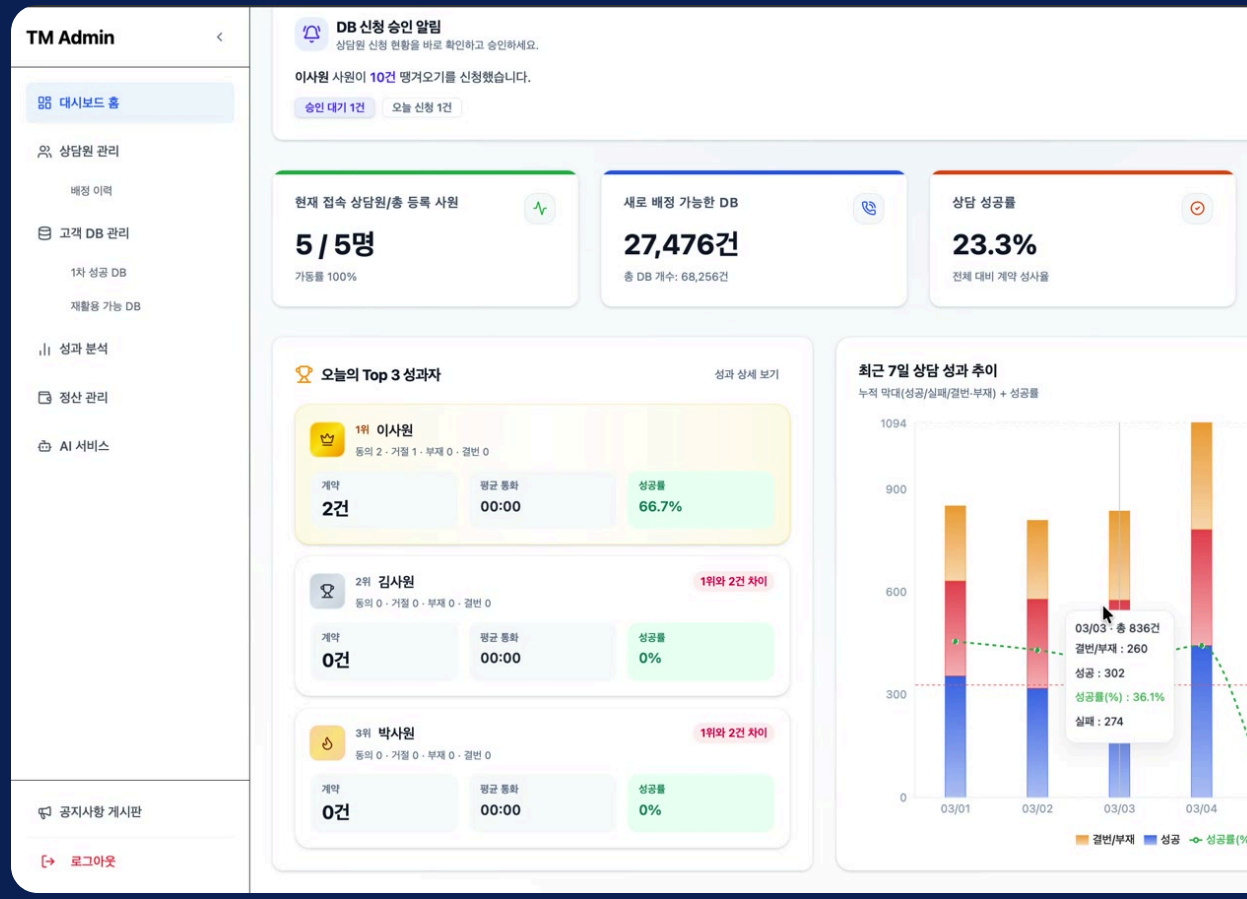
FULL-STACK CASE STUDY

GV TM Call Management Platform

One system for a sales call team: a web console for managers, an Android app for agents, and a backend that assigns leads, logs calls, and calculates pay.



Seonmin Kim
Sole developer, Global Vision Co., Ltd.
Jan to Mar 2026

THE PROBLEM

A call team run on spreadsheets and chat messages

BEFORE

By hand

Managers split customer lists in Excel every morning. Agents reported call results by chat. Pay was added up manually at the end of the period.

AFTER

One system

Leads are assigned automatically at 9:00. Agents save each result from their phone. Pay is calculated from the call logs.

3
surfaces: web, Android, API

121
commits and 21 pull requests in 8 weeks

7
backend domain apps, 10 data models

6
Docker services in one compose file

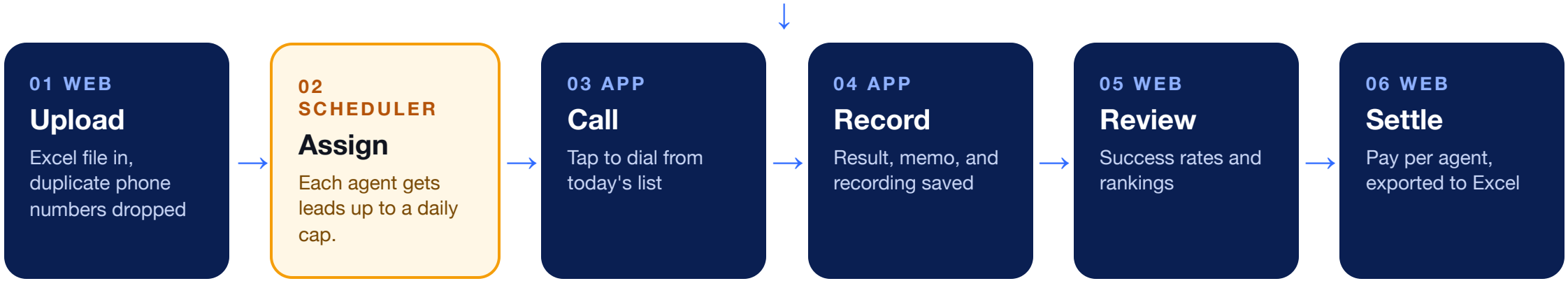
HOW IT WORKS

The life of one lead, in six steps

Admin web console
React, TypeScript, served by nginx

Agent Android app
React Native (Expo), built to APK

One REST API
Django REST Framework, JWT login



PostgreSQL

Celery + Redis workers

AWS S3 for recordings

One API for both clients. A call saved on the phone changes the dashboard and the pay calculation at once, with no sync job in between.

WHAT WAS HARD

Three problems I had to solve

0

leads given to two agents

The scheduler and a manager can assign at the same moment. Assignment runs in a transaction that locks rows and skips the ones already locked.

14 days

before a lead can come back

A declined lead returns after a cooldown, at most twice, and each daily batch mixes about 30% recycled leads. These are settings, not code changes.

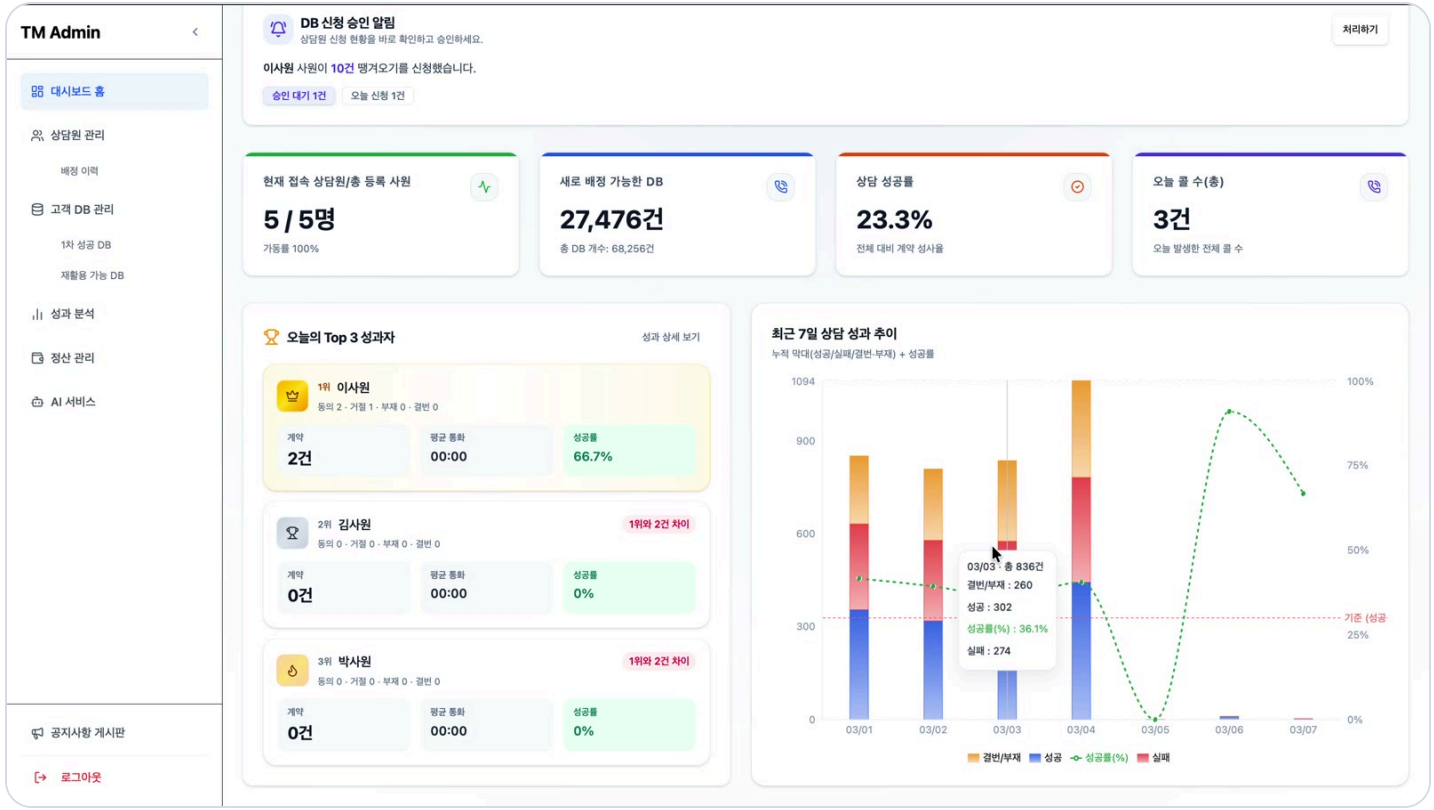
9:00

on working days only

The daily job runs Monday to Friday and checks the Korean public holiday calendar, so nobody gets a list on a day off.

The whole floor on one dashboard

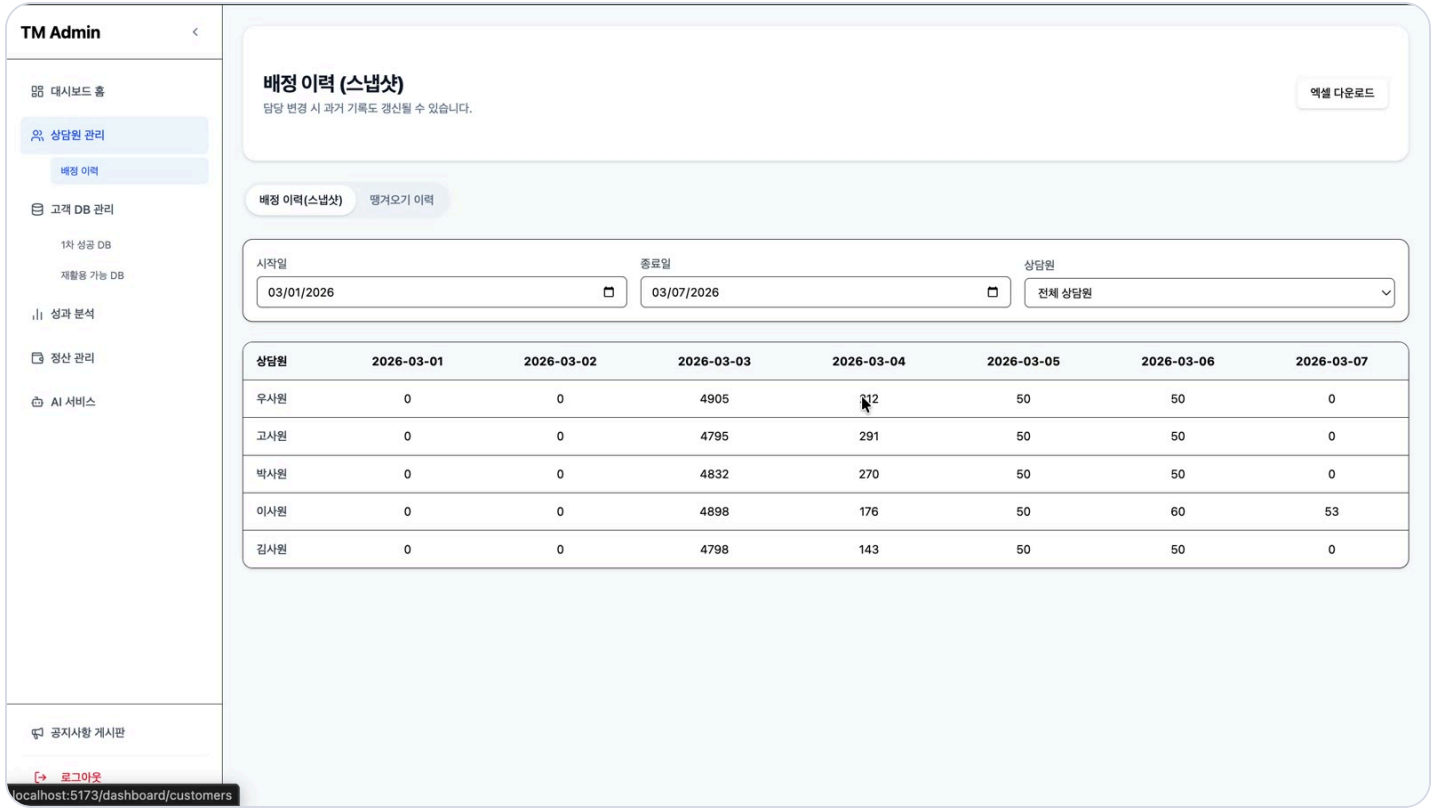
- **Top cards:** agents online, leads still available, success rate, calls today
- **Top 3** agents of the day
- **7-day chart** of results with the success rate as a line
- **Banner** when an agent asks for more leads



The interface is in Korean because the users were a Korean sales team. All screens show seeded test data.

A daily record of who got how many leads

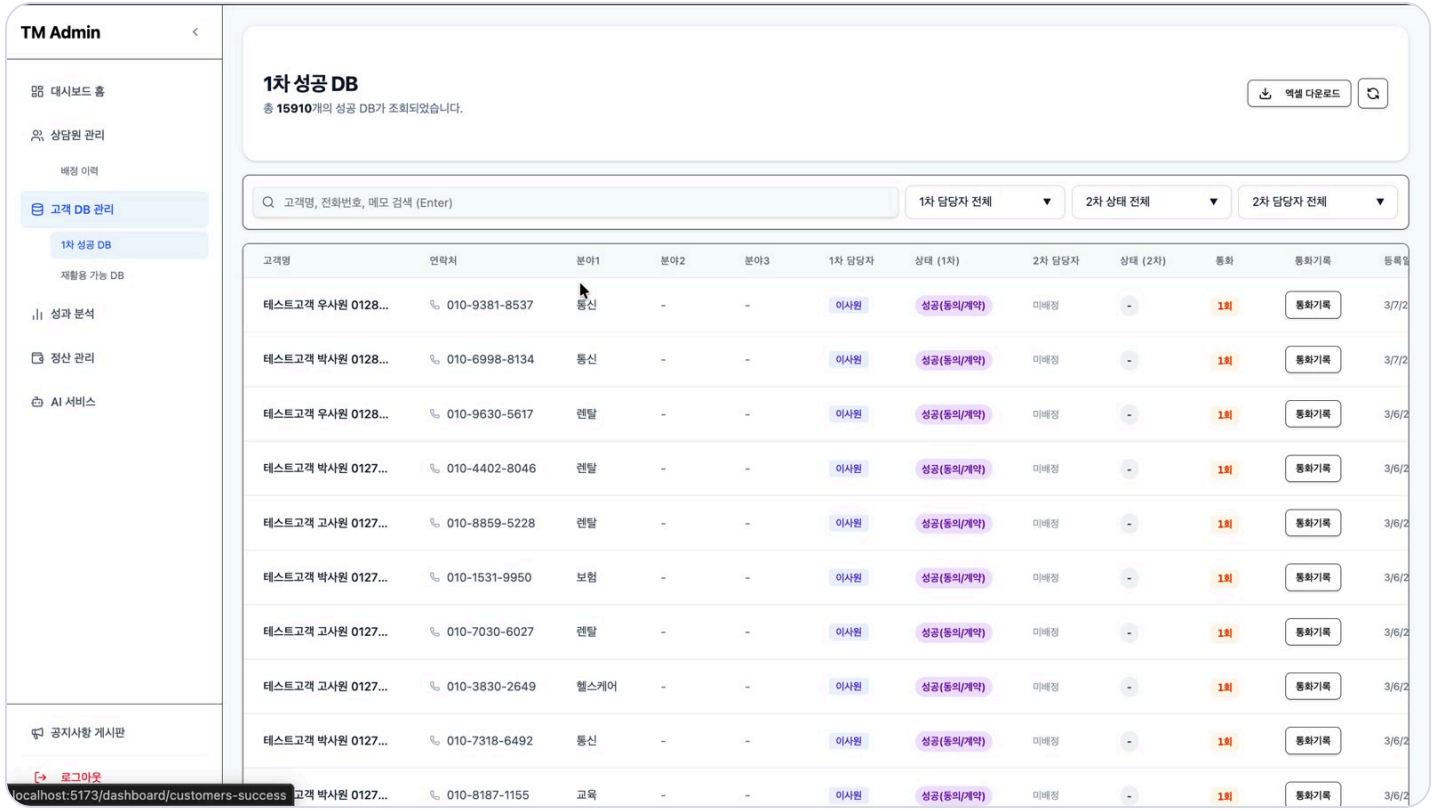
- **One row per agent,** one column per day
- **Snapshot** saved at assignment time, so history stays true after changes
- **Filter** by date range and agent
- **Excel download** for reporting



The interface is in Korean because the users were a Korean sales team. All screens show seeded test data.

Every successful call, searchable

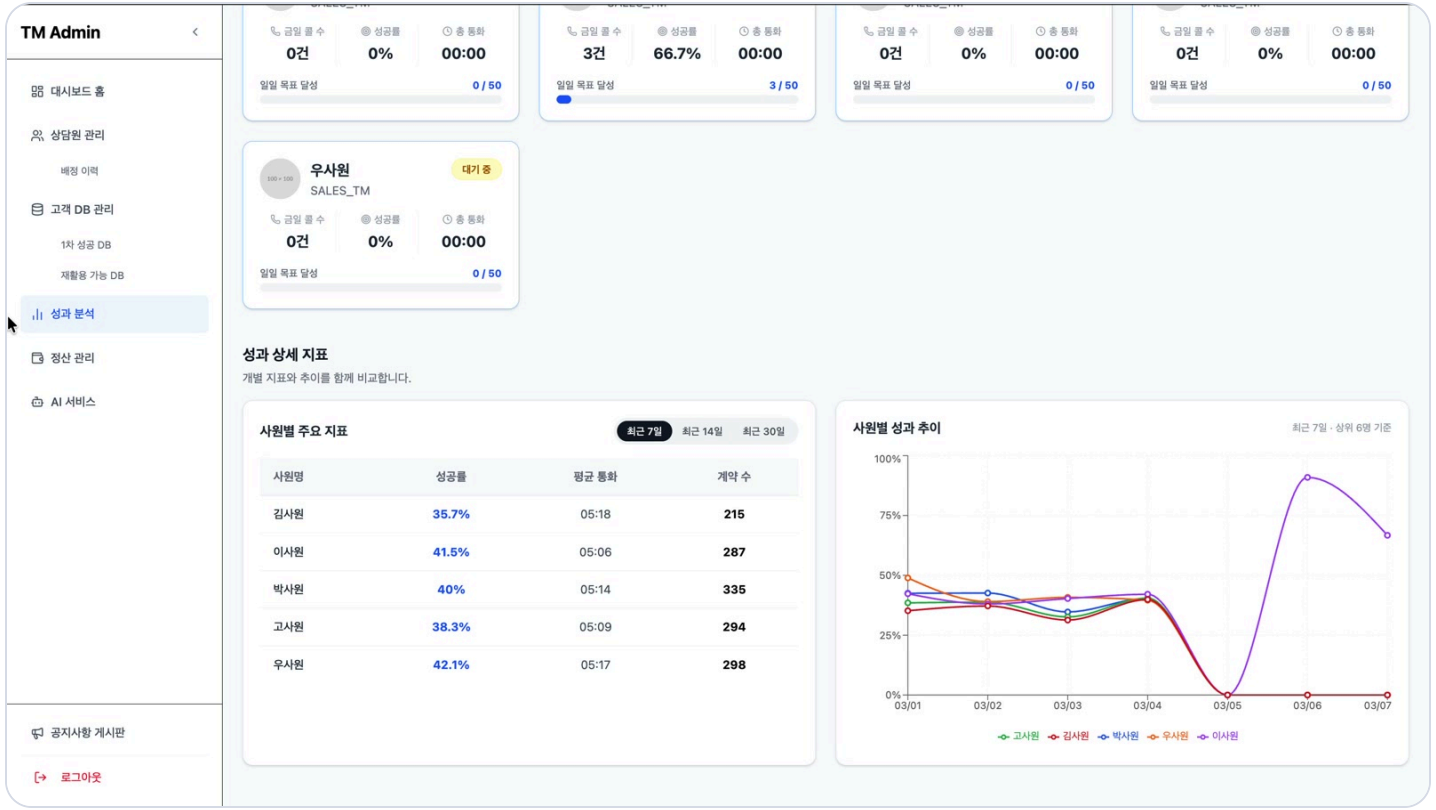
- **Search** by name, phone number, or memo
- **Filter** by first agent, status, and second-stage owner
- **Call history** opens per customer
- **Handoff** to a second sales stage is tracked in the same table



The interface is in Korean because the users were a Korean sales team. All screens show seeded test data.

Performance per agent, compared fairly

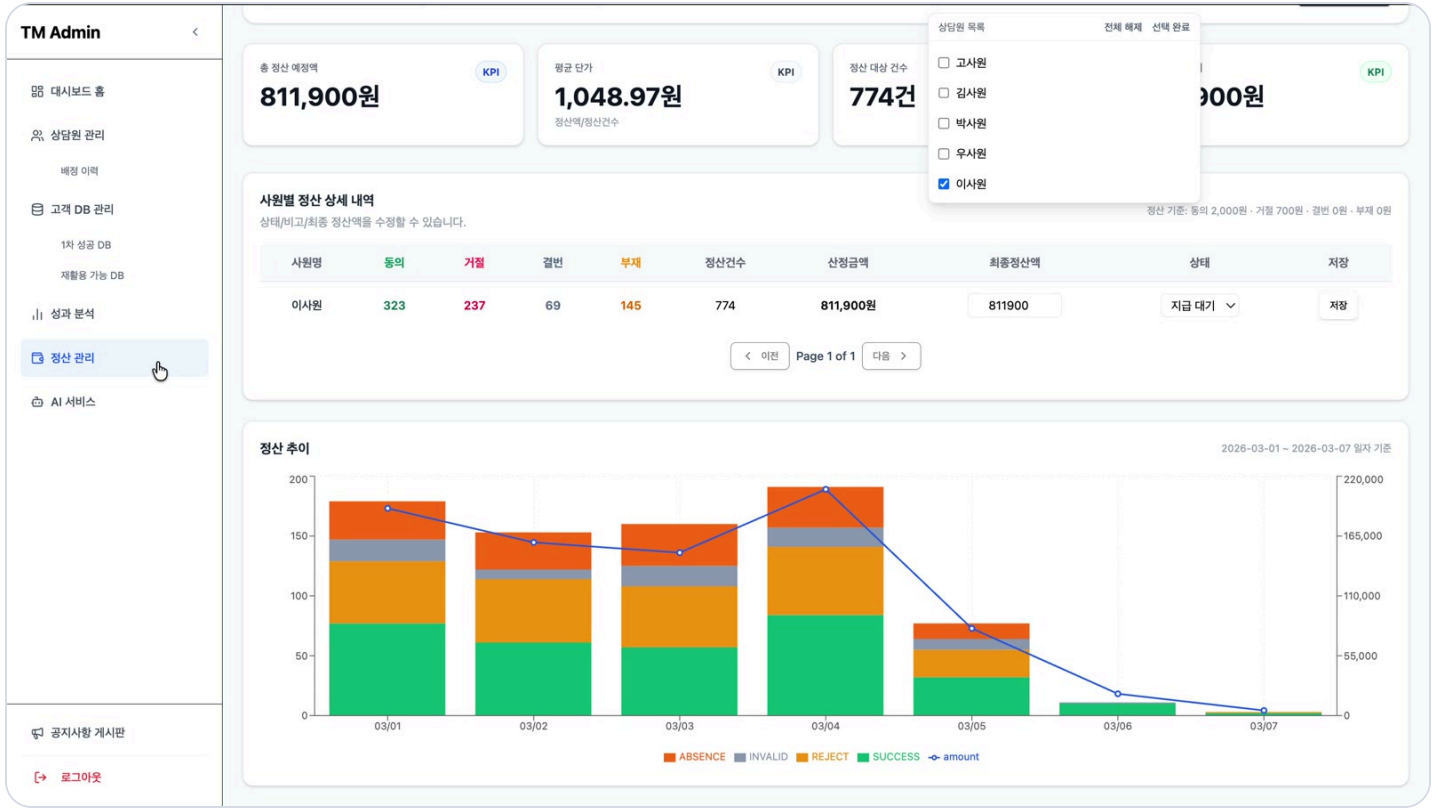
- **Success rate,** average call time, and contracts per agent
- **Window** of 7, 14, or 30 days
- **Trend lines** put every agent on the same chart
- **Daily goal** progress on each agent card



The interface is in Korean because the users were a Korean sales team. All screens show seeded test data.

Pay calculated from the call logs

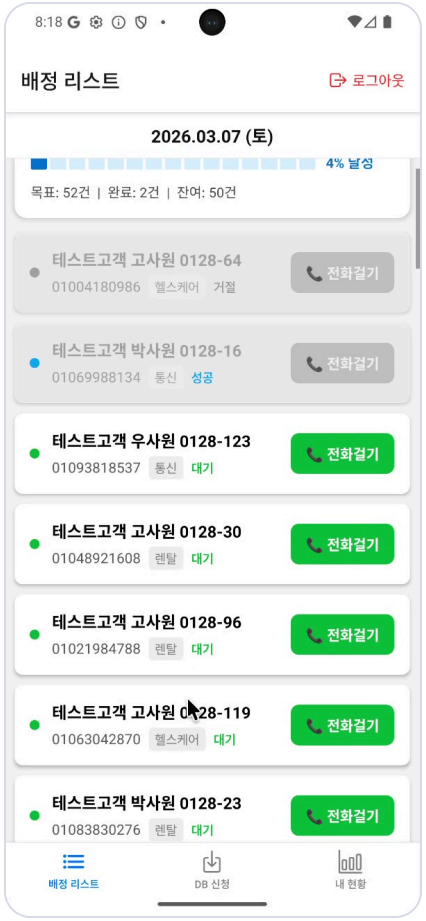
- **Counts by result** for each agent become the amount
- **Unit prices** are stored in the database and editable by managers
- **Payment status** is set per agent
- **Daily trend** chart and Excel export



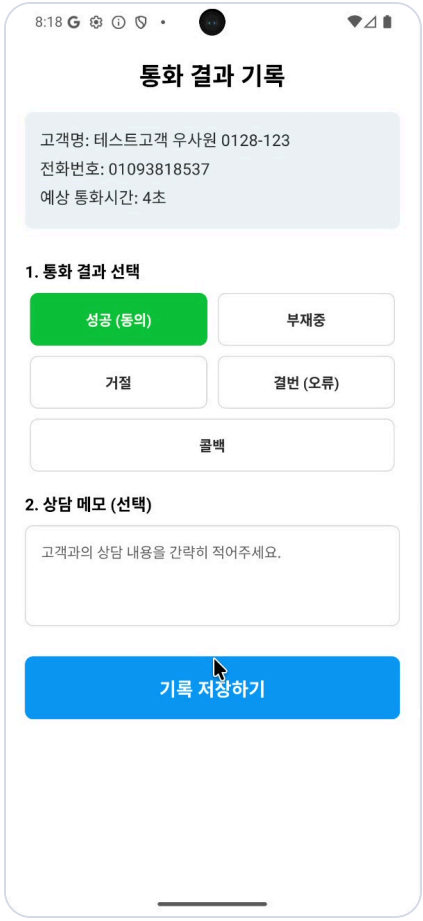
The interface is in Korean because the users were a Korean sales team. All screens show seeded test data.

Call, then record, in two taps

- **Today's list** shows the daily goal and each assigned customer
- **Tap to call.** Finished calls turn grey
- **One tap** for the outcome: agreed, no answer, declined, wrong number, call back
- **Recording** uploads straight from the phone to S3



Today's list

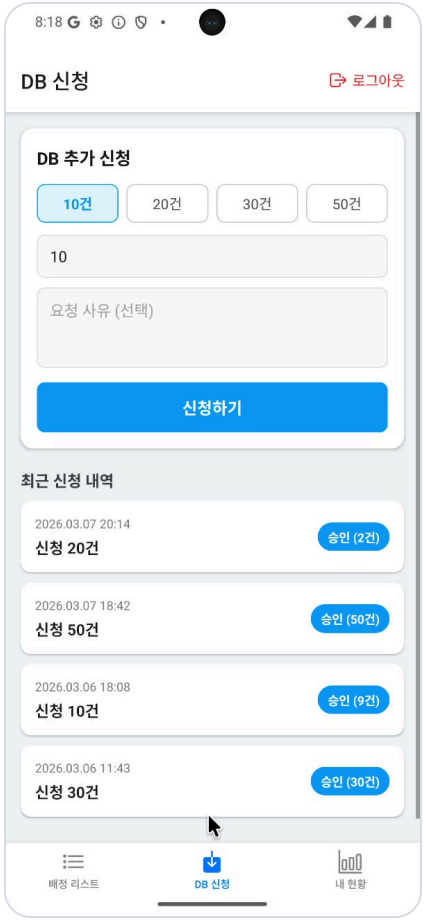


Call result

The interface is in Korean because the users were a Korean sales team. All screens show seeded test data.

Agents ask for leads and see their own numbers

- **Request** 10 to 50 more leads when the list runs out
- **Status** shows whether the manager approved, and how many
- **My stats** by day, week, month, or a custom range
- **Result breakdown** as a chart



Lead requests



My stats

The interface is in Korean because the users were a Korean sales team. All screens show seeded test data.

BEHIND THE SCREENS

How it was built and shipped

Background workers

Uploading thousands of rows inside a web request timed out. The upload now returns at once and a Celery worker parses the file.

Direct upload of recordings

The phone gets a short-lived signed link and sends the audio file straight to S3, so large files never pass through the API server.

One command to deploy

Six services in one Docker Compose file on the company server, redeployed by GitHub Actions. The Android app was built to an APK with Expo.

My role: everything

Requirements with the sales managers, database design, API, web console, Android app, deployment, and the user manual.

Django REST Framework · PostgreSQL · Celery · Redis · React · TypeScript · React Native · AWS S3 · Docker

Video **youtu.be/kht4BrjXxxQ**

Code **github.com/nimnusmik/gvms_TM**