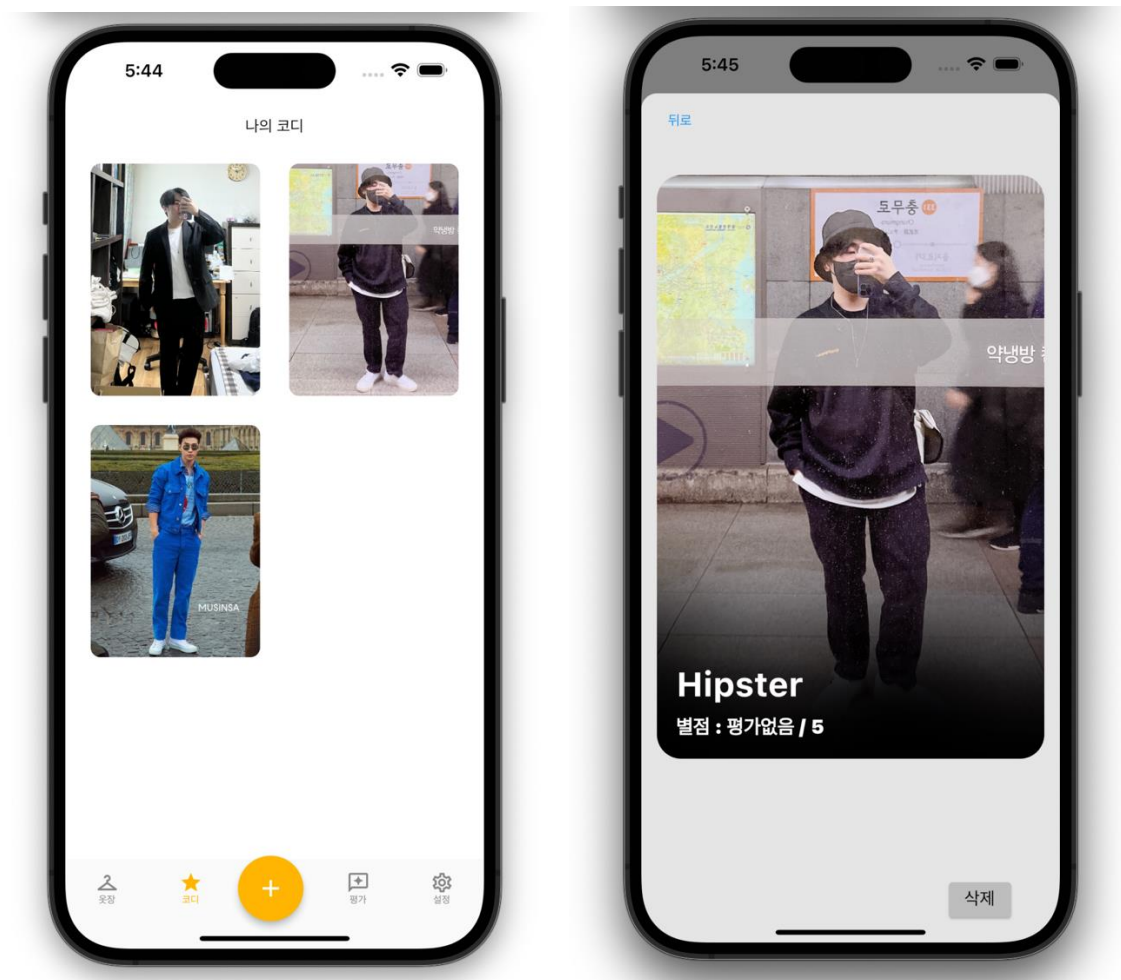


- 옷 추천 결과



최대 3개의 옷 추천 결과를 카드형식으로 받을 수 있다.

3.3 나의 코디



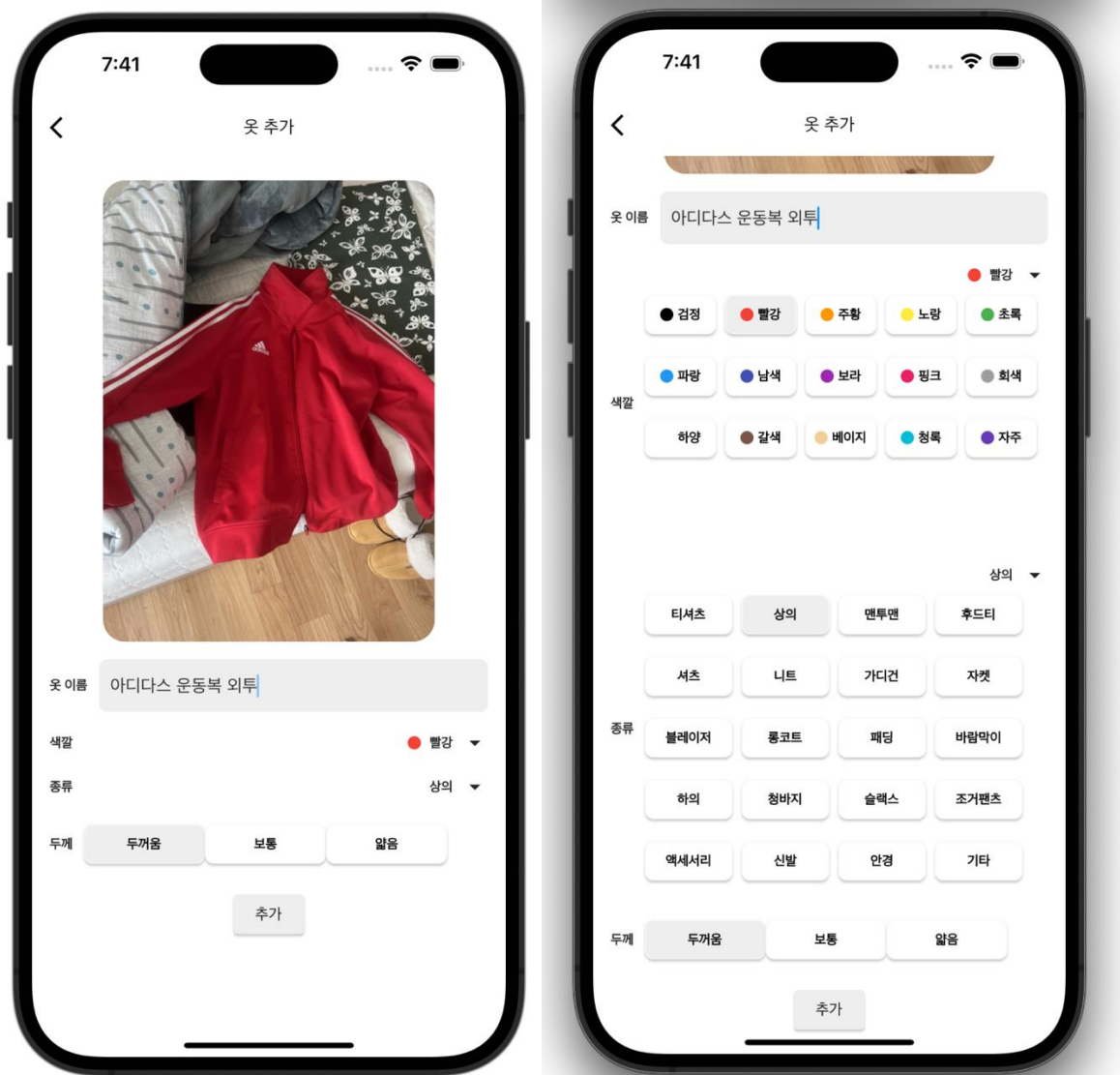
- 나의 코디 관리

옷장과 유사하게 Firebase에 저장된 나의 코디 리스트를 그리드 방식으로 출력한다. 항목을 누르면 세부 정보를 볼 수 있다.

- 점수 출력

코디 세부정보를 선택하면 코디 평가 페이지에서 타인에 의해 평가받은 별점의 평균을 구해 점수가 표시되게 된다. 만약 아직 평가를 받지 못했다면 '평가 없음'이 표시된다.

3.4 옷 업로드



- 나의 옷 업로드

카메라 모양의 사진영역을 누르면, 자신의 갤러리가 열리고, 가지고 있는 옷 사진을 업로드 할 수 있다. 옷 이름 항목에다가 제목을 입력한다.

- 종류, 색깔, 두께 선택

색깔 탭에서는 옷 색깔을 선택 할 수 있다. 종류 탭에서는 옷의 종류를 선택할 수 있다. 두께 탭에서는 두꺼움, 보통, 얇음 세 버튼을 통해 옷 두께를 선택할 수 있다.

옷 추가버튼을 클릭하면 입력한 옷의 정보를 Firebase에 업로드하고 추가된 옷을 옷장 페이지에서 확인할 수 있게 된다.

- **드롭다운 바 형식의 편한 UI**

유저에게 직접 항목을 입력하지 않고 버튼을 통해 간단히 선택할 수 있도록 편의성을 제공하고 사용감을 개선하였다. 옷 두께 또한 두꺼움, 보통, 얇음 세 버튼을 통해 간단하게 선택할수 있도록 유저의 편의성을 생각하여 구현하였다.

3.5 코디 업로드

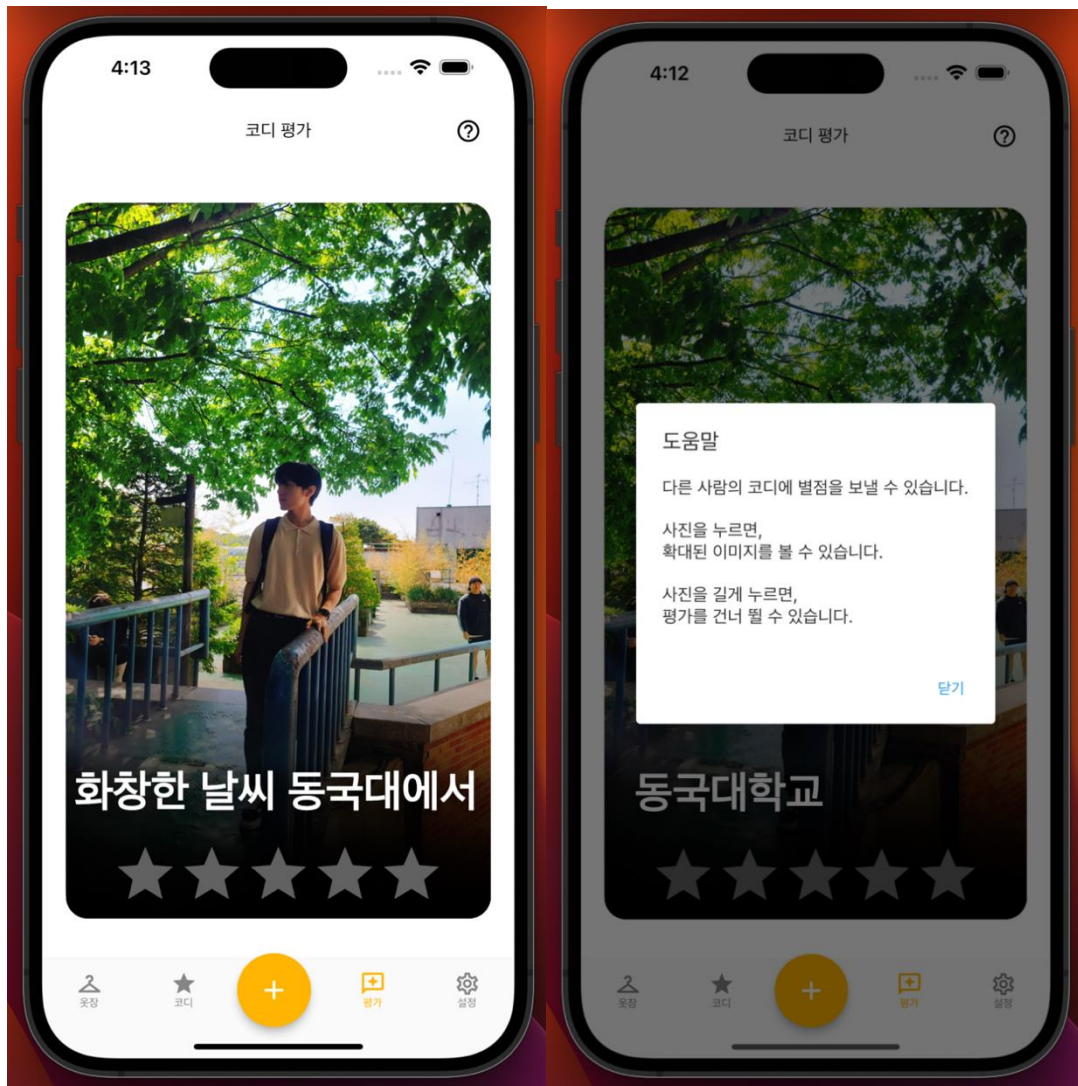


- 나의 코디 업로드

카메라 모양의 사진영역을 누르면, 자신의 갤러리가 열리고, 나의 코디 사진을 업로드 할 수 있다.

제목에서 자신의 코디에 어울리는 이름을 정할 수 있고 그 코디에 대한 간단한 설명을 코디 설명에 적을 수 있다. 업로드 버튼을 클릭하면 코디 정보를 firebase에 업로드하고 나의 코디 페이지에서 확인할 수 있다. 또한 다른 사람의 코디 평가 페이지에서 자신의 코디가 공개되고 별점을 평가받을 수 있게 된다.

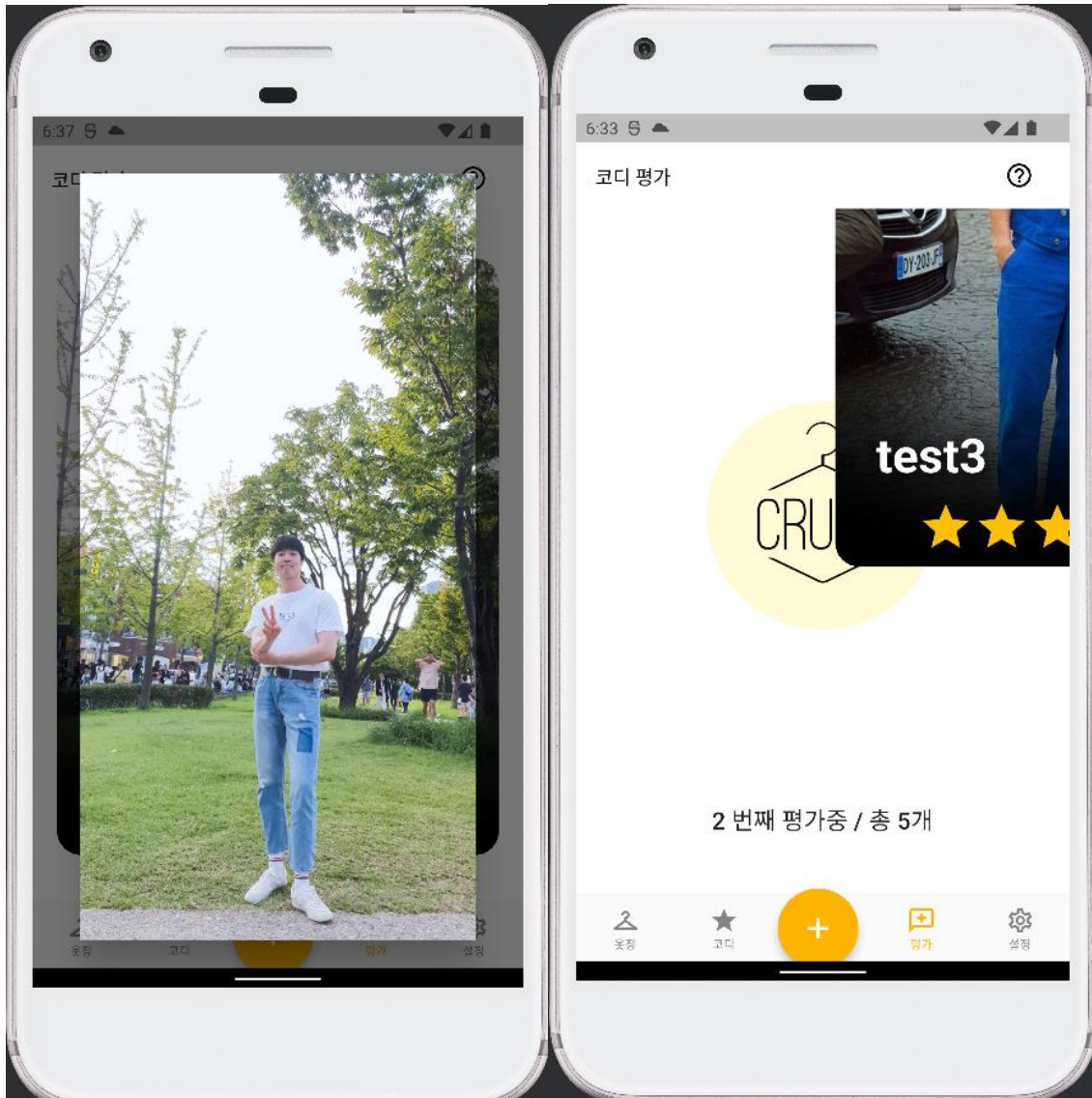
3.6 코디 평가



- **별점 평가 시스템**

Crush의 핵심 기능 중 하나인 유저간의 코디 평가 페이지이다. 코디 평가 페이지에서는 다른 유저들의 코디의 사진과 코디 이름, 그리고 평가를 할 수 있는 레이팅바가 출력된다. 사용자는 코디사진이 마음에 든다고 생각이 들면 높은 별점을 주고, 아니라고 생각이 들면, 낮은 점수를 부여하면 된다.

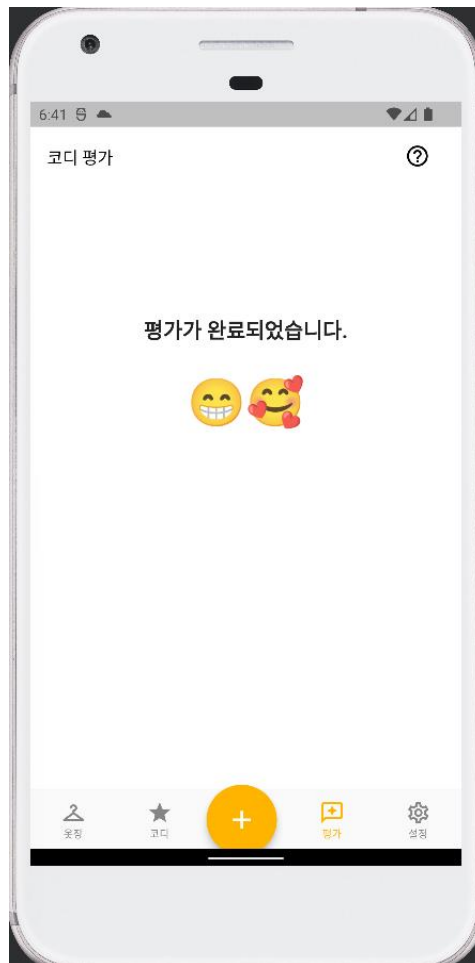
앱 바 오른쪽의 물음표 버튼을 통해 코디 평가 페이지를 이용하는데 도움이 되는 히든 제스처에 관한 도움말을 확인할 수 있다.



● 히든 제스처 및 애니메이션

- (1) 기본 레이아웃은 레이팅바 주변에 디자인적 요소를 위한 검정 오버레이가 존재한다. 이를 통해 심미적 효과는 얻었지만, 코디사진의 하단부가 가려지는 문제점이 존재했다. 그래서 사진을 누르면 확대된 이미지를 확인할 수 있는 기능을 추가하였다. 이를 통해 전체 사진을 확인할 수 있다.
- (2) 또한 사진을 길게 누르면, 평가를 하지 않고 건너뛰는 기능을 추가하였다.
- (3) 또한 심미적 완성도를 위해서 별점을 누르면 화려한 전환효과로 다음 사진으로 넘어간다.
- (4) 별점을 부여하면 화면의 상단 오른쪽으로 평가했던 사진이 넘어가게 되고, 다른 사진이 출력되는 사이에 자신이 지금까지 몇 개의 코디를 평가했고, 총 몇개를

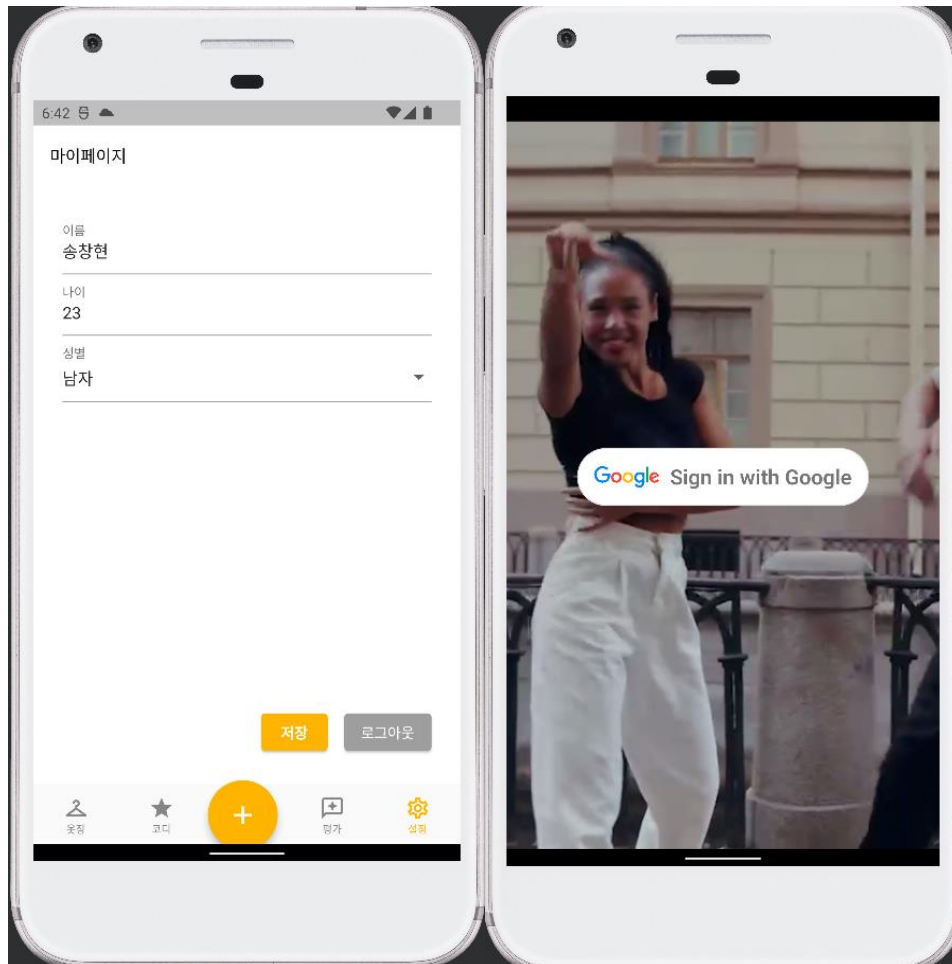
평가할 예정인지 출력하게 된다. 또한 Crush 로고도 출력한다. 이러한 역동적인 애니메이션의 적용을 통해 유저의 사용감을 높이도록 했다.



- 평가 완료

정해진 코디의 개수만큼 평가를 마치면 평가가 완료되었다는 문구가 출력되는 페이지가 표시된다. 너무 많은 코디를 평가하라고 기능을 설정하면 유저가 이탈할 수 있기 때문에 기본 설정으로 5개의 랜덤 코디 사진을 불러오도록 설정했다.

3.7 유저정보 설정



설정 페이지에서는 유저의 이름, 나이, 성별들을 변경하고 저장할 수 있다.

Crush는 구글 계정으로 로그인 할 수 있다.

기존에 이 앱을 사용한 적 없는 유저라면, 이름, 나이, 성별의 초기의 값은 구글 계정에서 설정되어 있는 값을 불러온다. 또한 직접 수정하고 저장할 수 있다.

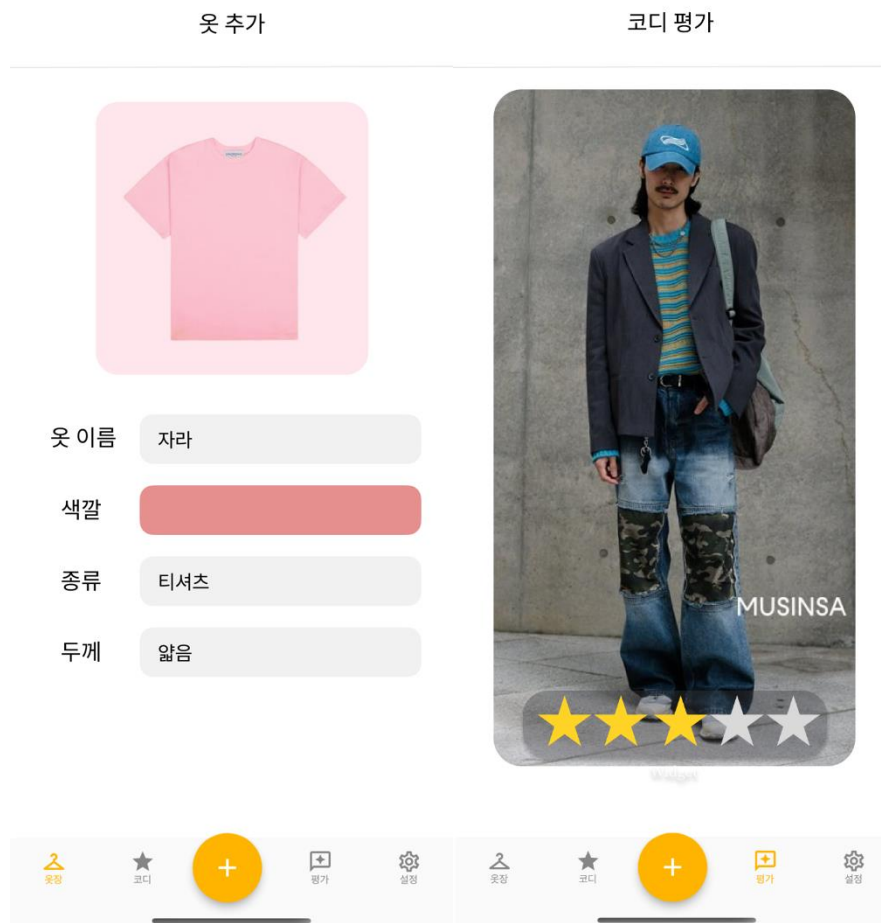
로그아웃 버튼을 누르면 구글 계정이 로그아웃 되며 초기 로그인 화면으로 돌아오게 된다.

나이, 성별 정보는 옷추천시에 활용되며, 이를 유저가 자연스럽게 변경할 수 있도록 도움말을 통해 유도한다.

4. UI/UX 완성도

4.1. 기존 설계와 달라진 UI/UX

* 기존 설계때의 UI

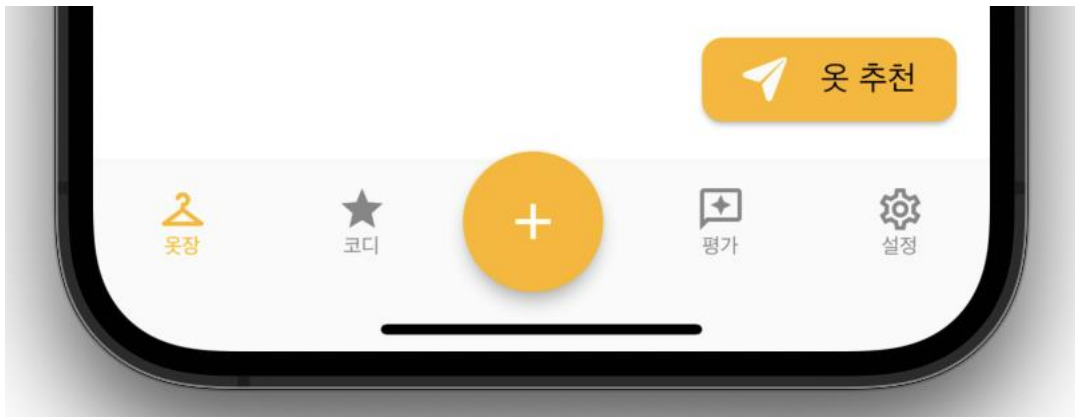


기존 설계에 비해서 전반적으로 UI/UX가 많이 개선되었다.

적으로 설계와 가장 달라진 부분은 옷 업로드 페이지이다. 설계 당시의 옷 업로드 페이지는 옷 색깔을 팔레트 페이지를 새로 열고 가장 비슷한 색을 선택하고 종류와 두께는 텍스트 형식으로 직접 입력하도록 설계하였다.

또한 코디 평가 페이지에도 많은 부분이 설계 당시와 달라지게 되었다. 설계 당시에는 옷 사진과 별점을 매길 수 있는 부분만 보이게끔 디자인 해 옷 평가 기능 자체에 충실하도록 설계했다.

4.2. 사용자 편의성 평가



앱의 기능을 네비게이션 바에 배치함으로써 기능들을 직관적으로 사용할 수 있다.

전반적인 UI 구현에서 직관적인 UX를 도입하였다. iOS의 애플식 디자인은, 사용자가 일단 직관적으로 어떤 행동을 하도록 유도한다. 예를 들어, 나의 코디에서 일단 사진을 눌러보면 확대가 되고, 길게 눌러보면 사진이 넘어가는 히든 제스처를 숨겨두는 것이다.

즉, '직관적인 사용자 경험(Intuitive User Experience)'을 구현했다. 이를 통해 사용자 친화적이고 높은 만족도를 제공하고자 했고, 히든 제스처와 같은 미묘한 기능들도 이를 향상시키는 데 기여한다.

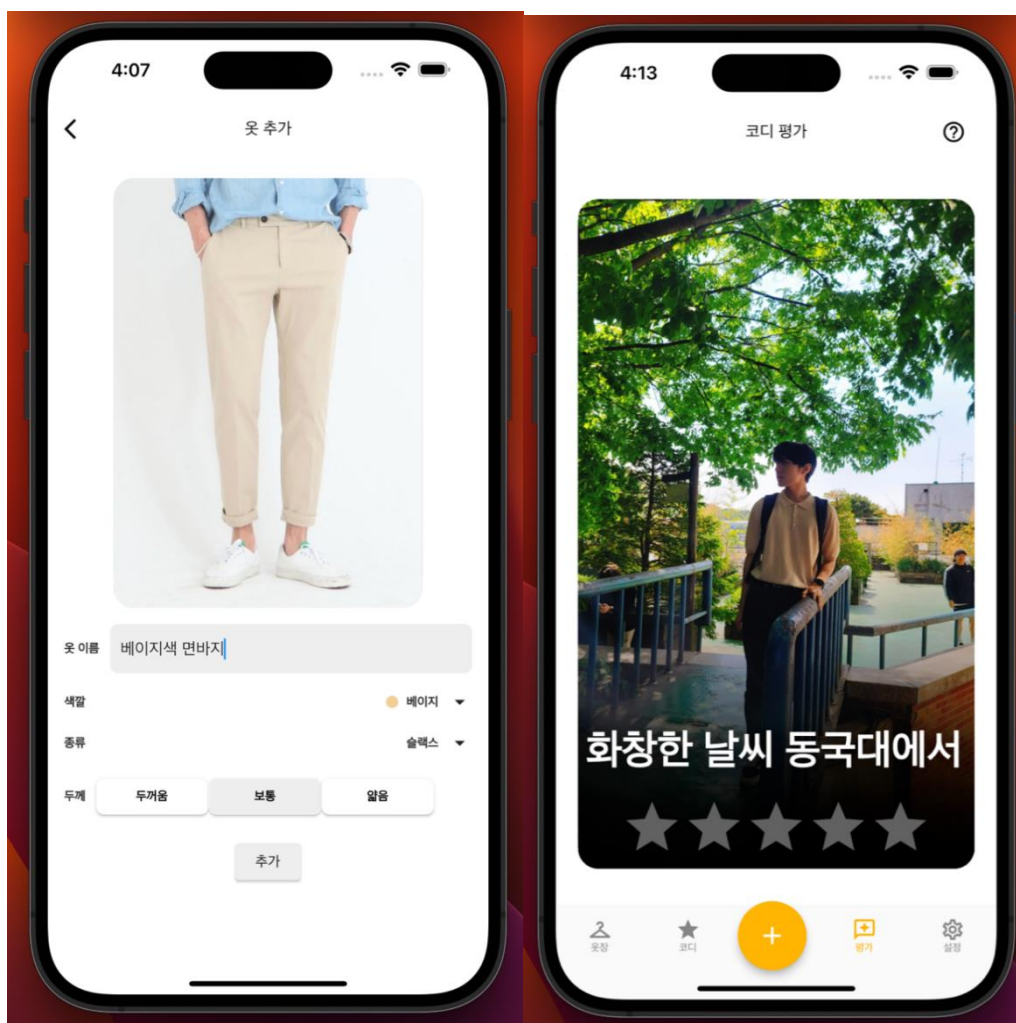
하지만, 이를 숨겨두면 사용자가 영원히 그 기능을 모를 수 있기 때문에 상단에 도움말을 배치하여 해당 기능이 존재함을 알렸다.

또한 설계 당시의 옷 업로드 페이지에서 색깔을 정하기 위해 새로운 페이지로 다시 넘어가는 것은 유저에게 혼란을 줄 수 있었고 옷 종류와 두께를 텍스트로 입력하는 것은 유저에게 어떻게 입력해야 하는지에 대한 고민을 줄 뿐더러 입력하는 방식에 따라 옷장 페이지에서 카테고리 별로 분류하는 것 또한 어려움이 발생할수 있겠다는 의견이 있었다. 또한 텍스트로 직접 입력하는 방식은 유저의 터치수가 굉장히 많이 늘어나기 때문에 앱을 이용하는 데 버튼 형식보다 시간이 많이 걸리기 때문에 불편한 사용감을 제공할 수 있겠다는 의견이 있었다.

이를 위해 구현 시 색깔 부분에 새 팔레트 페이지로 넘어가지 않는 색깔 선택 기능을 부여하기로 했고 옷 종류와 두께는 버튼 형식으로 선택함을 통해 유저들의 사용감을 개선해 보기로 했다.

코디 평가에서는 사진을 자세하게 볼 수 있는 기능과 코디를 눈에 더 잘 들어올 수 있도록 세련된 디자인과 코디 평가 후 사진이 바뀔 때 애니메이션을 추가해서 유저에게 역동적인 사용감을 부여했다.

4.3. UI/UX 반영 결과



옷 업로드 페이지를 실제로 구현할 때는 옷 색깔을 새로운 팔레트 페이지로 넘어가는 것이 아닌 드롭다운 형식의 바를 만들어 색깔 버튼을 클릭해서 선택할 수 있도록 만들었다. 페이지가 넘어가는 수를 줄임으로써 유저들의 사용감에 안정감을 주도록 했다. 또한 옷

종류도 직접 텍스트를 입력하는 것이 아닌 색깔과 비슷하게 드롭다운 형식의 바를 만들어 종류의 버튼을 선택할 수 있게 하였다. 이를 통해 유저의 터치 수를 줄이고 유저가 옷 종류에 대해서 고민하는 시간을 줄일 수 있도록 구현하였다. 두께 또한 세 개의 버튼으로 두께를 선택할 수 있게 하면서 유저의 터치 수를 줄이고 유저의 피로도를 줄일 수 있도록 구현하였다.

옷 평가 페이지에는 애플식 UI를 사용하는데 도움을 줄 수 있도록 숨겨져 있는 기능에 대한 설명이 담긴 도움말 버튼을 추가했고 화면을 출력할 때 사진에 셰이딩 효과를 줘 코디를 부각함과 동시에 사진의 이름과 별점 평가 버튼 또한 자연스럽게 보이도록 했다. 또한 별점을 부여한 뒤 화면 밖으로 날아가는 애니메이션을 추가함으로써 역동적인 사용감을 제공하고 사용자의 몰입도를 증가시키도록 구현하였다.

5. 기능완성도 및 성능 평가

5.1. 옷추천 시스템 정교화

앞서 우리의 Key Value는 유저에게 정교한 옷 추천을 제공하는 것이라고 하였다. 이를 위해 옷추천 성능을 최대한 높여야 한다.

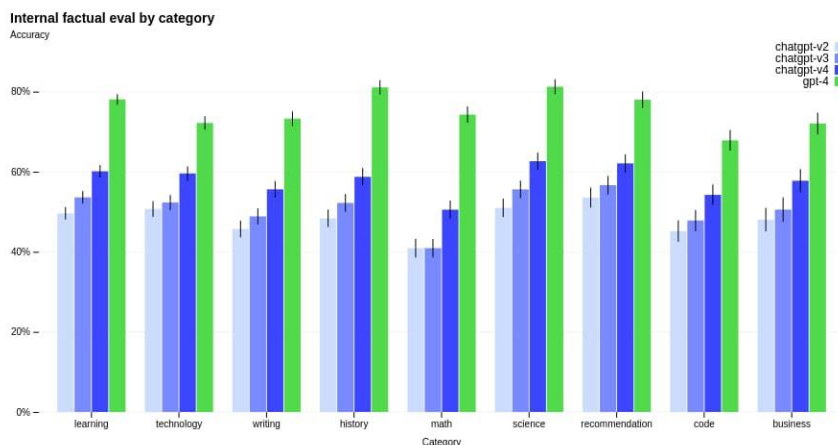
5.1.1 GPT-3.5 vs GPT-4



You're invited to use the OpenAI GPT-4 API!

Your OpenAI organization (org-m80cwXUfF4pt5P0NQUlITwVQe) can now access GPT-4 models with 32K context via the existing OpenAI API. You can learn more about GPT-4 in our [API documentation](#).

Open AI 로부터 개발자 인증을 받아서 GPT-4 및 개발자 API에 접근할 수 있는 권한을 얻었다. 이를 통해 기존 모델보다 진보된 모델을 사용할 수 있었다. 실제 두 모델에는 엄청난 성능차이가 있다. 추론 능력과 엄청나게 오르고, 환각현상이 크게 줄어들었다.



그렇다면, 두 모델이 우리의 옷 추천 시스템에서 큰 차이가 있는지 테스트 하기로 하였다. 테스트에 있어서 25세 여성이라는 페르소나를 부여하고 각 상황에 맞추어 옷 추천 테스트를 진행하고 결과를 비교하였다.

테스트1. 봄, 캐주얼, 맑음, 활기찬

GPT-3.5

첫번째 시도

["Brock 맨투맨", "와이드 조거팬츠", "두꺼운 맨투맨과 와이드 조거팬츠는 활기찬 캐주얼 스타일에 적합합니다."],

["검정 티셔츠", "검은 슬랙스", "검은색 티셔츠와 검은색 슬랙스는 맑은 봄 날씨에 적합한 캐주얼 스타일입니다."],

["우디 곰돌이 티셔츠", "편한 반바지", "우디 곰돌이 티셔츠와 편한 반바지는 캐주얼한 스타일에 적합합니다."]]

두번째 시도

["선글라스", "LA Dogers cap", "여행을 갈 때 적합한 조합입니다."],

["와이드 조거팬츠", "예쁜 스웨터", "활기찬 봄날에 적합한 조합입니다."],

["검은 슬랙스", "Brock 맨투맨", "캐주얼한 모임에 적합한 조합입니다."]]

GPT-4

첫번째 시도

["선글라스", "청바지", "검정 선글라스와 남색 청바지는 젊은 여성에게 어울리며 캐주얼한 봄날의 분위기를 자아냅니다."],

["편한 반바지", "우디 곰돌이 티셔츠", "회색 반바지와 하얀색 우디 곰돌이 티셔츠라는 조합은 활기찬 스타일과 경쾌한 봄날의 코디를 완성합니다."],

["와이드 조거팬츠", "검정 티셔츠", "회색 와이드 조거팬츠와 검정 티셔츠를 매치하면 소소한 캐주얼한 스타일과 봄날의 활기찬 룩을 연출할 수 있습니다."]]

두번째 시도

["선글라스", "언더아머 운동복", "검정 선글라스와 회색 언더아머 운동복 조합이 운동에 적합한 활기찬 스타일과 캐주얼한 분위기를 선사합니다."],

["편한 반바지", "예쁜 스웨터", "회색 반바지와 파랑색의 예쁜 스웨터는 봄 시즌에 어울리며 활기찬 스타일로 캐주얼한 룩을 완성합니다."],

["LA Dogers cap", "검정 티셔츠", "검정색 LA Dogers cap과 검정 티셔츠는 젊은 여성에게 어울리는 활기찬 캐주얼 스타일로 연출 가능합니다."]]

테스트2. 흐림, 포멀, 가을, 어두운

GPT-3.5

첫번째 시도

["Brock 맨투맨", "검정 슬랙스", "검정 캡과 함께 캐주얼한 룩을 완성하세요."], ["Zara 청자켓", "예쁜 스웨터", "베이지 슬랙스와 함께 포멀한 룩을 완성하세요."], ["와이드 조거팬츠", "오버핏 후드티", "편안한 룩을 원하시면 회색 톤의 조거팬츠와 함께 오버핏 후드티를 매치하세요."]]

두번째 시도

["Brock 맨투맨", "와이드 조거팬츠", "두꺼운 옷이 필요한 가을철에 어울리는 스타일입니다."], ["검은 슬랙스", "Zara 청자켓", "포멀한 모임에 어울리는 스타일입니다."], ["예쁜 스웨터", "슬랙스 바지", "포멀하지 않은 모임에 어울리는 스타일입니다."]]

GPT-4

첫번째 시도

["Zara 청자켓", "검은 슬랙스", "남색 두꺼운 청자켓과 검은 슬랙스는 가을 시즌 여성분의 포멀한 어두운 스타일에 적합하며 흐린 날씨에도 따뜻함을 유지해 줍니다. 청자켓은 세련된 무드를 연출하고 슬랙스는 심플한 매력을 더해줍니다."],
["검정 티셔츠", "베이지 슬랙스 바지", "검정 티셔츠와 베이지 슬랙스 바지는 어두운 스타일의 포멀한 룩을 완성할 수 있습니다. 가을 날씨에 적합한 두께와 흐린 날씨에도 잘 어울리며, 25세 여성분의 연령대에 적절한 스타일입니다."],
["APC 맨투맨", "와이드 조거팬츠", "검정 두꺼운 APC 맨투맨과 회색 와이드 조거팬츠는 가을 시즌 어두운 스타일의 아이템으로 포멀한 분위기를 연출하는데 도움이 되며, 날씨의 적합성도 괜찮습니다. 이 조합은 흐린 날씨에도 멋진 포인트로 작용할 것입니다."]]

두번째 시도

["Zara 청자켓", "슬랙스 바지", "남색 청자켓과 베이지 슬랙스는 가을 시즌에 어두운 스타일의 포멀한 룩을 완성할 수 있습니다. 청자켓은 두꺼운 두께로 날씨에 적합하며, 베이지 슬랙스가 세련된 분위기를 선사합니다."],
["검정 티셔츠", "검은 슬랙스", "검정 티셔츠와 검은 슬랙스는 어두운 스타일의 포멀한 룩을 연출할 수 있으며, 가을 날씨에 적합한 두께로 편안함을 더해줍니다. 이 룩은 흐린 날씨에도 잘 어울립니다."],

["예쁜 스웨터", "청바지", "파랑색 두꺼운 스웨터와 남색 청바지는 가을 시즌에 어두운 스타일의 캐주얼한 룩을 완성할 수 있습니다. 스웨터는 포멀한 분위기를 더하고, 청바지는 흐린 날씨에도 무난한 선택으로 알맞습니다."]]

- 결과 해석

테스트 결과, GPT-4가 전반적으로 옷추천 이유를 설명하는 부분에서 GPT-3.5에 비해 보다 우수함을 확인할 수 있었다. '청자켓은 세련된 무드를 연출하고 슬랙스는 심플한 매력을 더해줍니다,' '검정 두꺼운 APC 맨투맨과 회색 와이드 조거팬츠는 가을 시즌 어두운 스타일의 아이템으로 포멀한 분위기를 연출하는데 도움이 되며, 날씨의 적합성도 괜찮습니다.' 와 같은 부분에서 더 풍성한 표현을 작성함을 알 수 있었다.

특히, 테스트 1에서는 큰 차이는 못느꼈지만, 테스트 2에서 상황 이해도가 더 뛰어나다는 것을 알 수 있었다. GPT-4는 다양한 파라미터들을 더 잘 이해한다, 예를 들어 {흐림, 포멀, 가을, 어두운}과 같은 옵션에 대한 이해도가 더 높아 더 적합한 옷 추천을 제공할 수 있었다.

따라서, GPT-3.5 대신 우리의 옷 추천 모델에 GPT-4를 사용하기로 결정했다. GPT-4의 상대적인 성능 우위와 높은 이해도를 활용하여 유저에게 정교하고 개인화된 옷 추천 서비스를 제공할 수 있다. 이를 통해 우리의 Key Value인 유저에게 정교한 옷 추천을 제공하는 것을 실현할 수 있다.

5.1.2 파라미터 조절

모델에서 다음과 같은 파라미터 항목을 조절할 수 있다.

```
Parameters text-davinci-003 ▾ Copy
1  {
2    "model": "text-davinci-003",
3    "prompt": "Say this is a test",
4    "max_tokens": 7,
5    "temperature": 0,
6    "top_p": 1,
7    "n": 1,
8    "stream": false,
9    "logprobs": null,
10   "stop": "\n"
11 }
```

여기서 답변의 다양성을 결정하는 부분은 temperature, top_p 이다.

Open AI에서는 temperature의 설정값을 0.5~1.0으로 권장하고 있으며 기본설정값은 0.5이다. 값이 높아질수록 다양한 답변을 생성해낸다. 우리는 창의성을 확보하되, 어느정도 일관된 답변이 필요해서 temperature를 0.8로 설정했다.

```
ChatGptRequest chatGptRequest = ChatGptRequest.builder()
    .model("gpt-3.5-turbo")
    .messages(chatGptMessages)
    .maxTokens(500)
    .temperature(0.8)
    .topP(0.9)
    .build();
```

5.1.3 프롬프트 엔지니어링

Raw GPT4에는 아무런 역할이 부여되어 있지 않다. 예를 들어 대중들에게 익숙한 Chat-GPT의 경우 사용자에게 충실한 챗봇 역할을 하기 위해 역할이 사전에 설정되어 있다.

Crush에서 GPT 시스템을 적절히 하기 위해서는 이러한 역할 설정이 필요하다. 이것을 프롬프트 엔지니어링(Prompt engineering)이라고 한다. 즉, 사용자의 옷 정보를 분석하고, 그 결과를 파싱해서 출력하도록 사전설정을 해야하는 것이다.

이를 위해서는 System prompt, Assistant prompt, user message가 필요하다.

- System prompt

모델이 지켜야 하는 절대적인 규칙. 모델의 역할, 제한사항 등을 설정할 수 있다.

Crush에서 사용한 프롬프트는 다음과 같다.

```
"You are a competent fashion stylist. Look at a given set of clothes and their
conditions and recommend suitable combinations in Korean. You may add
appropriate advice, but it must be appropriate for the given options. You can
also recommend clothes that would be a good addition to the existing ones."
+ "Follow the output form unconditionally: [[₩"cloth1₩",₩"cloth2₩",₩"describe
Why we recommend it₩"],[ ₩"cloth1₩", ₩"cloth2₩", ₩"describe Why we
recommend it₩"], ...]. Up to 3 combinations."
```

- Assistant prompt

핵심 규칙과 동반되는 보조 규칙이다. 절대성을 가지지는 않지만 모델이 참고해야 하는 사항이다.

- User message

유저 메시지. Crush에서는 옷정보가 입력된다.

실제 위의 프롬프트가 원하는 결과를 출력하고 안정성이 있는지 테스트 하였다. 그 결과는 'API 안정성 테스트'에서 기술하겠다.

5.1.4. 파인튜닝

옷 추천에 있어서 기본적으로 GPT4가 각종 요소들을 고려해 옷 추천 조합을 잘 생성해내지만, 아직 몇가지 문제점이 남아있다. 우리가 원하지 않는 결과가 나올 수가 있고, 기본적인 색상 조합 등에서 원하지 않는 결과가 나올 수 있다. 또한 최신 트렌드정보는 잘 모를 수 있다. 이를 해결하기 위해 파인튜닝과 같은 효과를 얻어서 옷추천의 정확도를 높이고자 한다.

Prepare training data

Training data is how you teach GPT-3 what you'd like it to say.

Your data must be a [JSONL](#) document, where each line is a prompt-completion pair corresponding to a training example. You can use our [CLI data preparation tool](#) to easily convert your data into this file format.

```
1 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
2 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
3 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
4 ...
```

모델을 커스터마이징 하기 위해서는 위와 같은 형식으로 데이터를 준비해야 한다.

(1) 색조합

인터넷으로 아래와 같은 색 조합 표 등을 한번쯤 보았을 것이다.

모델이 아래와 같은 정보들을 참고할 수 있도록 데이터를 준비했다.

모델에게 같은 색의 조합에 대한 데이터와 일반적인 선호도를 입력했다.


```

{
{topcolor:white, bottomcolor:beige, result:excellent},
{topcolor:blue, bottomcolor:green, result:terrible},
{topcolor:blue, bottomcolor:red, result:good},
{topcolor:grey, bottomcolor:black, result:excellent},
{topcolor:yellow, bottomcolor:purple, result:poor},
{topcolor:pink, bottomcolor:brown, result:terrible},
{topcolor:navy, bottomcolor:kahki, result:good},
{topcolor:green, bottomcolor:orange, result:poor},
{topcolor:black, bottomcolor:purple, result:good},
{topcolor:white, bottomcolor:navy, result:excellent},
{topcolor:red, bottomcolor:gray, result:good},
{topcolor:black, bottomcolor:yellow, result:poor},
{topcolor:orange, bottomcolor:olive, result:terrible},
{topcolor:beige, bottomcolor:brown, result:excellent},
{topcolor:black, bottomcolor:white, result:excellent},
{topcolor:purple, bottomcolor:pink, result:poor},
{topcolor:white, bottomcolor:skyblue, result:good},
{topcolor:lightgray, bottomcolor:darkgray, result:excellent},
{topcolor:brown, bottomcolor:darkgreen, result:terrible},
{topcolor:blue, bottomcolor:gray, result:good}
{topcolor:grey, bottomcolor:green, result:terrible}
{topcolor: grey, bottomcolor:skyblue, result:terrible}
{topcolor:white, bottomcolor:white, result:good}
{topcolor:red, bottomcolor:red, result: poor}
{topcolor:blue, bottomcolor:blue, result: poor}
}

```

(2) 최신 정보

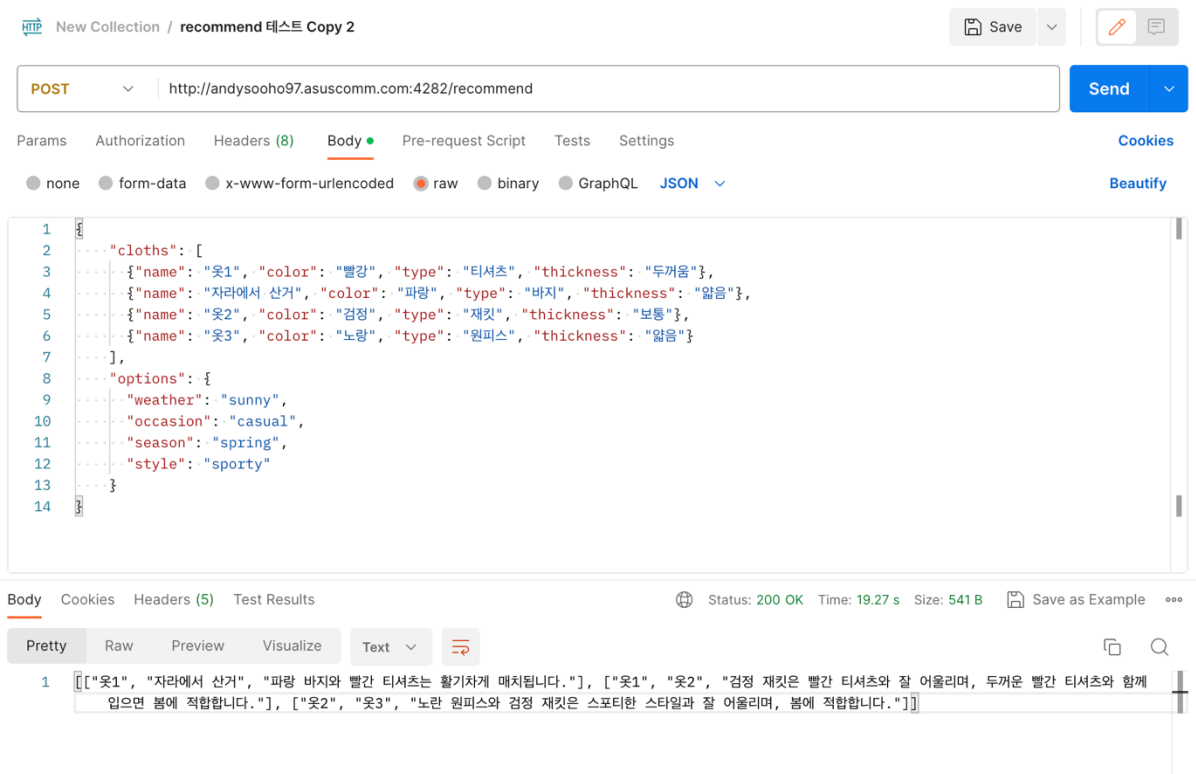
기존 모델은 최신 유행 정보등을 모르는 한계점이 있다. 이를 다음과 같은 형식으로 참고하도록 할 수 있다.

2023년의 봄, 여름 트렌드이다.

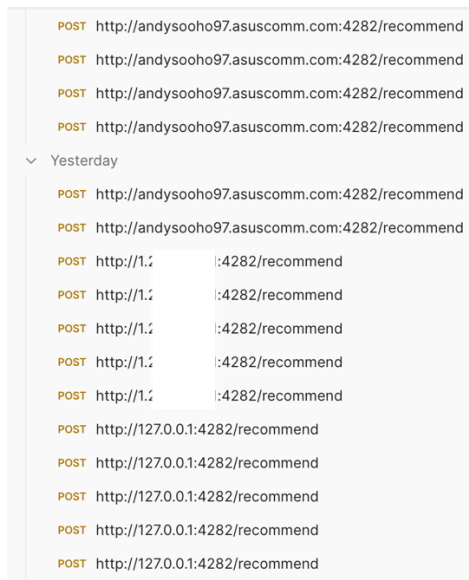
```
{
{"Utility Look": "A practical style featuring cargo pants and jackets with large pockets and utility details. Seen in brands like Miu Miu, Fendi, and Isabel Marant."},
{"Party Look": "A flashy and unique style using sequins and metallic materials. Showcased by brands like Valentino, Chanel, and Dolce & Gabbana."},
{"Casual Office Look": "A comfortable and versatile style using denim materials. Presented by brands like Miu Miu, Jacquemus, and Tibi."},
{"Feminine Look": "A sensual and soft style inspired by lingerie using lace, sheer, and satin materials. Suggested by brands like Chanel, Simone Rocha, and Bottega Veneta."},
{"Bright Colors": "A hopeful style using bright colors like yellow, purple, and pink. Incorporated by brands like Max Mara, Jacquemus, and Altuzarra."},
{"Denim": "A style featuring all-denim outfits or various items made of denim material. Recommended by brands like Miu Miu, Jacquemus, and Celine."}
}
```

5.2. API 안정성 테스트

앞서 진행한 프롬프트 엔지니어링의 안정성 테스트를 진행했다.



테스트용으로 간소화된 데이터를 만들었으며, 결과가 정상적임을 알 수 있었다.



또한, 20번의 연속 전송 결과, 모든 데이터가 안정적인 형태로 도착함을 확인하였다.

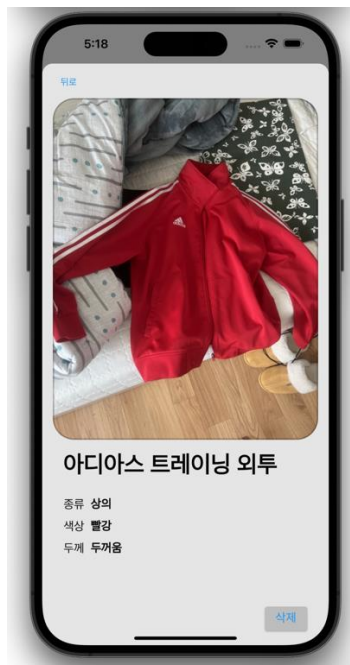
도커를 이용해 외부 서버에 직접 배포한 서버의 안정성도 문제가 없다.

5.3. DB 안정성 테스트

백엔드인 Firebase의 동작도 원하는 데로 다 작동해야한다.

● 정보 fetch

Users > 9QedclOELhW8NslElyXhYhigjq32 > clothes > 5KwWkqRMvZv... Google Cloud의 추가 기능		
9QedclOELhW8NslElyXhYhigjq32	clothes	5KwWkqRMvZvDHv9igY4c
clothes >	0vcHwq26tBYyIdgNZr1u 1leyCT6RgnF1l3E46ohQ 5KwWkqRMvZvDHv9igY4c > 8ddPDgb3gCk7afTuFkHs 9PaKzgx76bJXamMIGe0o AaFopEuM3dj2ZlgDZc7 Av0o5cFI0gJMoag0Sg86 06o1vcKXNhruiX0trpsV SjZYrqP3SZh6j4az6Fxs T3TEAg0H68vW3016Z8bw UNmjzR1lwGFEF3xuf1yP WzzYpEerkYS11tRTXEs6 itsBDZw5084cyTYImaLD kKHGaemISa8rRtsTlgZc ov5BCNMC0Uqo2pjWkiYJ	clothId: "5KwWkqRMvZvDHv9igY4c" color: "빨강" imageUrl: "https://firebasestorage.googleapis.com/v0/b/crush-5f8dc.appspot.com/o/clothes%2F9QedclOELhW8NslElyXhYhigjq3E880-4B98-9832-6046489B7926-18144-00000061700B9EE0.jpg?45c6-4cf1-8864-9422f58a087f" name: "아디아스 트레이닝 외투" thickness: "두꺼움" type: "상의"



정보를 올바르게 받아온다.

- 정보 갱신

정보가 정상적으로 갱신되는 것을 확인할 수 있다. 또한 삭제 또한 정상적으로 잘 이루어진다.

61g5J8xBpWXkzIV2Grbjr08VeTN2

clothes

age: 24
email: "royalex@gmail.com"
name: "멋쟁이"
sex: "male"

61g5J8xBpWXkzIV2Grbjr08VeTN2

clothes

age: 26
email: "royalex@gmail.com"
name: "안준수"
sex: "male"

유저 이름, 나이 변경 했을 때의 firebase 변화

- 객체 삭제

옷, 코디가 정상적으로 삭제된다.

- 별점 추가

별점을 선택하면 해당하는 코디의 ratings 서브 디렉토리에 별점이 잘 추가 된다

- 객체 업로드

옷, 코디 를 업로드 하면 Firebase에 정보가 잘 업로드 되고 사진파일도 FireStore에 잘 업로드 됨을 알 수 있다.

6. 기술 내재화 및 독창성

6.1. 독창적 아이디어

본 프로젝트에서 제안하는 독창적 아이디어는 사용자 소유 의류를 바탕으로 한 인공지능(AI) 기반의 개인 맞춤형 코디 추천 서비스와 별점을 통한 평가가 이루어지는 커뮤니티 기능이다. 이를 통해 개개인의 의류 데이터와 취향을 고려하여 최적화된 코디 제안을 제공하며, 사용자들 간 패션에 대한 피드백을 촉진하고 사용자 자신의 코디에 대한 다른 사용자들의 평가를 직관적으로 알 수 있도록 한다.

사용자 소유의 의류를 바탕으로 인공지능을 활용한 코디 추천 기능은 기존의 일반적인 패션 플랫폼에서 경험할 수 없는 차별화된 가치를 제공한다. 이를 통해 사용자들은 자신의 옷장에 있는 의류를 더 효과적으로 활용할 수 있으며, 개성 있는 스타일을 구축할 수 있다.

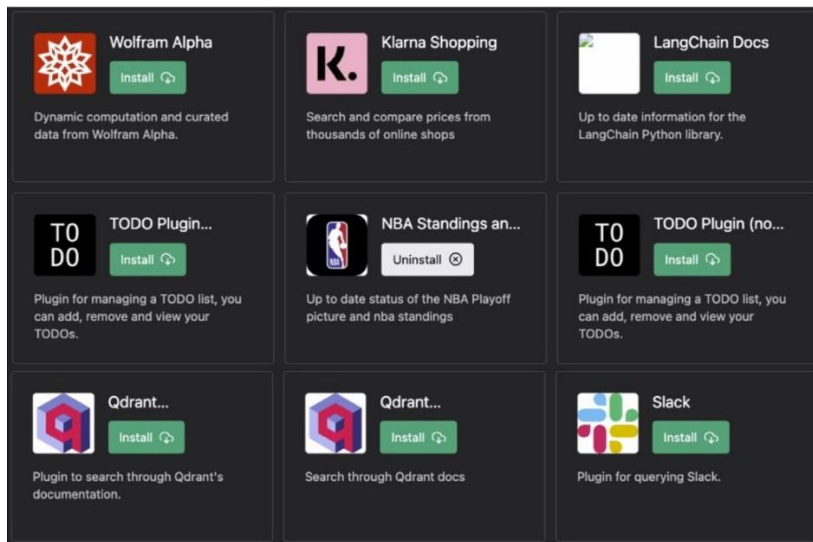
또한, 사용자들이 직접 자신의 코디 사진을 올리고 다른 유저들로부터 별점을 통한 평가를 받는 커뮤니티 기능을 도입함으로써, 패션에 대한 의견을 교환하고 서로의 스타일을 참고할 수 있는 환경을 제공한다. 이를 통해 사용자들은 서로의 코디에 대한 피드백을 주고 받을 수 있으며, 다양한 패션 정보를 공유한 결과로 패션 감각을 더욱 향상시킬 수 있다.

이러한 두 가지 독창적인 기능을 통해 기존의 서비스들과 차별화된 기능을 제공한다.

6.2. 현실세계를 잇는 차세대 인터페이스 설계

옷 조합 추천 시스템에 대규모 LLM인 GPT를 활용했다. 만약 추천 모델을 직접 구축했다면 좋았겠지만, 이를 위해서는 대규모의 데이터 셋이 필요하다는 현실적인 문제에 부딪혔었다. 비록 Gpt를 프로젝트에 사용했지만, 단순히 모델을 가져와서 사용한 것이 아니다.

일반적으로 LLM은 Chat-Gpt처럼 챗봇이라고 생각하는 사람들이 많다. Crush는 챗봇 형태로 LLM을 이용하는 것이 아니라, 내가 현재 가지고 있는 옷을 모은 옷장이라는 현실세계와 인공지능을 연결지어주는 새로운 인터페이스를 설계한 것이다.



이는 Open AI의 비전과도 일치한다. Open AI는 '플러그인'을 통해 AI와 다른 서비스 및 유저의 현실세계와 소통하려는 시도를 하고 있다.

Crush는 옷 추천이라는 사용자에게 훌륭한 기능을 제공하지만 그 이면에는 현실세계와 AI를 잇는 고도의 API 설계 기술이 바탕이 되어 있다.

7. 팀워크 및 역할 분배

7.1. 팀원별 역할 분배

2018112158 이승규	2017112211 김수호	2018112167 안준수	2018112134 송창현
<ul style="list-style-type: none">- 프로젝트 전체 관리- 구글 인증 연결- firebase 연결	<ul style="list-style-type: none">- 스프링 서버, API 통신- 나의 코디, 유저 정보 설정, 코디 업로드 구현	<ul style="list-style-type: none">- AI 옷 추천 기능 구현- 옷 평가 기능 구현	<ul style="list-style-type: none">- 옷장, 옷 업로드 기능 구현- 앱 전체 UI/UX 설계

- 이승규: 프로젝트 전체 관리, 구글 인증 연결, firebase 연결
- 김수호: 스프링 서버 통신, API 통신, 나의 코디 페이지 구현, 유저 정보설정 기능 구현, 코디 업로드 기능 구현
- 안준수: AI 옷 추천 기능 구현, 옷 평가 기능 구현
- 송창현: 옷장 기능 구현, 옷 추가 페이지 기능 구현, 앱 전체 UI/UX 설계

7.2. 협력 및 의사소통 과정

1. 깃허브(GitHub)

<input type="checkbox"/>	코디 평가 기능 구현 coordi eval enhancement		
	#48 by royalex1003 was closed last week		
<input type="checkbox"/>	앱 재실행시 유저정보 초기화 bug		2
	#46 by andysooho was closed 3 weeks ago		
<input type="checkbox"/>	Appbar 두개 출력 bug		
	#44 by andysooho was closed 3 weeks ago		
<input type="checkbox"/>	나의 코디 Model my coordi		1
	#42 by andysooho was closed last week		
<input type="checkbox"/>	코디 평가 view - 오버플로우 bug coordi eval		3
	#41 by andysooho was closed 3 weeks ago		
<input type="checkbox"/>	마이 페이지(My Page) 제작	1	
	#39 by andysooho was closed last month		
<input type="checkbox"/>	옷 추천 firestore 연동 closet recommend enhancement		1
	#37 by royalex1003 was closed 4 days ago		
<input type="checkbox"/>	splash_screen 제작 client		
	#35 by andysooho was closed 2 weeks ago		
<input type="checkbox"/>	코디 평가 UI coordi eval	1	
	#32 by royalex1003 was closed last month		
<input type="checkbox"/>	옷장 옷 출력 로직 수정 bug client	1	
	#29 by seungy0 was closed on May 10		
<input type="checkbox"/>	코디 평가 UI client	1	
	#25 by royalex1003 was closed on May 10		
<input type="checkbox"/>	나의 코디 View my coordi	1	
	#24 by andysooho was closed last month		
<input type="checkbox"/>	firestore 연동 client	1	
	#23 by seungy0 was closed on May 6		
<input type="checkbox"/>	tab_view 5개로 개선 enhancement		
	#20 by andysooho was closed on Apr 21		
<input type="checkbox"/>	Intro video 용량 압축 및 수정		2
	#19 by andysooho was closed last month		
<input type="checkbox"/>	로그인 화면 구현 client		3
	#12 by andysooho was closed on May 7		
<input type="checkbox"/>	애니메이션 효과 제작 및 적용		
	#11 by seungy0 was closed 2 weeks ago		
<input type="checkbox"/>	firebase auth 추가 client	1	
	#10 by seungy0 was closed on Apr 6		
<input type="checkbox"/>	firebase 연동 client network	1	
	#8 by seungy0 was closed on Mar 30		
<input type="checkbox"/>	옷 조합 기록 client		
	#7 by seungy0 was closed 2 weeks ago		
<input type="checkbox"/>	추천 조합 출력 closet recommend enhancement		
	#6 by seungy0 was closed last week		
<input type="checkbox"/>	옷 입력 기능 closet enhancement	1	1
	#5 by seungy0 was closed 2 weeks ago		
<input type="checkbox"/>	네트워크 통신 구현 client network		
	#3 by seungy0 was closed on May 7		
<input type="checkbox"/>	옷장 위젯 제작 client		1
	#2 by seungy0 was closed last month		
<input type="checkbox"/>	메인페이지 제작 client	1	1
	#1 by seungy0 was closed on Apr 21		

여러 사람이 동시에 작업하고 버전 관리를 할 수 있는 분산 버전 관리 시스템인 Git을 기반으로 한 깃허브를 이용해 프로젝트를 진행하였다.

깃허브를 통해 모든 멤버들이 동시에 작업을 하면서 변경사항을 실시간으로 공유할 수 있었기 때문에 효율적으로 협업이 가능했다. 각자의 독립적인 브랜치에서 작업을 하고 변경사항을 수월하게 통합할 수 있었다.

또한 깃허브를 이용한 버전 관리를 통해 어떤 팀원이 언제 코드를 변경했는지 추적할 수 있기 때문에 예상치 못한 에러가 발생할 경우 이전 버전으로 롤백하거나 원인을 쉽게 찾아 수정하기 수월했다.

또한 서로의 코드를 리뷰하면서 코드 품질을 향상시키고 버그를 사전에 찾아 낼 수 있었다. 이 과정을 통해서 서로의 지식을 공유하고 커뮤니케이션을 촉진하며 팀원 개개인의 성장도 이끌어 낼 수 있었다.

또한 작업 내용을 깃허브 중앙 서버와 개발자들의 로컬 환경에 복제해서 저장해서 작업하기 때문에 데이터 손실과 시스템 장애에도 대비할 수 있는 안정성과 신뢰성을 확보했다.

추가적으로 동시에 작업할 때의 서로의 작업 사이의 충돌을 발생하는 것을 최소화 하기 위해 서로의 작업 범위를 분리하고 서로 작업하기 전에 최신 변경 사항을 가져와 충돌을 미연에 방지했다. 또한 많은 횟수의 commit 과 pull request를 통해 코드를 수시로 통합했고 테스트를 자주 진행하여 프로젝트의 안정성과 품질을 유지했다.

마지막으로 깃허브의 추가 플러그인인 zenhub를 이용해 프로젝트의 이슈 관리와 목표와 일정 등을 명확히 정의하고 공유하여 프로젝트가 순탄히 진행될 수 있게 하였다.

이러한 깃허브의 장점들과 사용방식을 이용해 팀 내에서 충분한 커뮤니케이션과 협력을 하면서 프로젝트를 진행시켜나갔다.

2. 노션(Notion)



문서 작성, 작업 관리, 프로젝트 관리, 지식 공유, 일정 조율 등을 통합한 협업 도구인 노션을 통해서 프로젝트를 진행하였다.

노션을 통해 프로젝트의 계획, 요구 사항, 기능 명세서, 설계, 기술 문서를 작성했고 테이블을 통해 수행해야 할 작업을 관리했다. 할일 목록과 작업 상태, 우선 순위, 담당자 등을 정의하고 업데이트를 함을 통해서 개별적으로 진행해야 하는 작업의 분담을 수월하게 할 수 있었다.

또한 회의록 기능을 통해 회의 일정을 조정하고 회의록을 작성함으로써 팀 내의 토의 내용과 결정 사항을 기록하고 공유하여 팀 내의 정보 공유와 의사 결정을 원활하게 했다.

이러한 노션의 기능을 이용해 프로젝트 관련 정보를 중앙 집중화 할수 있었고 프로젝트 자료들을 저장, 공유할 수 있었다. 프로젝트를 진행하면서 필요한 자료들을 팀원들이 쉽게 찾을 수 있었으며 정보를 업데이트 하기에에도 용이했다.

또한 실시간 협업 기능을 제공하기 때문에 여러 팀 구성원이 동시에 문서를 편집하고 업데이트 할 수 있었다. 프로젝트 설계 단계에서 팀원들의 의사 소통과 협업이 유연하게 이루어질 수 있도록 해 주었다.

노션을 여러가지 뷰와 템플릿을 제공하기 때문에 작업 관리 또한 유연하게 할 수 있었다. 팀원 각자의 스타일에 맞게 할일 목록을 구성하고 각자의 작업 상황을 시각적으로 파악할 수 있었다.

이러한 노션의 뛰어난 프로젝트 관리와 상호 협력을 위한 도구를 이용해 팀의 생산성과 효율성을 높일 수 있었다.

8. 결론 및 향후계획

8.1. 프로젝트 총평

이번 Crush 프로젝트를 통해 사용자들은 지금까지는 볼 수 없었던 AI 코디 추천과 옷장 기능, 그리고 사용자 간의 코디 평가 기능을 한 플랫폼에서 만나 볼 수 있게 되었다.

AI 코디 추천 기능은 사용자들이 개인이 소지한 옷과 분위기, 날씨 등 여러가지 패러미터를 기반으로 최적의 코디를 제안받을 수 있다. 이를 통해 사용자들은 패션에 대한 고민 없이 멋진 코디를 손쉽게 찾을 수 있다 또한, 옷장 기능을 통해 사용자들은 자신의 옷들을 체계적으로 관리하고 필요한 때에 쉽게 찾을 수 있다. 옷의 종류 탭을 제공하여 부위 별로 옷을 찾아내고 조합할 수 있어 사용자들의 패션 관리를 효과적으로 지원한다.

또한, Crush 프로젝트에서 제공되는 사용자 간의 코디 평가 기능은 사용자들끼리 코디를 공유하고 피드백을 주고받을 수 있는 소통의 장을 제공한다. 이를 통해 사용자들은 서로의 코디에 대한 의견을 나누고 옷 선택에 대한 도움을 받을 수 있으며, 패션에 대한 주관과 영감을 얻을 수 있다.

프로젝트 전반적으로 사용자 중심의 디자인과 사용자 경험에 집중한 구현이 이루어져 있다. 사용자들은 직관적이고 편리한 인터페이스를 통해 쉽게 원하는 기능을 이용할 수 있으며, AI 코디 추천과 옷장 기능을 통해 패션 관리와 스타일링에 대한 부담을 줄일 수 있다.

Crush 프로젝트는 사용자들에게 새로운 패션 경험을 제공하면서도 편리하고 실용적인 기능을 제공하는 플랫폼으로, 패션에 관심이 있는 사람들에게 큰 가치를 제공할 것이다. 사용자들은 이 플랫폼을 통해 지금까지는 볼 수 없었던 기능들과 존재하는 기술의 융합을 경험하고 패션 센스를 더욱 발전시킬 수 있을 것이다.

Crush 프로젝트는 사용자 뿐만 아니라 개발자에게도 귀중한 경험이 된 프로젝트였다.

Crush 프로젝트를 통해 다양한 기술과 기술 스택을 적용하는 과정에서 문제를 해결하고 의사소통하며 협력하는 방법을 배울 수 있었다. 깃허브와 노션 등의 플랫폼을 이용해

팀원들과의 조율과 협업, 버전 관리, 코드 리뷰 등을 경험하면서 소프트웨어 개발의 과정을 이해할 수 있었다.

또한 간단한 시점의 변화로 혁신적이고 새로운 기술을 만들어 낼 수 있음을 배울 수 있었다. Crush 프로젝트의 옷 평가 커뮤니티의 방식은 지금까지는 데이팅 앱에서 많이 사용되었던 커뮤니티 방식이었으나 유저간의 옷 평가로 대상을 바꿈으로써 지금까지 만나보지 못했던 새로운 기술이 탄생하였다. 이 외에도 여러가지 기술의 융합으로 새로운 가치를 만들어 낼 수 있음을 느낄 수 있었다.

Crush 프로젝트는 개발자들에게 귀중한 경험과 성장의 기회를 제공한 프로젝트였다. 기술적인 면에서의 성장과 함께 기획 능력과 창의성을 향상시킬 수 있는 소중한 경험을 얻을 수 있었다.

8.2. 향후 개선사항 및 앞으로의 계획

앞으로 구글 플레이 스토어에 올려 실제 유저들의 데이터를 얻고 이렇게 얻은 데이터를 분석하여 다른 사업과 연계할 수 있는 방법을 찾고자 한다. Crush 프로젝트의 코디 평가 커뮤니티를 통해 나이, 성별에 맞춰 어떠한 옷과 코디를 선호하는지 데이터화 할 수 있으며 그 데이터를 다른 사업에 이용할 수 있도록 데이터베이스를 제공하고자 한다.

Crush 프로젝트를 유저들의 편의성을 제공하는 패션 통합 플랫폼으로 멈추는 것이 아닌 패션에 대해 더 많은 가능성을 탐색하고 발전 시킬 수 있는 아이템으로 성장시키고자 한다.

부록

9.1. 참고문헌

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/>

9.2. 소스코드

https://github.com/seungy0/crush_client