

## 소프트웨어공학 Teamproject

---

팀 역할 및 개인 기여도 평가

2팀 이예린 김재이 김아리

이예린 – Team Leader / Product Manager

주요 역할(Role)

프로젝트 총괄 기획 및 PM (Product Manager)

요구사항 분석 및 서비스 설계

발표 자료(PPT) 제작 및 팀 운영

서비스 디자인 참여

요구사항 분석 & 데이터 구조 설계

Firebase 기반 저장·처리 구조 설계

SDLC 문서 및 사용 매뉴얼 전체 작성

기능 구현 일부 참여 및 테스트 주도

기여한 상세 내용(Contributions)

1) 프로젝트 기획 및 서비스 전략 수립

◆ 핵심 컨셉 정의

“월간 달력 기반으로 사진 + 글을 한눈에 확인하는 일기장”이라는 서비스 방향 최초 제안

기존 일기 앱과의 차별성 분석 후 UX 매력 요소 도출

커플 모드 기능(공유 일기) 컨셉 주도

◆ 사용자 가치 분석

대학생 사용자 페르소나 정의

글·사진·감정 기록 3요소 통합 경험 설계

“한눈에 돌아보기” UX의 핵심 기능 우선순위 설정(MoSCoW 분석 진행)

◆ 전체 사용자 플로우 설계

로그인 → 홈 → 달력 → 일기 작성/수정 → 커플 공유 흐름 설계

팀원과 기능 흐름을 맞추기 위한 UI/UX 프로세스 문서화

2) 요구사항 분석 및 SDLC 기반 프로젝트 구조 설계

◆ 요구사항 명세서 작성

기능 요구사항(FR-001 ~ FR-00X) 전체 문서화

시스템 요구사항(성능·안정성·보안) 정의

제약조건 및 비기능 요구사항 정리

◆ 데이터 구조 설계

일기 데이터 모델(day, imageURL, text, emotion 등) 정의

커플 공유 구조(coupleId, sharedDiaryList) 기획

사용자(User) 구조와 Firebase Auth 기반 연결 방식 설계

## 데이터 관계 다이어그램(DRG/ERD) 작성

### ◆ 오류/예외 상황 설계

사진 저장 실패, 인터넷 끊김, 로그인 세션 문제 등 예외 케이스 정의  
후속 유지보수 측면에서 필요한 처리 로직 문서화

### 3) Firebase 연동 기획 및 기능 일부 구현

### ◆ Firebase 설계 담당

Storage, Database, Auth 연동 구조 초안 설계

디렉토리 구조 및 경로 명명 규칙 수립

데이터 저장 방식(이미지는 Storage, 메타데이터는 DB) 구조화

### ◆ Firebase 실사용 테스트

DB 구조 정상 동작 여부 검증

크로스 디바이스 저장/조회 테스트 진행

커플 공유 DB 데이터 sync 구조 검토

### ◆ 기능 구현 일부 참여

Firebase 초기 설정 지원

로그인 state 관리 로직 협업

데이터 구조 재정의 과정에서의 refactoring 제안 및 적용

### 4) SDLC 문서 및 프로젝트 산출물 전체 작성

### ◆ SDLC 문서 총괄

프로젝트 개요서

요구사항 명세서

분석·설계 문서

UI/UX 설계 문서

테스트 보고서(시나리오, 과정, 결과 포함)

유지보수 계획서

사용자 매뉴얼

### ◆ 시각 자료 제작

데이터 모델 구조도

시스템 아키텍처 다이어그램

기능 흐름도 및 화면 구성도

유지보수/테스트 시각 흐름도

문서 기반으로 PPT에 들어가는 요소까지 제작.

### 5) 발표 자료(PPT) 제작 및 프레젠테이션 구조 설계

◆ PPT 제작 전체 담당

개요 → 문제 정의 → 요구사항 → 설계 → 구현 → 테스트 → 회고 흐름 구성

UI 시연, 기능 소개, 데이터 모델 시각화 등 전체 슬라이드 디자인

팀원 역할·기여도 페이지 정리

발표 운영

발표 흐름 총괄

팀원 발표 분량 조율

질의응답 대비 자료 준비

6) 기능 테스트 & 사용자 관점 품질 관리

◆ 테스트 시나리오 작성

일기 CRUD, 커플 모드, 이미지 저장, 로그인 등 기능별 테스트 항목 작성

오류 상황 기반 테스트 케이스 포함

◆ 실제 테스트 주도

테스트 중 발생한 오류 재현 및 원인 정리

UI/UX 문제 발견 시 개선 포인트 제안

테스트 결과를 문서화하여 보고서에 반영

7) 프로젝트 운영 및 팀 커뮤니케이션 관리

◆ 팀 회의 리딩

개발 일정 및 작업 분배

산출물 관리 및 마감 일정 조율

GitHub 충돌 문제 시 해결 흐름 안내

팀원들 의견 중재

기능 우선순위 변경 및 범위 조정 역할 수행

김재이 - Full-Stack Developer

주요 역할(Role)

풀스택 개발(Front-end & Back-end) 전반 담당

Flutter 기반 프론트엔드 개발 및 UI 구현

Firebase 기반 백엔드(Realtime Database · Auth · Storage) 연동

전체 데이터 구조 설계 및 CRUD 기능 개발

앱 빌드 환경 구성(Android) 및 오류 해결

기능 테스트 및 성능 개선 참여

시스템 아키텍처 설계 및 기술 스택 선정

성능 최적화 (로딩 시간 60% 단축)

### 1) Firebase 기반 데이터 구조 설계 & 연동 구현

#### ◆ 데이터 모델 구조 설계

일기 데이터(day, emotion, text, imageURL) 구조 필드 정의  
커플 공유 데이터(coupleId, sharedDiaries) 구조 논리 설계  
사용자(User) 데이터 흐름과 인증(Auth) 연계 설계  
모든 기능이 충돌 없이 동작하도록 데이터 참조 경로 구조화

#### ◆ Firebase Authentication 연동

구글 로그인 기능 연동  
사용자 uid 기반 개인 데이터 저장 흐름 제작  
로그인/로그아웃 상태 변화에 따른 화면 전환 처리

#### ◆ Firebase Realtime Database 구현

일기 CRUD(Create · Read · Update · Delete) 구현  
날짜별 일기 목록 조회 기능 구성  
커플 관계 데이터 read/write 로직 개발  
실시간 변경 감지(onValue) 적용으로 UI 자동 갱신

#### ◆ Firebase Storage 이미지 처리

이미지 파일 업로드 경로 설계(date/userId 기반)  
업로드 → 다운로드 URL 발급 → DB 저장까지 전체 프로세스 구현  
Storage 권한 오류 해결 및 경로 규칙 설정

### 2) Flutter UI 개발 및 화면 설계 구현

#### ◆ 프론트엔드 전체 구조 세팅

lib/screens, widgets, services 기반의 구조화된 폴더링 구성  
화면 간 네비게이션 흐름 설계 및 라우팅 구현

#### ◆ 주요 화면 구현

달력(Calendar) 탭 UI 구성 및 썸네일 표시 로직 협업  
일기 작성 화면에서 텍스트 입력 + 이미지 선택 기능 개발  
일기 상세보기 화면에서 Firebase 데이터 실시간 반영  
감정 선택 기능 UI 요소 제작(라디오 버튼/아이콘 기반)

#### ◆ 인터랙션 개선 및 사용자 경험(UX) 보완

입력 시 키보드 오버레이 문제 해결  
로딩 인디케이터(Spinner) 처리  
오류 메시지/성공 메시지 전송 UI 구현

### 3) 핵심 기능 개발 프로세스 및 문제 해결

#### ◆ 데이터 동기화 문제 해결

초기 개발 단계에서 특정 날짜의 일기가 중복 출력되는 문제 발생

→ Firebase query 정렬(orderByChild) 및 조건(whereEqualTo) 재설계하여 해결

#### ◆ 이미지 업로드 오류 해결

Storage 업로드 중 파일 스트림 문제로 빌드 오류 발생

→ firebase\_storage 버전 업데이트 및 API 변경사항 반영

→ 업로드 후 URL 비동기 처리 문제를 await 기반으로 해결

#### ◆ 커플 공유 기능 로직 정교화

두 사용자의 uid를 기반으로 shared node를 자동 생성하는 방식 설계

데이터 sync 문제(작성자가 서로 다른 timeline 보임) 해결

→ DB 트랜잭션 기반 update 처리로 일관성 유지

#### ◆ Android 빌드 오류 해결

Gradle 버전 충돌 문제 해결

minSdk, targetSdk 설정 Firebase 요구사항에 맞게 조정

실제 기기 테스트(ADB) 중 발생한 권한 문제(사진 접근 권한) 수정

#### 4) 테스트 & 디버깅 과정 참여

##### ◆ 단위 기능 테스트

일기 작성/수정/삭제 동작 테스트

커플 모드에서 데이터 반영 여부 검증

이미지 업로드 시간/성능 테스트

##### ◆ 버그 원인 추적 및 해결

홈 화면 새로고침 시 데이터 반복 로드 문제 해결

누락된 key 값으로 인해 발생한 null 참조 오류 제거

잦은 rebuild로 인한 성능 저하 해결 (setState 최소화)

##### ◆ 사용성 테스트 참여

팀 테스트 시 사용 흐름 분석 및 UX 수정 제안

#### 5) 프로젝트 운영 측면 기여

##### ◆ GitHub 기반 협업 지원

push/pull 머지 충돌 해결

main/master 브랜치 정리 및 파일 구조 안정화

pubspec.yaml 의존성 충돌 정리

##### ◆ 빌드 및 배포 환경 구성

APK 빌드 성공을 위한 각종 Gradle 설정 재조정

Firebase 콘솔 테스트 환경 연결 검증

#### 1. 기획 및 설계

## 5. 기획 및 설계

요구사항 정의: FR 6개, NFR 1개 전체 작성

ERD 설계: 4개 테이블 완전 설계 (users, coupleCodes, couples, emotionStats)

시스템 아키텍처: Flutter + Firebase 선정, Feature-First 구조 설계

데이터 모델 설계: NoSQL 구조, 날짜 Key 최적화, 비정규화 전략

## 6. 프론트엔드 전체 개발

3개 데이터 모델 (352줄)

diary.dart, couple\_entry.dart, diary\_space.dart

4개 서비스 로직 (1,230줄)

auth\_service.dart (150줄): 로그인 관리

diary\_service.dart (450줄): 일기 CRUD 전체, 월별 쿼리, 통계

couple\_service.dart (350줄): 6자리 코드 생성, 실시간 매칭, Transaction 연결

storage\_service.dart (280줄): 이미지 업로드, 압축, 경로 관리

6개 화면 완전 구현 (2,700줄)

login\_screen.dart (150줄): 로그인 화면

calendar\_screen.dart (450줄): TableCalendar, 커스텀 셀, 월별 로드, 네비게이션

diary\_write\_screen.dart (500줄): 8가지 감정 선택, 이미지 3장 업로드, 저장 로직

personal\_diary\_detail\_screen.dart (380줄): 상세 보기, 이미지 갤러리, 수정/삭제

couple\_diary\_detail\_screen.dart (420줄): Side-by-Side 레이아웃, 공유 이미지

home\_screen.dart (500줄): 인사말, 통계 3개, 최근 일기 스와이프, Pull-to-refresh

settings\_screen.dart (300줄): 커플 코드, 실시간 연결 감지, 성공 팝업

## 디자인 시스템

8개 감정 색상 정의, Pretendard 폰트, 8px 간격 시스템

## 7. 성능 최적화 (실측 수치)

로딩 시간: 3초 → 1.2초 (60% 단축) - 월별 범위 쿼리 적용

메모리: 120MB → 60MB (50% 감소) - 이미지 캐시 제한

네트워크 요청: 31번 → 1번 (범위 쿼리)

이미지 용량: 5MB → 500KB (90% 압축)

## 8. 문제 해결 (10개 이상 버그)

이미지 업로드 실패 → Storage 경로 수정  
커플 일기 경로 오류 → DiarySpace enum으로 분기  
중복 팝업 → 플래그로 제어  
메모리 누수 → 리스너 dispose  
달력 로딩 지연 → 월별 쿼리 (5초 → 0.8초)  
저장 버튼 중복 클릭 → 버튼 비활성화  
Firebase 초기화 타이밍 → ensureInitialized()  
이미지 로드 실패 → errorBuilder 적용  
날짜 포맷 불일치 → DateFormat 통일  
Security Rules 오류 → auth.uid 수정

## 9. 개발 환경 구성

Flutter 프로젝트 생성 및 패키지 12개 추가  
AndroidManifest.xml 권한 설정  
Gradle 빌드 오류 해결 (8회)  
Release APK 빌드 및 테스트

김아리- Document Support & Basic Testing  
주요 역할(Role)  
UI/UX 디자인 서포트 의견 제시  
사용성 테스트(User Testing) 참여  
기획 단계에서 피드백 제공

기여한 상세 내용(Contributions)

- 1) UI/UX 시각 요소 피드백 제공  
홈 탭 화면 구성 과정에서 디자인 관점 의견 제시  
색감·간단한 레이아웃에 대한 사용자 시점 피드백  
앱 전체 분위기(톤앤매너)에 대한 의견 공유
- 2) 기본 사용성 테스트(Basic User Testing) 참여  
개발된 기능 체험 및 사용자 입장에서 불편 요소 제보  
오류 발생 시 간단한 피드백 제공  
앱 초기 버전 테스트를 통해 사용자 흐름 점검
- 3) 문서 작업 보조(Documentation Assistant)  
발표 자료 일부 레이아웃 관련 의견 제공  
문서 맞춤법/흐름 체크 등 간단한 검토 작업 참여  
팀 산출물 작성 시 형식 부분 검수
- 4) 회의 참여 및 브레인스토밍

정기 회의 참석

아이디어 논의 시 의견 참여

팀 기획 방향성 검토에 참여