AI Site Visit Reporting System – System Design Document

1. Overview

Architecture firms often rely on voice notes recorded by site engineers during field visits. However, converting these raw audio recordings into structured, actionable reports is time-consuming, inconsistent, and error-prone.

This system uses AI to automate transcription, analysis, and report generation from site visit audio and floor plans. The goal is to streamline field documentation, improve communication between stakeholders, and reduce manual report creation time.

2. Architecture and Tech Stack

The architecture is composed of the following components:

- Flutter-based mobile app for site engineers to upload audio and floor plans.
- Audio and documents are stored in Amazon S3.
- AWS SQS queues trigger background transcription jobs using Whisper.
- The transcribed text is analyzed by GPT-4o Mini to extract observations, decisions, and action items.
- Floor plan PDFs are OCR-processed and cross-referenced with extracted insights.
- Data is stored in a PostgreSQL database.
- Office staff reviews reports on a React-based dashboard.
- System metrics are monitored using Prometheus and Grafana.

Tech Stack:

Component	Technology	Reason
Mobile App	Flutter	Cross-platform UI, single codebase
Dashboard	React (Next.js)	Fast, modular, production-ready
Transcription	Whisper	High-accuracy speech-to-text
AI Analysis	GPT-4o Mini	Structured extraction from natural language
Queueing	AWS SQS	Reliable async task dispatch

Database	PostgreSQL	Relational storage with strong consistency
Monitoring	Prometheus + Grafana	Real-time observability

3. Core API Endpoints

- POST /visits – Create a new site visit entry.
- POST /upload/audio – Upload and trigger transcription of an audio note.
- POST /upload/floorplan – Upload a PDF floor plan for the visit.
- GET /visits/{id}/report – Retrieve the final AI-generated visit report.
- PATCH /visits/{id}/actions – Update action items post-review.
- GET /projects – View all projects and their visit history.
- GET /status/{id} – Check processing status of a visit.

4. Database Tables

- projects – Project metadata.
- visits – Links to projects and users, stores visit-level data.
- audio_files – Audio file storage references.
- floor_plans – Links to floor plan PDFs.
- ai_insights – Output from LLM (summary, metadata, decisions).
- action_items – Room-wise actions, contractors, and responsibilities.

5. AI Workflow

After an audio file is uploaded, Whisper is used for transcription. The resulting text is parsed by GPT-4o Mini, which extracts structured fields such as room-wise issues, stakeholder decisions, and follow-up actions. This output is stored in the database.

If a floor plan is uploaded, room names are extracted using OCR and matched to the transcription via fuzzy logic to enrich the report with spatial references.

The dashboard allows users to view, edit, and validate AI-generated data. Any inconsistencies or failures (e.g., bad transcription or JSON errors) are handled with retries or surfaced for manual review.