

캡스톤 제안서(계획서)

프로젝트	제목	AI를 활용한 자동 코디추천 어플리케이션 개발		
팀장	팀명	Team WD		
	성명	김민성	학번	201958003
	연락처	010-2268-9396		
	E-MAIL	ava9797@hs.ac.kr		
구분	성명	학번	E-MAIL	연락처(H.P)
팀원 인적사항	김로은	202258011	kdhok2285@hs.ac.kr	010-3683-2285
	이유빈	202258016	miny2482@hs.ac.kr	010-7566-4295
	정민준	202058010	nicemj0914@hs.ac.kr	010-5364-2817
지도교수	박기홍 (서명)			
<p>본인과 팀원은 2025학년도 1학기 캡스톤 디자인1 과목의 설계 프로젝트에 대한 캡스톤 계획서를 다음과 같이 제출합니다.</p> <p>2025 년 4 월 28 일</p> <p>팀 장 : 김민성 (서명)</p> <p>한신대학교 컴퓨터공학부</p>				

목 차

1. 문제 및 목적	2
2. 관련 연구	3
3. 주요기능 및 기대효과	8
4. 개발 환경	14
5. 위험 요소	14
6. 일정 계획	16
7. 참고 문헌	16

1. 문제 및 목적

가. 문제

- 옷 코디에 오랜 시간 걸리는 사람들을 위한 코디 추천
 - 코디에 어려움을 겪는 사람들은 옷 입을 때 어떤 옷을 입어야 할지 오랫동안 고민한다. 패션 유튜브나 블로그 포스팅만으로는 따라 하기 어려운 기존의 문제점을 보완하기 위해 사용자의 취향을 고려하여 코디를 보다 쉽게 할 수 있도록 도와준다.
- 일정과 날씨 등을 고려한 스마트 코디 추천
 - 일정과 날씨를 고려하지 못하고 외출하여 당혹한 경험이 간혹 발생할 수 있다. 이러한 문제점을 보완하기 위해 날씨 예보와 일정을 분석하여 TPO에 적절한 옷을 입고 외출할 수 있도록 도움을 준다.
- 한눈에 보이는 나만의 가상 옷장
 - 많은 옷을 보유한 경우, 옷장에 모든 옷이 한눈에 들어오지 않기 때문에 방치되는 옷이 생기거나 중복된 옷을 구매하는 경우가 있다. 이를 보완하기 위해 보유하고 있는 옷들을 촬영해서 어플리케이션 내의 가상 옷장에 추가하여 보다 편리하게 옷을 관리할 수 있다.

나. 목적

- 현대 소비자들은 자신의 개성과 라이프스타일을 반영한 맞춤형 패션 추천을 선호한다. 이 프로젝트는 AI 기반 맞춤형 코디 추천을 통해 이러한 수요를 충족하고, 실생활에 적용 가능한 스타일링을 제공한다.
- 사용자가 보유한 옷을 체계적으로 관리하고, 코디 기록과 리뷰를 저장해 스타일을 분석할 수 있도록 지원한다. 이를 통해 사용자들은 효율적인 옷 관리와 개인 스타일 최적화를 경험할 수 있다.

2. 관련 연구

가. 동일, 유사 서비스에 대한 국내 사례

1. 비교 대상 서비스

현재 코디 추천기능을 서비스 중인 어플리케이션은 Acloset, 핏프티, onthelook, outfics, 옷뚝 5개 업체로, 그중 프로젝트의 최종 목표인 “AI 코디추천”와 근접한 다음 4개 서비스를 비교한다.

- Acloset ((주)룩코(한국), 2020.06~)
- 핏프티 (, 2015.04.21.~)
- onthelook ((주)식회사 온더룩(한국), 2019.7.~)
- outfics (Onout.Inc(한국), 2024.12.~)
- 옷뚝 (SAMATTI(한국), 2019.10.~)

2. 각 서비스별 기능 비교

항목	Acloset	핏프티	온더룩	outfics	옷뚝
옷 등록 방식	사진촬영 시, 자동 배경 제거 및 상세정보 입력	옷 등록 X (코디만 추천)	옷 등록 X (코디만 추천)	옷 등록 X (코디만 추천)	사진촬영 시, 자동 상세정보 입력
날씨 추천	O	O	-	-	-
TPO 추천	-	-	-	-	-
스타일 추천 (캐주얼 등)	-	O 스타일 별로 확인가능	O 선택한 코디를 바탕으로 스타일 추천	-	-
옷 색상 분류	O	-	-	O 코디별 대표 색상분류	-
캘린더 코디 추가	O	-	-	-	O
코디 노출 방식	옷 사진만 노출	모델 사진 노출	모델 사진 노출	모델 사진 노출	
옷장 기능	O 소유한 옷의 통계 확인 가능	-	-	-	O
커뮤니티 기능	-	-	코디 '좋아요' 기능 코디에 댓글 작성기능	코디 '좋아요' 기능	코디 '좋아요' 기능. '트렌드 톱' 기능으로 옷 관련 소통 가능. 타인과 옷, 코디 공유.

2. 관련 연구 (계속)

나. 서비스 개발 모델

이 프로젝트는 구현해야 하는 기능 구현을 단계별로 나누고 우선순위를 설정한 후, 우선한 단계가 끝날 시에 다음 단계를 진행하는 방식으로 진행하며, “폭포수 모델^[1]”을 참조한다. 다만, 한정된 시간으로 인해 요구 분석 및 설계는 프로젝트 시작 시에 결정하며, 프로젝트 진행 중에는 해당 결정사항을 최소한으로 수정하는 것을 목표로 한다. 또한, 역시 한정된 시간과 자원으로 인해, 한 단계 완료 후 시행하는 테스트 등의 QA는 다음 단계의 설계와 동시에 이루어지도록 일정을 수립한다.

2. 관련 연구 (계속)

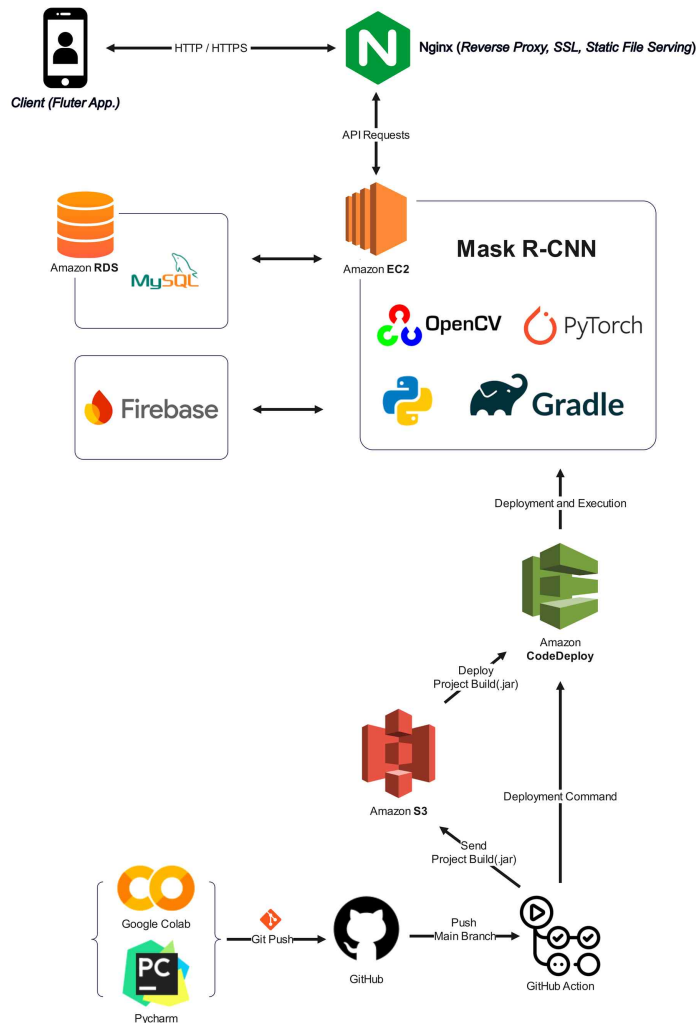
다. 서비스 구축에 사용 예정인 기술 연구

1. 백엔드

- 클라우드 인프라 구성

빠르고 쉽게 확장 가능한 백엔드 인프라 구성을 위하여 이 프로젝트는 Amazon Web Service를 사용할 예정이다. AWS 클라우드는 여러 서비스 제품을 제공하는데, 그중 아래의 서비스를 사용한다.

서비스 제공 업체	서비스 명	설명
Amazon Web Service	EC2	서버를 구동하기 위한 컴퓨팅 리소스 인스턴스
	S3	사용자의 동기화용 DB 및 사용자 업로드 파일 저장소
	RDS	사용자 계정 및 명함 등을 저장하는 데이터베이스
Google Firebase	Cloud Messaging	사용자의 기기에 알림을 푸시할 수 있는 서비스



다. 서비스 구축에 사용 예정인 기술 연구 (계속)

1. 백엔드 (계속)

- AWS EC2 내 서버 구성

AWS의 EC2 인스턴스 내부에서 실행될 서버는 Python으로 작성 중이며, 다음과 같은 프레임워크와 라이브러리, 프로그램이 사용되고 있다.

가. PyTorch^[2]

실시간 이미지 프로세싱 및 딥러닝 프레임워크로, Python을 비롯해 C++, Java 등 다양한 언어에서 사용할 수 있도록 바인딩을 지원하며, Windows, Linux, macOS 등 주요 운영체제뿐 아니라 모바일 플랫폼인 iOS 및 Android에서도 활용 가능하다. 최근에는 객체 탐지, 얼굴 인식, 딥페이크 생성 등 다양한 인공지능 기반 이미지 처리 분야에서 활발히 사용되고 있으며, 본 프로젝트에서는 Mask R-CNN 모델을 활용하여 의류 정보를 구분하고 영역을 검출하는 데 사용된다.^[5]

나. OpenCV

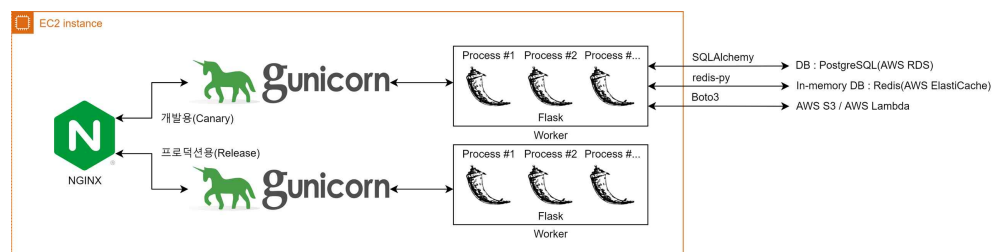
실시간 이미지 프로세싱 라이브러리로, Python, C, Java 등 다양한 언어로 바인딩을 제공하고, Windows/Linux/mac OS/iOS/Android 등의 크로스 플랫폼을 지원한다. 최근에는 딥러닝을 이용한 얼굴인식 및 딥페이크 등 다양한 영역에 사용되고 있으며, 본 프로젝트에서는 사진에서 명함을 구분하기 위해 사용한다.

다. Gradle^[4]

Gradle은 Java, Kotlin, Groovy 등의 언어로 작성된 프로젝트를 자동으로 빌드하고 관리할 수 있는 오픈소스 빌드 자동화 도구로, Android Studio의 기본 빌드 시스템으로도 사용된다. 선언적 빌드 스크립트와 유연한 플러그인 구조를 갖추고 있으며, 프로젝트 의존성 관리, 테스트 실행, 패키징 및 배포 자동화에 강력한 기능을 제공한다. 본 프로젝트에서는 GitHub Actions와 연동되어 코드 푸시시 자동으로 .jar 파일을 빌드하고, 이를 AWS CodeDeploy를 통해 배포하는 데 사용된다.

라. 프론트 웹 서버: NGINX

NGINX는 2002년 이고르 시쇼브(Игорь Сысоев)가 개발하여 2004년 오픈소스로 공개하고, 현재까지 커뮤니티에서 개발 중인 웹 서버로^[5], 위의 HTTP 서버를 리버스 프록시로 연동하고, 정적인 파일을 담당하거나 SSL 인증서 연동을 통해 HTTPS를 지원한다. (Certbot을 통해 Let's Encrypt에서 bca.mudev.cc 도메인의 SSL 인증서를 발급한다) 또한, 개발용 서버와 프로덕션용 서버를 같은 EC2 인스턴스에 실행시킨 후, NGINX로 각 라우트에 매핑한다.



다. 서비스 구축에 사용 예정인 기술 연구 (계속)

2. 프론트엔드

클라이언트 앱 클라이언트는 Flutter 기반의 크로스플랫폼 앱으로, HTTP/HTTPS를 통해 NGINX를 경유하여 서버 API와 통신한다.

- 플랫폼: Flutter (Android/iOS 지원)
- 언어: Dart
- 라이브러리 : HTTP 통신을 위한 http 패키지

Firestore 연동을 위한 firebase_core, firebase_messaging 등
OpenCV와 연동 가능하도록 플랫폼별 네이티브 연동 구현 고려

라. 프로젝트 필요성에 관한 설문조사

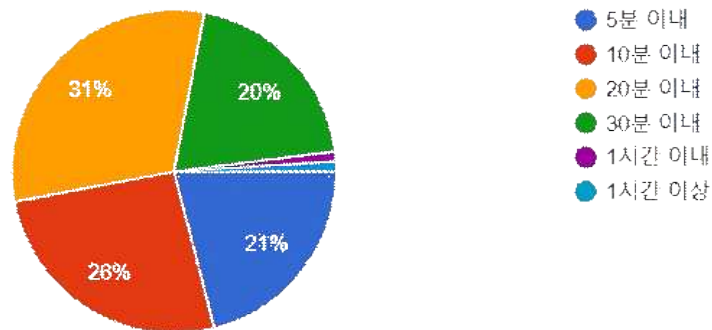
1. 개요:

프로젝트의 필요성을 파악하고자 주요 사용자층인 10대와 20대 100명을 대상으로 구글폼 설문조사를 진행하였다. 응답자는 10대 5명 20대 95명이 답변을 해주었고, 성별의 분포는 여성 61명 남성 39명이었다.

2. 설문 결과:

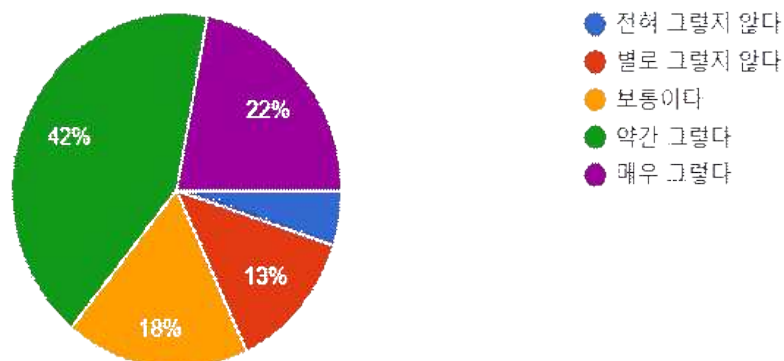
1. 외출 전 옷 고르는 시간은 얼마인가?

- 평균 약 18분이 소요되는 것으로 나타나, 사용자들이 의류 선택 과정에서 상당한 시간을 소비하고 있음을 확인했다.



2. 소유한 옷 중 자주 입지 않는 옷이 많은 편인가?

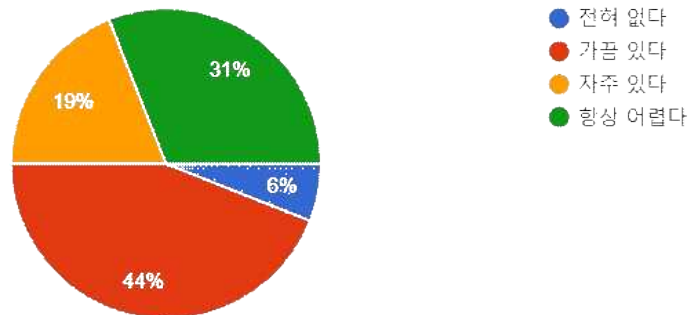
- 평균 점수가 5점 만점에 3.7점으로, 상당수의 사용자가 자주 입지 않는 옷이 대부분 있음을 알 수 있었다.



라. 프로젝트 필요성에 관한 설문조사 (계속)

3. 특정 옷을 마지막으로 입은 시점을 기억 못하는 경우가 있는가?

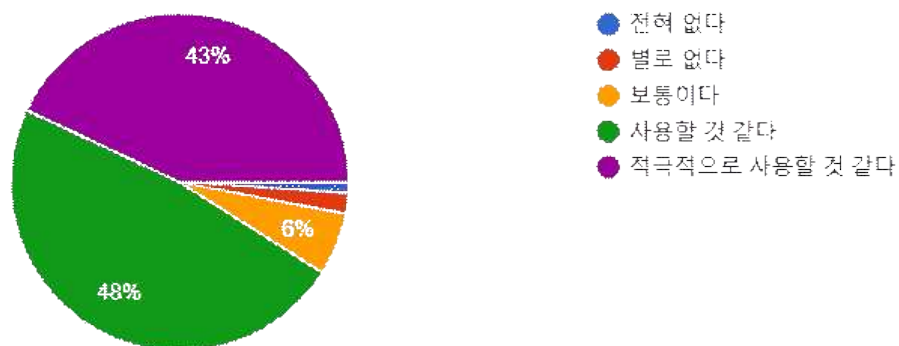
- '가끔 있다'(43%)가 1위, '자주 있다'(26%)가 2위로, 사용자들이 옷 관리



및 착용 기록에 어려움을 겪고 있음을 확인하였다.

4. 옷 코디하는 것이 어렵다고 느낀 적이 있는가?

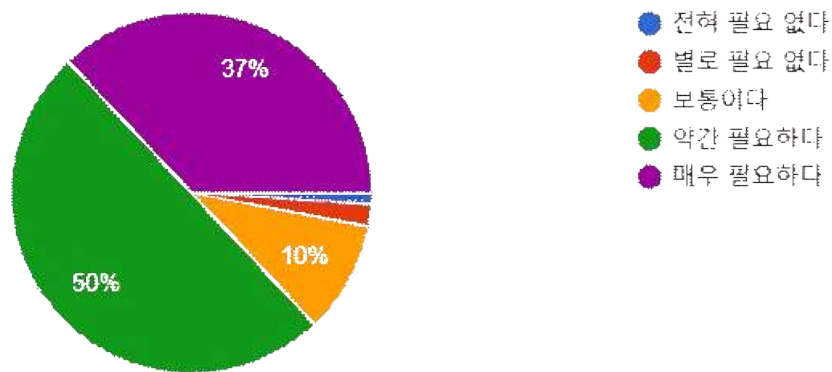
- '가끔 어렵다'(41%)가 1위, '항상 어렵다'(31%)가 2위로, 코디 과정을 어려워하는 사용자들이 많음을 확인했다.



라. 프로젝트 필요성에 관한 설문조사 (계속)

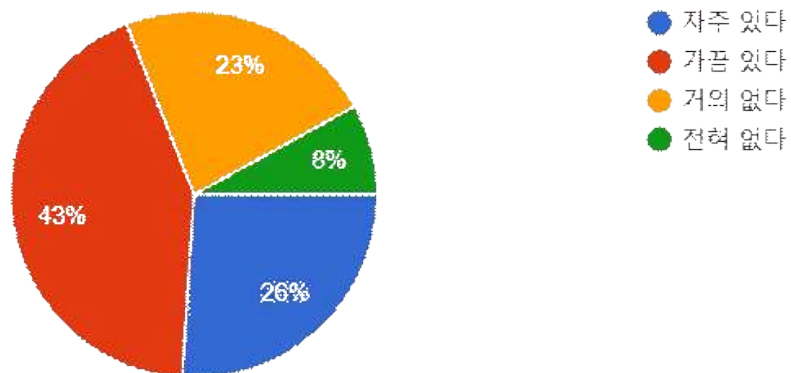
5. AI가 잘 활용되지 않는 옷을 추천해준다면 사용할 의향이 있는가?

- '약간 필요하다'(50%)가 1위, '매우 필요하다'(37%)가 2위로, 잘 활용되지 않는 옷을 추천해주는 기능에 대한 사용자들의 수요가 있음을 확인할 수 있었다.



6. AI가 날씨와 개인 취향을 반영해 코디를 추천해준다면 사용할 의향이 있는가?

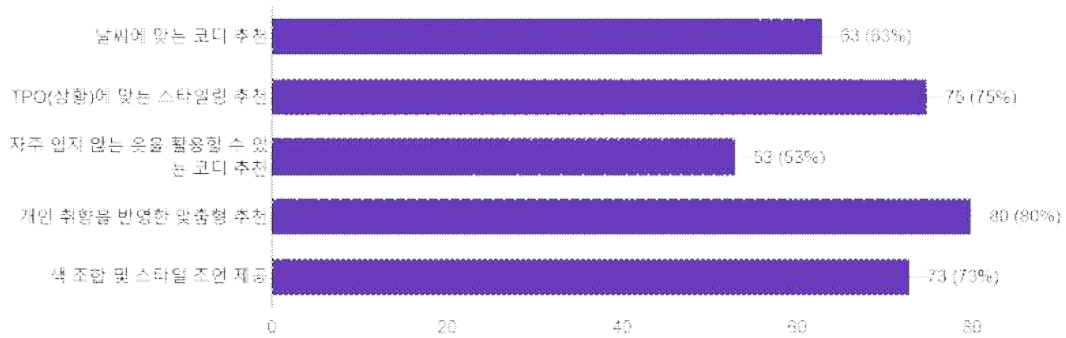
- '사용할 것 같다'(48%)가 1위, '적극적으로 사용할 것 같다'(43%)가 2위로, 맞춤형 코디 추천 기능에 대한 사용자들의 높은 기대감과 긍정적인 사용 의향을 확인할 수 있었다.



라. 프로젝트 필요성에 관한 설문조사 (계속)

7. 코디 추천에 가장 필요하다고 생각하는 기능이 무엇인가?

- '개인 취향을 반영한 맞춤형 추천'과 'TPO(상황)에 맞는 스타일링 추천'이 가장 필요한 기능으로 확인되었다.



3. 결론:

설문 조사를 통해 사용자들이 일상 속에서 의류 선택과 코디에 있어 다양한 불편함을 겪고 있음을 확인할 수 있었다. 많은 응답자가 옷을 고르는 데 평균 18분 이상을 소비하고 있으며, 자주 입지 않는 옷이 많고, 마지막 착용 시점을 기억하지 못하는 경우도 잦은 것으로 나타났다. 또한, 옷을 코디하는 데 어려움을 느끼는 사용자들도 많아, AI 기반 맞춤형 코디 추천 서비스, 특히 개인 취향 및 상황(TPO)을 반영한 기능에 대한 수요가 높다는 점을 파악할 수 있었다.

설문 결과는 사용자의 실질적인 요구와 불편함을 해결할 수 있는 어플리케이션의 필요성을 보여준다. 사용자들의 수요를 중심으로 한 서비스 개발은 사용자 만족도를 높일 것으로 기대된다.

3. 주요기능 및 기대효과

가. 주요기능

- 사용자가 촬영한 의류 이미지를 분석해 카테고리(상의, 하의 등)로 분류 및 저장
- 날씨, 취향, TPO(Time, Place, Occasion) 등에 맞춘 맞춤형 코디 제안
- 특정 날짜에 입은 코디를 저장하고, 캘린더 형태로 착용 이력 관리
- 착용한 코디에 대한 리뷰 및 만족도를 기록하여, 추후 스타일 개선에 활용
- Weather API를 활용해 지역별 날씨에 따른 코디 추천
- 사용자의 착용 데이터와 선호도를 기반으로 개인화된 스타일 추천
- 사용자가 상황(출근, 데이트, 여행 등)을 선택하면 해당 TPO에 맞춘 코디 추천

나. 비기능 요구 사항

- 회원 가입 및 로그인 후에만 주요 기능을 사용할 수 있도록 회원제 기반으로 운영
- 사용자 수가 증가해도 추천 속도와 데이터베이스 응답 시간이 크게 저하되지 않아야 함
- 외부 API가 실패해도 기본적인 코디 추천 기능 동작
- 사용자 맞춤 튜토리얼 또는 온보딩 기능을 제공
- 앱은 직관적이고 쉬운 UI/UX를 제공하여, 첫 이용자도 5분 이내에 가상 옷장을 만들 수 있어야 함

다. 기대효과

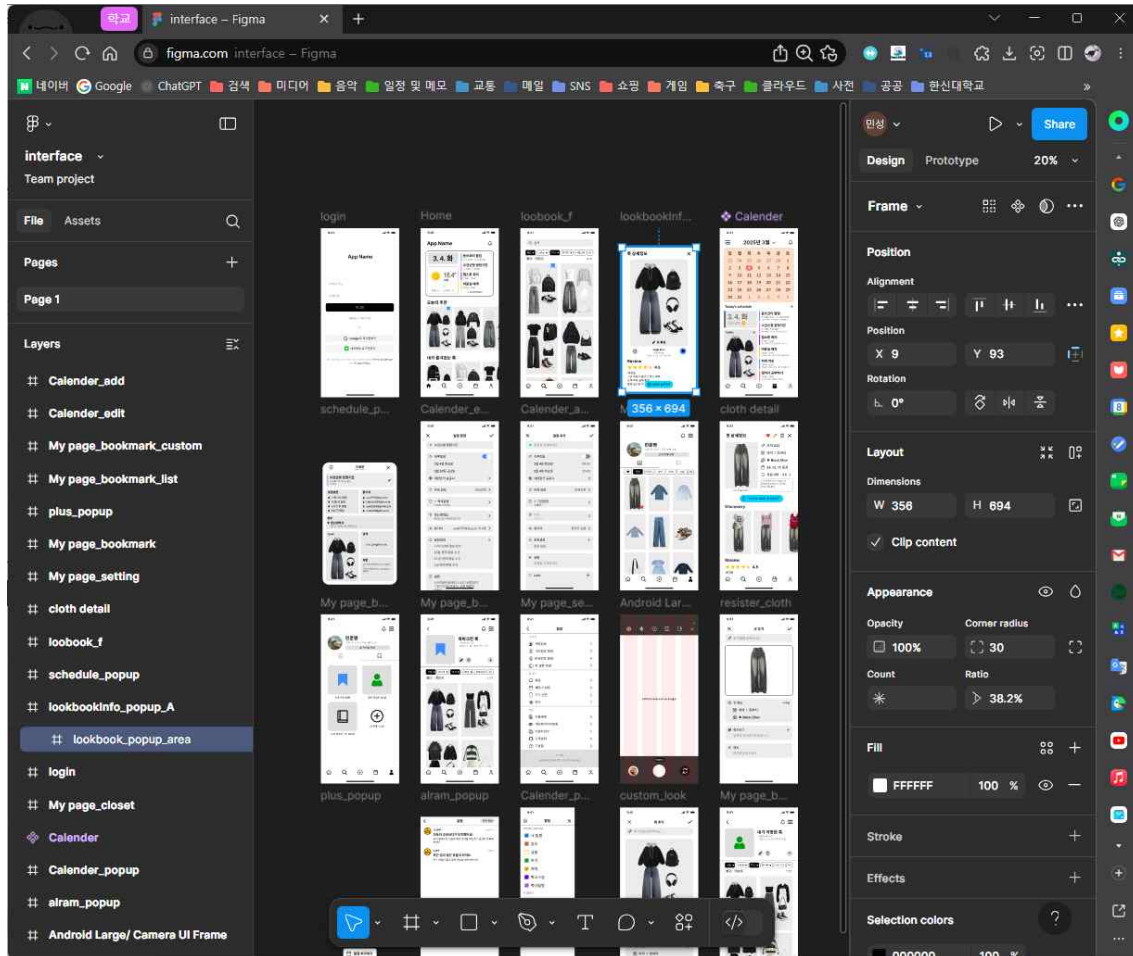
- 외출 전 코디 고민을 줄여 준비시간 절약
- 코디 추천 서비스로 발견하지 못한 여러 가지 새로운 룩 발견 가능
- 스마트폰에서 어떤 옷이 있는지 한눈에 볼 수 있어, 효율적인 옷 관리 및 경제적인 옷 구매 가능

라. GUI 프로토타입

- UI/UX 디자인 툴(Figma)을 사용한 프로토타입 제작

3. 주요기능 및 기대효과 (계속)

라. GUI 프로토타입 (계속)



<Figma를 통한 프로토타입 UI/UX 제작 및 디자인 중인 화면>

4. 개발 환경

- 빠른 클라우드 인프라 구축을 위해 AWS 사용
(EC2, S3, RDB, Codedeploy 등)
- 사용자, 의류 등 정보를 저장하기 위한 데이터베이스는 MySQL를 사용
- 클라이언트와 통신에서 보안성, 유연성 향상을 위해 Firebase와 백엔드 서버를 통합 운영
- 클라이언트가 요청한 데이터를 백엔드 서버에서 Python을 기반으로 OpenCV, Pytorch 등 여러 딥러닝 라이브러리를 활용해서 의류 데이터 분석 후 전송
- 백엔드와 클라이언트 통신에서 Static 파일 전송 및 라우트 관리 등의 역할로 Nginx를 사용
- 백엔드는 클라이언트와 HTTP/HTTPS 프로토콜을 이용하여 통신한다.
- 어플리케이션은 Flutter와 Dart를 사용하여 개발 예정
- 커뮤니케이션 도구로는 카카오톡, Google Meet, 아이디어 회의 시 문서 공유는 Notion, Google Drive를 사용
- 오류나 기능 추가 등의 이슈 관리, 소스코드 버전 관리는 Github를, 앱 디자인 도구로는 Figma를 사용

5. 위험 요소

1. 데이터 수집의 어려움

이 프로젝트의 성공을 위해서는 충분하고 다양한 사용자 옷 데이터 확보가 필수적임. 만약 데이터의 양이 부족하거나 다양성이 떨어진다면, AI 추천 시스템의 정확도가 크게 저하됨. 특히 계절별/상황별(TPO)에 맞는 세분화된 데이터가 부족할 경우, 특정 사용자층이나 상황에 대한 추천 품질이 떨어지는 문제가 발생할 수 있음.

2. 추천 정확도 부족

AI 추천 시스템이 사용자의 개인적인 취향, TPO, 현재 날씨 등의 다양한 요소를 정교하게 파악하여 추천에 반영하지 못하면, 사용자는 추천 결과에 대한 신뢰를 잃게 될 수 있음. 만약 AI가 반복적으로 사용자의 기대와 동 떨어진 코드를 추천한다면, 사용자는 서비스에 실망하고 결국 앱 사용을 중단할 가능성이 높아짐.

5. 위험 요소 (계속)

3. 기술적 복잡도

사용자에게 맞춤형 코드를 추천하기 위해서는 날씨 정보를 가져오는 날씨 API, 사용자의 일정을 파악하는 일정 API, 사용자 옷 이미지를 분석하는 이미지 분석 모델, 그리고 이 모든 정보를 바탕으로 최적의 코드를 제안하는 추천 시스템 등 여러 기술 요소가 통합되어야 함. 이러한 복잡한 시스템 구성 요소 중 단 하나라도 제대로 작동하지 않거나 오류가 발생한다면, 전체 서비스의 사용자 경험(UX)에 심각한 부정적인 영향을 미칠 수 있음.

5. 개인정보 및 보안 이슈

이 서비스는 사용자의 옷 사진뿐만 아니라, 일정 정보, 현재 위치 정보 등 민감한 개인 데이터를 다룸. 따라서 수집 및 활용되는 데이터에 대한 강력한 보안 대책 마련이 필수적이며, 사용자 개인정보 보호 및 프라이버시 침해 가능성에 대한 문제에 매우 민감하게 대응해야 함. 데이터 유출이나 오용 발생 시 서비스 신뢰도에 치명적인 손상을 입을 수 있음.

6. 유지보수 및 확장성

서비스 초기에는 사용자 수가 적어 시스템 운영에 큰 문제가 없을 수도 있지만 사용자가 점차 늘어나고 데이터 양이 방대해짐에 따라, 서버 트래픽 증가, 데이터 관리 부담 증가 등 시스템 부하가 커지게 됨. 만약 서비스 설계 단계부터 이러한 사용자 증가에 대비한 시스템 유지보수 및 확장성에 대한 고려가 부족했다면, 서비스 품질 저하를 야기하거나 안정적인 운영에 어려움을 겪게 될 수 있음.

6. 일정 계획

- 주차 별 일정 계획

구분	추진내용	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
계획	프로젝트 목표 및 기능 정의										
분석	사용자 요구사항, 필요한 데이터와 API 조사										
설계	개념도 및 아키텍처 다이어그램										
설계	어플리케이션 흐름도 설계										
설계	UI/UX 디자인 설계										
설계	데이터베이스 스키마 설계										
개발	이미지 의류 추출 및 대표 색상 분석 기능 개발										
개발	의류 카테고리 자동 분류 기능 개발										
개발	최적 의류 색상 조합 분석 기능 개발										
개발	AI 모델 학습 및 구현										
개발	데이터 베이스 구축 및 연동										
개발	일정 카테고리(TPO) 자동 분석 기능 개발										
개발	상황(날씨, 피부톤, TPO, 등)에 따른 코디 추천 기능 개발										
개발	개발한 기능 기반 Flutter App 개발										
테스트	기능별 유닛 테스트 진행										
테스트	앱 통합 테스트 및 버그 수정										
종료	어플리케이션 완성 및 최종 보고서 작성										
오프라인 미팅계획	매주 화요일 대면 회의 진행										

7. 참고 문헌

- [1] 이성구, "소프트웨어 공학 기본 원리", 홍릉과학출판사, 2020.
- [2] [Paszke A., 외 20명, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", NeurIPS, 2019.](#)
- [3] [He K., Gkioxari G., Dollár P., Girshick R., "Mask R-CNN", Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961-2969.](#)
- [4] Tim Berglund, "Building and Testing with Gradle", O'Reilly Media, 2011.
- [5] [About NGINX, NGINX team, 2020](#)