PRODUCT BUILD | QA | UX | APPLICATION MODERNIZATION

Major Property Rental Provider

How We Built an AI Chatbot That Transforms 10,000+ Daily Maintenance Requests into Clear, Actionable Jobs in Seconds

Our client is one of the largest single-family rental housing providers in the United States. As a publicly traded company operating thousands of homes across more than twenty states, they manage a vast volume of day-to-day resident service requests.

gravity9 has been a trusted partner for many years. We previously helped modernize their field operations with a mobile app used by technicians to record and manage property issues. As the platform grew, the client faced a recurring operational problem: residents submit maintenance requests in completely free-text form, often unclear, sometimes in Spanish, always inconsistent.

The client needed a way to turn this raw, messy natural language into clean, structured maintenance data. They asked us to design an AI chatbot that could understand residents the way a human would, and then convert every request into a precise, standardized work order ready for scheduling.

Utilized Technology Stack

Cloud: Microsoft Azure

Frontend Framework: React

Backend Framework: ASP.NET Core

AI Framework: Semantic Kernel

LLM Model: GPT-5 Mini

Vector Database: Microsoft CosmosDB

Review of Challenges

The request are typically submitted in natural language, often without formal structure, leading to significant operational challenges. Equally, they ranged from rambling paragraphs to three word messages. Some used slang, some mixed languages, others left out essential details.

The client's operations team had to read, interpret, translate, classify, prioritize, and validate every single message manually; thousands per day. Even small inconsistencies resulted in duplicated tickets, incorrect categorization, or reduced response times.

The new system needed to:

- read what a resident writes
- ask follow-up questions
- translate when required
- identify one or more maintenance issues
- match them to the client's official classification
- detect duplicates
- package everything into a structured, consistent request. Whilst ensuring it happened in real time.

Our Solution

We built an AI chatbot that acts like an expert maintenance coordinator. Residents can describe a problem in their own words. The chatbot reads it, clarifies it, translates it if needs be, and classifies it using the client's approved condition and asset definitions.

Every decision the chatbot makes is grounded in the client's controlled database meaning it doesn't invent new categories or guess outside defined rules. It simply interprets human language and maps it to the company's world.

The result: residents speak naturally, and the system delivers clean, accurate maintenance requests.

Our Approach

Because AI ecosystems change almost monthly, we followed a careful, structured path.

We began with a proof of concept using GPT-4.1 Mini to see if the model could handle translation, classification, summarization, and follow-up questioning using real resident examples. Once GPT-5 Mini was released, we upgraded to increase speed and accuracy which is critical for chat-based workflows.

Next, we evaluated two leading agent frameworks: LangChain and Semantic Kernel. Semantic Kernel aligned better with the client's .NET-based ecosystem, so we selected it, while also preparing for future migration to the Microsoft Agent Framework once it stabilizes.

We explored retrieval-augmented generation (RAG) versus prompt-only grounding to determine which method produced the most accurate classifications. These experiments helped us optimize both the AI behavior and the database design.

After validating the approach, we entered a full compliance and security review. The client operates at national scale, handling sensitive resident data, so we implemented strong controls through Azure AI Foundry including Managed Identity, Private Endpoints, Content Safety, and role-based access security.

Once approved, we transformed the proof of concept into a production ready platform. We built a responsive chat interface, restructured the POC into micro-services, added duplicate detection AI, instrumented telemetry using Grafana, and tested extensively with both QA teams and real resident data.

The rollout occurred gradually using LaunchDarkly feature flags so we could monitor performance and adjust behavior before expanding system wide.

Subsequent Outcomes

The finished solution is an AI-driven conversational assistant that takes unstructured human language and turns it into clean, consistent maintenance requests.

Residents now enjoy a simple dialogue-based experience where the AI asks only the questions needed to understand their problem. They no longer need to decode the company's internal categories; the system handles it entirely.

Behind the scenes, maintenance teams receive structured, accurate, duplicate checked requests with clear summaries, asset matches, and priority classifications. This improves triage speed, scheduling efficiency, technician preparedness, and overall service consistency.

The chatbot effectively bridges the gap between how people talk and how operations run.

Client Feedback

The client praised the clarity and consistency the chatbot brings to their maintenance workflows. Early stage reviewers described it as "a major step forward for resident experience and operational accuracy."

They also highlighted the strength of the collaboration and the disciplined way complex AI behaviors were simplified into predictable, business-ready outcomes.

"10,000+ daily maintenance requests transformed from free text into clean, actionable jobs, in seconds."

WE'RE BUILDING BETTER DIGITAL PRODUCTS