

Technical Operations Engineer

Take-Home Assignment: Property & Unit Onboarding Tool

Background

Our customer success team helps onboard new customers by importing their property and unit data into our system.

Customers usually send us a spreadsheet export from their property management system. Even when we provide a template, the files are not always perfect, and small issues can lead to duplicate properties, incorrect units, or manual cleanup after the fact.

We'd like an internal tool that makes this process safer and easier for the team doing the onboarding.

The Request

This is roughly how the request might come through from customer success:

“When I get a property spreadsheet, I want to be able to upload it, quickly sanity-check what’s about to be added, fix anything that looks wrong, and feel confident I’m not creating duplicate properties or bad data.”

The details are intentionally a little fuzzy. Part of this exercise is deciding what needs to exist to support that experience.

What You’re Building

Build a small Ruby on Rails application that helps an internal user import properties and units from a CSV file.

You can assume the CSV will always be in our expected format (an example file is attached).

The app should include, at minimum, the following models:

Property

- Represents a single building

- A property can have many units
- The building name can be treated as a unique identifier for a property

Unit

- Belongs to a property
- Has a unit number
- Units are created based on the rows in the uploaded CSV

Beyond this, the structure and flow of the application is up to you.

How the Tool Should Feel to Use

From the perspective of someone on the customer success team, the tool should make it easy to:

- Upload a CSV file and understand what the system thinks it is going to import
- Catch obvious mistakes before anything is permanently saved
- Avoid accidentally importing the same property more than once
- Feel confident when finalizing an import

Exactly how you accomplish this is intentionally left open-ended.

Technical Notes

- Please use Ruby on Rails
- A relational database is fine (Postgresql is acceptable)
- The user interface can be simple and unpolished
- Authentication is not important — assume this is an internal tool

Make reasonable assumptions where things are unclear, and document them.

What We're Looking For

We are not looking for a production-ready system. We care more about:

- How you interpret vague, real-world requests
 - Thoughtful handling of data safety and duplication
 - Technical decisions that feel appropriate for an internal operations tool
 - Clear communication through code and documentation
-

Deliverables

Please provide:

- A GitHub repository with your source code
- A README that explains:
 - How to run the application locally
 - Any assumptions or tradeoffs you made
 - How you thought about identifying duplicate properties
 - What you would improve if this were used daily by the team

Optional:

- Screenshots or a short walkthrough of the tool in use
 - If you choose to deploy the application somewhere and share a link, we're happy to look at it. This is entirely optional and not required for evaluation — a local setup is more than sufficient.
-