건설사 현장 작업일보·프로젝트 관리 시스템

Jake Tae-moon Jo · 개발 리드 · https://jake-jo-portfolio.vercel.app/ko/work/construction-pmis/

범용 건설 프로젝트 관리 시스템(PMIS)으로 시작해 고객의 종이 작업일보에 맞춰 다시 설계하고, 이미 만든 재무 기능을 범위에서 덜어 낸 뒤 고객 인수 테스트(UAT)까지 진행했습니다.

고객 인수 테스트	1일	2번	12종
까지 진행	일보 재설계 → 코어 병합	이미 만든 기능을 범위에서 뺀 결정	역할 4 × 언어 3 사용자 매뉴얼

결과

범용 관리 시스템으로 시작해 작업일보 중심으로 다시 지었고, 고객 인수 테스트까지 진행했습니다.

고객의 수정 요청은 모두 24시간 안에 반영했습니다.

작업일보 중심의 현장 PMIS를 3개 언어 웹으로 만들었습니다. 주 보고자 지정, 작성, 제출 검사, 현장소장 승인, 디렉터의 승인 뒤 수정, 일보 PDF, 승인된 행의 실적 합산까지 한 흐름입니다.

7월 8일 스테이징을 고객 도메인에 올렸고, 고객 인수 테스트까지 진행했습니다.

고객의 종이 작업일보 샘플 2건을 기준으로 다시 설계한 다음 날(5월 10일), 일보 코어 7단계를 병합했습니다.

이미 만든 일정·EVM과 재무를 범위에서 빼, 제품을 일보와 자재의 물리적 흐름에 맞췄습니다.

역할 4개 × 언어 3개, 사용자 매뉴얼 12종을 만들었습니다. 스크린샷 450장은 캡처 스크립트로 다시 찍습니다.

기간

2026.04 – 2026.07

역할

사람은 본인 1명 · AI 에이전트 팀 설계·운영

팀

본인 1명 + AI 에이전트 팀(11개 역할 정의)

플랫폼

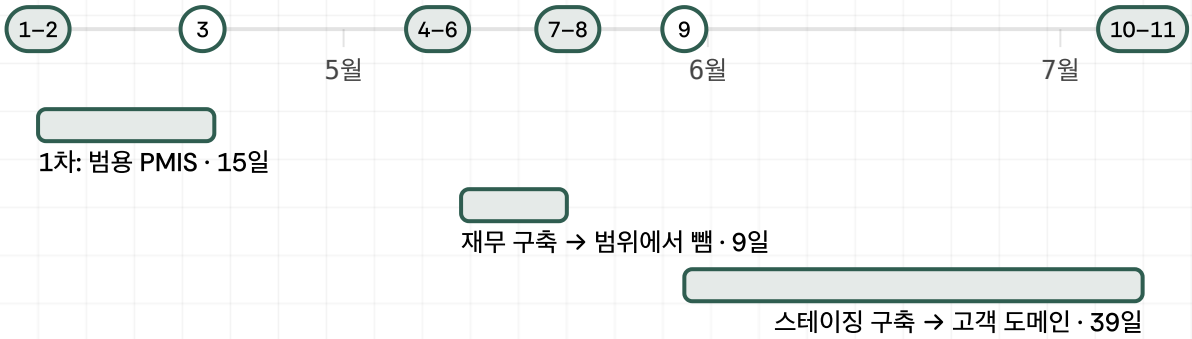
웹(본사·현장 역할별 화면, 3개 언어), 일보 PDF

배경

종합건설사. 고객의 종이 작업일보 샘플 2건을 입력 화면의 기준으로 삼음

범용 관리 시스템에서 고객 인수 테스트까지

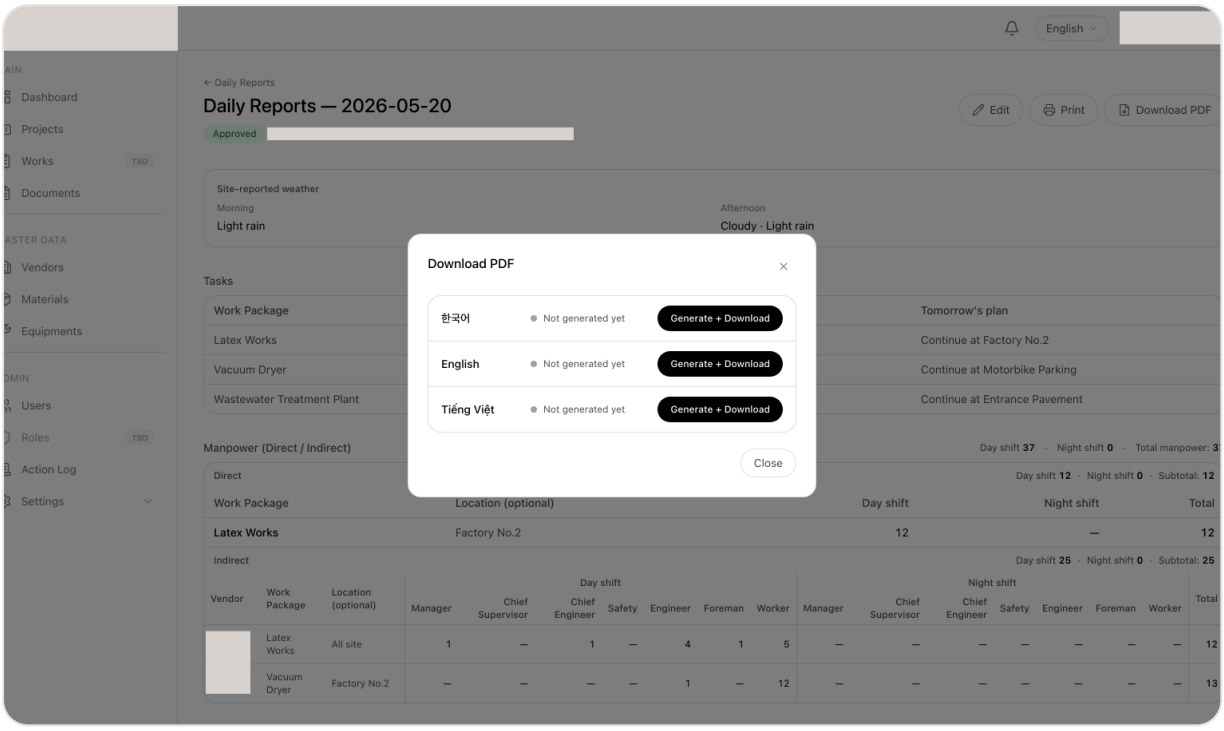
범용 관리 시스템으로 시작해 일보 중심으로 다시 짓고 재무를 덜어 낸 뒤, 고객 인수 테스트까지 갔다.



1	2026년 4월 5일	**첫 커밋** 에이전트 팀 7개 역할로 시작
2	2026년 4월 7일	**첫 배포** 문서 관리·접근 제어까지
3	2026년 4월 19일	**일정 성과 분석** 상태 신호·S-curve·임계 경로(4월 20일까지)
4	2026년 5월 9일	**일보 중심으로 다시 설계** 고객의 종이 일보 샘플 2건 기준, 일정·EVM은 보관함으로
5	2026년 5월 10일	**일보 코어 병합** 7단계, 첫 매뉴얼
6	2026년 5월 11일	**재무 구축** 발주·입고·지급·원가(5월 12일까지)
7	2026년 5월 20일	**재무를 범위에서 뺌** 「발주·지급·원가는 외부 회계 시스템」, 자재 3종 카탈로그
8	2026년 5월 24일	**매뉴얼 다시 씀** 역할 4 × 언어 3 = 12종
9	2026년 5월 30일	**스테이징 구축** 증분 배포 스크립트(5월 31일까지)
10	2026년 7월 8일	**고객 도메인 스테이징** 인증서 적용
11	2026년 7월 9일	**메일 알림, 마지막 커밋**

화면

일보 상세의 PDF 내려받기 대화상자(영어 화면). 한국어·영어·베트남어 세 언어 중 골라 생성한 뒤 내려받는다. 시드 데이터이며 로고와 협력사명은 가렸다.



문제와 배경

고객의 종이 작업일보를, 프로젝트마다 정한 한 사람이 쓰고 현장소장이 승인하는 웹 흐름으로 옮기는 일이었습니다.

고객사는 종합건설사입니다. 사용자는 본사 관리자, 디렉터, 현장소장, 직원입니다.
제품의 중심은 작업일보입니다. 프로젝트마다 정한 주 보고자 한 명이 하루 한 건을 쓰고, 현장소장이나 디렉터가 검토해 승인합니다. 일보에는 작업, 직영 인력, 협력사 인력(7개 직종 × 주간·야간), 장비, 자재 사용, 사진, 도면, 기타 첨부가 들어갑니다. 협력사·장비·자재 마스터와 알림, 사진·도면 업로드는 모두 이 흐름을 받치려고 있습니다.

처음부터 이 모양은 아니었습니다. 4월에는 작업 WBS·간트, 문서 승인, 일정 성과 분석(EVM)을 갖춘 범용 PMIS로 시작했습니다. 5월 9일 고객의 종이 작업일보 샘플 2건을 기준으로 일보 중심으로 다시 설계했고, 5월 20일에는 이미 만든 재무 기능을 범위에서 뺐습니다(핵심 결정 1·2).

권한이 까다로웠습니다. 같은 직원이 어떤 프로젝트에서는 일보를 쓰고, 다른 프로젝트에서는 읽기만 합니다. 현장소장도 프로젝트마다 다릅니다. 승인된 일보는 감사의 단위이자 실적 합산의 근거여서, 승인 뒤에 누가 무엇을 고칠 수 있는지도 정해야 했습니다. 화면은 3개 언어로 만들고, 알림은 받는 사람의 언어로 보내야 했습니다.

일보 한 건이 지나가는 길

일보는 프로젝트마다 정한 주 보고자 한 명이 하루 한 건 쓰고 현장소장이 승인하며, 승인된 행만 실적에 들어간다.

① **주 보고자 지정** 〇사람
디렉터가 현장소장·멤버를 정하고 프로젝트당 1명 지정(DB가 활성 1명만 허용)

② **오늘 초안** 〇사람
프로젝트×날짜당 1건, 전날 「내일 계획」이 오늘 실적으로 미리 채워진다

③ **섹션 저장** □시스템
저장은 막지 않는다. 행마다 잠금 확인 → 변경 → 감사 로그 → PDF 캐시 무효화, 다른 프로젝트 id는 거절

④ **제출** 〇사람
편집기가 규칙 11종으로 먼저 막고 서버가 그중 6종을 다시 본다. 제출 뒤 날씨를 붙인다

⑤ **검토** 〇사람
현장소장이 승인하거나 사유를 적어 반려. 검토 중에는 일보와 첨부를 아무도 못 고친다

⑥ **승인** □시스템
읽기 전용. 디렉터만 상태를 유지한 채 고치고 감사 로그에 남는다

⑦ **일보 PDF** □시스템
서버가 HTML을 만들어 헤드리스 크롬에 넣고 도면 PDF를 덧붙여 캐시

⑧ **실적 합산** □시스템
승인된 행만 사용 내역·작업계획 실적에 더한다

되돌아가는 길
↺ ⑤ 검토 → ③ 섹션 저장: 반려되면 고쳐 다시 제출(횟수 제한 없음)

제 역할

만드는 쪽의 사람은 저 한 명이었고, 에이전트 팀을 첫 커밋부터 정의해 운영하면서 디자인 방향과 병합은 제가 정했습니다.

만드는 쪽의 사람은 저 한 명이었습니다. git 작성자는 저 한 사람이고, 커밋 290개 중 177개(61%)에 AI 공동 작성 표시가 있습니다. 병합 커밋을 빼면 239개 중 177개(74%)입니다. 손으로 짠 부분과 에이전트가 쓴 부분을 커밋으로 가를 수는 없어서, 역할을 「사람은 본인 1명 · AI 에이전트 팀 설계·운영」으로 적었습니다.

에이전트 팀은 첫 커밋(4월 5일)부터 정의 파일로 들어 있었습니다. 오케스트레이터, 디자이너, 프런트엔드, 백엔드, 리뷰어, API 문서, QA 7개 역할로 시작했습니다. 이후 TDD 가이드, git 흐름(4월 19일), 매뉴얼(5월 10일), 데브옵스(5월 22일)를 더해 11개 역할이 됐습니다. 오케스트레이터·백엔드·프런트엔드·데브옵스는 Opus에, 나머지 7개는 Sonnet에 배정했습니다. 정의 파일은 공개 템플릿과 다르게, 이 프로젝트에 맞춰 썼습니다. 파이프라인은 브랜치 시작 → 디자인 → 실패하는 테스트 먼저 →

구현 → 테스트 통과(커버리지 80%) → 리뷰·API 문서 → QA(e2e) → 매뉴얼 → PR 순서로 지시문에 적었습니다.

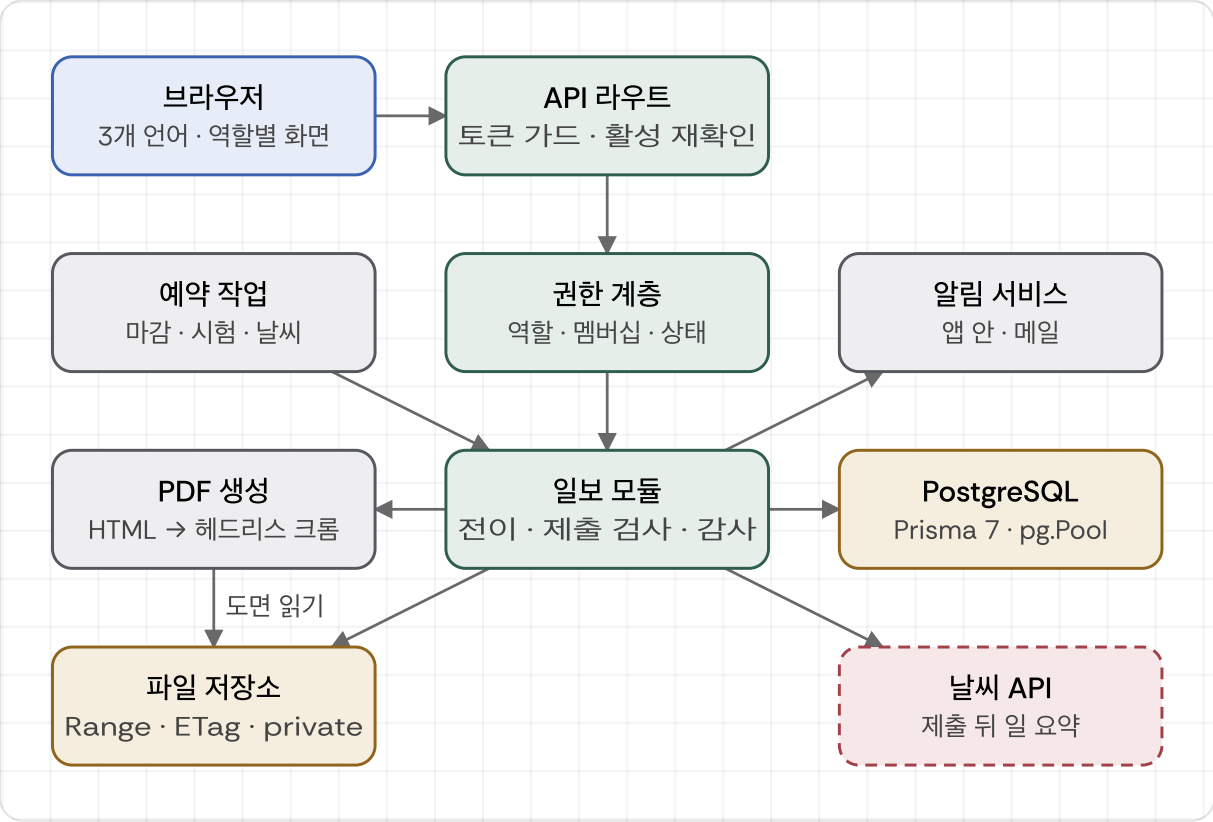
PR은 110개였고, 병합한 106개는 모두 제 계정으로 병합했습니다. 사람의 PR 리뷰는 없었습니다. 사람이 정한 것이 기록에 드러나는 곳도 있습니다. 5월 12일에는 IBM Carbon 디자인 시스템으로 옮기는 PR을 「이 제품이 갈 디자인 방향이 아니다」라며 닫았습니다. 디자인은 shadcn을 쓰지 않는 회색조 자체 시스템으로 갔습니다. 5월 14일에는 고객 미팅용 제안서(인력관리 모듈, 16쪽)를 만들어 화면 설계 승인을 요청했습니다. 이 모듈은 구현하지 않았습니다.

결정 기록 파일에는 42건이 있고, 그중 25건에 대안이 적혀 있습니다. 머리말에 적힌 대로, 에이전트가 추천 기본값으로 정해 진행한 기록입니다. 범위 전환 설계 문서에는 5월 20일에 확정된 결정 8건이 따로 있습니다. 결정 카드의 대안과 이유는 이 두 문서와 커밋 본문에서 왔습니다.

시스템 구조

시스템 구성

일보의 행 변경은 일보 모듈의 한 레시피(잠금 확인 → 변경 → 감사 로그 → PDF 캐시 무효화)를 거친다.



브라우저

3개 언어(언어마다 키 1,600개), 모든 시각은 한 현지 시간대로 고정해 보인다. 역할마다 첫 화면이 다르다. 디렉터는 포트폴리오 대시보드, 현장소장은 검토 대기함, 직원은 프로젝트 타일이다.

API 라우트

자체 JWT. 액세스 120분, 리프레시 7일 (httpOnly 쿠키 + DB). 쓸 때마다 회전하고, 재사용을 감지하면 그 사용자의 세션을 모두 끊는다. 인증 가드는 요청마다 DB에서 활성 여부를 다시 본다.

예약 작업

제출 마감 알림(주 보고자·현장소장), 콘크리트 사용 28일 뒤 시험 알림, 날씨 시간별 백필.

권한 계층

기본 역할 4개 + 프로젝트 멤버십 + 프로젝트당 현장소장 1명 + 주 보고자 지정(역할 4개와 별개인 지정, DB 부분 유일 인덱스로 활성 1명) + 일보 상태에 따른 잠금. 문서 편집 버튼을 막을지는 서버가 판정해 상세 응답에 싣는다.

알림 서비스

알림을 한 서비스로 모아, 받는 사람의 언어로 작성 시점에 만든다. 화면이 숨으면 폴링을 멈춘다. 7월 9일 메일과 서명 토큰 웹 보기를 더했다.

PDF 생성

일보당 한 장. 서버에서 HTML을 만들어 헤드리스 크롬에 넣고 도면 PDF를 뒤에 붙인다. 변경 경로 18곳에서 캐시를 지우고, 동시 생성은 하나로 합치며 30초에 끊는다.

일보 모듈

상태 전이는 순수 함수 하나다. 저장은 막지 않고 제출만 검사한다(서버 강제 6종). 행 변경은 잠금 확인 → 변경 → 감사 로그 → PDF 캐시 무효화를 거치고, 다른 프로젝트의 id는 저장 단계에서 거절한다.

PostgreSQL

PostgreSQL + Prisma 7. 감사 로그는 실패해도 응답을 막지 않고, 수정·삭제는 속성 단위 차이로 남긴다.

파일 저장소

파일은 저장소 추상화 뒤에 둔다. 다운로드는 스트리밍 + Range + 약한 ETag + private 캐시다. 공유 CDN이 다른 사용자에게 바이트를 주지 않게 했다.

날씨 API

제출이 성공하면 일 요약 14개 값과 파생 지표(콘크리트 양생 지수 등)를 저장한다. 실패해도 제출은 막지 않고, 재시도 버튼과 백필이 있다.

`prisma/migrations/…_major_reporter_active_unique/migration.sql` ·
프로젝트당 활성 주 보고자 1명은 DB 안의 부분 유일 인덱스가 지킨다. 지정이 끝난 기록은 effectiveTo가 채워져 인덱스에서 빠진다. (마이그레이션 전체)

```
-- Ensure at most one active MajorReporter per project
-- (effectiveTo IS NULL).
CREATE UNIQUE INDEX "MajorReporter_projectId_active_key"
  ON "MajorReporter"("projectId")
  WHERE "effectiveTo" IS NULL;
```

`src/lib/auth/document-permissions.ts` · 일보에 붙은 문서는 일보 상태를 따른다. 승인 뒤에는 디렉터·시스템 관리자만, 검토 중에는 아무도 못 고치고, 초안·반려일 때는 활성 주보고자도 고친다. (주석은 원문을 줄여 옮김. 원문은 오류 객체를 그 자리에서 만든다. 여기서는 설명용 deny로 줄였다)

```
// Documents attached to a daily report follow the report's state:
//   DRAFT / REJECTED → active Major Reporter or SYSTEM_ADMIN / DIRECTOR
//   SUBMITTED        → nobody (must be reviewed first)
//   APPROVED         → SYSTEM_ADMIN / DIRECTOR only
export async function gateDocumentMutation(
  caller: Caller,
  doc: DocumentForPermission,
): Promise<MutationGate> {
  const report = await findAttachedReport(doc.id);
  if (!report) {
    if (await canEditDocument(caller, doc)) return { ok: true };
    return deny(403, 'FORBIDDEN');
  }
  const isAdmin = canSeeAllProjects(caller.role);
  if (report.status === 'APPROVED') {
    if (isAdmin) return { ok: true };
    return deny(403, 'REPORT_APPROVED_DIRECTOR_ONLY');
  }
  if (report.status === 'SUBMITTED') {
    return deny(400, 'REPORT_UNDER_REVIEW');
  }
  // DRAFT or REJECTED
  if (isAdmin) return { ok: true };
  const isReporter =
    await isActiveReporter(caller.userId, report.projectId);
  if (isReporter) return { ok: true };
  return deny(403, 'FORBIDDEN');
}
```

`src/lib/daily-report/mutate.ts` · 일보의 하위 행을 바꾸는 경로들이 이 함수 하나를 거친다. 편집 잠금 → 변경 → 승인 상태면 수정 시각 → PDF 캐시 무효화 + 감사 로그 순서다. (주석은 원문을 줄여 옮김. 거절 응답 매핑은 설명용 refuse로, 감사 로그 인자는 펼쳐 썼다)

```
/**
 * Centralises the four cross-cutting concerns of a daily-report
 * child mutation:
 *   1. Edit gate    — status + actor checks
 *   2. Mutation     — the caller's callback, against the gated report
 *   3. lastEditedAt — bumped iff the report was APPROVED at gate time
 *   4. Side-effects — PDF cache invalidation + audit-log entry
 */
export async function performReportChildMutation<T>(
  input: PerformChildMutationInput<T>,
): Promise<PerformChildMutationOutcome<T>> {
  const { reportId, request, caller, audit, mutate } = input;
  const gate =
    await gateReportChildEdit(reportId, caller.userId, caller.role);
  if (!gate.ok) return refuse(gate);
  const report = gate.report!;
  const result = await mutate({ report });
  if (report.status === 'APPROVED') {
    await prisma.dailyReport.update({
      where: { id: reportId },
      data: { lastEditedAt: new Date(), lastEditedById: caller.userId },
    });
  }
  invalidateDailyReportPdfCache(reportId).catch(() => {});
  const d = typeof audit === 'function' ? audit(result) : audit;
  logAction({ userId: caller.userId, ...d, request });
  return { ok: true, result };
}
```

핵심 결정

여섯 결정 중 둘은 이미 만든 기능을 범위에서 뺀 것이고, 넷은 고객 일보의 실제 모양과 승인된 기록을 지키는 쪽입니다. 그중 권한, 잠금 판정, 장비 소수점의 세 건은 에이전트의 추천 기본값으로 진행했습니다.

일정·EVM 중심의 범용 PMIS를 작업일보 중심으로 다시 설계

문제

4월에 만든 것은 작업 WBS·간트, 문서 승인, 일정 성과 분석(상태 신호·S-curve·임계 경로)을 갖춘 범용 PMIS였습니다. 그 뒤 고객의 종이 작업일보 샘플 2건이 설계 입력으로 들어왔습니다. 고객 양식에는 작업·인력·협력사 인력·장비·자재·사진·도면 섹션이 있었습니다.

검토한 선택지

자동 집계 + 수기 보충: 활동 로그로 하루의 변경을 모으고, 현장소장이 날씨·특이사항을 덧붙인다(4월 7일 기획안 C안)

✓ 고객 양식 그대로: 주 보고자가 섹션을 직접 채우고 현장소장이 승인한다

결정

고객 양식을 따르는 쪽으로 했습니다. 프로젝트마다 주 보고자 한 명이 하루 한 건을 쓰고, 현장소장·디렉터가 승인합니다. 협력사·장비·자재 마스터, 알림, 사진·도면 업로드는 모두 이 흐름을 받치는 자리로 옮겼습니다. 스키마를 새로 쓰고 마이그레이션 이력을 버렸으며, 일정·작업·인력·출결·문서 승인은 보관함(archive)으로 옮겼습니다. C안을 버린 이유는 기록에 없고, 기획안에는 「SPEC으로 대체됨」만 남아 있습니다.

결과

다음 날(5월 10일) 일보 1~7단계를 병합했습니다(PR #12~#19). EVM은 「향후 시각화 단계」로 보관했습니다.

대가·남은 위험

4월에 만든 일정 분석과 작업 관리가 제품에서 빠졌습니다.

이미 만든 재무 기능을 범위에서 빼고, 자재는 3종 카탈로그로

문제

5월 11~12일에 발주(3단계 승인), 입고(증빙), 지급 요청, 원가 요약(엑셀·PDF)을 만들었습니다. 현장 PMIS가 돈의 흐름까지 맡을지, 자재의 흐름만 맡을지가 남았습니다.

검토한 선택지

재무를 PMIS 안에 둔다(5월 11~12일에 만든 발주·입고·지급·원가)

✓ 재무를 보관하고, 현장 PMIS는 자재의 물리적 흐름만 맡는다

결정

재무 메뉴와 모델을 모두 보관했습니다. 설계 문서에 남은 이유는 둘입니다. 「발주·지급·원가는 외부 회계 시스템에서 별도로 처리한다(PMIS 범위 아님)」와 「자재가 3종으로 고정되면 입력 부담이 줄고 시험·추적 흐름이 단순해진다」입니다. 현장 PMIS는 「공급사 → 트럭 → 현장 위치 → 사용 → 시험」이라는 자재의 흐름에 집중하기로 했습니다. 자재는 콘크리트·철근·석재 3종 카탈로그로 고정했고, 일보에는 「사용(=입고) 등록」 한 가지 이벤트만 두었으며, 작업에는 2단계 위치를 붙였습니다. 같은 날 확정된 결정 8건에는 미룬 것도 적혀 있습니다. 위치는 2단계까지(3단계 이상은 차기), 콘크리트 시험은 완료 처리만(강도·슬럼프 구조화는 차기), 알림은 앱 안에서만(메일·SMS는 후속)입니다.

결과

5월 20일 PR 9개(#57~#65)로 실행했습니다. 콘크리트는 사용일 28일 뒤 시험 일정을 예약 작업이 알립니다. 미뤘던 메일 알림은 7월 9일에 더했습니다.

대가·남은 위험

재무 화면 12경로와 API 10경로, 테이블을 걷어 냈습니다. 작업 기록 기준으로 단위 테스트 통과 수는 1,855개(5월 18일)에서 790개(5월 20일)로 줄었습니다.

권한은 역할·멤버십·주 보고자 지정으로, 첨부 잠금은 일보 상태로

에이전트 기본값 · 확정 기록 없음

문제

같은 직원이 어떤 프로젝트에서는 일보를 쓰고, 다른 프로젝트에서는 읽기만 합니다. 현장소장도 프로젝트마다 다릅니다. 일보에 붙은 사진·도면·문서는 문서 목록에서 따로 열어 고칠 수 있는 문서이기도 했습니다.

검토한 선택지

문서의 기존 생성자·역할 기준만 따른다 | 별도의 「동결」 플래그를 둔다

✓ 일보 상태를 따른다: 초안·반려는 주 보고자와 디렉터·시스템 관리자, 검토 중은 아무도, 승인 뒤는 디렉터·시스템 관리자만

결정

역할은 시스템 관리자·디렉터·매니저·직원 4개만 두고, 나머지는 프로젝트 안에서 정했습니다. 프로젝트 멤버십, 프로젝트당 현장소장 1명, 그리고 주 보고자입니다. 주 보고자는 역할 4개와 별개인 지정이고, 프로젝트당 활성 1명을 DB 부분 유일 인덱스가 지킵니다(아래 코드). 일보에 붙은 문서의 수정 권한은 일보 상태에 묶었습니다. 결정 기록의 이유는 「일보가 감사 단위」라는 것입니다. 검토 중에는 증빙이 바뀌면 안 되고, 승인 뒤에는 일보 본문과 같은 규칙으로 디렉터·시스템 관리자만 고칩니다. 5월 18일에는 흩어져 있던 권한 판단을 canManageProjectMembers 같은 이름 있는 동사로 모았습니다(경로 5곳).

결과

검토 중인 일보의 사진·도면·문서는 아무도 고치지 못하고(REPORT_UNDER_REVIEW), 승인 뒤에는 디렉터·시스템 관리자만 고칩니다. 일보에 붙지 않은 문서는 원래의 문서 권한을 따릅니다.

대가·남은 위험

현장소장 1명 규칙은 앱 검사(409)가 지키고, 주 보고자 규칙은 DB가 지킵니다. 권한 판단도 두 갈래입니다. 협력사·장비 마스터는 표 기반 RBAC를, 프로젝트·일보는 멤버십 판단 함수를 씁니다.

잠금 판정은 서버 한 곳에서, 화면은 결과만 받는다

에이전트 기본값 · 확정 기록 없음

문제

문서 상세 화면은 잠긴 문서의 편집 버튼을 꺼야 했습니다. 그런데 화면 쪽 추정과 서버의 실제 판정이 달라질 수 있었습니다.

검토한 선택지

화면이 역할·작성자·첨부 종류로 잠금을 다시 계산한다

✓ 서버의 판정 함수 결과(canMutate, lockedCode)를 상세 응답에 싣는다

결정

서버 판정을 응답에 싣기로 했습니다. 결정 기록의 이유는 「화면의 추정이 활성 주 보고자 검사를 빠뜨려, 버튼 상태가 서버와 달라졌다」입니다. 이제 화면과 API가 같은 판정 함수를 씁니다. 같은 원칙으로 권한 예외 값도 JWT에 넣지 않고 요청마다 DB에서 읽습니다. 관리자가 토큰을 다시 발급하지 않고도 권한을 거둘 수 있게 하려는 것입니다.

결과

문서 상세 응답에 canMutate와 lockedCode가 실리고, 화면은 그 값으로 버튼 상태를 정합니다. 같은 방향으로 5월 18일에는 제출 검사 규칙도 서버와 편집기가 한 모듈을 함께 쓰게 했습니다. 규칙 11종 중 서버가 6종을 강제하고, 날씨 누락은 편집기만 봅니다.

장비 수량은 소수 둘째 자리까지 에이전트 기본값 · 확정 기록 없음

문제

고객의 일보 샘플에는 장비 수량이 「0.5일」처럼 적혀 있었습니다. 처음 스키마는 장비의 주간·야간 수량을 정수로 받았습니다.

검토한 선택지

정수로 두고 보고자가 반올림한다

✓ 소수(12,2)로 바꾼다

결정

소수로 바꿨습니다. 결정 기록의 이유는 「정수만 받으면 보고자가 데이터를 지어내야 한다」입니다. 5월 21일에는 오전·오후·추가 세 칸으로 받던 장비 시간을 시작·종료 시각 한 쌍(24시간제, 자정 넘김 허용)으로 바꿨습니다. 오전·점심(제외)·오후·추가 구간과 작업일(작업 시간 ÷ 7)은 표시할 때 계산하고, 기존 행은 마이그레이션으로 채웠습니다.

결과

일보의 장비 행은 대당 번호판과 시작·종료 시각 한 쌍만 받고, 구간과 작업일은 화면이 계산합니다.

일보 PDF는 서버가 데이터로 HTML을 만들어 헤드리스 크롬에서

문제

일보마다 한 장짜리 PDF가 필요했고, 일보에 붙은 도면 PDF도 뒤에 이어 붙여야 했습니다. 같은 시기에 만든 원가 요약 PDF는 브라우저 인쇄로 만들었습니다.

검토한 선택지

브라우저 인쇄(원가 요약 PDF의 방식): 새 런타임 의존성이 없다

인쇄 화면 주소를 헤드리스 크롬으로 열어 찍는다

✓ 서버가 DB 데이터로 HTML을 만들어 헤드리스 크롬에 바로 넣는다

결정

서버에서 만들기로 했습니다(5월 12일). puppeteer-core로 헤드리스 크롬을 띄우고, 도면 PDF는 pdf-lib로 뒤에 붙입니다. 처음에는 인쇄 화면 주소로 이동해 찍었는데, 헤드리스 브라우저에는 로그인 맥락이 없어 로그인 화면이 찍혔습니다. 다음 날 DB 데이터로 서버에서 HTML을 만들어 바로 넣는 방식으로 바꿨습니다. 만든 PDF는 파일로 캐시하고, 일보를 바꾸는 경로 18곳에서 캐시를 지웁니다. 같은 일보의 동시 생성은 하나로 합치고, 30초가 넘으면 끊습니다.

결과

다운로드는 캐시가 있으면 바로 내려가고, 없으면 그 자리에서 만듭니다. 일보 한 건이 도면까지 붙은 PDF 한 파일이 됩니다.

대가·남은 위험

브라우저 인쇄안이 피하려던 런타임 의존성이 생겼습니다. 5월 13일에는 배포 번들에서 크롬 파일이 빠지고 글꼴이 없어 글자가 깨져, 파일 추적 설정과 글꼴 번들로 고쳤습니다.

품질과 검증

테스트 1,001개 통과가 작업 기록에 남아 있지만, 테스트 규칙은 에이전트 지시문으로만 있었고 야간 QA는 실패를 삼키게 짜여 있었습니다.

테스트는 정적으로 세면 단위 45파일(테스트 블록 542개), 통합(API) 28파일(255개), 컴포넌트 23파일(190개), e2e 45개 스펙(429개)입니다. e2e는 데스크톱과 모바일 두 환경으로 나뉩니다. 이 사례를 쓰면서 다시 돌리지는 않았습니다.

실행 결과는 대부분 커밋 본문에 남은 작업 기록입니다. 5월 25일 한 커밋에는 단위·통합·컴포넌트 테스트 1,001개 통과가 적혀 있습니다. 같은 날 다른 커밋에는 데스크톱 e2e 358개 통과, 불안정 4개, 건너뜀 54개, 실패 0개가 적혀 있습니다. 5월 21일 커밋에는 사용 사례 33개 중 28개를 전부, 5개를 일부 덮어 e2e 범위가 약 85%라고 적혀 있습니다(작업 기록 기준).

5월 24일에는 낡은 e2e 실패 86건을 정리했고, 다시 써야 하는 테스트 43개(스펙 12개)는 건너뜀으로 남겼습니다.

실행 기록 파일이 남은 것은 하나입니다. 5월 21일 시드 데이터로 돌린 데모 배포를 대상으로 스모크 테스트 6개가 모두 통과했습니다. 5월 20일부터는 야간 GitHub Actions가 같은 배포에 역할별 화면 QA를 돌렸습니다. 다만 QA 단계가 비정상 종료를 삼키게 짜여 있어, 실행의 「성공」은 QA 통과를 뜻하지 않습니다.

파이프라인 규칙(테스트 먼저, 커버리지 80%, 리뷰어·QA·매뉴얼을 통과한 뒤 커밋)은 지시문으로만 있었습니다. 에이전트 훅도, PR에서 도는 CI도 없었습니다.

리뷰 기록은 둘 있습니다. 5월 18일 에이전트 명령으로 돌린 아키텍처 리뷰에서 나온 후보로 리팩터 PR 5개(#47~#51)를 냈습니다. 제출 검사 규칙을 서버와 편집기가 한 모듈로 공유한 것과 권한 판단을 이름 있는 동사로 모은 것이 여기서 나왔습니다. 5월 25일에는 디자인 리뷰 지적 6건을 반영했습니다.

기능 하나, 처음부터 끝까지

일보 한 건이 주 보고자 지정에서 PDF와 실적 합산까지 어떻게 지나가는지 보여 드립니다.

1. **지정 — 디렉터**

   디렉터가 프로젝트를 만들고, 현장소장을 지정하고, 직원을 넣고, 주 보고자를 지정합니다. 새로 지정하면 이전 활성 기록을 닫고, 이전 주 보고자와 새 주 보고자 모두에게 알림을 보냅니다.

2. **초안 — 주 보고자**

   주 보고자가 오늘 초안을 만듭니다. 일보는 프로젝트와 날짜마다 하나입니다. 전날의 「내일 계획」이 오늘의 「금일 실적」으로 미리 채워집니다.

3. **저장**

   저장은 막지 않습니다. 행 변경은 공통 레시피를 거칩니다. 편집 잠금을 확인하고, 바꾸고, 승인 상태였으면 수정 시각을 남기고, PDF 캐시를 지우고 감사 로그를 씁니다. 다른 프로젝트의 공종·위치·협력사 id는 이 단계에서 거절합니다.

4. **제출**

   편집기가 공유 규칙 11종으로 먼저 막고, 서버는 그중 6종을 다시 검사합니다. 통과하면 상태가 SUBMITTED로 바뀌고, 현장소장에게 받는 사람의 언어로 알림이 갑니다. 응답을 보낸 뒤 날씨 일 요약을 가져와 붙입니다.

5. **검토 중 잠금**

   검토가 끝날 때까지 일보에 붙은 사진·도면·문서는 아무도 고치지 못합니다(REPORT_UNDER_REVIEW).

6 **반려와 재제출**

현장소장이 반려하려면 사유를 적어야 합니다. 주 보고자의 편집기에는 그 사유가 보이고, 제출 버튼이 「재제출」로 바뀝니다. 재제출 횟수에는 제한이 없습니다.

7 **승인**

승인된 일보는 읽기 전용입니다. 디렉터만 그 자리에서 고칠 수 있고, 상태는 그대로 두며 알림은 보내지 않습니다. 수정 시각과 감사 로그는 남습니다.

8 **PDF**

캐시가 있으면 바로 내려 줍니다. 없으면 서버가 HTML을 만들어 헤드리스 크롬에서 PDF로 찍고, 도면 PDF를 뒤에 붙여 캐시합니다.

9 **실적 합산**

승인된 행만 프로젝트의 자재 사용 내역과 작업계획 타임라인의 실적에 더합니다.

10 **예약 알림**

마감 시각까지 제출이 없으면 예약 작업이 주 보고자와 현장소장에게 알립니다. 콘크리트를 쓴 날로부터 28일 뒤에는 시험 알림이 갑니다.

회고

만든 기능을 두 번 범위에서 뺐고, 기록된 사고마다 원인과 바뀐 것을 남겼습니다.

만든 뒤 범위에서 뺀 것이 두 번입니다. 일정·EVM은 4월 19~20일에 만들어 5월 9일에 보관했고, 재무는 5월 11~12일에 만들어 5월 20일에 보관했습니다. 재무 계획 단계에서 저는 「전부 하고, 끝에 확인하겠다」고 맡겼습니다. 두 번에 걸쳐 보관함으로 옮긴 파일은 251개입니다.

문서가 코드보다 늦게 따라온 일이 있었습니다. 매뉴얼 캡처 스크립트가 404·403 페이지를 오류 없이 넘겨, 일보 전환 때 없어진 기능의 화면을 매뉴얼이 계속 가리켰습니다. 5월 24일 스크립트가 실패하면 바로 멈추게 고치고, 매뉴얼 12종을 처음부터 다시 썼습니다.

리팩터가 검사 하나를 지운 적이 있습니다. 5월 10일 작업 행의 공종을 선택 항목으로 바꾸면서, 다른 프로젝트의 공종·위치·협력사 id를 일보 행에 넣지 못하게 막던 유일한 검사가 사라졌습니다. 5월 25일 저장 단계에 프로젝트 범위 검사를 넣어 라우트 7개에서 막았습니다.

결정 기록 머리말에는 「7단계 뒤에 다시 보고 고친다」고 적었지만, 다시 본 기록은 없습니다. 게이트도 지시문뿐이었습니다. 그 뒤에 시작한 급여 선지급 플랫폼에서는 훅,

보안 리뷰어, 스토리 ID, 결정 표시를 새로 더했습니다.

사고에서 바뀐 것

기록된 사고마다 원인과 바뀐 것이 남았고, 5월 13일 하루에 배포·PDF 사고 다섯 건을 고쳤다.

5건
5월 13일 하루에 고친 배포·PDF 사고



1 — 2026년 4월 7일
빌드에서만 나는 타입 오류
로컬은 타입 검사를 건너뛰었다 → 빌드에 prisma generate, 타입 경로 통일

2 — 2026년 5월 13일
배포·PDF 사고 5건
DB 연결 한도 초과, 빈 DB로 나간 데모 배포, 크롬 파일 누락, 로그인 화면이 찍힌 PDF, 글꼴 누락 → 연결 풀 조정, 시드 명시 호출(사용자 0명이면 중단), 파일 추적 포함, 서버에서 HTML 생성, 글꼴 번들

3 — 2026년 5월 22일
문서 등록 페이지 404
일보 전환 때 같이 빠진 페이지 복구

4 — 2026년 5월 24일
매뉴얼이 없는 기능을 가리킴
캡처 스크립트가 404를 삼켰다 → 실패하면 바로 멈추게, 매뉴얼 12종 다시 씀

5 — 2026년 5월 25일
다른 프로젝트 id가 행에 들어가던 문제
5월 10일 리팩터가 지운 검사 → 저장 단계 프로젝트 범위 검사 (라우트 7개)

6 — 2026년 5월 30일
스테이징 첫 구축 함정 11개
런북 v0.3에 반영

일하는 방식

이 프로젝트도 역할별 AI 에이전트 팀으로 만들었습니다. 팀 구성과 게이트는 프로젝트마다 다르고, 이 프로젝트의 것은 위 「제 역할」에 적었습니다. 아래는 EWA 플랫폼을 만들며 정리한 방식입니다.

2026년 EWA 플랫폼을 만들며 세우고 운영한 방식 보기 →

https://jake-jo-portfolio.vercel.app/ko/process/

연락처

whxoans@gmail.com
+82 10-9233-7440