

2인 개발팀 세부 작업 분해 (Task Breakdown)

팀 구성 및 역할

- **Developer A (Full-stack Lead)**: Spring Boot, React, 시스템 설계
- **Developer B (AI/Data Specialist)**: Python/FastAPI, AI 모델, 데이터 분석

Sprint 1: 기반 인프라 구축 (Week 1-2)

Developer A 작업 상세

TASK-A001: 개발 환경 구축 (Day 1-2, 8 points)

세부 작업: 1. 프로젝트 구조 설정 (4시간) - Git 저장소 초기화 및 브랜치 전략 수립 - 프로젝트 루트 디렉토리 구조 생성 - README.md 및 기본 문서 작성

1. **Docker** 환경 구성 (8시간)
2. `docker-compose.yml` 작성 (MariaDB, Redis, Nginx)
3. 각 서비스별 Dockerfile 작성
4. 환경변수 관리 (.env 파일)
5. 네트워크 및 볼륨 설정
6. **Spring Boot** 프로젝트 초기화 (4시간)
7. Spring Boot 3.x 프로젝트 생성
8. 필수 의존성 추가 (Security, JPA, Web)
9. `application.yml` 기본 설정
10. 기본 패키지 구조 생성

산출물: - `docker-compose.yml` - Spring Boot 기본 프로젝트 - 환경 설정 파일들

TASK-A002: 사용자 인증 시스템 구현 (Day 3-5, 5 points)

세부 작업: 1. 데이터베이스 설계 (4시간) - User 엔티티 설계 - 데이터베이스 스키마 생성 - JPA 엔티티 및 Repository 구현

1. **Spring Security** 설정 (6시간)

2. SecurityConfig 클래스 구성
3. JWT 토큰 생성/검증 로직
4. 인증/인가 필터 구현
5. **회원가입/로그인 API** (6시간)
6. AuthController 구현
7. 회원가입 로직 (이메일 중복 체크, 비밀번호 암호화)
8. 로그인 로직 (JWT 토큰 발급)
9. 이메일 인증 기능

산출물: - User 엔티티 및 Repository - AuthController 및 Service - JWT 토큰 유틸리티

TASK-A003: React 프론트엔드 기초 (Day 6-8, 3 points)

세부 작업: 1. **React 프로젝트 설정** (4시간) - Create React App 또는 Vite 설정 - 기본 폴더 구조 및 라우팅 설정 - CSS 프레임워크 선택 및 설정 (Tailwind CSS)

1. **인증 페이지 구현** (8시간)
2. 로그인 페이지 컴포넌트
3. 회원가입 페이지 컴포넌트
4. 폼 유효성 검사
5. API 통신 설정 (axios)
6. **기본 레이아웃** (4시간)
7. 헤더/네비게이션 컴포넌트
8. 기본 레이아웃 컴포넌트
9. 라우터 Guard 구현

산출물: - React 기본 프로젝트 - 로그인/회원가입 페이지 - 기본 레이아웃 컴포넌트

TASK-A004: 초기 설정 기능 (Day 9-10, 8 points)

세부 작업: 1. **사용자 설정 API** (6시간) - UserSettings 엔티티 설계 - 설정 CRUD API 구현 - 프로필 이미지 업로드 기능

1. **초기 설정 UI** (8시간)
2. 온보딩 페이지 컴포넌트
3. 닉네임/목표 설정 폼
4. 알림 설정 UI
5. 프로필 이미지 업로드

6. 소셜 로그인 연동 (2시간)

7. Google OAuth 설정

8. Kakao 로그인 설정

산출물: - UserSettings API - 온보딩 페이지 - 소셜 로그인 기능

Developer B 작업 상세

TASK-B001: AI 서버 기반 구조 (Day 1-3, 5 points)

세부 작업: 1. **FastAPI 프로젝트 설정** (4시간) - FastAPI 프로젝트 초기화 - 기본 디렉토리 구조 생성 - 필수 라이브러리 설치 (requirements.txt)

1. **MongoDB 연결 설정** (4시간)

2. MongoDB 컨테이너 설정

3. PyMongo 연결 설정

4. 기본 데이터 모델 정의

5. **기본 API 엔드포인트** (4시간)

6. Health Check API

7. 기본 라우터 구조

8. 로깅 설정

산출물: - FastAPI 기본 프로젝트 - MongoDB 연결 모듈 - 기본 API 구조

TASK-B002: Spring Boot 연동 구조 (Day 4-6, 8 points)

세부 작업: 1. **API 통신 인터페이스 설계** (4시간) - Spring Boot -> Python API 호출 구조 - 공통 응답 포맷 정의 - 에러 핸들링 표준화

1. **Kafka 기본 설정** (6시간)

2. Kafka 컨테이너 설정

3. Producer/Consumer 기본 구현

4. 토픽 정의 및 설정

5. **비동기 처리 구조** (6시간)

6. 백그라운드 작업 큐 설정

7. 작업 상태 관리 시스템

8. 결과 캐싱 (Redis 연동)

산출물: - API 통신 모듈 - Kafka 설정 및 기본 코드 - 비동기 처리 기반 구조

TASK-B003: STT 기본 기능 (Day 7-10, 8 points)

세부 작업: 1. 음성 파일 처리 모듈 (6시간) - 음성 파일 업로드 API - 파일 형식 변환 (wav, mp3 지원) - Supabase 스토리지 연동

1. STT 엔진 연동 (8시간)
2. Google Speech-to-Text API 연동
3. 한국어 최적화 설정
4. 음성 품질 전처리
5. STT API 구현 (2시간)
6. `/api/ai/stt` 엔드포인트 구현
7. 실시간 처리 상태 반환
8. 에러 처리 및 재시도 로직

산출물: - 음성 파일 처리 모듈 - STT API 엔드포인트 - Supabase 연동 코드

Sprint 2: 일기 시스템 구축 (Week 3-4)

Developer A 작업 상세

TASK-A005: 일기 기본 CRUD (Day 1-3, 8 points)

세부 작업: 1. 일기 데이터 모델링 (4시간) - Diary 엔티티 설계 - 감정 enum 정의 - 일기-이미지 연관관계 설정

1. 일기 API 구현 (8시간)
2. DiaryController 및 Service
3. 일기 생성/조회/수정/삭제 API
4. 페이징 및 검색 기능
5. 일기 작성 UI (4시간)
6. 일기 작성 페이지
7. 감정 선택 컴포넌트
8. 임시저장 기능

산출물: - Diary 엔티티 및 API - 일기 작성 UI 컴포넌트

TASK-A006: 타임라인 뷰 구현 (Day 4-6, 8 points)

세부 작업: 1. 캘린더 뷰 컴포넌트 (8시간) - react-calendar 커스터마이징 - 날짜별 감정/소비 요약 표시 - 캘린더 이벤트 처리

1. **리스트 뷰 컴포넌트** (6시간)
2. 시간순 일기 목록
3. 무한 스크롤 구현
4. 필터링 기능
5. **타임라인 API** (2시간)
6. 기간별 데이터 조회 API
7. 통계 데이터 API

산출물: - 캘린더 뷰 컴포넌트 - 리스트 뷰 컴포넌트 - 타임라인 API

TASK-A007: 일기 수정/삭제 기능 (Day 7-8, 5 points)

세부 작업: 1. 수정 페이지 구현 (6시간) - 일기 수정 페이지 - 변경 이력 관리 - 수정 권한 검증

1. **삭제 기능 구현** (4시간)
2. Soft Delete 구현
3. 삭제 확인 모달
4. 관련 데이터 처리
5. **변경 로그 시스템** (2시간)
6. 변경 이력 엔티티
7. 감사 로그 기능

산출물: - 일기 수정/삭제 기능 - 변경 이력 관리 시스템

Developer B 작업 상세

TASK-B004: 이미지 업로드 시스템 (Day 1-4, 8 points)

세부 작업: 1. **이미지 업로드 API** (6시간) - 파일 업로드 엔드포인트 - 이미지 크기/형식 검증 - SupaBase 스토리지 연동

1. **이미지 전처리** (6시간)
2. 이미지 리사이징
3. 포맷 최적화

- 4. 메타데이터 추출
- 5. **이미지 관리 시스템** (4시간)
- 6. 이미지 URL 생성
- 7. 임시 파일 정리
- 8. 용량 관리

산출물: - 이미지 업로드 API - 이미지 전처리 모듈

TASK-B005: STT 고도화 (Day 5-8, 8 points)

세부 작업: 1. **음성 품질 향상** (6시간) - 노이즈 제거 알고리즘 - 음성 품질 분석 - 전처리 파이프라인 개선

- 1. **실시간 처리 최적화** (6시간)
- 2. 비동기 처리 개선
- 3. 스트리밍 처리 지원
- 4. 캐싱 전략 구현
- 5. **정확도 개선** (4시간)
- 6. 도메인 특화 어휘 추가
- 7. 후처리 규칙 적용
- 8. 신뢰도 점수 계산

산출물: - 향상된 STT 시스템 - 실시간 처리 최적화

Sprint 3: 가계부 연동 (Week 5-6)

Developer A 작업 상세

TASK-A008: 오픈뱅킹 연동 (Day 1-6, 21 points)

세부 작업: 1. **오픈뱅킹 API 연동** (12시간) - 금융사 API 인증 구현 - 계좌 정보 조회 API - 거래 내역 조회 API - API 호출 제한 관리

- 1. **거래 내역 데이터 모델링** (8시간)
- 2. Transaction 엔티티 설계
- 3. 금융사별 데이터 표준화
- 4. 중복 거래 방지 로직
- 5. **자동 데이터 수집 시스템** (12시간)

6. 스케줄링 시스템 구현
7. 실시간 거래 알림
8. 데이터 동기화 로직
9. 에러 복구 메커니즘
10. **가계부 UI 구현** (8시간)
11. 거래 내역 목록 페이지
12. 카테고리별 필터링
13. 월별/일별 요약 뷰
14. 수동 입력 기능

산출물: - 오픈뱅킹 연동 모듈 - 거래 내역 관리 시스템 - 가계부 UI 컴포넌트

TASK-A009: 데이터 동기화 시스템 (Day 7-8, 3 points)

세부 작업: 1. **실시간 동기화** (4시간) - WebSocket 연결 관리 - 실시간 데이터 푸시 - 동기화 상태 표시

1. **충돌 해결 로직** (4시간)
2. 데이터 충돌 감지
3. 자동/수동 해결 방식
4. 백업 및 복구

산출물: - 실시간 동기화 시스템 - 데이터 충돌 해결 로직

Developer B 작업 상세

TASK-B006: 소비 카테고리 분류 AI (Day 1-6, 13 points)

세부 작업: 1. **분류 모델 개발** (12시간) - 거래 내역 텍스트 전처리 - 기계학습 모델 훈련 (Random Forest, SVM) - 가맹점명 기반 분류 규칙 - 모델 성능 평가 및 개선

1. **분류 API 구현** (6시간)
2. `/api/ai/classify/expense` 엔드포인트
3. 실시간 분류 처리
4. 신뢰도 점수 반환
5. **학습 데이터 관리** (6시간)
6. 사용자 피드백 수집
7. 지속적 학습 파이프라인
8. 모델 버전 관리

Sprint 4: 기본 분석 기능 (Week 7-8)

Developer A 작업 상세

TASK-A010: 차트 시각화 시스템 (Day 1-4, 8 points)

세부 작업: 1. 차트 라이브러리 설정 (4시간) - Recharts 라이브러리 설정 - 기본 차트 컴포넌트 구현 - 반응형 디자인 적용

- 데이터 시각화 컴포넌트 (8시간)
- 파이차트 (카테고리별 소비)
- 라인차트 (시간별 변화)
- 바차트 (기간별 비교)
- 인터랙티브 기능
- 대시보드 페이지 (4시간)
- 종합 대시보드 레이아웃
- 위젯 구성 시스템
- 필터링 및 기간 선택

산출물: - 차트 컴포넌트 라이브러리 - 대시보드 페이지

Developer B 작업 상세

TASK-B007: 데이터 분석 파이프라인 (Day 1-4, 8 points)

세부 작업: 1. 데이터 전처리 모듈 (6시간) - 일기 텍스트 정제 - 거래 데이터 정규화 - 시계열 데이터 변환

- 분석 알고리즘 구현 (8시간)
- 상관관계 분석
- 패턴 발견 알고리즘
- 이상치 탐지
- 배치 처리 시스템 (2시간)
- 주기적 분석 스케줄러
- 분석 결과 저장

8. 성능 모니터링

산출물: - 데이터 전처리 모듈 - 분석 알고리즘 - 배치 처리 시스템

TASK-B008: 기본 패턴 분석 (Day 5-6, 13 points)

세부 작업: 1. 감정-소비 연관 분석 (8시간) - 감정별 소비 패턴 분석 - 통계적 유의성 검증 - 개인별 패턴 추출

1. 시계열 분석 (6시간)
2. 소비 트렌드 분석
3. 주기성 패턴 발견
4. 계절성 요인 분석
5. 분석 결과 API (2시간)
6. 분석 결과 조회 API
7. 개인화된 인사이트 생성

산출물: - 패턴 분석 모델 - 분석 결과 API

TASK-B009: 인사이트 리포트 (Day 7-8, 8 points)

세부 작업: 1. 리포트 생성 엔진 (6시간) - 템플릿 기반 리포트 생성 - 자연어 인사이트 생성 - 개인화된 메시지

1. 리포트 스케줄링 (4시간)
2. 주간/월간 리포트 자동 생성
3. 이메일 발송 시스템
4. 리포트 히스토리 관리
5. 리포트 UI 연동 (2시간)
6. 리포트 조회 페이지
7. PDF 출력 기능
8. 공유 기능

산출물: - 리포트 생성 시스템 - 리포트 UI 연동

각 스프린트 별 Daily Standup 가이드

Daily Standup 구조 (15분)

1. 어제 완료한 작업 (각자 2-3분)

2. 오늘 계획 작업 (각자 1-2분)
3. 블로커/도움 필요사항 (각자 1분)
4. 협업 필요 사항 논의 (3-5분)

협업 체크포인트

- API 인터페이스 변경: 상대방 작업에 영향 즉시 공유
- 공통 데이터 모델: 함께 설계 및 리뷰
- 기술적 의사결정: 페어 프로그래밍 또는 함께 논의

일간 기술 리뷰

매일 오후 (1시간): - 코드 리뷰 및 아키텍처 점검 - 성능 이슈 및 개선 방안 논의 - 다음 주 기술적 챌린지 사전 논의