

날씨와 연동한 의상 추천 애플리케이션



부산대학교 정보컴퓨터공학부

중간보고서

지도교수: 유영환

부과명: 네트워크/통신

팀 이름: ByteStorm

호노카 - 202255574

마디나 - 202155622

신상협 - 201424478

목차

1. 과제 목표.....	3
2. 요구조건 및 제약 사항 분석에 대한 수정사항.....	3
2.1 기존 요구조건.....	3
2.2 요구조건 수정사항	3
3. 설계 상세화 및 변경 내역.....	4
3.1 전체 구성도	4
3.2 Front-End 설계	5
3.3 Back-End 설계	5
3.4 AI 설계	5
4. 갱신된 과제 추진 계획.....	6
5. 구성원별 진척도.....	7
호노카 - 프론트엔드, UX 설계, 백엔드 일부 기능	7
마디나 - 백엔드, 데이터베이스	7
신상협 - AI / 데이터 엔지니어링/ 백엔드	8
6. 보고 시점까지의 과제 수행 내용 및 중간 결과	9

1. 과제 목표

본 프로젝트의 궁극적인 목표는 사용자의 날씨, 일정, 보유 의류 데이터를 종합적으로 반영하여 개인 맞춤형 패션 코디네이션을 추천해주는 애플리케이션을 개발하는 것이다.

기존의 날씨 기반 의상 추천 서비스는 대체로 기온에 따른 일률적인 추천을 제공하는 반면, 본 프로젝트는 AI 추천 엔진을 기반으로 사용자 일정(예: 회의, 데이트 등)과 옷장에 등록된 실제 소유 의류를 활용하여, 현실적이고 실용적인 아웃핏 조합을 제안한다.

2. 요구조건 및 제약 사항 분석에 대한 수정사항

2.1 기존 요구조건

본 프로젝트의 초기 요구조건은 다음과 같은 기술 스택 구성을 기반으로 계획:

- 프론트엔드: Flutter 또는 React Native
- 백엔드: Django 또는 Node.js
- 데이터베이스: Firebase Realtime DB + PostgreSQL
- AI 추천 엔진: OpenCV, MediaPipe, BERT4Rec, XGBoost, GPT-4 또는 LLaMA

2.2 요구조건 수정사항

착수 이후 멘토 자문(2025 년 6 월 7 일) 내용을 반영하여 다음과 같은 방향으로 수정 및 확정하였다:

① 프론트엔드 프레임워크 확정

- 최종적으로 Flutter 로 통일.

② 백엔드 기술 명확화

- Django 와 Node.js 병기를 제거하고 Django 단독으로 개발 진행 중.

③ 데이터베이스 및 캐시 구조 정비

- Firebase + PostgreSQL 병용 대신 PostgreSQL 중심 구조로 통일.
- Firebase 는 사용자 인증 기능만 유지하고, 나머지 데이터는 PostgreSQL 에 저장.
- Redis 기반 캐시 도입은 후반부 시스템 부하 분석을 통해 필요 여부 판단.

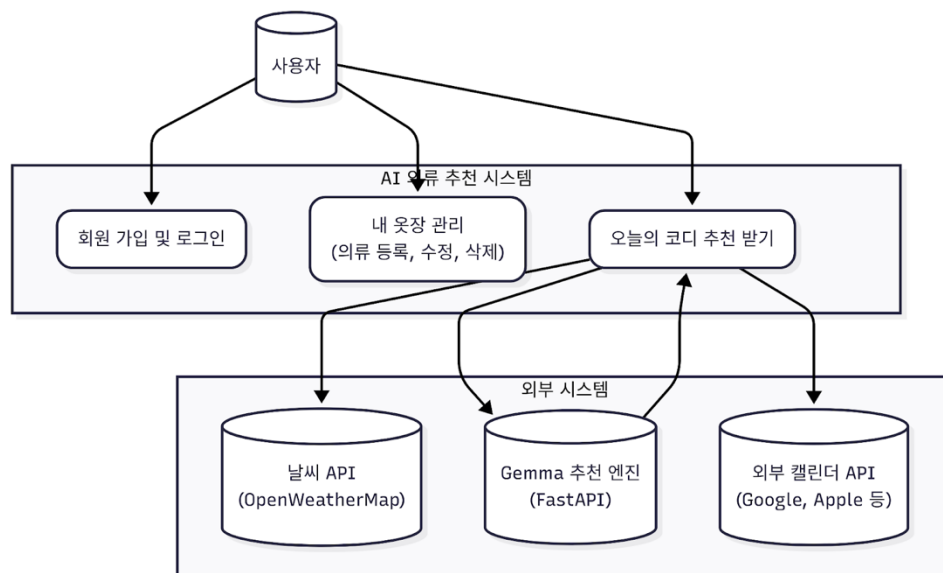
④ AI 추천 엔진 구조: Fine-tuned LLM 기반으로의 전환 및 고도화

- 아키텍처 전환: 기존의 전통적인 추천 시스템 구조에서 탈피하여, 최신 소형 언어 모델(LLM)인 Gemma 를 기반으로 한 추천 엔진 구조로 성공적으로 전환했습니다.
- 효율적 미세 조정: 이 과정에서 **PEFT(Parameter-Efficient Fine-Tuning)**와 LoRA(Low-Rank Adaptation) 기법을 도입하여, 제한된 자원 내에서도 효율적인 모델 Fine-tuning 을 완료했습니다.

3. 설계 상세화 및 변경 내역

3.1 전체 구성도

사용자, 프론트엔드, 백엔드, AI 추천 엔진, 날씨 API, 외부 캘린더 API 를 포함한 전체 시스템 아키텍처를 재설계하였다. 각 구성 요소 간의 데이터 흐름(예: 사용자 요청 → 날씨 조회 → 일정 조회 → 모델 연동 → 결과 반환)을 시각화하였다.



3.2 Front-End 설계

사용 기술: Flutter (Dart)

주요 설계 내용:

- 로그인/회원가입, 옷 등록, 추천 결과 화면을 Flutter 로 구현
- 서버(REST API)와 통신하는 HTTP 요청 모듈 설계
- 옷 이미지와 추천 결과를 카드 형식으로 출력
- 아침 푸시 알림 기능 등 UI 도 추가할 예정

3.3 Back-End 설계

사용 기술: Django, PostgreSQL, Firebase 인증

주요 설계 내용:

- 회원가입/로그인, 옷 CRUD, 추천 요청 처리용 REST API 설계
- FastAPI (serve.py)와 연동하여 추천 결과를 수신하고 반환
- 추천 결과를 JSON 으로 반환하고 DB 에 저장

3.4 AI 설계

- 추천 모델 구조: Fine-tuned Gemma (LLM) 단일 아키텍처
 1. 대신, 하나의 Fine-tuned 된 Gemma 모델이 상황(날씨, 일정)의 문맥적 이해, 각 의류 아이템의 속성 파악, 그리고 이 둘 간의 적합성(Compatibility) 점수 계산까지 모든 과정을 종단간(End-to-End)으로 처리하는 통합 아키텍처를 채택했습니다.
- 추천 흐름:
 1. 입력: 사용자의 옷장 정보(ID 목록)와 날씨/일정 정보를 입력으로 받습니다.
 2. 문맥적 프롬프트 생성: 각 옷 ID 를 styles.csv 와 매핑하여 '색상', '종류', '계절' 등 풍부한 속성을 포함한 텍스트로 변환합니다. 이 정보를 날씨/일정과

결합하여, 가능한 모든 옷 조합에 대해 자연어 기반의 상세한 질문 프롬프트를 동적으로 생성합니다.

3. LLM 기반 추론 및 점수 계산: 생성된 각 프롬프트를 Gemma 모델에 입력하여, 주어진 상황에 해당 옷 조합이 얼마나 '적절한지'에 대한 확률 점수를 추론합니다.
4. 최적 조합 추천: 계산된 모든 조합의 점수 중, 가장 높은 점수를 획득한 최고의 조합 하나를 최종 결과로 반환합니다. 추천 사유는 모델이 특정 조합에 부여한 높은 점수 자체에 내재되어 있습니다.

4. 갱신된 과제 추진 계획

6월					7월					8월					9월					10월	
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5		
기획 및 요구사항 정리, 아키텍처 설계																				졸업과제 발표회	
				UI 프로토타입 작성, 데이터 구조 구상																	
			Flutter 구현																		
			AI 모델 구현, 백엔드 API 설계 및 연동 테스트																		
					프론트-백 통합 및 추천 시스템 완성, 디버깅																
										베타 테스트, 피드백 반영 및 UI 개선											
															발표 자료 준비 및 최종 시연						

5.구성원별 진척도

호노카 - 프론트엔드, UX 설계, 백엔드 일부 기능

프론트엔드

- AdobeXD 를 사용하여 UX 프로토타입 제작
- UI sets : <https://xd.adobe.com/view/8f06e601-4ef4-405b-80ec-905b5feca5bf-da55/>
- 초기 UX 에 대해 팀원 피드백을 반영하여 일부 UI 를 수정
- UX 설계와 함께 상세한 사양서(기능 명세서)도 작성
- Flutter 를 활용한 화면 설계는 현재 진행 중

백엔드 (일부 기능)

- Django 초기 프로젝트 생성 및 기본 설정 완료
- 모델을 설계(ClothingItem, Schedule)

마디나 - 백엔드, 데이터베이스

백엔드 주요 기능 구현 진행 중

- Django 프로젝트 내부에서 FastAPI 를 활용하여 AI 추천 엔진과 통신할 수 있는 엔드포인트(recommend_api.py) 구현
- OpenWeatherMap API 연동 테스트 및 mock 데이터 기반 통신 기능 구현 완료
- 일정/날씨 기반 추천 로직 연동 준비

데이터베이스 모델 설계 및 연동

- ClothingItem, Schedule, RecommendationLog 모델 정의 및 마이그레이션 완료
- 프론트엔드에서 전달되는 옷 정보를 ClothingItem 모델을 통해 데이터베이스에 저장할 수 있도록 API 연결 (DRF 기반 ListCreateAPIView 구현)

AI 추천 관련 기능 구현

- 사용자의 일정 및 날씨 정보를 AI 모델에 전달하는 로직 FastAPI 에서 테스트 완료
- 추천 결과로 반환된 코디 정보(JSON)를 Django 에서 저장 및 출력할 수 있도록 RecommendationLog 연동 구현 중

추가 예정 작업

- 로그인 및 프로필 관련 기능 구현은 Firebase 또는 Django Auth 방식 중 선택하여 적용 예정
- 실제 Google/Apple 캘린더 API 와 연동하지 못하는 경우를 고려하여 mock 데이터를 활용한 로직 우선 구현

신상협 - AI / 데이터 엔지니어링/ 백엔드

- 데이터셋(styles.csv, images) 초기 분석 및 제약 사항 파악.
- 새로운 학습 데이터셋 생성 스크립트(generate_dataset.py) 개발 및 지속적인 개선 주도.
- 데이터셋 품질 및 효율성 문제(0 개 샘플, 라벨 불균형, 논리적 오류) 해결에 핵심 기여.
- 모델 학습 루프(노트북) 구현 및 디버깅 (차원 불일치, not defined 오류 해결).
- 모델 학습 진행 및 가중치 저장 형식 (.pth 파일 생성) 통일.
- FastAPI 서버(serve.py) 기본 구조 구축.
- 모델 클래스 정의 및 헬퍼 함수(load_item_tensors) 통합.
- FastAPI recommend_outfit 엔드포인트 설계 및 구현.
- 클라이언트로부터의 요청 처리 및 모델 추론 결과 반환 로직 구현.
- FastAPI 서버 측 디버그 출력(분류 상세, 조합 진행 상황) 강화.
- 신규 모델 아키텍처 설계 및 구현:
- 기존 모델의 일반화 성능 한계(학습 데이터에 없는 조합 평가 불가)를 명확히 분석.
- 문제 해결을 위해 LLM(Large Language Model) 기반의 Fine-tuning 이라는 새로운 접근법을 제시하고 Gemma 모델을 최종 선정.
- 정형 데이터(JSON, CSV)를 Gemma 모델 학습에 적합한 자연어 지시(Instruction) 텍스트 형식으로 변환하는 전처리 파이프라인 전체를 설계 및 개발.
- styles.csv 의 의류 메타데이터를 활용하여, 단순 ID 가 아닌 '색상', '종류' 등의 풍부한 속성으로 학습 데이터의 품질을 극대화.
- LLM Fine-tuning 파이프라인 구축 및 최적화:

- Hugging Face transformers 라이브러리를 기반으로 Gemma 모델의 Fine-tuning 전체 코드를 구현.
- 제한된 컴퓨팅 자원(Colab GPU)에서 대규모 언어 모델을 효율적으로 학습시키기 위해 PEFT(Parameter-Efficient Fine-Tuning) 및 LoRA(Low-Rank Adaptation) 기법을 도입하고 성공적으로 적용.
- 학습 과정에서 발생한 핵심 오류(Gated Repo 인증, Quantized 모델 학습 불가, Loss 미반환, 라이브러리 의존성 충돌 등)를 체계적으로 디버깅하고 해결.
- 현실적인 시간 내에 학습이 가능하도록 데이터셋 규모, 배치 사이즈, 에포크 등 주요 하이퍼파라미터를 최적화하여 학습 시간을 수십 시간에서 수십 분 단위로 단축.

6. 보고 시점까지의 과제 수행 내용 및 중간 결과

프론트엔드:

UI sets : <https://xd.adobe.com/view/8f06e601-4ef4-405b-80ec-905b5feca5bf-da55/>

UI 및 설계를 바탕으로 Flutter 로 구현 중

AI:서버 (serve.py)의 fast_api.py 의 요청에 따른 옷장 분류

작업

```
--- 새로운 추천 요청 접수 ---
상황 : Business Presentation, 20.0°C, Indoors, Men
옷장 아이템 수 : 12개

--- 옷장 분류 결과 ---
> Topwear: 4개 | IDs: [15970, 9452, 13089, 4959]
> Bottomwear: 2개 | IDs: [39386, 10258]
> Outerwear: 1개 | IDs: [18569]
> Footwear: 4개 | IDs: [59435, 9036, 9204, 6806]
```

1 옷 조합들에 대한 평가 및 옷들

```
분석할 총 조합 수 : 64개 (기본 : 32개, 아우터 포함 : 32개)
> 조합 1/64 | 점수 : 0.0118 | 조합 : 상의 (Navy Blue Shirts), 하의 (Blue Jeans), 신발 (Black Formal Shoes)
> 조합 2/64 | 점수 : 0.0118 | 조합 : 상의 (Navy Blue Shirts), 하의 (Blue Jeans), 신발 (Black Formal Shoes)
> 조합 3/64 | 점수 : 0.0120 | 조합 : 상의 (Navy Blue Shirts), 하의 (Blue Jeans), 신발 (Black Casual Shoes)
> 조합 4/64 | 점수 : 0.0123 | 조합 : 상의 (Navy Blue Shirts), 하의 (Blue Jeans), 신발 (Brown Casual Shoes)
> 조합 5/64 | 점수 : 0.0131 | 조합 : 상의 (Navy Blue Shirts), 하의 (Navy Blue Trousers), 신발 (Black Formal S
hoes)
> 조합 6/64 | 점수 : 0.0131 | 조합 : 상의 (Navy Blue Shirts), 하의 (Navy Blue Trousers), 신발 (Black Formal S
hoes)
> 조합 7/64 | 점수 : 0.0131 | 조합 : 상의 (Navy Blue Shirts), 하의 (Navy Blue Trousers), 신발 (Black Casual S
hoes)
> 조합 8/64 | 점수 : 0.0133 | 조합 : 상의 (Navy Blue Shirts), 하의 (Navy Blue Trousers), 신발 (Brown Casual S
hoes)
> 조합 9/64 | 점수 : 0.0109 | 조합 : 상의 (Red Shirts), 하의 (Blue Jeans), 신발 (Black Formal Shoes)
> 조합 10/64 | 점수 : 0.0109 | 조합 : 상의 (Red Shirts), 하의 (Blue Jeans), 신발 (Black Formal Shoes)
> 조합 11/64 | 점수 : 0.0113 | 조합 : 상의 (Red Shirts), 하의 (Blue Jeans), 신발 (Black Casual Shoes)
> 조합 12/64 | 점수 : 0.0113 | 조합 : 상의 (Red Shirts), 하의 (Blue Jeans), 신발 (Brown Casual Shoes)
> 조합 13/64 | 점수 : 0.0118 | 조합 : 상의 (Red Shirts), 하의 (Navy Blue Trousers), 신발 (Black Formal Shoes)
> 조합 14/64 | 점수 : 0.0118 | 조합 : 상의 (Red Shirts), 하의 (Navy Blue Trousers), 신발 (Black Formal Shoes)
> 조합 15/64 | 점수 : 0.0120 | 조합 : 상의 (Red Shirts), 하의 (Navy Blue Trousers), 신발 (Black Casual Shoes)
> 조합 16/64 | 점수 : 0.0122 | 조합 : 상의 (Red Shirts), 하의 (Navy Blue Trousers), 신발 (Brown Casual Shoes)
> 조합 17/64 | 점수 : 0.0117 | 조합 : 상의 (Grey Sweatshirts), 하의 (Blue Jeans), 신발 (Black Formal Shoes)
> 조합 18/64 | 점수 : 0.0117 | 조합 : 상의 (Grey Sweatshirts), 하의 (Blue Jeans), 신발 (Black Formal Shoes)
> 조합 19/64 | 점수 : 0.0121 | 조합 : 상의 (Grey Sweatshirts), 하의 (Blue Jeans), 신발 (Black Casual Shoes)
> 조합 20/64 | 점수 : 0.0122 | 조합 : 상의 (Grey Sweatshirts), 하의 (Blue Jeans), 신발 (Brown Casual Shoes)
> 조합 21/64 | 점수 : 0.0124 | 조합 : 상의 (Grey Sweatshirts), 하의 (Navy Blue Trousers), 신발 (Black Formal
Shoes)
> 조합 22/64 | 점수 : 0.0124 | 조합 : 상의 (Grey Sweatshirts), 하의 (Navy Blue Trousers), 신발 (Black Formal
Shoes)
> 조합 23/64 | 점수 : 0.0126 | 조합 : 상의 (Grey Sweatshirts), 하의 (Navy Blue Trousers), 신발 (Black Casual
Shoes)
> 조합 24/64 | 점수 : 0.0128 | 조합 : 상의 (Grey Sweatshirts), 하의 (Navy Blue Trousers), 신발 (Brown Casual
Shoes)
```

테스트 시나리오 에 대한 test_api.py 에서의 결과

```
--- 테스트 시나리오 : 시나리오 1: 남성 비즈니스 미팅 ---
상황 : Business Presentation (Indoors, 20.0°C)
옷장 아이템 ID 리스트 : [15970, 9452, 39386, 10258, 59435, 13089, 9036, 9204, 4959, 18569, 50584, 6806]

[AI 추천 결과]
- 최고 조합 : {'description': '상의 (Green Tshirts), 하의 (Navy Blue Trousers), 신발 (Brown Casual Shoes)
', 'ids': [4959, 10258, 6806]}
- 호환성 점수 : 0.0134
- 분석한 총 조합 수 : 64
- 소요 시간 : 231.23초
=====
```