

# AI 기반 추억 앨범 및 치매 예방 서비스



202155514 김나림

202155540 김채현

지도교수 조준수

---

## 목 차

<b>1. 서론</b>	<b>1</b>
1.1. 연구 배경	1
1.2. 기존 문제점	1
1.3. 연구 목표	2
<b>2. 연구 배경</b>	<b>2</b>
2.1. 사회적·이론적·기술적 배경	2
2.2. 기존 방식과의 차이점	2
<b>3. 연구 내용</b>	<b>3</b>
3.1. 시스템 흐름도	3
3.2. 감정 분석 시스템	4
3.2.1. 표정 분석	4
3.2.2. 텍스트 분석	9
3.2.3. 발화 감정 통합	15
3.2.4. 대화 감정 흐름 분석	16
3.2.5. 대규모 언어 모델(LLM: Large Language Model)	18
3.2.6. Model Serving	20
3.3. 플랫폼 구현	21
3.3.1. 로그인 및 권한 관리	21
3.3.2. 질문 데이터베이스	22
3.3.3. 대화 플로우 및 앨범 생성	23
3.3.4. 프라이버시 보호 및 접근성 강화	24
3.4 시스템 설계	25
<b>4. 연구 결과 분석 및 평가</b>	<b>39</b>
4.1. 성능 평가	39
4.2. 시스템 결과물	42
4.3. 개발 일정 및 역할 분담	43
<b>5. 결론 및 향후 연구 방향</b>	<b>44</b>
<b>6. 참고 문헌</b>	<b>45</b>

## 1. 서론

### 1.1. 연구 배경

대한민국은 이미 초고령 사회에 진입했으며, 고령화 가속은 치매 유병·관리 비용 증가라는 중대한 사회적 과제를 동반한다. 한정된 인력·시설 중심의 전통적 돌봄만으로는 수요를 흡수하기 어렵기에, 확장 가능하고 비용 효율적인 예방·정서 지원 솔루션이 필요하다. 본 연구는 임상적으로 근거가 축적된 회상 요법(Reminiscence Therapy)을 멀티모달 대화형 AI와 결합해, 고령 사용자의 기억·정서 지원을 지속 가능하고 접근성 높은 디지털 서비스로 제공하는 것을 목표로 한다.

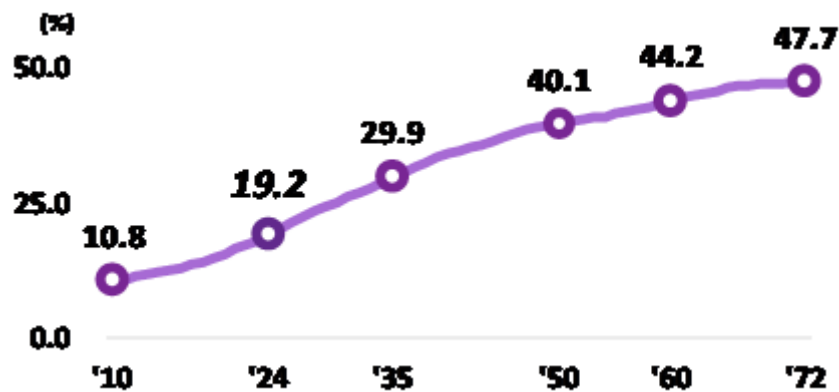


그림1. 통계청, 장래인구추계: 2022~2072년 <고령인구(65세 이상) 비중>

### 1.2. 기존 문제점

기존 회상 요법은 대면 중심으로 이루어져 시간과 장소의 제약이 크고, 전문가의 개입이 필수적이어서 인력과 비용의 부담이 따른다. 이를 보완하기 위해 일부 디지털 프로그램이 시도되었으나, 대부분 단순한 정보 제공이나 정적 콘텐츠 수준에 머물러 사용자와의 깊은 정서적 교감을 형성하는 데 한계가 있었다. 또한, 감정은 복합적인 요소로 발현되기 때문에 텍스트나 표정 등 단일 양식만으로는 사용자의 감정 상태를 정확히 해석하기 어렵다. 더불어, 고령층에게는 긴 텍스트 입력 방식의 인터페이스가 사용성 저하의 요인이 되어 디지털 접근성에도 장벽이 존재한다.

---

### 1.3. 연구 목표

본 연구는 이러한 문제점을 극복하기 위해 멀티모달 대화형 AI를 기반으로 한 디지털 회상 요법 플랫폼을 설계하고 구현하는 것을 목표로 한다. 구체적으로

1. 안내형 디지털 회상 요법을 통해 시니어의 장기 기억을 자극하고 인지 기능을 강화한다.
2. 얼굴 표정과 언어적 단서를 동시에 해석할 수 있는 멀티모달 감정 인식 기술을 적용하여 실시간으로 적응적인 정서 지원을 제공한다.
3. 대화 결과물을 공유 가능한 감성 일기의 형태로 생성함으로써 가족과의 소통을 촉진하고, 이를 통해 고립감을 해소하며 세대 간 유대감을 강화한다.

## 2. 연구 배경

### 2.1. 사회적·이론적·기술적 배경

회상요법(Reminiscence Therapy, RT)은 과거 경험과 장기 기억을 떠올리게 함으로써 고령자의 인지 기능과 정서를 개선하는 비약물적 치료로 효과가 입증되어 왔으며, 본 연구는 이를 AI 기반 대화로 확장하여 촉진자(facilitator) 역할을 수행하고 결과를 일기로 구조화해 사회적 연결을 강화하고자 한다.

특히 고령층을 대상으로 하는 연구에서는, 일반 성인보다 표정 변화가 미묘하고 발화 패턴도 다르다는 특성이 보고되고 있다. 이 때문에 시니어 데이터에 특화된 모델 개발이나 데이터셋 보정이 중요하게 다뤄지고 있으며, 이는 본 연구와도 직접적으로 맞닿아 있다. 본 프로젝트 또한 이러한 최신 연구 흐름을 반영하여, YOLO 기반 얼굴 표정 분석과 KoBERT 기반 텍스트 분석을 결합한 멀티모달 감정 인식 구조를 채택하였다.

### 2.2. 기존 방식과의 차이점

본 연구는 노인 공학(gerontechnology), 계산 심리학(computational psychology), 그리고 멀티모달 인공지능(multimodal AI)이라는 세 가지 학문 분야의 중요한 교차점에 위치한다.

기존에도 고령자를 위한 대화형 에이전트, 멀티모달 감정 인식 등에 대한 개별적인 연구는 존재해왔다. 그러나 이 핵심 요소를 하나의 통합된 시스템으로 구현한 연구는 거의 찾아보기 어렵다. 본 연구의 독창성은 바로 이러한 '통합'에 있다. 즉, (1) 임상적으로 검증된

치료를 (2) 접근성 높은 대화형 인터페이스를 통해 전달하고, 이 인터페이스를 (3) 멀티모달 인식 엔진을 통해 적응적이고 공감적으로 만들며, 최종적으로 (4) 실제 세상의 사회적 연결을 촉진하는 것을 목표로 하는 시스템을 구축했다는 점에서 차별성을 가진다.

### 3. 연구 내용

#### 3.1. 시스템 흐름도

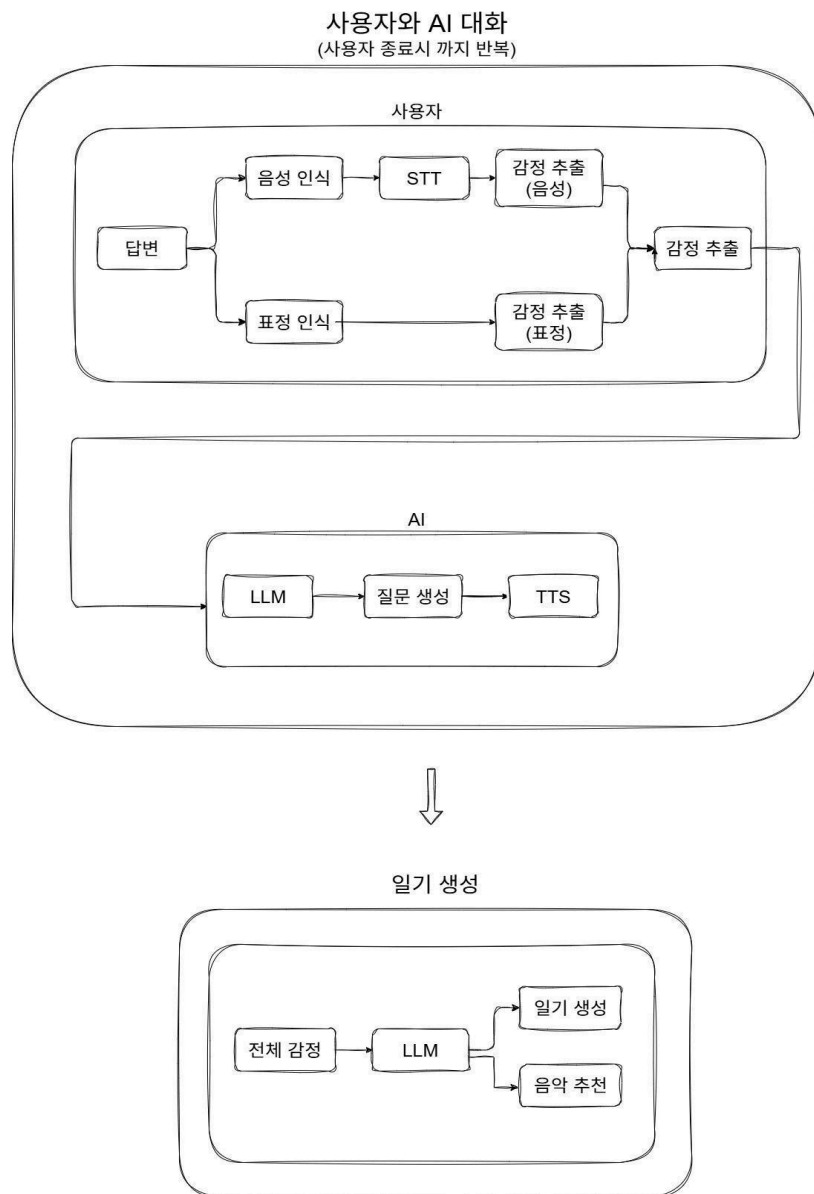


그림2. 시스템 흐름 다이어그램

---

본 시스템은 **사용자-AI 대화 단계**와 **일기 생성 단계**로 구성된다. 대화 단계에서는 사용자의 음성과 표정을 인식하여 발화 단위 감정을 추출하고, 이를 기반으로 AI가 회상 중심의 대화를 이어간다. 이어서 발화별 감정을 종합해 한 대화 단위의 통합된 감정 흐름을 도출하며, 이 흐름과 대화 요약이 일기 생성 모듈의 입력으로 활용된다. 마지막으로 이를 바탕으로 일기와 감정에 어울리는 음악과 배경 색상을 추천한다.

## 3.2. 감정 분석 시스템

본 프로젝트의 감정 분석 시스템은 **음성 기반 발화 분석**과 **표정 기반 이미지 분석**을 모두 수행하는 **멀티모달 감정 인식 방식**으로 설계되었다. 감정 데이터의 라벨링 체계는 *기쁨*, *슬픔*, *당황*, *불안*, *상처*, *분노*의 여섯 가지 범주로 구성되며, 발화와 표정에서 추출된 정보를 통합하여 보다 신뢰도 높은 감정 인식을 실현한다.

### 3.2.1. 표정 분석

#### 1. 데이터 구축

본 프로젝트의 표정 감정 분석 모델을 학습시키기 위해 AI-Hub의 ‘한국인 감정인식을 위한 복합 영상’을 활용하였다. 시니어 맞춤형 모델을 구축하기 위해 노년층의 얼굴 표정과 다양한 감정 상태를 포함하는 해당 데이터셋을 자료로 선정하였다. 원천 데이터는 전 연령대를 아우르지만 본 연구에서는 시니어 대상 최적화를 위해 **50대 및 노년층의 데이터만을 선별**하여 학습에 사용하였다.

- 데이터 출처: [AI-Hub - 한국인 감정인식을 위한 복합 영상](#)
- 데이터 전처리 및 정제:
  - **품질 필터링**: 블러·저해상도·강한 가림·조명 불균형 등 기준을 만족하는 이미지와 얼굴박스/신뢰도 임계치 미만 샘플 제외하였다.
  - **데이터 형식 변환**: 모델 학습에 필요한 JSON 포맷의 바운딩 박스 좌표를 YOLOv8 모델의 표준인 .txt 포맷(정규화된 상대 좌표)으로 변환하는 자동화된 파이프라인을 구축하였다. 바운딩 박스는 서로 겹침(IoU)  $\geq 0.7$ 인 박스들만 모아 중앙값으로 합쳤고, 표정 영역을 놓치지 않도록 약 6% 패딩을

적용한 뒤 YOLO 형식(cx, cy, w, h)으로 변환하였다.

- **Train/Val/Test 구성:**

- **분할 원칙:** 같은 사람이 학습·검증·테스트에 동시에 등장하지 않도록 사람 단위(subject-wise)로 분리하였다.
- **밸런싱:** 검증 세트는 클래스 균형을 맞추도록(subject 라운드로빈) 구성하였다.

이러한 정제 과정을 거쳐 최종적으로 구축된 데이터의 감정 라벨별 분포는 아래 [표 1]과 같다.

	기쁨	분노	상처	슬픔	당황	불안
<b>train</b>	1373	1385	1398	1403	1400	1375
<b>val</b>	214	203	193	195	183	192
<b>test</b>	196	203	199	209	204	199
<b>합계</b>	<b>1783</b>	<b>1791</b>	<b>1790</b>	<b>1607</b>	<b>1787</b>	<b>1766</b>

표1. YOLO 데이터의 감정 라벨별 분포

## 2. 모델 선정 배경

본 연구에서는 노인 표정 감정 분석을 위해 **YOLOv8**을 채택하였다. YOLOv8은 얼굴 탐지와 표정 분류를 하나의 프레임워크에서 동시에 수행할 수 있어 별도의 전처리 모듈 없이 활용 가능하다. 또한 경량성과 빠른 추론 속도를 갖추어 실시간 대화 환경에 적합하다.

## 3. YOLO 학습 실험 및 계층적 분류 전략

본 프로젝트에서는 YOLOv8을 활용해 6개 감정 클래스를 학습한 결과, joy·angry는 양호했으나 anxious·embarrassed는 저조하였다. anxious를 제외한 5개 클래스 학습에서는 embarrassed 성능이 개선되었으나 hurt 성능이 저하되었고, anxious와 embarrassed를 통합한 complex 클래스를 도입하자 complex 성능은 양호했지만 hurt가 다시 악화되었다. 이를 통해 anxious-embarrassed-hurt 간 경계 간섭이 성능 저하의 주요 원인임을 확인하였다.

---

따라서 본 연구에서는 계층적(hierarchical) 분류 전략을 적용하였다.

- 1차 분류기: joy, angry, sad, hurt, complex
- 2차 분류기: 1차에서 complex로 분류된 샘플만 입력하여 anxious와 embarrassed로 세분화.

이와 같은 2단계 구조는

1. 성능 저하를 유발하는 anxious–embarrassed–hurt 간 혼동을 줄이고,
2. 모델이 우선적으로 “큰 범주(Complex)”를 학습한 뒤 세부 감정을 분류하도록 유도하여, 시니어 표정 인식의 신뢰도를 높이는 최적화 전략으로 활용되었다.

#### 4. 계층적 학습 및 최적화 전략

- 1차 분류기 학습
  - 클래스: joy, angry, sad, hurt, complex
  - anxious와 embarrassed는 단일 단계에서 동시에 분류할 경우 성능이 불안정하게 나타났으므로, 우선 두 클래스를 complex라는 상위 범주로 통합하였다.
- 2차 분류기 학습
  - 클래스: anxious, embarrassed
  - 1차 분류기에서 complex로 분류된 샘플만 입력받아 두 감정을 세분화하였다.
  - 이 방식은 1차에서 큰 틀을 구분하고 2차에서 세부 감정을 판별하는 구조로, anxious와 embarrassed의 경계 모호성을 완화할 수 있었다.
- 추론 게이트(Inference Gate) 적용
  - 최종 추론 시 1차와 2차 결과를 결합하는 규칙을 도입하였다.
  - 예를 들어, 1차에서 complex로 분류된 경우 무조건 2차로 전달하고, 2차 분류기의 신뢰도가 낮으면 다시 1차 결과를 유지하였다. 또한 joy·sad·angry 등 주요 클래스에서 신뢰도가 낮은 경우, 보조적으로 2차 분류를 수행하여 잠재적인 anxious/embarrassed를 보완하도록 하였다.



- 임계치 최적화 및 보정

- 검증 데이터셋을 활용해 각 클래스별 **F1-score**가 최대가 되는 **confidence threshold**를 산출하였다.
- 특히 **anxious**와 **embarrassed**는 임계치를 높여 잘못된 분류를 줄였으며, **joy-sad**는 **recall**을 중시하여 임계치를 상대적으로 낮게 설정하였다.
- 추가적으로, **softmax** 확률에 대한 온도 보정(**Temperature Scaling**)을 적용하여 예측 신뢰도를 개선하였다.

이와 같은 2단계 구조와 후처리 기법을 통해, 모델은 큰 범주 → 세부 감정 순서로 안정적인 분류를 수행하며, 시니어 표정 인식에서 중요한 **anxious-embarrassed-hurt** 구간의 혼동을 효과적으로 줄일 수 있었다.

## 5. 평가

### (1) 성능 지표 (Performance Metrics): Macro F1-Score

다중 감정 분류에서는 클래스 불균형이 흔해 단순 정확도(**Accuracy**)는 다수 클래스만 잘 맞춰도 높게 나와 성능을 과대평가할 수 있다. 반면, **Macro F1-score**는 각 감정 클래스의 **F1**을 동일 가중으로 평균하기 때문에, 희소한 감정(예: 불안·당황)까지 공정하게 반영한다.

### (2) 계층적 분류 전략 vs 단일 6클래스 분류 성능 비교

- 단일 6클래스 분류

라벨	F1-Score
기쁨	92
분노	83
상처	66
슬픔	70

당황	59
불안	52
최종	70.3

표2. 단일 6클래스 분류 모델 성능 측정값

#### - 계층적 분류

라벨	F1-Score
기쁨	93
분노	81
상처	63
슬픔	71
complex	67
최종	72.0

표3. 계층적 분류 모델 성능 측정값

따라서 계층적 분류 전략은 단일 6클래스 분류 대비 전체 성능 지표(F1-score)를 향상시켰으며, 특히 당황-불안과 같이 경계가 모호한 감정을 복합 범주(complex)로 우선 분류함으로써 모델의 혼동을 줄이는데 기여하였다. 이는 시니어 표정 인식에서 자주 발생하는 미세한 감정 구분 문제를 완화하고, 보다 신뢰성 있는 감정 추론을 가능하게 하는 효과적인 최적화 전략임을 확인할 수 있었다.

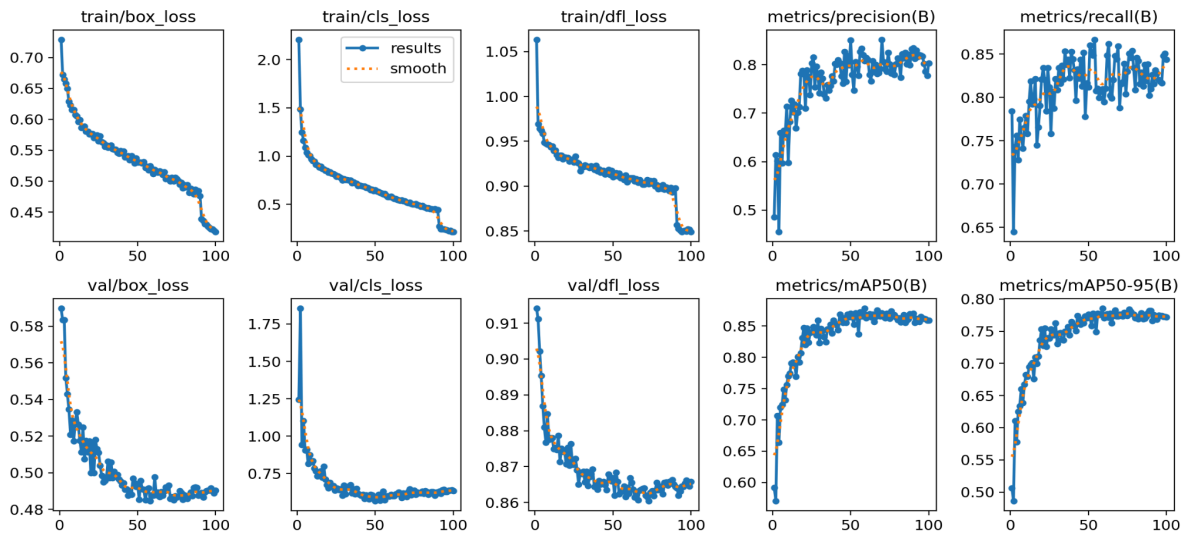


그림3. 계층적 분류 모델의 학습 곡선

### 3.2.2. 텍스트 분석

#### 1. 데이터 구축

본 연구의 텍스트 감정 인식 모델 학습을 위해 AI Hub에서 제공하는 감정 대화 말뭉치 데이터셋을 활용하였다. 해당 데이터셋은 클라우드소싱 방식을 통해 약 1,500명의 일반인으로부터 수집되었으며, 음성 15,700건 및 텍스트 27만 건의 대규모 말뭉치로 구성되어 있다. 본 연구에서는 이 중 텍스트 데이터만을 학습에 사용하였다.

- 데이터 출처: [AI hub - 감정 대화 말뭉치](#)
- 데이터 구조
  - 형식: 사용자의 발화와 시스템의 응답이 순차적으로 이어지는 대화 형식이다.
  - 구성: 각 대화는 최대 3턴(turn), 즉 3번의 상호작용을 포함하는 구조를 가진다.
- 데이터 전처리 및 정제:
  - 대상 데이터 선별: 서비스의 주요 대상인 시니어를 고려하여, 전체 데이터 중 연령 정보가 '노년'으로 분류된 데이터만을 선별하였다.
  - 품질 관리: 대화의 맥락적 일관성을 확보하고 데이터의 품질을 높이기 위해, 3턴의 대화 구조가 누락된 불완전한 샘플은 제외하였다.

위와 같은 정제 과정을 거쳐 최종 데이터를 구축했으며, 감정 라벨별 학습 데이터 분포는 아래 [표 2]와 같다.

	기쁨	분노	상처	슬픔	당황	불안
train	2230	4036	4067	3822	3891	3880
val	418	410	372	356	303	477
합계	2,648	4,446	4,439	4,178	4,194	4,357

표4. KoBERT 최종 학습 및 검증 데이터의 감정 라벨 분포

## 2. 모델 선정 배경

본격적인 모델 학습을 위하여 텍스트 감정 분석 모델 선정을 진행하였으며, 초기 모델이었던 KoBERT의 한계점을 보완할 수 있는 **KLUE-BERT**를 최종 모델로 채택하였다.

구분	BERT	KoBERT	KLUE-BERT
개발사	Google	SKT	Upstage
주요 학습 데이터	다국어 위키피디아	한국어 뉴스, 위키백과	KLUE 벤치마크 데이터셋(뉴스, 댓글, 웹 문서 등)
토큰나이저	WordPiece	WordPiece	형태소 기반 서브워드 (Morpheme-based)
주요 특징	- 다양한 언어 처리 가능	- 한국어 데이터에 특화 - 한국어 문장 구조 이해도 우수	- 신조어 및 구어체 처리 능력 탁월 - 다양한 한국어 NLP 태스크에서 최고 성능(SOTA) 기록
본 프로젝트 적합성	- 한국어 교착어 특성 반영에 한계	- 최신 구어체 및 감정 표현 대응력에서 다소 아쉬움	- 시니어와의 일상 대화 및 감정 표현을 이해하는 데 가장 적합하다고 판단

표5. 언어 모델 비교

연구 초기에는 **KoBERT** 모델을 채택하여 학습을 진행하였으나, 테스트 과정에서 다음과 같은 한계점을 식별하였다.

1. 감정 분류 태스크에서의 성능 한계: **KoBERT**는 뉴스, 백과사전 등 정제된 문어체 텍스트에서는 우수한 성능을 보였으나, 본 프로젝트의 목적인 시니어의 일상 대화와 같은 비정형적인 구어체 기반의 감정을 분류하는 데에는 다소 미흡한 성능을 나타냈다.

2. 토큰나이저(**Tokenizer**) 호환성 및 확장성: 다양한 전처리 및 토큰화 기법을 적용하여 모델 성능을 최적화하는 과정에서, 최신 연구에서 활용되는 토큰나이저와의 호환성 및 확장성 측면에 일부 제약이 있음을 확인하였다.

이러한 문제점을 해결하고, 구어체 텍스트에 대한 이해 능력을 강화하여 최종 모델의 성능을 향상시키기 위해 **KLUE-BERT** 모델로 교체하였다.

### 3. 이원화(**Two-stage**) 학습 파이프라인 설계

본 프로젝트의 텍스트 감정 인식 시스템은 사용자의 발화를 단편적으로 분석하는 것을 넘어, 대화의 전후 맥락까지 종합적으로 고려하는 고도화된 모델을 목표로 설계되었다. 이를 위해, **KLUE-BERT** 모델을 순차적으로 학습시켜 유기적으로 연동하는 이원화(**Two-stage**) 구조의 학습 파이프라인을 구축하였다.

(1) 1단계 (**U1 모델**): 대화의 시작점에서 사용자의 즉각적인 감정을 포착하는 **U1** 모델을 학습시킨다. 사전학습된 **KLUE-BERT**를 기반으로 **fine-tuning**하여, 최종 **[CLS]** 토큰 출력 위에 드롭아웃(**p=0.3**)과 단일 선형 분류기를 결합한 구조로 아키텍처를 구성하였다. 학습 시에는 차등 학습률(**Differential Learning Rate**) 전략을 적용하여, 사전학습된 인코더에는 낮은 학습률(**2e-5**)을, 새롭게 초기화된 분류기에는 상대적으로 높은 학습률(**5e-5**)을 부여하였다. 모든 인코더 계층을 학습 가능 상태로 두어 감정 분류라는 특정 태스크(**Task**)에 모델 전체가 깊이 있게 적응하도록 설계하였다.

또한, 클래스 간 데이터 불균형 문제를 완화하기 위해 학습 데이터 분포를 기반으로 클래스 가중치(Class Weight)를 산출하여 손실 함수(Loss Function)에 반영하였다. 최적화 알고리즘으로는 AdamW를, 학습률 스케줄링 기법으로는 Cosine Warmup을 각각 적용하였다.

**(2) 2단계 (CTX 모델):** 1단계에서 완성된 U1 모델을 교사 모델(Teacher Model)로 활용하여, 대화의 맥락을 이해하는 CTX 모델을 학습시키는 의사 라벨링(Pseudo-Labeling) 기법을 핵심 전략으로 채택하였다.

#### 4. 학습 결과 및 분석

##### (1) 성능 평가 지표 : Macro F1-Score

모델의 텍스트 감성 분류 성능을 평가하기 위한 핵심 지표로 Macro F1-Score를 채택하였다. 정확도(Accuracy)는 클래스 불균형이 존재할 경우 다수 클래스의 성능에 의존하여 모델을 왜곡 평가할 수 있다. 반면, Macro F1-Score는 정밀도(Precision)와 재현율(Recall)의 조화 평균인 F1-Score를 모든 클래스에 동일한 가중치를 부여하여 산술 평균을 낸다. 이를 통해 데이터 수가 적은 소수 클래스의 성능 저하를 명확하게 파악할 수 있으며, 전체 감정에 대한 모델의 균형 잡힌 성능을 측정하는 데 가장 적합하다.

##### (2) 정량적 성능 평가

###### ● U1 모델 학습 결과

라벨 (Label)	F1-Score
상처	0.639094
슬픔	0.674797
불안	0.739035
당황	0.648884
분노	0.724816
기쁨	0.948749

표6. U1 모델 성능 측정값

U1 모델은 검증 데이터셋(Validation Set)에 대해 Macro F1-Score 72.9% 의

성능을 기록했다. 이는 주어진 6종 감정 분류 태스크에 모델이 성공적으로 최적화되었음을 보여주며, CTX 모델 성능 비교를 위한 기준선(Baseline) 역할을 수행한다.

● CTX 모델 학습 결과

라벨 (Label)	F1-Score
상처	0.77957
슬픔	0.681178
불안	0.770161
당황	0.702238
분노	0.727273
기쁨	0.985714

표7. CTX 모델 성능 측정값

CTX 모델은 검증 데이터셋에 대한 Macro F1-Score가 0.7845를 기록하여, U1 모델의 0.729 대비 약 7.6%p의 의미 있는 성능 향상을 보였다. 이는 단일 발화 정보만을 사용하는 것보다 대화의 전후 맥락을 종합적으로 반영하는 것이 실제 사용자의 감정 추론에 더 효과적이라는 핵심 가설을 명확히 입증하는 결과이다.

(3) 정성적 성능 평가: 문맥 이해 능력 검증

CTX 모델이 실제로 대화의 문맥을 얼마나 깊이 있게 이해하는지에 대한 질적 평가를 수행하였다.

● 대화 흐름 일관성 실험 (Experiment A)

이 실험은 CTX 모델이 대화 내 감정의 흐름을 얼마나 안정적이고 논리적으로 추론하는지 확인하는 것을 목표로 한다. 실험 과정에서 사용된 용어는 다음과 같이 정의한다.

대화 구성 요소	
U1, U2, U3:	각각 대화의 첫 번째, 두 번째, 세 번째 사용자 발화(User Utterance)를 의미한다.

S1, S2:	각각 U1, U2에 대한 시스템의 응답(System Response)을 의미한다.
---------	-----------------------------------------------

대화예측(Prediction) 시점	
P1	문맥 없이 U1 단일 문장만으로 감정을 예측한 결과이다.
P2	U1, S1의 문맥을 바탕으로 U2의 감정을 예측한 결과이다.
P3	U2, S2의 문맥을 바탕으로 U3의 감정을 예측한 결과이다.

결과(예측 일치율)	
P1 vs P2	71.0%
P2 vs P3	55.3%
P1, P2, P3 모두 동일 감정으로 분류된 비율	38.8%

다수의 사례에서 모델은 대화의 핵심 감정을 안정적으로 유지하는 경향을 보였으며, 일부 사례에서는 감정이 합리적으로 변화하는 지점을 정확히 포착(예: 슬픔 → 슬픔 → 분노)하여 인간의 맥락 이해 패턴과 유사한 추론 능력을 나타냈다.

● 위치 불변성 실험 (Experiment B)

동일한 발화가 대화의 어느 위치에 놓이더라도 모델이 일관된 감정으로 분류하는지를 검증하였다.

입력 형태	
Q1	U1 문장만 단독 입력.
Q2	CTX(U1, S1, U1)
Q3	CTX(U2, S2, U1)



구분	결과
세 위치 모두 동일 감정으로 분류된 비율	63.8%
실제 정답과의 일치율(Q1)	55.3%
실제 정답과의 일치율(Q2)	74.1%
실제 정답과의 일치율(Q3)	72.8%

약 2/3(63.8%)의 경우 동일한 감정으로 분류되어 모델이 발화 자체의 의미를 안정적으로 유지함을 보였다. 또한, 정답과의 일치율은 문맥이 보강될수록 상승(Q1 → Q2/Q3)하는 경향을 나타냈다. 이는 **문맥 정보가 제공될 때 모델의 감정 예측이 실제 정답에 더 근접한다**는 점을 나타낸다.

### 3.2.3. 발화 감정 통합

본 연구에서는 노인 대상 발화 상황에서의 안정적 감정 인식을 위해, 표정(이미지) 기반 모델과 텍스트(발화) 기반 모델의 결과를 **결정/점수 레벨에서 결합(late fusion)**하였다.

#### 1. 개별 모델 출력

- YOLOv8 모델은 발화 동안 캡처된 다수의 얼굴 프레임에 대해 프레임별 최상위 감정 라벨과 신뢰도를 얻는다. 프레임 신뢰도  $c_i$ 를 가중치로 하여 라벨별 누적치를 만들고( $c_i^\gamma$  가중,  $\gamma \approx 1.5$ ), 디리클레 스무딩( $\alpha \approx 0.5$ )을 적용해 발화 단위 표정 분포  $p_{\text{face}}(c)$ 를 추정한다. 얼굴 모델에만 존재하는 중립은 제거하고 6개 감정으로 재정규화한다.
- KoBERT 모델은 발화 문장을 입력받아 동일한 감정 클래스에 대한 확률 분포  $p_{\text{text}}(c)$ 를 출력한다(softmax).

#### 2. 확률 정규화 및 보정

- 두 모델의 출력 확률은 스케일 차이가 존재하므로, 온도 보정(Temperature Scaling)을 통해 캘리브레이션을 수행하였다.

- 이를 통해 YOLO와 KoBERT 확률이 동일한 기준에서 해석 가능해졌으며, 이후 가중 합산을 통한 멀티모달 통합에 활용할 수 있었다.

### 3. 가중 합산 방식(Late Fusion)

- 최종 감정 확률은 다음과 같이 산출하였다.

$$p_{\text{fused}}(c) = \text{normalize}(w_{\text{text}} \cdot p_{\text{text}}(c) + w_{\text{face}} \cdot p_{\text{face}}(c))$$

모달 품질을 반영한 가중치  $w_{\text{text}}$ ,  $w_{\text{face}}$ 로 선형 결합하여 발화 단위 통합 분포  $p_{\text{fused}}(c)$ 를 얻는다. 여기서  $w_{\text{text}} = \lambda_{\text{text}} \cdot q_{\text{text}}$ ,  $w_{\text{face}} = \lambda_{\text{face}} \cdot q_{\text{face}}$ .

- $q$ 는 각 모달의 품질이며, 한쪽 품질이 낮으면 해당 가중을 자동으로 낮추는 게이팅이 된다.

### 4. 불일치 처리 (Conflict Handling)

표정과 발화의 감정이 다르게 나올 경우, 두 결과를 품질 가중치로 합성한 분포에서 최종 감정을 결정하였다. 다만 실제 대화 맥락에서는 **말(텍스트)의 의미 신호가 더 중요**하다고 판단하여, 불일치 시 텍스트 쪽 가중치를 조금 더 높여 반영하였다. 또한 두 모델 모두 확신이 낮을 경우에는 **중립 상태**로 처리하여 대화 생성기에 감정을 직접 언급하지 않고 자연스럽게 이어가도록 하였다.

### 5. 최종 감정/신뢰도

$$\hat{y} = \arg \max_c p_{\text{fused}}(c), \quad \text{confidence} = p_{\text{fused}}(\hat{y})$$

통합 분포의 최대 확률 감정을 최종 예측으로, 해당 확률을 신뢰도로 제시한다.

이와 같은 **Late Fusion** 기반의 멀티모달 통합 전략은 표정과 텍스트 중 한 쪽 신호가 불명확할 때 다른 쪽 정보를 보완할 수 있어, 노인 대상 대화 상황에서 보다 안정적이고 신뢰도 높은 감정 인식을 가능하게 하였다.

### 3.2.4. 대화 감정 흐름 분석

발화 단위의 통합 확률 분포( $p_{\text{fused}}$ )를 활용하여 대화 전체의 감정 흐름을 파악하였다.

#### 1. 발화 단위 통합 확률 분포 산출

각 발화에서 텍스트 기반 확률 분포( $p_{\text{text}}(c)$ )와 얼굴 기반 확률 분포( $p_{\text{face}}(c)$ )를 가중 합산하여 최종 통합 분포를 계산하였다.

$$p_{\text{fused}}(c) = \text{normalize}(w_{\text{text}} \cdot p_{\text{text}}(c) + w_{\text{face}} \cdot p_{\text{face}}(c))$$

여기서  $w_{\text{text}}$ ,  $w_{\text{face}}$ 는 모달리티별 신뢰도와 가중치 계수를 반영한 값이다.

최종 감정 라벨은 다음과 같이 결정된다.

$$\hat{y} = \arg \max_c p_{\text{fused}}(c), \quad \text{confidence} = p_{\text{fused}}(\hat{y})$$

#### 2. 대화 시퀀스 구성

대화는 발화 순서에 따라  $\{p_{\text{fused}}^{(1)}, p_{\text{fused}}^{(2)}, \dots, p_{\text{fused}}^{(T)}\}$ 의 시퀀스로 표현된다.

각 턴별 분포에서 지배적 감정( $\hat{y}^{(t)}$ )을 추출하고, 그 신뢰도를 함께 기록한다.

턴 번호 $t$	지배적 감정 $\hat{y}^{(t)}$	신뢰도 $p_{\text{fused}}^{(t)}(\hat{y})$	분포 예시
1	슬픔	0.72	{슬픔: 0.72, 기쁨: 0.10, ...}
2	슬픔	0.64	{슬픔: 0.64, 기쁨: 0.15, ...}
3	기쁨	0.70	{기쁨: 0.70, 슬픔: 0.18, ...}
4	기쁨	0.81	{기쁨: 0.81, 불안: 0.07, ...}

표8. 지배적 감정 예시

#### 3. 세그먼트화

---

연속된 구간을 묶어 세그먼트(segment)를 정의한다. 세그먼트는 시작 턴, 종료 턴, 지배적 감정, 평균 신뢰도, 정·부정 점수 등으로 요약된다.

세그먼트	턴 범위	지배적 감정	평균 신뢰도	정서 경향(Valence)
S1	1-2	슬픔	0.68	부정적 (-1)
S2	3-4	기쁨	0.76	긍정적 (+1)

표9. 세그먼트 예시

#### 4. 정량적 지표 산출

- **전환 횟수(flips)**: 감정이 바뀐 횟수
- **긍정 비율(positive ratio)**: 전체 발화 중 긍정 감정이 차지하는 비율
- **최장 부정 구간(longest negative run)**: 연속된 부정 감정 턴의 길이
- **피크 위치(peak arousal turn)**: 가장 강한 정서적 반응이 나타난 턴

#### 5. 패턴 분류

마지막으로, 위의 지표와 세그먼트를 기반으로 대화의 전반적인 흐름을 패턴으로 분류하였다.

- **상승형(Upward)**: 부정 → 긍정 전환
- **하강형(Downward)**: 긍정 → 부정 전환
- **안정형(Stable)**: 큰 변동 없이 유지
- **급반전형(Flip)**: 짧은 간격으로 빠른 전환
- **U자형(U-shape)**: 긍정 → 부정 → 긍정

### 3.2.5. 대규모 언어 모델(LLM: Large Language Model)

#### 1. 언어 모델 선택

본 연구에서는 대화 요약, DiaryPlan 생성, 일기 작성, 음악 추천 등 다양한 생성 작업에 **GPT-4o mini**를 활용하였다. GPT-4o mini는 GPT-4 수준의 성능을 유지하면서도 빠른 추론 속도와 비용 효율성을 갖추어, 하루 여러 차례 실행되는

---

치매 예방 서비스에 적합하다.

## 2. 대화 생성

본 연구에서의 대화 생성 모듈은 단순한 질문·응답을 넘어서, 회상 요법(**Reminiscent Therapy**)의 목적을 달성하기 위해 설계되었다. 사용자가 과거 경험을 떠올리고 정서적으로 안정감을 얻을 수 있도록, 언어 모델은 다음의 입력 정보를 종합하여 응답을 생성하였다.

- **사용자 감정 정보:** 발화 단위로 산출된 통합 감정(예: 기쁨, 슬픔 등)과 해당 신뢰도를 입력으로 제공하였다.
- **대화 맥락:** 직전까지의 사용자 발화와 AI 발화를 포함한 대화 히스토리를 함께 전달하여, 모델이 맥락을 유지하면서도 자연스러운 흐름의 대화를 이어갈 수 있도록 하였다.

## 3. 일기 생성

### (1) 대화 내용 요약

대화 세션은 여러 발화(turn)로 구성되어 있으며, 그대로 LLM에 투입하기에는 길이가 지나치게 길고 불필요한 반복이 포함될 수 있다. 따라서 먼저 대화 요약 단계를 통해 핵심 내용을 정제하였다.

### (2) DiaryPlan 생성

단순 요약만으로는 대화의 감정적 흐름을 충분히 반영하기 어렵기 때문에, **대화 요약과 함께 감정 흐름(emotionFlow, flowPattern)을 결합하여 DiaryPlan을 생성하였다.** DiaryPlan은 일기의 정서적 톤과 구조(시작-전환-마무리)를 설계하는 중간 표현 역할을 한다.

### (3) 일기 작성

DiaryPlan을 기반으로 최종 일기 텍스트를 생성한다. DiaryPlan에 담긴 감정적 톤과 주제 구조를 참고하여, 사용자의 발화와 대화 맥락을 하나의 자연스러운 회상 서술로 재구성하였다.

### (4) 음악 추천

---

생성된 일기의 정서적 톤과 감정 요약을 근거로 음악을 추천하였다. 노인 세대가 친숙하게 느낄 수 있는 곡을 제안하여, 일기 읽기 경험을 한층 풍부하게 하였다.

### 3.2.6. Model Serving

학습이 완료된 YOLOv8 기반 표정 감정 분석 모델과 KoBERT 기반 발화 감정 분석 모델은 **FastAPI 프레임워크**를 이용하여 REST API 형태로 구현하였다. 두 모델 API는 Google Cloud VM 인스턴스 환경에 배포되어 서비스 백엔드에서 실시간으로 호출할 수 있도록 구성하였다. 이를 통해 웹·모바일 환경에서 발생하는 사용자 대화를 클라우드 서버에 전송하면, 서버는 모델 추론을 수행한 뒤 감정 분석 결과를 API 응답으로 반환한다.

추가적으로, **LangChain 프레임워크**를 도입하여 LLM 호출 과정을 체계적으로 관리하였다. LangChain은 프롬프트 템플릿 관리, 멀티스텝 체인 구성(DiaryPlan 생성 → 일기 작성 → 음악 추천), 외부 API 연동 등을 지원함으로써, LLM 기반 모듈이 서비스 로직과 원활히 통합될 수 있도록 한다. 이를 통해 모델 서빙 계층과 생성 응용 계층이 유기적으로 연결되며, 유지보수성과 확장성이 향상되었다.

### 3.3. 플랫폼 구현

#### 3.3.1. 로그인 및 권한 관리

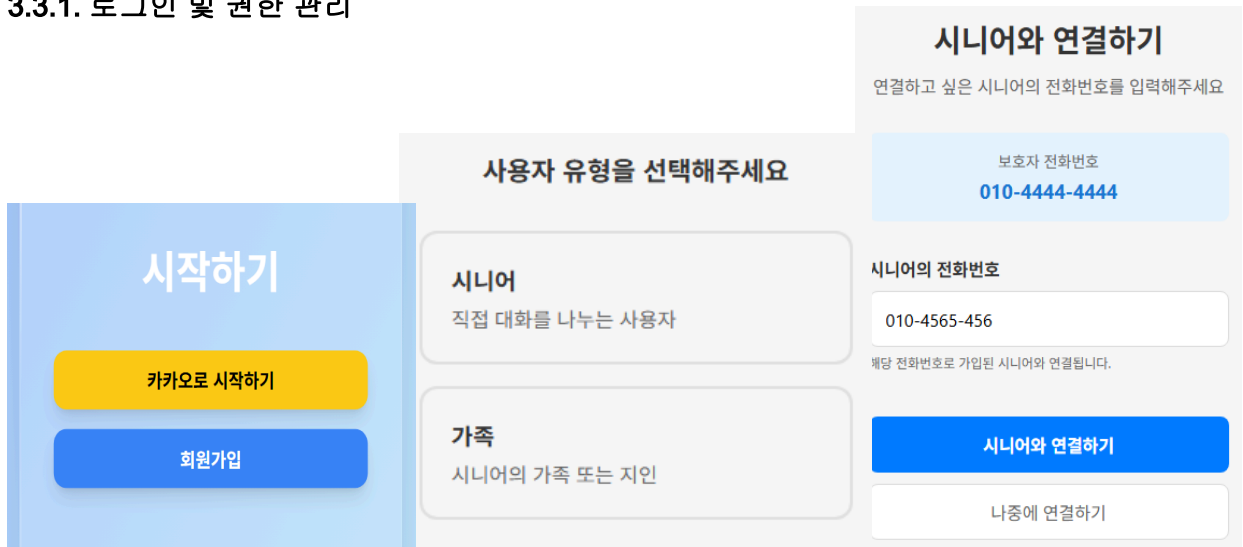


그림4. 회원가입 및 로그인 화면

그림5. 사용자 유형 선택

그림6. 시니어-가족 연결

- 카카오 로그인 및 인증 구조

- 시니어 사용자의 접근성을 극대화하기 위해 카카오 소셜 로그인을 도입하였다. 복잡한 정보 입력 과정 없이 안전하게 로그인을 처리하며, 로그인 성공 후 Spring Security가 사용자 정보를 기반으로 권한을 관리하는 구조를 갖췄다.

- 역할 기반 기능 차별화

- 본 서비스는 시니어와 가족이라는 두 가지 역할을 기반으로 기능이 차별화된다. 특히, 보호자는 시니어와의 연결을 통해 서비스의 핵심 기능을 함께 이용하게 된다.

- 연결 방식: 보호자가 회원가입 후, 시니어의 전화번호를 통해 연결을 신청한다.. 시니어가 이 연결 요청을 승인해야만 관계가 성립된다.
- 관계 구조: 가족과 시니어는 다대다(Many-to-Many) 연결 관계이다.
- 기능 차별화: 시니어 사용자는 회상 요법과 일기 생성 등 서비스의 핵심 기능에 집중하고, 보호자는 시니어가 공유를 허용한 일기를



열람하거나 사진 첨부, 댓글 작성 등 소통을 지원하는 역할을 수행한다.

### 3.3.2. 질문 데이터베이스

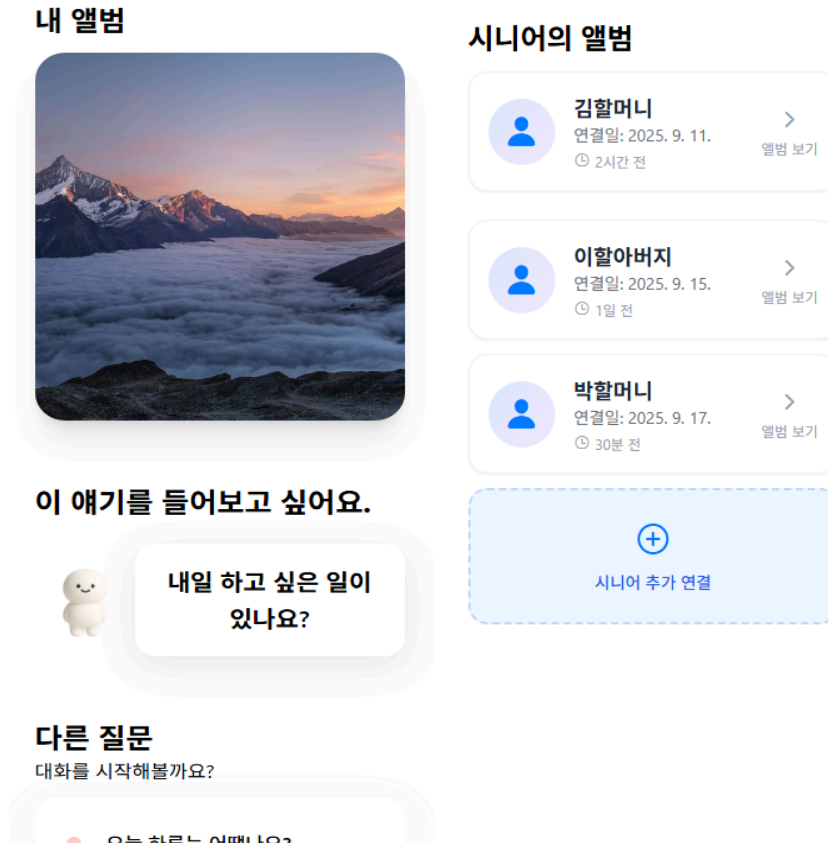


그림7. 시니어 메인 화면

그림8. 보호자 메인 화면

메인 페이지에서 기본 질문을 db에 저장하여 제공하며(최대 50개), 시니어는 무작위 질문 또는 선택한 질문으로 대화를 시작할 수 있다. 본 질문 리스트는 단순 Q&A 모음이 아니라, 회상요법의 임상 가이드를 검토하여 체계적으로 선별·재구성한 치료적 질문 체계이다. 우리는 문헌에서 공통으로 강조되는 축(형성기 기억→성인기의 관계·역할→전환점과 대처→삶의 의미와 유산)을 따라 다섯 범주(A~E)를 정의하고, 각 범주에 맞는 질문을 균형 있게 배치하였다.

- A 범주 (기반과 초기 생애): 어린 시절과 형성기 경험을 탐색하며, 구체적 기억을

---

통해 이야기의 출발점을 마련한다.

- ex) “제일 오래된 기억이 뭐예요?”, “어디서 자라셨어요?”, “학교생활은 어땠나요?”
- **B 범주 (성인기의 궤적)**: 직업, 결혼, 양육 등 사회적 역할과 성취를 중심으로 개인의 정체성과 책임감을 살펴본다.
  - ex) “배우자는 어떻게 만나셨어요?”, “평생 해온 일은 뭐예요?”
- **C 범주 (인생의 전환점)**: 도전과 위기 극복 경험을 회상하여 회복탄력성과 강점을 확인한다.
  - ex) “가장 힘들었던 도전은요?”, “큰 사건을 겪었을 때 어떻게 이겨내셨나요?”, “가장 큰 모험은요?”
- **D·E 범주 (통합과 의미, 유산과 마무리)**: 삶의 사건들을 연결해 의미를 부여하고, 교훈과 유산을 정리하며 삶 전체를 조망한다.
  - ex) 다음 세대에 남기고 싶은 건 뭐예요?”, “사람들이 어떻게 기억해주길 바라세요?”

### 3.3.3. 대화 플로우 및 앨범 생성

- **캡처·인식**
  - **표정 캡처 주기**: 발화 시작~종료까지 **3초 간격**으로 얼굴 프레임 캡처 → 프레임별 최상위 라벨·신뢰도 집계 → 한 발화의 **표정 분포**로 합산.
  - **얼굴 프레임 이탈 알림**: 발화 중 얼굴 박스 미검출 비율이 임계치(예:  $\geq 40\%$ )를 넘으면 “카메라 정면을 봐주세요” 실시간 안내.
  - **사전 테스트**: 대화 시작 전 **카메라/마이크 점검**(얼굴 인식 OK, 버튼/누르고-떼기 동작 연습) 후 시작 가능.
- **무반응(Non-response) 처리**
  - **정의**
    - **무반응 A**: 표정 없음 & 말 없음(얼굴 미검출 + 음성 무입력).
    - **무반응 B**: 말만 있음(표정 없음) → **텍스트 감정만** 분석하여 진행.
    - **무반응 C**: 표정만 있음(말 없음) → **감정 분석 생략**(대화 흐름만 유지).
  - **종료 규칙**: 세션 중 **무반응 A가 3회** 누적되면 안전하게 **자동 종료**(종료 전 1회 경고 팝업).
- **STT/TTS 구성**
  - **STT**: *OpenAI Whisper API* (ko 실시간 스트리밍 / 잡음 강건, 짧은 지연).

- 
- **TTS:** *Google Cloud Text-to-Speech* (ko-KR WaveNet 음성, 속도/피치 조절).
  - **기록·후처리**
    - **음악추천:** 일기 톤/패턴을 반영해 추천 음악을 플레이어 형태로 제공하여, 사용자가 직접 재생, 정지 등의 제어를 할 수 있다.
    - **표지 추천:** 일기 톤/패턴을 반영해 일기 조회 시 감정에 알맞는 표지가 적용된다.
    - **대화 내역 확인:** 사용자는 AI와 대화 내역을 채팅 형식으로 확인할 수 있다.
    - **상호작용:** 보호자의 댓글, 사진 업로드를 통하여 회상 경험을 강화한다.

### 3.3.4. 프라이버시 보호 및 접근성 강화

사용자의 민감 정보를 보호하고 시니어 계층의 접근성을 보장하기 위해 다음과 같은 설계 원칙을 적용하였다.

#### 1. 데이터 프라이버시 보호 설계 (Privacy by Design)

- **기본 비공개 원칙 (Private by Default):**

생성되는 모든 대화 기록과 감성 일기는 별도의 설정 없이 자동으로 비공개 처리된다.

- **명시적 동의 기반 공유 (Opt-in Consent):**

데이터 공유는 반드시 사용자가 공유를 원하는 기록에 한해서만 선택적으로 공개할 수 있으며, 동의 없이는 가족이라도 데이터 열람이 원천적으로 차단된다.

#### 2. 고령자 친화적(Age-friendly) UI/UX 원칙

금융위원회와 금융감독원의 '고령자 친화적 모바일 금융앱 구성 가이드라인'을 일부 준수하여 시니어 사용자의 편의성을 극대화하였다.

- **접근 경로 최소화:**

주요 기능으로 진입하는 과정을 2회 터치 이하로 설계하여 직관성을 높였다.

- **인지적 부담 완화:**

각 단계마다 충분한 설명과 시간을 제공하고, 불필요한 시각 효과 및 애니메이션을 배제하여 사용자가 기능에만 집중할 수 있도록 인터페이스를 구성하였다.

3.4 시스템 설계

3.4.1 개발 환경

Frontend	React Native, Expo, TypeScript, Context API
backend	Spring boot, MySQL
Infra	Docker, AWS (S3, RDS), Google Cloud Platform
AI	FastAPI, Langchain,, Google Cloud Platform

3.4.2. 시스템 구조

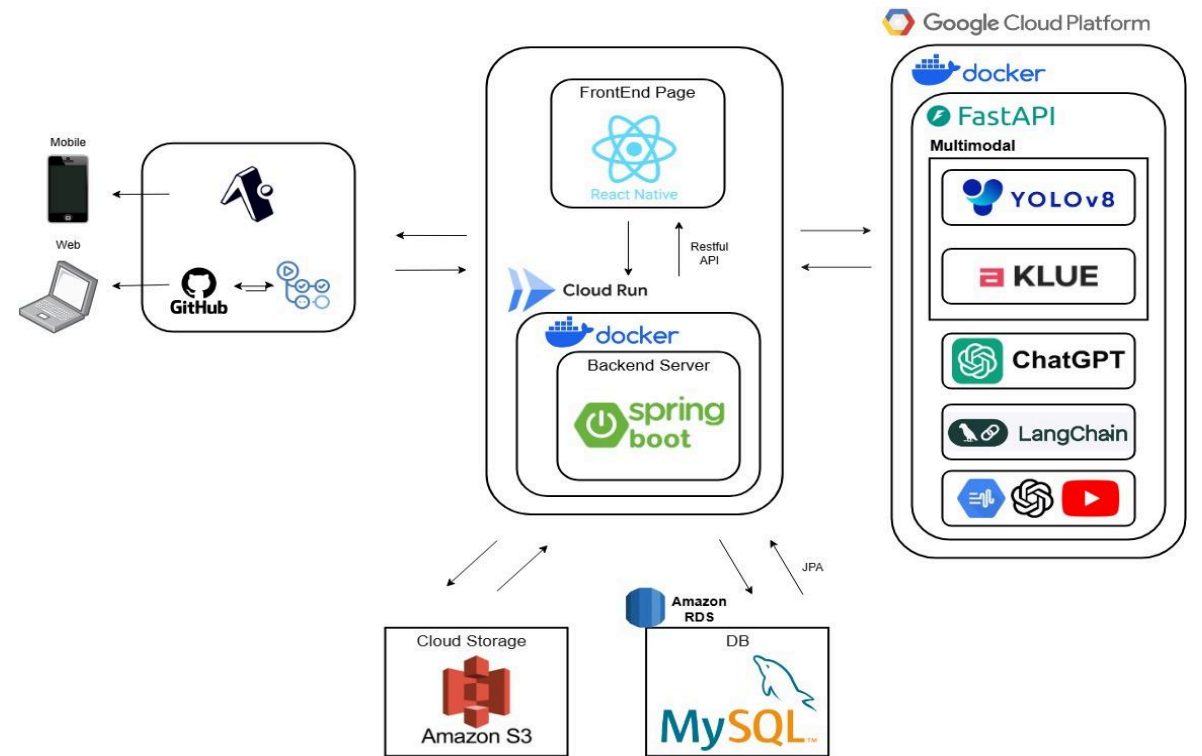


그림9. 시스템 아키텍처

## 1. 프론트엔드

### (1) 상태 관리

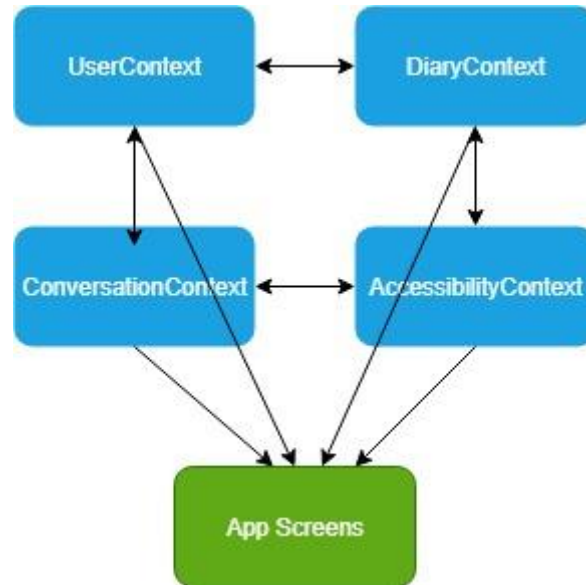


그림10. 상태 관리 구조

시스템에서 사용자의 대화 플로우가 다단계로 이루어지므로, 화면 간 데이터 공유가 잦고 상태를 지속할 필요성이 있다. 따라서 **Props Drilling**을 피하기 위하여, 전역적으로 상태를 관리하도록 하였다.

- **UserContext**: 인증/사용자 정보 관리
- **ConversationContext**: 대화 흐름 관리
- **DiaryContext**: 일기 데이터 관리
- **AccessibilityContext**: 접근성 설정 관리

### (2) 배포

사용자 접근성 확대를 위해, 개발된 프론트엔드 애플리케이션을 **웹(Web)**과 **모바일(Mobile)** 환경에 맞추어 각각 배포하였다.

**웹 (Web)**: GitHub Pages를 통해 정적 웹사이트를 배포하였으며, GitHub Actions를 연동하여 소스코드가 업데이트될 때마다 자동으로 빌드 및 배포가 진행되는 CI/CD 파이프라인을 구축 했다.

---

모바일 (Mobile): iOS 환경은 Expo Go를 통해 실시간 테스트 및 동일 네트워크에서 실행을 지원하며, Android 환경은 APK 파일을 생성하여 사용자가 직접 기기에 설치하는 방식으로 배포 하였다.

## 2. 백엔드

서버 애플리케이션은 Google Cloud Run 위에 Docker 컨테이너로 배포되어, 서버리스 환경에서 자동 스케일링과 트래픽 기반 비용 최적화를 달성하였다. 컨테이너 이미지는 단일 아티팩트로 관리되어 롤백/점진 배포가 용이하며, 무중단으로 리비전을 전환할 수 있다.

- 애플리케이션 레이어 (Spring Boot)

- 백엔드는 Spring Boot 기반의 RESTful API로 구현되어 프론트엔드와 외부 AI 서비스(YOLO, KoBERT)를 연결한다.
- 컨트롤러-서비스-리포지토리 계층을 분리하고, IoC/DI와 AOP를 활용해 의존성 관리와 트랜잭션 경계를 명확히 했다.
- 대화·메시지·감정분석 엔티티는 JPA/Hibernate로 관리되며, 감정 분포/흐름 등 구조적 데이터는 JSON 컬럼으로 저장해 유연성을 확보했다.

- 비동기 파이프라인 (@Async)

- 사용자 대화가 종료되면 대화 요약 → DiaryPlan 생성 → 일기 생성 → 음악 추천을 비동기 작업(@Async)으로 실행하여 응답 지연을 최소화했다.
- 각 작업의 상태는 DB 필드(예: processing\_status, emotion\_flow)로 기록되어 재시도/오류 추적과 운영 모니터링이 용이하다.
- 동일 대화에 대한 중복 실행을 방지하도록 idempotent 키(대화 ID 기반으로 보호한다.

- 모델 연동(External AI Integration)

- 표정 감정(YOLO)·발화 감정(KoBERT) 모델은 별도 FastAPI로 서빙되며, 백엔드는 HTTP 호출 시 타임아웃/재시도(지수 백오프)와 간단한 회로 차단(서킷) 전략으로 안정성을 높였다.
- 모델 품질 신호(신뢰도 등)를 함께 수집하여 멀티모달 결합(품질 가중 late fusion)에 활용한다.

- API 명세

- 대화 관리 API (/api/conversations)

Method	URL	Description	Parameters
POST	/api/conversations/start	대화 시작	userId, questionId
GET	/api/conversations/{conversationId}	대화 세션 조회	conversationId
GET	/api/conversations/user/{userId}	사용자별 대화 세션 목록	userId
GET	/api/conversations/user/{userId}/active	사용자의 활성 대화 세션	userId
GET	/api/conversations/question/{questionId}	질문별 대화 세션 목록	questionId
PUT	/api/conversations/{conversationId}/end	대화 세션 종료	conversationId
GET	/api/conversations/{conversationId}/messages	대화 메시지 목록	conversationId
GET	/api/conversations/{conversationId}/processing-status	처리 상태 확인	conversationId
GET	/api/conversations/{conversationId}/diary	일기 조회	conversationId

- GPT API (/api/gpt)

Method	URL	Description	Parameters
POST	/api/gpt/generate	감정 기반 대화 생성	conversationMessageId
POST	/api/gpt/conversation-summary	대화 내용 요약	conversationId,

			summaryLength
--	--	--	---------------

○ 감정분석 API (/api/emotion-analysis)

Method	URL	Description	Parameters
GET	/api/emotion-analysis/message/{conversationMessageId}	특정 메시지의 감정 분석 결과 조회	conversationMessageId
GET	/api/emotion-analysis/conversation/{conversationId}	대화 세션의 모든 감정 분석 결과 조회	conversationId
GET	/api/emotion-analysis/emotion/{emotionId}	특정 감정으로 필터링된 결과 조회	emotion
DELETE	/api/emotion-analysis/{analysisId}	감정 분석 결과 삭제	analysisId
POST	/api/emotion-analysis/facial	표정 감정 분석 결과 저장	conversationMessageId, facialEmotionData(JSON)
POST	/api/emotion-analysis/speech	말 감정 분석 결과 저장	conversationMessageId, speechEmotionData(JSON)
POST	/api/emotion-analysis/combine	통합 감정 계산 및 저장	conversationMessageId

○ 마이크 API (/api/microphone)

Method	URL	Description	Parameters
GET	/api/microphone/session/{sessionId}	마이크 세션 조회	sessionId



	onId}		
<b>DELETE</b>	/api/microphone/session/{sessionId}	마이크 세션 삭제	sessionId
<b>GET</b>	/api/microphone/session/user/{userId}	사용자별 마이크 세션 목록	userId
<b>POST</b>	/api/microphone/speech/start	음성 녹음 시작	conversationId
<b>POST</b>	/api/microphone/speech/end	음성 녹음 종료	conversationId

○ 카메라 API (/api/camera)

Method	URL	Description	Parameters
<b>GET</b>	/api/camera/session/{sessionId}	카메라 세션 조회	sessionId
<b>DELETE</b>	/api/camera/session/{sessionId}	카메라 세션 삭제	sessionId
<b>GET</b>	/api/camera/session/user/{userId}	사용자별 카메라 세션 목록	userId

○ 대화 메시지 API (/api/conversation-messages)

Method	URL	Description	Parameters
<b>POST</b>	/api/conversation-messages	대화 메시지 생성	conversationId, content, senderType
<b>GET</b>	/api/conversation-messages/{messageId}	특정 메시지 조회	messageId
<b>GET</b>	/api/conversation-messages/conversationId	대화별 메시지 목록	conversationId

	nversation/{conversationId}		
--	-----------------------------	--	--

○ 질문 API (/api/questions)

Method	URL	Descriptions	Parameters
GET	/api/questions	모든 질문 목록 조회	없음
GET	/api/questions/{questionId}	특정 질문 조회	questionId
GET	/api/questions/random	랜덤 질문 조회	없음

○ TTS API (/api/tts)

Method	URI	Descriptions	Parameters
POST	/api/tts/generate	텍스트를 음성으로 변환	Text, voiceName, audioEncoding, speakingRate

○ S3 파일 관리 API (/api/s3)

Method	URL	Description	Parameters
POST	/api/s3/upload	파일 업로드	File, folder
GET	/api/s3/download/{fileId}	파일 다운로드	fileName, folder
DELETE	/appi/s3/delete/{fileId}	파일 삭제	fileId, folder
GET	/api/s3/exists/{fileId}	파일 존재 확인	fileId, folder
POST	/api/s3/upload/diary-cover	일기장 표지 업로드	file

POST	/api/s3/upload/guardian-photo	가족 사진 업로드	File, userId
------	-------------------------------	-----------	--------------

○ 앨범 API(/api/album)

Method	URL	Description	Parameters
GET	/api/album/{conversationId}/photos	사진 목록 조회	conversationId
GET	/api/album/{conversationId}/photos/cover	앨범 표지 사진 조회	conversationId
GET	/api/album/{conversationId}/comments	댓글 목록 조회	conversationId
POST	/api/album/{conversationId}/comments	댓글 추가	conversationId, content, author
POST	/api/album/{conversationId}/photos	사진 추가	conversationId, imageUrl, uploadedBy
PUT	/api/album/{conversationId}/photos/{photoId}/set-cover	앨범 표지 설정	conversationId, photoId

### 3. 데이터베이스

- Amazon RDS의 MySQL을 사용해 운영 DB와 로컬 DB로 환경을 분리하여 운용하였다. 운영 DB는 관리형 백업·모니터링을 통해 안정성을 확보하고, 개발 DB는 기능 검증과 테스트 데이터 투입에 활용한다.

- 정규화와 성능의 균형: 코어 엔터티는 3NF 기반 설계로 중복을 최소화하되, 조회/재사용 빈도가 높은 분석 결과는 JSON 컬럼으로 비정규화하여 읽기 성능과 유연성을 확보했다. 특히 알고리즘이 바뀌어도 기존 데이터와의 호환성이 유지되도록 원본 응답과 파생 지표를 함께 저장한다.
- 대화 종료 시 계산된 감정 흐름(EmotionFlow)을 conversations.emotion\_flow(JSON)에 결과 캐시로 저장하여, 재계산 없이 후속 단계(요약·일기·음악 추천)에서 즉시 활용 가능하도록 했다.

테이블 명	필드	타입	제약	설명
users	id	BIGINT	PK	사용자 고유 식별자
	user_id	VARCHAR	NOT NULL, UNIQUE	사용자 ID
	name	VARCHAR	NOT NULL	사용자 이름
	email	VARCHAR	NOT NULL, UNIQUE	이메일 주소
	phone	VARCHAR		전화번호
	gender	ENUM		성별 (MALE, FEMALE)
	user_type	ENUM	NOT NULL	사용자 타입 (SENIOR, FAMILY)

	profile_image	VARCHAR		프로필 이미지 URL
	created_at	TIMESTAMP	NOT NULL	계정 생성 시간
	updated_at	TIMESTAMP		계정 수정 시간
user_links	<u>id</u>	BIGINT	PK	연결 고유 식별자
	familiy_user_i d	BIGINT	FK	가족 사용자 ID
	senior_user_i d	VARCHAR(25 5)	FK	시니어 사용자 ID
	status	VARCHAR(20 )	NOT NULL	연결 상태
	created_at	TIMESTAMP	NOT NULL	연결 생성 시간
	updated_at	TIMESTAMP		연결 수정 시간
conversations	<u>id</u>	BIGINT	PK	대화 세션 고유 ID

	user_id	BIGINT	FK	사용자 ID
	question_id	BIGINT	FK	선택된 질문 ID
	camera_session_id	VARCHAR(255)	FK	연결된 카메라 세션 ID
	microphone_session_id	VARCHAR(255)	FK	연결된 마이크 세션 ID
	status	ENUM	NOT NULL	대화 상태
	summary	JSON		대화 요약
	diary	TEXT		일기 내용
	processing_status	ENUM		백그라운드 처리 상태
	flow_pattern	VARCHAR(255)		감정 흐름 패턴
	emotion_flow	JSON		감정 흐름 상세 분석 결과
	created_at	DATETIME	NOT NULL	대화 시작 시간
	ended_at	DATETIME		대화 종료 시간
conversation_messages	<u>id</u>	BIGINT	PK	메시지 고유 ID
	conversation_id	BIGINT	FK	대화 세션 ID
	sender_type	ENUM	NOT NULL	발신자 타입 (User, AI)
	content	TEXT	NOT NULL	메시지 내용
	timestamp	DATETIME	NOT NULL	메시지 전송 시간
user_emotion_analysis	<u>id</u>	BIGINT	PK	감정 분석 고유

analysis				ID
	conversation_message_id	BIGINT	FK	대화 메시지 ID
	user_id	BIGINT	FK	사용자 ID
	facial_emotion	JSON		표정 감정 분석 결과
	speech_emotion	JSON		발화 감정 분석 결과
	combined_emotion	VARCHAR(255)		표정 + 음성 통합 감정
	combined_confidence	DOUBLE		통합 감정의 신뢰도
	combined_distribution	JSON		감정별 통합 확률 분포
	analysis_timestamp	DATETIME	NOT NULL	감정 분석 수행 시간
music_recommendations	<u>id</u>	BIGINT	PK	음악 추천 고유 ID
	conversation_id	BIGINT	FK	대화 세션 ID
	user_id	BIGINT	FK	
	title	VARCHAR(255)	NOT NULL	노래 제목
	artist	VARCHAR(255)	NOT NULL	가수 이름
	mood	VARCHAR(255)		음악 분위기
	youtube_link	TEXT		유튜브링크
	youtube_video_id	VARCHAR(255)		유튜브비디오ID

	id			
	created_at	DATETIME		추천 생성 시간
questions	<u>id</u>	BIGINT	PK	질문 고유 ID
	content	TEXT	NOT NULL	질문 내용
camera_sessions	<u>id</u>	BIGINT	PK	세션 고유 ID
	session_id	VARCHAR(255)	NOT NULL, UNIQUE	WebSocket 세션 ID
	user_id	BIGINT	FK	사용자 ID
	status	VARCHAR(255)		카메라 상태
	created_at	DATETIME	NOT NULL	세션 생성 시간
	ended_at	DATETIME		세션 종료 시간
microphone_sessions	<u>id</u>	BIGINT	PK	세션 고유 ID
	session_id	VARCHAR(255)	NOT NULL, UNIQUE	WebSocket 세션 ID
	user_id	BIGINT	FK	사용자 ID
	status	VARCHAR(255)		마이크 상태
	audio_format	VARCHAR(255)		오디오 포맷
	sample_rate	INT		오디오 샘플 레이트
	created_at	DATETIME	NOT NULL	세션 생성 시간
	ended_at	DATETIME		세션 종료 시간
album_photos	<u>id</u>	BIGINT	PK	사진 고유 식별자



	converstaion_id	BIGINT	NOT NULL, FK	대화 ID
	image_url	VARCHAR(500)	NOT NULL	S3 저장 이미지 URL
	is_cover	BOOLEAN	NOT NULL, (DEFAULT FALSE)	앨범 표지 여부 (TRUE/FALSE)
	uploader_user_id	BIGINT	FK	사진 업로더 ID
	created_at	DATETIME	NOT NULL	사진 업로드 일시
	updated_at	DATETIME		사진 수정 일시
album_comments	id	BIGINT	PK	댓글 고유 식별자
	converstion_id	BIGINT	NOT NULL, FK	대화 ID
	content	TEXT	NOT NULL	댓글 내용
	commenter_user_id	BIGINT	FK	댓글 작성자
	created_at	DATETIME	NOT NULL	댓글 생성 일시
	updated_at	DATETIME		댓글 수정 일시

#### 4. 스토리지

본 프로젝트는 **Amazon S3**를 이미지 자산 전용 저장소로 사용한다. 감정별로 분류된 표지 이미지와 가족 업로드 사진을 객체 단위로 관리하며, 버킷 내에 체계적으로 보관한다. 이를 통해 대용량 데이터의 효율적 보관이 가능하며, 필요 시 빠르게 불러올 수 있다.

## 4. 연구 결과 분석 및 평가

### 4.1. 성능 평가

#### 1. 대화 생성

“삶에서 바꾸고 싶은 게 있다면 무엇인가요?”라는 질문에 대한 사용자의 발화가 주어졌을 때, 회상요법 특화 모델과 일반 GPT 모델의 응답을 비교한 결과는 다음과 같다.

사용자 발화	회상요법 특화 모델	일반 GPT 모델
내 삶에서 바꿀 수 있다면, 딸애가 내게 '엄마 밥은 맛없다'고 말했던 그날이었지. 그때는 섭섭한 마음에 아무 말 못하고 돌아섰는데, 그날의 서운함이 지금까지도 가슴에 남아있어.	그날 딸애가 그러면 마음이 상했겠어요. 함께 있던 사람들이 누구였나요? 어떤 일이 있었던 거죠?	그날의 서운함이 아직도 선명하시다니, 그만큼 마음을 다해 밥을 해오신 거겠죠.
내 삶에서 바꿀 수 있다면, 아이들이 다 장성해서 집을 떠나던 날이었지. 텅 빈 방을 보며 내 품을 떠난 아이들 생각에 가슴이 텅 빈 것 같아서 얼마나 쓸쓸했는지 몰라.	가슴이 텅 빈 느낌은 참 힘든 것 같아요. 어린 시절을 떠올리며 아이들과 보낸 소중한 순간이 떠오르나요? 무슨 일이 있었나요?	아이들 키워 다 떠나보내는 날, 부모라면 누구나 겪는 가장 쓸쓸한 순간일 거예요.
내 삶에서 바꿀 수 있다면, 백화점 회전문을 처음 봤던 그날이었지. 겁이 나서 멈춰서 있는데, 뒤에서 사람들이 밀려와서 꿈쩍 못하고 서 있는 게 너무 당황스러웠네.	뒤에서 사람들이 밀려와 순간 당황스러우셨겠어요. 누구와 함께 그날을 보냈나요? 어떤 소리와 냄새가 났나요?	회전문이 사람을 잡을 때가 있죠. 다들 한 번쯤은 그 앞에서 얼어붙어 본 경험 있을 거예요.

표10. 대화 생성 관련 응답 비교

위 예시에서 확인되듯, 본 연구의 회상요법 기반 모델은 회상요법(Reminiscent Therapy)의 원칙을 반영하여 사용자가 스스로 과거 경험을 자연스럽게 이어갈 수 있도록 질문을

제시한다. 또한 발화 단위에서 산출된 감정값을 반영함으로써, 단순한 응답 생성에 그치지 않고 사용자의 정서를 세밀하게 공감하는 특징을 보였다. 이를 통해 대화는 단순한 질의응답을 넘어 실제 회상 대화에 가까운 자연스럽고 몰입도 높은 흐름을 형성할 수 있었다.

## 2. 일기 생성

사용자 발화가 주어졌을 때, 회상요법 특화 일기 생성 모델과 일반 GPT 모델의 결과는 다음과 같다. 회상요법 특화 모델은 사용자-AI 대화 내용과 감정 흐름 정보를 함께 반영하여 일기를 생성한 반면, 일반 GPT 모델은 대화 내용만을 입력으로 사용하였다.

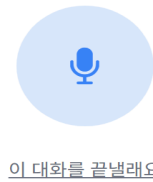
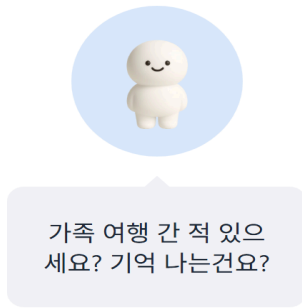
회상요법 특화 모델	일반 gpt 모델
<p>고등학교 시절, 중요한 수학 시험을 망쳤다.. 시험 준비를 하면서 자신감이 있었지만, 시험장에서 머리가 하얘졌다. 계산 실수를 반복하며, <b>결과에 속상함이 밀려왔다.</b></p> <p>시험이 끝난 후, 나는 그 결과를 받아들이기 힘들었다. 학교에서의 <b>긴장감이 여전히 남아 있었고, 불안한 마음이 가득했다.</b> 하지만 그 경험을 통해 실수에서 교훈을 얻었다는 생각이 들었다.</p> <p>이제는 실패도 성장의 일부라는 깨달음이 있다. 그때의 아쉬움이 나를 더 강하게 만들었다고 느낀다. 이런 경험에 감사하며, 앞으로도 계속 배우고 성장할 것이다.</p>	<p>고등학교 시절에 중요한 수학 시험을 본 날이 아직도 기억난다. 그때는 나름대로 열심히 준비했다고 생각했는데, 막상 시험장에 들어가니 갑자기 머릿속이 하얘져서 평소에 풀던 문제들도 잘 안 풀리고 계산 실수도 몇 번이나 하게 되었다. 결국 시험지를 제출하고 나올 때는 이미 잘못 봤다는 걸 스스로 알고 있었고, 결과를 확인했을 때는 역시나 점수가 기대만큼 나오지 않아 속상한 마음이 들었다. 그래도 그 경험을 지나고 나니 이후로는 문제를 풀 때 조금 더 신중하게 계산을 확인하게 되었고, 실수를 줄이는 법을 조금은 알게 된 것 같다.</p>
명절에 할머니 댁에서 가족이 모여 함께	명절이 되면 늘 온 가족이 할머니 댁에

<p>식사한 적이 있었다. 큰상에 다양한 음식이 가득 차려졌고, 모두가 웃으며 나누던 순간이 정말 따뜻했다. 웃음소리와 아이들 떠드는 소리가 집안을 가득 메웠었다.</p> <p>하지만 설거지와 뒷정리할 때는 힘든 순간도 있었다. 그때는 모두가 피곤해 보였고, 서로의 얼굴에 지친 기색이 엿보였다. 그래도 <b>가족과 함께하는 시간이 가장 소중하다는 생각이 들었다.</b> 우리가 함께 나눈 음식과 웃음은 그 어떤 것보다 값진 기억이었기 때문이다.</p> <p>이런 시간을 통해 가족의 소중함을 다시 한 번 깨달았다. 함께 웃고, 나누며 보낸 시간이 <b>내 마음을 따뜻하게 해주었다.</b> 앞으로도 이런 소중한 순간들을 계속 간직하고 싶다.</p>	<p>모였다. 커다란 상 위에는 전과 떡국, 나물 같은 음식들이 가득 차 있었고, 방 한쪽에서는 삼촌들이 고스톱을 치며 웃고 떠드는 소리가 들려왔다. 아이들은 집안을 뛰어다니며 장난을 치고, 어른들은 이야기꽃을 피우면서 집안이 온통 활기로 가득했다. 식사가 끝나면 설거지와 뒷정리로 조금은 힘들었지만, 그래도 모두 함께 있어서 그 시간마저 즐겁게 느껴졌다.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

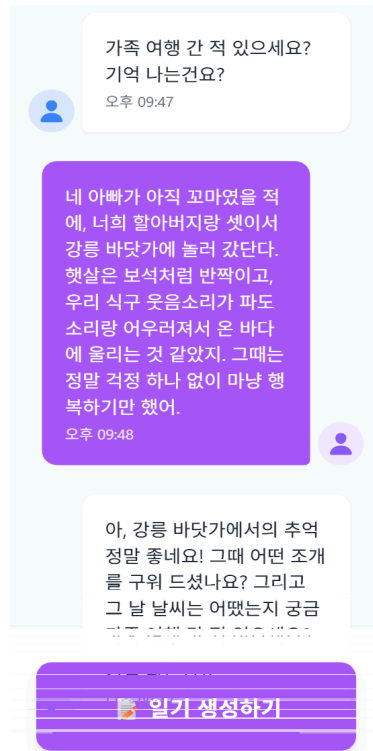
표11. 일기 생성 관련 응답 비교

위 예시에서 확인되듯, 회상요법 특화 모델은 단순히 대화 내용을 요약하는 데 그치지 않고, 발화 단위 감정과 대화의 흐름을 반영하여 문장을 생성하였다. 이를 통해 모델은 사용자의 정서를 세밀하게 공감하면서도, 회상 경험을 자연스럽게 이어갈 수 있는 문맥을 형성하였다.

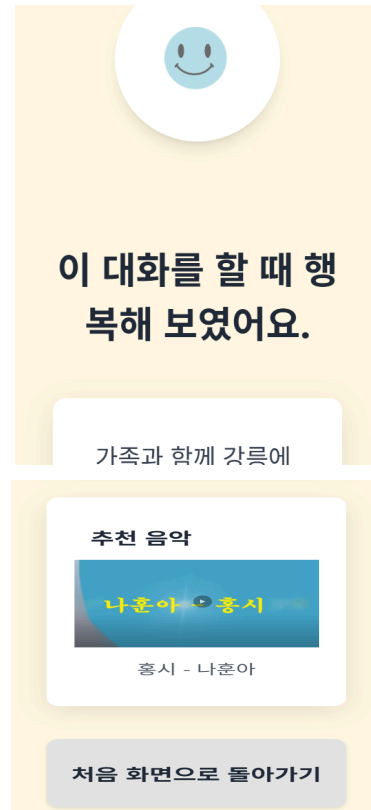
## 4.2. 시스템 결과물



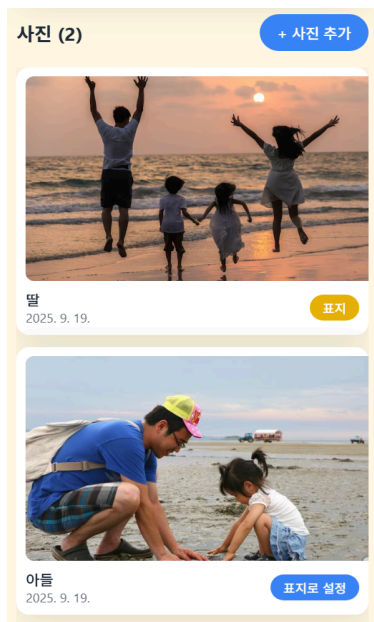
AI와 대화 화면



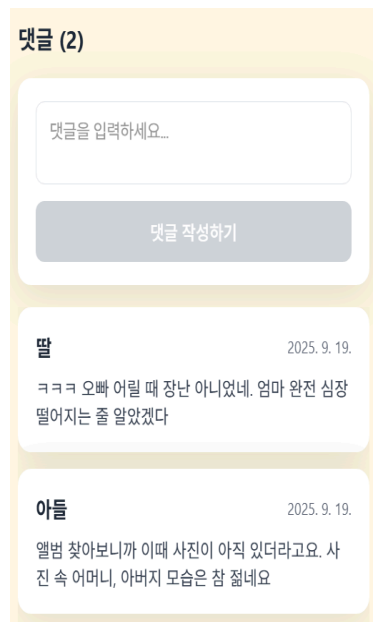
대화 내역 확인



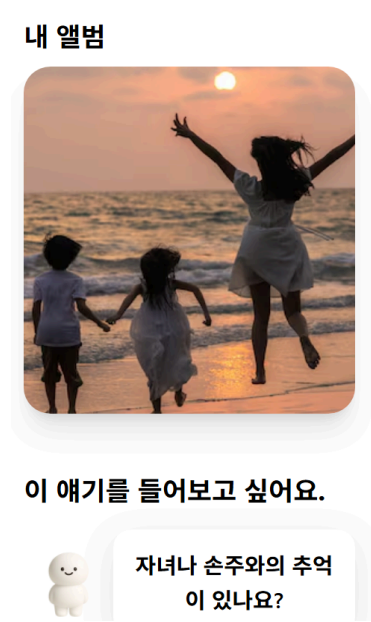
생성된 일기 페이지



보호자의 사진 등록



보호자의 댓글 작성



앨범 표지가 설정된 메인 화면

### 4.3. 개발 일정 및 역할 분담

#### 1. 개발 일정

구분	추진 목표	5	6	7	8	9
기획	기능 구체화					
	DB 설계					
AI 모델 개발	텍스트 감정 분류 모델					
	표정 감정 분석 모델					
	감정 통합 로직 구현					
	LLM 구축					
백엔드 개발	REST API 설계					
	기능 개발					
프론트 개발	UI/UX 화면 설계					
	기능 개발					
배포 및 개선	AWS 및 Docker 활용 배포					

#### 2. 역할 분담

학번	성명	역할
202155514	김나림	아이디어 기획 KLUE-BERT fine-tuning 기반 시니어 표정 감정 분석 모델 구축 프론트엔드 개발 및 배포 로그인 기능, 시니어-가족 연결 관계 개발
202155540	김채현	YOLO 기반 시니어 표정 감정 분석 모델 구축 및 배포 감정 통합 로직 구현 대화의 감정 흐름 로직 구현 LLM 구축 백엔드 개발 및 배포 DB 구축 및 배포

---

## 5. 결론 및 향후 연구 방향

### 5.1. 결론

본 연구는 급속한 고령화 사회에서 시니어의 정서적 고립감 해소와 치매 예방을 위한 실질적인 AI 솔루션을 제시하고자 하였다. 이를 위해, 사용자의 표정과 텍스트를 동시에 분석하는 멀티모달(Multi-modal) 감정 인식 모델을 성공적으로 개발하였으며, 이를 기반으로 회상 요법을 수행하는 AI 기반 추억 앨범 및 치매 예방 서비스를 구축하였다.

최종적으로 개발된 웹 서비스는 소셜 로그인, 사용자 유형 구분, 자동 일기 생성 등 사용자 편의성을 극대화한 기능들을 포함하고 있으며, 이는 기술적 성과를 넘어 실제 사용자가 체감할 수 있는 가치를 제공한다는 점에서 본 연구의 가장 큰 의의를 찾을 수 있다.

### 5.2. 향후 연구 방향

본 연구의 성과를 바탕으로, 다음과 같은 방향으로 연구를 확장하고 서비스를 고도화할 수 있을 것이다.

**1. 음성 감정 분석(Prosody Analysis) 추가:** 현재의 텍스트-표정 기반 멀티모달 시스템에 사용자의 목소리 톤, 억양, 말의 속도 등 음성적 특징(Prosody)을 분석하는 모델을 추가하여, 3가지 데이터를 종합적으로 판단하는 고도화된 감정 인식 시스템으로 발전시킬 수 있다. 이는 사용자의 감정 상태를 더욱 정밀하게 파악하는 데 기여할 것이다.

**2. 개인화된 대화 모델(Personalized LLM):** 현재의 GPT 기반 대화 모델을 넘어, 사용자 개인의 대화 기록 데이터를 바탕으로 모델을 추가 미세조정(Fine-tuning) 하여, 사용자의 말투, 자주 사용하는 단어, 과거 대화 내용을 기억하는 등 개인화된 대화 모델을 구축할 수 있다. 이는 AI와의 유대감을 형성하고 더욱 깊이 있는 교감을 가능하게 할 것이다.

---

## 6. 참고 문헌

[1] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, and E. Cambria, "DialogueRNN: An Attentive RNN for Emotion Detection in Conversations," *Proc. of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 6818–6825, 2019.

[2] W. Jiao, H. Yang, I. King, and M. R. Lyu, "HiGRU: Hierarchical Gated Recurrent Units for Utterance-level Emotion Recognition," *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pp. 397–406, 2019.

[3] W. Jiao, H. Yang, I. King, and M. R. Lyu, "HiGRU: Hierarchical Gated Recurrent Units for Utterance-level Emotion Recognition," *NAACL-HLT 2019*, pp. 397–406, 2019.

[4] J. Deng, W. Jiao, H. Yang, et al., "Context-Dependent Emotion Recognition Using Multimodal Fusion of Facial Expression and Speech," *IEEE Transactions on Affective Computing*, vol. 12, no. 4, pp. 828–840, 2021.

[5] A. Pascual, J. Serrano, and S. Escalera, "Light-FER: A Lightweight Facial Expression Recognition Network for Edge Devices," *Pattern Recognition Letters*, vol. 145, pp. 55–61, 2021.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL-HLT 2019*, pp. 4171–4186, 2019.

[7] S. Park, J. Lee, et al., "KoBERT: A Pre-trained Korean BERT Model for Natural Language Processing," *arXiv preprint arXiv:2008.03979*, 2020.

[8] R. Ekman and W. V. Friesen, "Facial Action Coding System: A Technique for the Measurement of Facial Movement," *Consulting Psychologists Press*, 1978.