

어린이가 작성한 이야기 기반 그림 동화 생성기



부산대학교 정보컴퓨터공학부

팀 스토리픽셀

201824527 여현준

201824551 이승민

지도교수 탁성우

목차

1. 요구 조건 제약 사항 분석에 대한 수정 사항	3
1.1 과제 목표	3
1.2 제약 사항 1	4
1.3 제약 사항 2	4
1.4 제약 사항 3	5
2. 설계 상세화	6
2.1 구성도	6
2.2 프론트 엔드	7
2.3 백 엔드	8
2.4 데이터 베이스 구성	9
2.5 AI API 설정 및 파인튜닝	10
2.6 유스케이스	13
3. 과제 추진 계획 갱신	14
4. 구성원 역할별 진척 현황	15
5. 과제 수행 내용 및 중간 결과	16
5.1 수행내용	16

1. 요구 조건 제약 사항 분석에 대한 수정 사항

1.1 과제 목표

본 과제의 궁극적 목표는 **어린이가 작성한 이야기를 바탕으로 자동으로 그림 동화를 생성하는 인공지능 기반 시스템**을 개발하는 것이다.

첫 번째, 어린이가 쓴 이야기의 데이터를 자연어로 입력받을 수 있는 인터페이스를 설계한다. 어린이가 자유롭게 이야기를 작성할 수 있는 직관적인 입력 환경을 제공해야 한다.

두 번째, 입력된 이야기의 핵심 내용을 이해하기 위한 자연어 처리 모듈을 구현한다. 어린이가 자연어로 작성한 이야기에서 핵심 단어를 정확하게 추출하고, 문장 구조 및 어조 분석을 통해 이야기 속에 담긴 감정이나 분위기를 효과적으로 파악해야 한다.

세 번째, 분석한 내용을 바탕으로 장면별 이미지 생성이 가능한 txt-2-img AI 모델을 적용한다. 어린이가 입력한 내용을 바탕으로 어울리는 스타일이나 색감을 적용한다.

네 번째, 이야기 흐름에 맞게 이미지와 텍스트를 배치한다. 원본 텍스트와 이미지를 조합하여 페이지 형식의 동화 콘텐츠로 자동 구성한다.

1.2 제약 사항 1

1.2.1 DALL-E의 파인튜닝 미지원

과제 진행 동안 claude, ChatGPT, Gemini 등의 LLM과, Stable Diffusion, DALL-E, Gemini의 txt-2-img 모델들을 테스트 해본 결과 본 프로젝트에 필요한 요소인 스토리 및 감정의 이해는 Gemini가, 프롬프트를 바탕으로 만드는 그림의 정확도 및 퀄리티는 DALL-E가 가장 좋은 결과를 보여주었으므로 이를 활용할 예정이다.

그러나 Gemini는 Google Cloud Vertex AI 또는 Google AI Studio를 활용한 파인튜닝이 가능 한 반면, OpenAI의 DALL-E는 따로 사용자가 직접 모델의 가중치를 조정하거나 학습데이터를 재학습 시키는 기능을 제공하지 않는다.

따라서 Gemini를 통해 프롬프트 엔지니어링 또는 이미지 인페런싱 등 우회적인 방법을 통해 원하는 결과를 도출 예정이고, 결과를 테스트 해본 후 프로젝트에 적합하지 않다면 Stable Diffusion 등의 파인튜닝을 지원하는 다른 API의 활용을 검토 예정이다.

1.3 제약 사항 2

1.3.1 파인튜닝용 자료의 부족 및 입수 경로 확보의 어려움

파인튜닝의 핵심은 고품질 학습 데이터셋의 준비이다. 본 프로젝트의 경우 “어린이의 일기 및 그림”, 그리고 “그림 동화”가 그 예시인데, 실제 어린이들의 일기나 그림을 모아 놓은 자료를 찾기가 어렵고, 입수 경로를 확보하기 힘들다는 제약 사항이 있다. 또한, 동화책 등의 저작권으로 인해 파인튜닝에 어려움이 있다.

이에 주변 지인이나 유치원 등의 교육기관에 협력을 요청할 계획이며, 저작권이 없는 동화 및 이미지를 계속 수집하여 학습 시킬 예정이다.

1.4 제약 사항 3

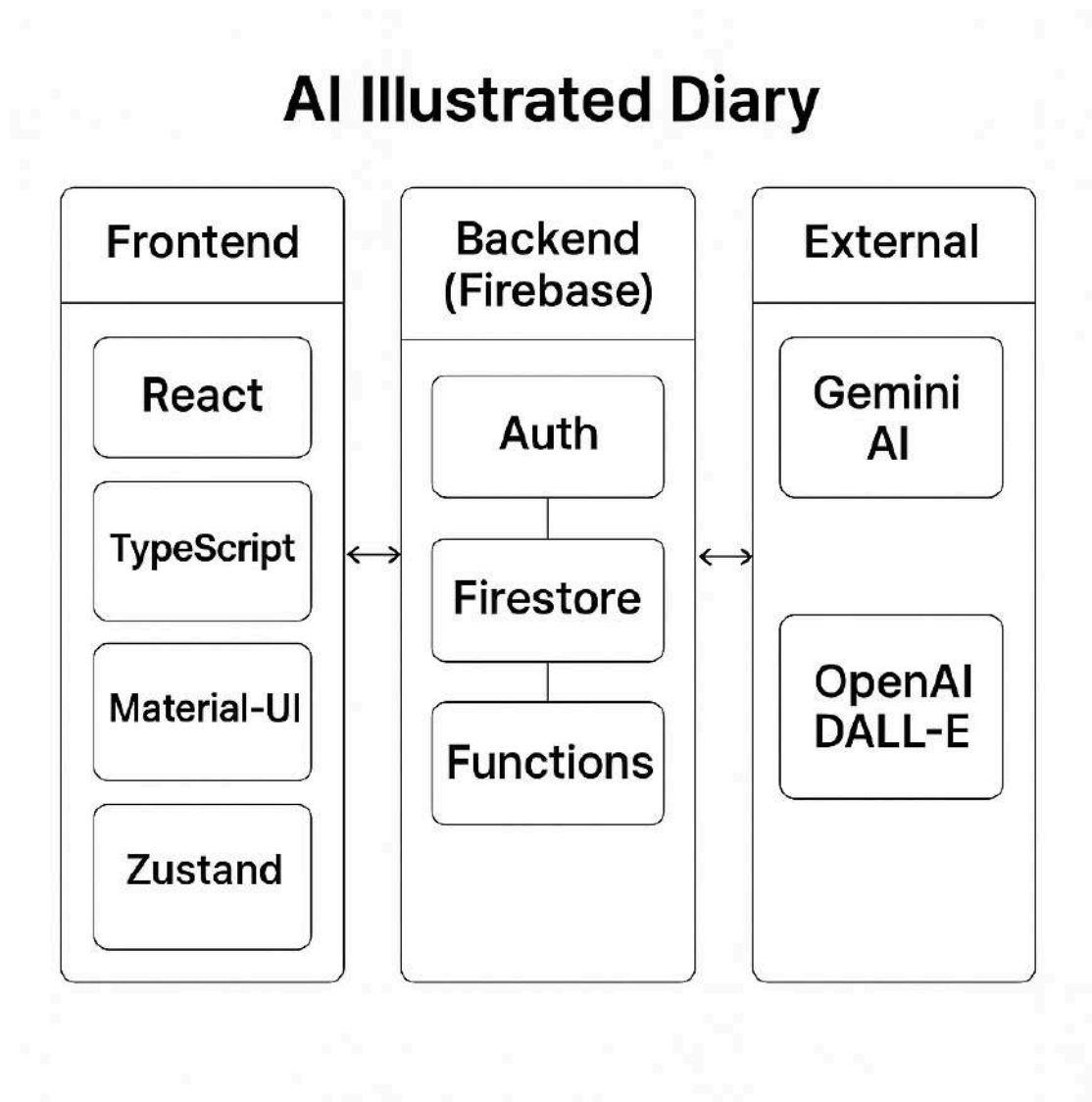
1.4.1 Firebase 및 DALL-E(ChatGPT) 활용에 따른 비용 발생

현재 개발 테스트 단계에서는 이용자가 개발자 외에는 존재 하지 않아 비용이 크지 않지만, 추후 배포를 통해 사용자가 늘어나면 그에 대한 서비스 사용량이 증가해 높은 비용이 발생할 가능성이 있다.

DALL-E의 경우 이미지 생성 횟수에 제한이 있고, 이를 해결하기 위해 비용이 발생한다.

2. 설계 상세화

2.1 구성도



2.2 프론트 엔드

2.2.1 핵심 프레임워크

기술	버전	용도	선택이유
Next.js	14.0.0	React 프레임워크	SSR/SSG, APP Router, 최적화
React.js	18	UI 라이브러리	컴포넌트 기반 개발
Type Script	5	정적 타입	타입 안정성, 개발 생산성

2.2.2 UI/UX 라이브러리

기술	버전	용도	선택이유
Material-UI	5.14.0	UI 컴포넌트	일관된 디자인, 접근성
Emotion	11.11.0	CSS-in-JS	동적 스타일링
Framer Motion	12.13.3	애니메이션	부드러운 페이지 전환
MUI Icons	5.18.0	아이콘	Material Design 아이콘

2.2.3 상태 관리

기술	버전	용도
Zustand	4.4.0	전역 상태 관리

2.2.4 유틸리티

기술	버전	용도
date-fns	4.1.0	날짜 처리
MUI Date Pickers	8.8.0	날짜 선택

2.3 백 엔드

2.3.1 핵심 백엔드

기술	버전	용도	선택이유
Firebase Auth	11.10.0	인증 시스템	Google OAuth, 보안
Firebase Firestore	11.10.0	NoSQL 데이터베이스	실시간 동기화, 확장성
Next.js API Routes	14.0.0	서버 API	서버리스 함수

2.3.2 AI 서비스

서비스	라이브러리	버전	용도
Google Gemini	@google/generative -ai	0.24.1	텍스트 분석
OpenAI DALL-E	openai	5.9.0	이미지 생성

2.4 데이터 베이스 구성

2.4.1 데이터베이스 인덱스

인덱스	필드	정렬	용도
'userId_createdAt_desc'	'userId' (오름차순) + 'createdAt' (내림차순)	복합 인덱스	사용자별 일기 목록 조회
'userId_category'	'userId' + 'category'	복합 인덱스	카테고리별 필터링
'userId_diaryDate'	'userId' + 'diaryDate'	복합 인덱스	날짜별 필터링

2.5 AI API 설정 및 파인튜닝

2.5.1 API 및 프롬프트 설정

클라이언트 (use client)

↓ API 요청

Next.js API Route (/api/gemini, /api/openai)

↓ 서버 사이드 처리

AI 서비스 (Gemini, OpenAI)

1. Gemini AI 기능

텍스트 분석: 일기 내용, 동화 스토리 분석

프롬프트 엔지니어링: 한국어 최적화된 프롬프트

2. OpenAI DALL-E 기능

이미지 생성: 동화 일러스트, 일기 그림 생성

```

async function callGPTAPI(
  apiKey: string,
  prompt: string,
  image?: string,
  maxTokens: number = 100
): Promise<string> {
  const model = process.env.NEXT_PUBLIC_GPT_MODEL || 'gpt-3.5-turbo';

  let requestBody: any = {
    content: [
      {
        parts: [
          {
            text: prompt,
          }
        ]
      }
    ],
    generationConfig: {
      maxOutputTokens: maxTokens,
      temperature: 0.8,
      topP: 0.9,
      topK: 40
    }
  };

  // 이미지 기반 생성은 필수 필드인 image
  if (image) {
    requestBody.content[0].parts.push({
      inlineData: {
        mimeType: 'image/jpeg',
        data: image
      }
    });
  }

  const response = await fetch(
    `https://generativelanguage.googleapis.com/v1beta/models/${model}:generateContent?key=${apiKey}`,
    {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(requestBody),
    }
  );
}

```

```

// API 호출을 위한 함수
// 프롬프트와 이미지 URL을 전달합니다
export const FAIRYTALE_API_PROMPT = `
당신은 훌륭한 이야기꾼입니다.
주어진 프롬프트를 바탕으로 그림 동화 이야기를 작성해주세요.
`;

// 프롬프트를 처리하는 함수
1. 각 문장은 독립적인 그림으로 표현 가능해야 합니다.
2. 각 문장은 그림이 들어갈 수 있도록 넓게 남겨주세요.
3. 각 문장은 명확한 시작과 끝을 포함해야 합니다.
4. 각 문장은 문장이 될 인물의 심정을 잘 표현해야 합니다.
5. 문장의 흐름이 자연스럽게 흘러야 합니다.
6. 문장 시작에는 그림장면을 묘사하는 지시어가 있어야 합니다.
7. 각 문장의 끝에는 문장을 마무리하고 다음 문장을 포함해야 합니다.
8. content에는 문장의 시작과 끝을, originalText에는 해당 문장의 내용을 포함해주세요.

문장은 반드시 다음 7000 문자를 넘지 않도록 작성해주세요.
`;

// API 호출을 위한 함수
// 프롬프트와 이미지 URL을 전달합니다
export const FAIRYTALE_API_PROMPT = `
당신은 훌륭한 이야기꾼입니다.
주어진 프롬프트를 바탕으로 그림 동화 이야기를 작성해주세요.
`;

// 프롬프트를 처리하는 함수
1. 각 문장은 독립적인 그림으로 표현 가능해야 합니다.
2. 각 문장은 그림이 들어갈 수 있도록 넓게 남겨주세요.
3. 각 문장은 명확한 시작과 끝을 포함해야 합니다.
4. 각 문장은 문장이 될 인물의 심정을 잘 표현해야 합니다.
5. 문장의 흐름이 자연스럽게 흘러야 합니다.
6. 문장 시작에는 그림장면을 묘사하는 지시어가 있어야 합니다.
7. 각 문장의 끝에는 문장을 마무리하고 다음 문장을 포함해야 합니다.
8. content에는 문장의 시작과 끝을, originalText에는 해당 문장의 내용을 포함해주세요.

문장은 반드시 다음 7000 문자를 넘지 않도록 작성해주세요.
`;

```

기본 API 설정 코드와 프로젝트에 사용된 프롬프트의 일부

2.5.2 파인튜닝 계획

파인튜닝의 목적

어린이의 언어 이해를 증진하고, 감정 및 분위기 추출 정확도 향상.

상황 및 사물 정보 정교화 후 DALL-E 친화적 프롬프트 생성

파인튜닝 데이터셋 구축

실제 어린이의 일기를 수집하여 각 일기에 해당하는 분위기 및 감정, 그리고 DALL-E용

프롬프트를 직접 작성 하여 JSON 형태로 생성한다. 데이터셋의 규모는 클 수록 좋다.

DALL-E에 작성할 프롬프트를 테스트 하여 적합한 그림이 나오는지 테스트 후 수정 과정을 거친다.

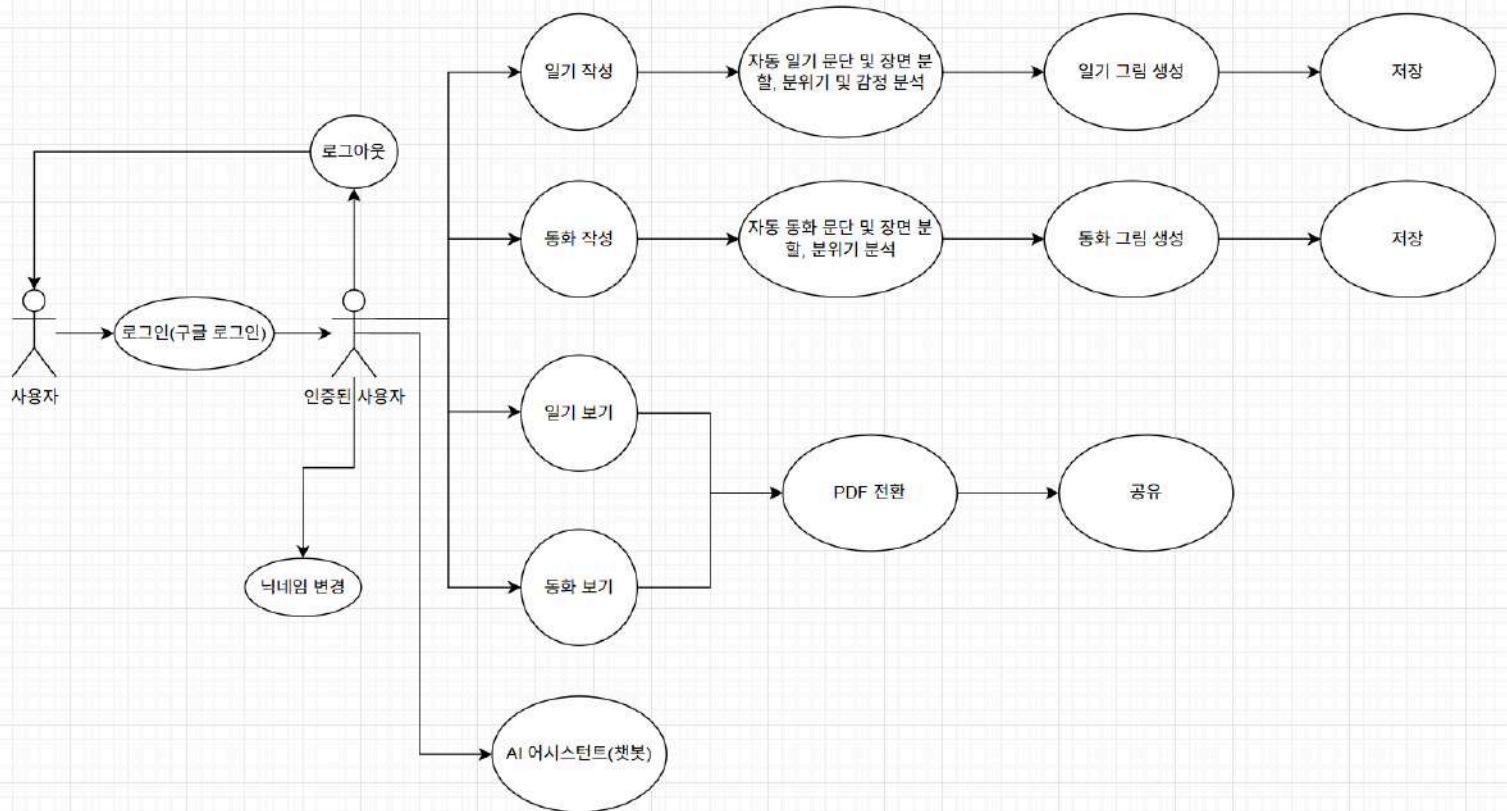
기반 모델: Gemini 2.5 Flash

파인튜닝 방법: 감독학습(Supervised Fine-tuning)

하이퍼파라미터 설정: 데이터셋의 특성과 모델의 학습 진행 상황에 따라 에포크와 학습률, 어댑터 크기 등을 조정

모니터링 및 평가: API 대시보드를 통해 파인튜닝 진행 상황을 모니터링 하고 성능 검증

2.6 유스케이스



3. 과제 추진 계획 갱신

5월		6월				7월					8월				9월			
3	4	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4
요구사항 분석 및 기능 정의, 초기기획, 환경 구축																		
			LLM 및 txt-2-img 검토 및 기본 API와 프롬프트 설정															
							중간 보고서 작성											
							프론트엔드 및 백엔드 개발 및 파인튜닝											
															테스트 및 수정			
																배 포		
																최종 보고서 작성		

4. 구성원 역할별 진척 현황

이름	진척 현황
여현준	일기/동화 CRUD 기능 AI 장면 분할 및 이미지 생성 적용 반응형 UI/UX Firebase 백엔드
이승민	AI api 관련 설정 Firebase 데이터베이스 구조 정리 Google OAuth 인증 AI 장면 분할 및 이미지 생성 파인튜닝 및 프롬프트 작성

5. 과제 수행 내용 및 중간 결과

5.1 수행 내용

기능 영역	완료율	주요 성과
프로젝트 기반 구조	100%	Next.js 14 프로젝트 설정, TypeScript 설정
인증 시스템	100%	Firebase Auth 연동, 로그인/로그아웃 구현
일기, 동화 관리	70%	일기 캘린더 뷰 구현
일기, 동화 생성	70%	작성 및 저장 기능 구현
UI/UX	80%	Material-UI 컴포넌트, 반응형 디자인
AI	80%	Gemini, DALL-E API 연동
상태 관리	80%	Zustand 스토어 구현
데이터베이스	80%	Firestore 구조 정립

5.2 해결해야 할 과제

이슈	상태
모바일 반응형 개선	진행 중
AI 이미지 생성 속도	진행 중
Firebase 비용 최적화	진행 중
브라우저 호환성	대기

