

# 소프트웨어공학 Teamproject

유지보수 계획서 (Maintenance Plan)

2팀 이예린 김재이 김아리

## 1. 문서 개요 (Introduction)

본 문서는 「한 눈에 보는 일기장」 서비스가 출시된 이후 안정적으로 운영되기 위한 유지보수 전략, 업데이트 로드맵, 장애 대응 프로세스, 보안 관리 정책을 정의한다.  
Firebase 기반 Mobile App 특성을 고려하여, 데이터 안정성·서비스 가용성·성능·보안·운영 효율성을 달성하는 것을 목표로 한다.

## 2. 유지보수 목적 및 범위 (Purpose & Scope)

### 2.1 목적

서비스 운영 단계에서 발생하는 오류, 성능 문제, 사용자 불편 사항을 신속하게 해결  
기능 개선 및 환경 변화(Android 버전, Firebase SDK 업데이트 등)에 대응  
장기적인 확장 및 데이터 품질 유지

### 2.2 유지보수 범위

#### (1) 기능 유지보수

일기 생성/수정/삭제 기능 유지  
감정 기록 및 시각화 기능의 정확성 검증  
커플 모드(초대 코드, 공유 달력, 데이터 동기화) 안정성 유지

#### (2) 인프라 유지보수

Firebase Realtime Database 구조 업데이트  
Storage 이미지 최적화 및 불필요 파일 정리  
Firebase Authentication 정책 점검

#### (3) 성능 개선

화면 로딩 시간 개선  
Storage 업로드 최적화  
달력 렌더링 성능 향상

#### (4) 보안 유지

Firebase Rules 정기 점검  
데이터 접근 통제 강화  
개인정보 보호 기준 준수

#### (5) 환경 변화 대응

Android OS 업데이트 대응  
Flutter/Android Gradle/Firebase SDK 버전 업데이트

## 3. 조직 및 역할 (Maintenance Roles)

역할	담당자	주요 업무
PM/운영 책임자	이예린	유지보수 일정 관리, 변경 승인, 정책 수립
개발 담당자	김재이	기능 수정, SDK 업데이트, 버그 핫픽스

QA/테스터          김아리    업데이트 기능 검증, 사용자 피드백 수집

#### 4. 정기 유지보수 계획 (Regular Maintenance Schedule)

##### 4.1 월간 점검 (Monthly)

점검 항목          내용

Storage 사용량 점검      불필요 이미지 삭제, 용량 과다 사용자 정리

오류 로그 분석    Firebase Crashlytics 기반 크래시 원인 분석

DB 동기화 점검    실시간 반영 여부 및 지연 시간 측정

사용자 피드백 검토      기능 불편사항 및 UX 개선점 수집

##### 4.2 분기별 점검 (Quarterly)

점검 항목          내용

보안 규칙 점검    Firebase Rules 업데이트 및 취약점 점검

코드 리팩토링      중복 코드 제거, 모듈화, 구조 개선

성능 최적화      달력 렌더링 속도 개선, Storage 업로드 방식 점검

신규 기능 후보 평가      사용자 요청 기반 우선순위 선정

##### 4.3 반기별 점검 (Semi-Annual)

점검 항목          내용

데이터베이스 구조 재평가 스키마 확장성 및 비용 효율성 검토

API 및 SDK 버전 업 대응 Firebase 및 Android 버전 마이그레이션

이미지 저장 정책 점검    Thumbnail 도입, 압축 정책 개선

장기 로드맵 조정 서비스 확대 계획 점검 및 재수립

#### 5. 장애 대응 계획 (Incident Response Plan)

##### 5.1 장애 발생 프로세스

Crashlytics 통해 자동 장애 감지

심각도(High, Medium, Low) 분류

개발 담당자에게 즉시 알림 전달

긴급 패치(Hotfix) 또는 다음 업데이트로 배포

장애 원인 분석 및 재발 방지 대책 문서화

##### 5.2 심각도 기준

등급    정의    대응 시간

High    앱 실행 불가, 데이터 손실, 커플 연결 불가 24시간 이내 수정

Medium 기능 일부 오류, 이미지 업로드 실패      72시간 이내 패치

Low    UI 불편, 문구 오류      정기 업데이트 포함 해결

##### 5.3 로그 관리 정책

로그 종류          목적    도구

Crash 로그          장애 원인 분석    Firebase Crashlytics

기능 사용 이벤트 사용자 행동 분석 Firebase Analytics  
 Storage 업로드 로그      업로드 실패 추적 Storage Listener  
 커플 연결 이벤트 기능 안정성 검증 Realtime DB Log  
 ※ 개인정보(텍스트·이미지·감정 등)는 로그에 포함하지 않음.

## 6. 보안 및 개인정보 보호 관리 (Security & Privacy)

### 6.1 데이터 보호 원칙

개인 식별 정보 수집 최소화  
 UID 기반 익명 사용자 구조  
 사용자 데이터는 사용자가 소유하며 언제든지 삭제 가능

### 6.2 접근 통제

DB와 Storage는 “사용자 본인 또는 커플 쌍만 읽기/쓰기 가능” 규칙 적용  
 커플 해제 시 즉시 권한 삭제

### 6.3 데이터 삭제 정책

사용자가 일기 삭제 시 DB & Storage에서 완전 삭제  
 커플 모드 해제 시 공유 데이터 read-only로 보관하거나 즉시 삭제 선택 가능

## 7. 업데이트 및 기능 개선 계획 (Update Roadmap)

### 7.1 출시 직후(1개월)

UI 안정화, 성능 개선  
 버그 패치 및 Storage 최적화  
 감정 색상 시각화 방식 보완

### 7.2 1차 업데이트 (3개월 후)

월별 감정 통계 차트 제공  
 일기 검색 기능(날짜/키워드) 추가  
 커플 모드 UX 개선

### 7.3 2차 업데이트 (6개월 후)

소셜 로그인(Google, Kakao)  
 테마 커스터마이징(라이트/다크 모드)  
 홈 위젯, 알림 기능 추가  
 데이터 백업/복구 기능 강화

## 8. 리스크 관리 (Risk Management)

리스크	영향도	발생 가능성	대응 방안
Firebase Storage 용량 초과	중	중	이미지 자동 압축, 만료 정책 도입
커플 모드 데이터 충돌	중	낮음	사용자 권한 분리(읽기/쓰기)
이미지 업로드 지연	높음	중	비동기 처리 강화 및 로딩 UI

일정 지연            높음        중            스프린트 관리 및 역할 재조정

## 9. 비용 전망 및 운영 전략 (Cost Analysis & Operations)

### 9.1 비용 구조

항목	비용	비고
Firebase Realtime DB	무료 티어	1GB/월 기준
Firebase Storage	무료 티어	5GB/월 기준
Figma	무료	디자인 협업
GitHub	무료	버전 관리
배포(apk)	무료	App 제공 가능

### 9.2 결론

초기 MVP는 무료 티어 내 안정적 운영 가능

사용자 증가 시 Storage 비용 증가 → 이미지 최적화 정책 필수

## 10. SLA / 서비스 품질 기준 (Service Level Agreement)

항목    목표

서비스 가용성    99% 이상

달력 로딩 속도    2초 이내

이미지 업로드 성공률    98% 이상

데이터 동기화 지연    5초 이내

장애 대응 시간    High: 24시간 이내

Firebase Status Dashboard 기준으로 장애 상황 대응.

## 11. 장기 운영 전략 (Long-term Operations)

감정 데이터 기반 통계 및 AI 기능으로 확장

커플 모드 고도화(기념일 자동 생성, 공통 타임라인)

서버리스 구조 유지로 비용 최적화

UI/UX 지속 개선(사용자 피드백 반영)

## 12. 결론 (Conclusion)

본 유지보수 계획서는 서비스 안정성 확보, 장애 대응 체계 구축, 보안 강화, 지속적인 기능 개선을 목표로 한다.

정기 점검·로드맵 기반 개발·데이터 정책을 적용함으로써 「한 눈에 보는 일기장」의 장기적 성장과 사용자 만족도를 지속적으로 확보할 수 있다.