데이터 시각화를 활용한
사용자 감정 분석 플랫폼 구현

중간 보고서

팀명	소요
팀장	성도범
팀원	송시우 성민기 김세엽

목차

1. 서론

1.1 배경

- 현대 SNS 커뮤니티 서비스의 한계(사용자 감정 반영 부족)
- 감정 분석 기반 추천/맞춤형 상호작용의 필요성

1.2 목적 및 범위

- 사용자 감정을 분석하는 기능을 구현한 플랫폼 개발
- GPT API를 활용한 텍스트 감정 분석 모듈과, 이를 활용한 감정 점수 관리
- 중간 단계로 API 연동 상태 보고 및 프론트엔드/백엔드 설계 계획 수립

1.3 팀 소개

- 성도범(조장): 프론트엔드 & 백엔드 개발
- 성민기(조원): 프론트엔드 개발
- 송시우(조원): 인공지능 모델 개발 및 학습
- 김세엽(조원): 백엔드 개발

2. 플랫폼 개요 및 요구사항 분석

2.1 플랫폼 주요 기능

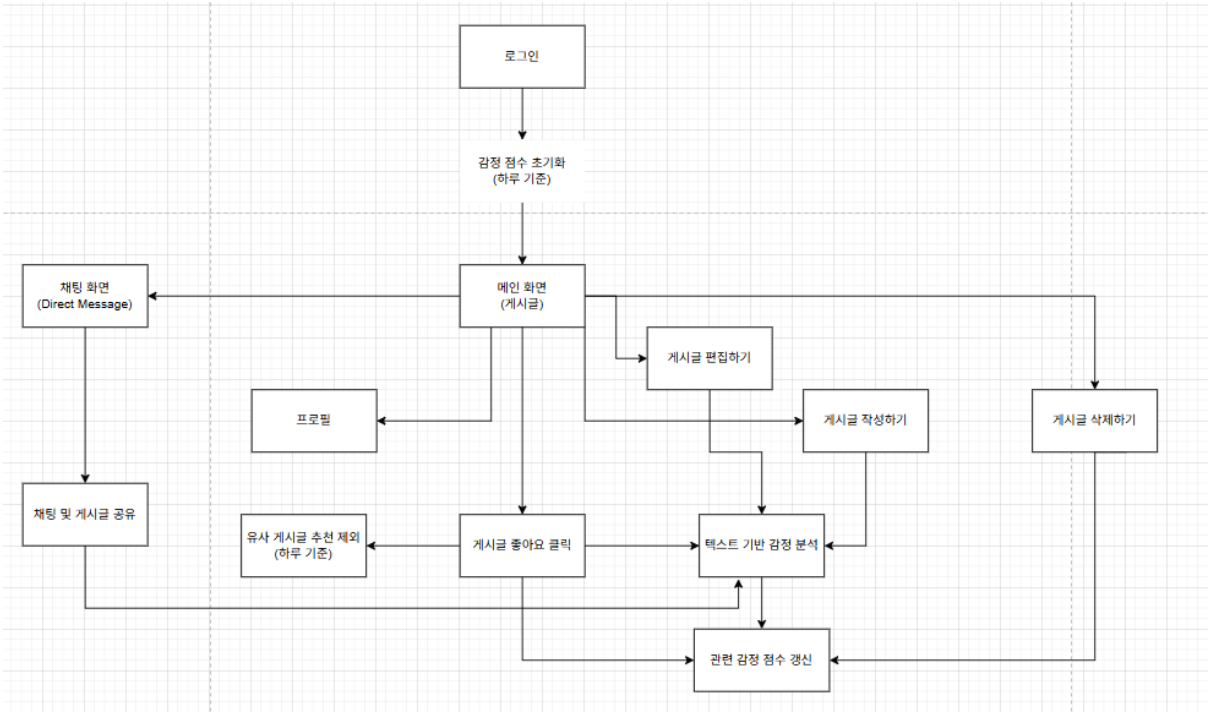
1. 로그인 / 인증 기능
2. 감정 점수 초기화
3. 메인 화면(게시글 피드)
4. 게시글 조회, 작성, 편집, 삭제
5. 게시글 좋아요 → 감정 점수 반영
6. 텍스트 기반 감정 분석 → 감정 점수 갱신
7. 유사 게시글 추천 제외 로직
8. DM(Direct Message) 및 게시글 공유 기능

2.2 요구사항 정리(기능/비기능)

1. 기능 요구사항: 위 8가지 워크플로우
2. 성능 요구사항: 1일 1회 감정 점수 초기화
3. 보안 요구사항: JWT 기반 인증, 중요한 API에 권한(Bearer) 검증

3. 시스템 아키텍처 설계

3.1 전체 아키텍처 다이어그램



3.2 컴포넌트 역할

- 클라이언트(UI): React 기반 SPA(Single Page Application)
- API 서버:
  - Spring MVC를 이용한 URL 매핑 및 컨트롤러 구성
  - Spring Security + JWT로 인증 인가 처리
  - Service 레이어에서 로직 구현
  - Repository 레이어(Spring Data JPA)로 DB 접근
- 감정 분석 엔진: FastAPI, HuggingFace Transformers (모델 로드 및 토크나이저) Tensorflow/Pytorch (필요 시 모델 fine-tunning, 추론)
- DB: Spring Data JPA(Hibernate) 기반 엔티티 매핑

4. 인공지능 모델 연동

4.1 연동 구조

1. 클라이언트

   게시글 작성/편집 완료 시 텍스트 수집하여 HTTP 요청으로 백엔드 서버에 전송

2. 백엔드서버 처리

   Controller에서 입력 검증 → Service 계층으로 전달

   A. 환경변수 OPENAI_API_KEY 로드. HTTP 클라이언트로 OpenAI Chat API 호출. 감정 분석하여 JSON형식으로 감정과 점수를 반환하게 프롬프트로 요청. 응답으로 받은 JSON에서 파싱

   B. 로컬 모델 사용 시 컴포넌트에서 웹 구동 시점에 모델을 로드하여 관리.

3. 모델 응답으로 감정 레이블(기쁨, 놀라움, 두려움, 사랑스러움, 슬픔, 화남, 중립) 및 점수 산출하여 DB에 저장

4.2 모델 선정 기준

현재는 가장 접근성이 좋고 성능이 좋은 GPT API를 이용해 감성 분석을 진행하였지만 API 비용을 고려하여 로컬 모델의 성능이 충분하다면 로컬 모델을 이용해서 감정 분석을 진행할 예정. 다양한 언어 모델을 테스트하며 짧은 응답 속도와 한국어 감정 분류 태스크에 특화된 모델을 고려 중. AI Hub의 감정이 태깅된 데이터셋들을 이용해서 추가 fine tune 작업 수행.

5. API 명세 요약

5.1 인증 관련

1. POST /auth/login → JWT 발급
2. POST /auth/refresh → 액세스 토큰 갱신

5.2 게시글 관련

- GET /posts → 게시글 리스트 반환
- POST /posts → 게시글 작성
- PUT /posts/{id} → 게시글 수정
- DELETE /posts/posts/{id} → 게시글 삭제
- POST /posts/{id}/like → 게시글 좋아요
- POST /posts/{id}/share → 게시글 공유

5.3 채팅 관련

1) GET /chat → 채팅 목록 반환
2) POST /chat/{id}/message → 채팅 전송

5.4 외부 컴포넌트 관련

6. 프론트엔드(UI/UX) 설계

6.1 화면 목록 및 흐름

1. 로그인/회원가입 화면
2. 메인 화면(게시글 목록)
3. 게시글 작성 Modal 화면
4. 게시글 편집 Modal 화면
5. 게시글 삭제 화면
6. 게시글 상세 view
   A. 좋아요 버튼
   B. 감정 아이콘 버튼
7. DM(Direct Message) 화면

6.2 와이어 프레임 예시

1. 로그인 → 메인 화면→ 게시글 작성
2. 좋아요 클릭 시 애니메이션 + 점수 반영 표시
3. 게시글 카드, 감정 점수 바, Modal 컴포넌트, 채팅 리스트

6.3 기술 스택

- React + TypeScript, Redux, Tailwind CSS등

7. 백엔드 설계 및 데이터베이스 모델링

7.1 주요 테이블 ERD

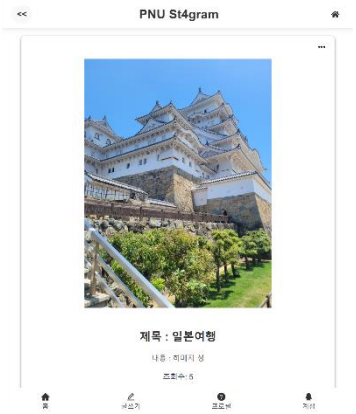
- User (id, username, password, todayScore, createdAt, updatedAt)
- Post (id, userId, title, content, label, score, createdAt, updatedAt)
- Like (id, userId, postId, createdAt)
- Share (id, userId, postId, createdAt)
- ChatRoom (id, participant1Id, participant2Id, createdAt
- Message (id, chatRoomId, senderId, content, sentAt)
- EmotionScoreHistory (id, userId, date, score(감정과 매핑))

7.2 API 서버 구조

- Controller → Service → Repository Layer

8. 진행 현황

작업 항목	상태
GPT API 연동 및 테스트	완료
API 명세서 작성	완료
DB 모델링 및 ERD 작성	진행 중
프론트엔드 와이어프레임 설계	진행 중
백엔드 기본 골격(프로젝트 셋업)	예정
프론트엔드 컴포넌트 구현	진행 중




```
[분석 대상 문장]: 가격이 착한것도 아니고 제품을 어케 만들면 일주일만에 계기판이 고장이나고 사후처리가 저따위임?

[프롬프트 입력 내용]

다음 문장의 감정을 분석해 주세요. 감정은 다음 여섯 가지 중 하나로만 대답해 주세요:
Joy, Sadness, Anger, Fear, Disgust, Neutral

문장: "가격이 착한것도 아니고 제품을 어케 만들면 일주일만에 계기판이 고장이나고 사후처리가 저따위임?"

감정:

[GPT 전체 응답]
Anger
[추출된 감정]: Anger
→ 최종 감정 결과: Anger
--------------------------------------------------

[분석 대상 문장]: 어뜩해... 맘아파.ㅠ

[프롬프트 입력 내용]

다음 문장의 감정을 분석해 주세요. 감정은 다음 여섯 가지 중 하나로만 대답해 주세요:
Joy, Sadness, Anger, Fear, Disgust, Neutral

문장: "어뜩해... 맘아파.ㅠ"

감정:

[GPT 전체 응답]
Sadness
[추출된 감정]: Sadness
→ 최종 감정 결과: Sadness
```

9. 향후 일정 및 작업 계획

기간	주요 내용
6월 초	- 프론트엔드 UI 컴포넌트 개발 및 최종 디자인 마무리- 사용자 피드백 반영용 프로토타입 정리
6월 초 ~ 7월 중순	- 프론트엔드·백엔드 통합 협업 개발 • API 연동 검증 및 예외 처리 보완 • 인증·인가 로직 및 권한 체크 완성 • DM·공유 기능 종합 테스트
5월 중순 ~ 7월 중순	- GPT API 학습 파이프라인 가동 및 성능 튜닝 • 추가 프롬프트 실험 및 데이터셋 확장 • 응답 지연 최소화 방안 검증
7월 중순	- 전체 기능 개발 완료 및 내부 통합 테스트- 최종 보고서 작성