Project: Abu Taxi – Automated Document Notification & Follow-up System

---

Overview

Client: Abu Taxi — a taxi company operating in Poland with a multilingual driver workforce (Polish, Ukrainian, English speakers)
Deliverable: A fully automated outbound calling system that notifies drivers when documents are ready, reminds them to come to the office, and manages follow-up campaigns across four distinct document types — all driven by a Google Sheets backend with zero manual dialing required.

Problem Statement

Abu Taxi manages a large base of drivers who need to regularly visit the office to sign invoices, wage statements (KW), and legal residency documents. The office team was manually calling drivers to remind them — a repetitive, time-consuming task that was inconsistent, multilingual (Polish, Ukrainian, English), and impossible to scale. Key problems:

- Drivers would miss calls or not respond, requiring multiple follow-up attempts with no systematic tracking
- Different document types (outstanding invoices, current invoices, permits, faktury) had different urgency levels and follow-up rules, but there was no system to handle them separately
- If a driver didn't show up within 7 days, the process had to restart — and this was being tracked manually on spreadsheets with no automation
- The team had no visibility into whether a call was answered, went to voicemail, or resulted in a commitment to come in
- Drivers spoke different languages and a single-language reminder system wasn't effective

---

Solution

We built a Google Sheets-triggered outbound calling campaign using VAPI as the voice layer and n8n as the orchestration engine. The system handles four separate document workflows, each tied to its own Google Sheet tab, with a shared AI agent (Elliot) making the calls.

The Voice AI Agent (VAPI) — Elliot

The agent is named Elliot and is designed as a professional but warm notification assistant. The system prompt defines a strict scope: Elliot informs, confirms, and records — nothing more. He does not engage in lengthy conversations, does not make promises about legal or processing timelines, and redirects anything outside his scope to a human team member.

The call flow is structured in four steps:

Step 0 is a greeting. Step 1 is identity confirmation — Elliot first verifies he is speaking to the right person by name before delivering any information. If someone else picks up, he leaves a callback request. If voicemail is detected, he leaves a short scripted message and logs the outcome. This identity check was a deliberate design requirement to protect client data and avoid delivering sensitive document information to the wrong person.

Step 2 is the document notification — delivered concisely with office address and hours. Step 3 captures pickup intent — Elliot asks whether the person plans to come today, tomorrow, or another day, and logs the response as a structured value (today, tomorrow, a specific date, undecided, or no_response). Step 4 closes the call cleanly, handles any in-scope questions, escalates to a human callback if the caller is frustrated, and always ends with the `end_call` tool.

The agent is fully multilingual — English, Polish, and Ukrainian — with the language variable injected at call time. Elliot never identifies himself as an AI unless sincerely asked, at which point he says he is an automated assistant from Abu Taxi.

At the end of every call, Elliot logs a structured data payload: client name, document type, reached_client status, call language, pickup intent, escalation flag, and timestamp. This feeds back into the workflow for tracking.

The n8n Automation Backend

The backend consists of three functional layers: outbound campaign management, post-call logging, and the 7-day re-trigger system.

Outbound Campaign Triggers (4 separate sheets, same pattern)

Four Google Sheets Trigger nodes poll every minute, one for each document sheet: Outstanding Invoices (zaległe braki Rachunki i KW), Current Invoices (Bieżące Rachunki), Permits (pozwolenia), and Faktury (invoices). When a row changes, the workflow waits 1 minute, then filters only rows where the activator column equals "yes" — this is the checkbox the office staff ticks to authorize a call campaign for that driver.

Filtered rows are passed into a batch loop (SplitInBatches, 8 per batch). For each batch, a Phone Number Normalizer code node strips whitespace and ensures international format (prefixes with + if missing). The VAPI call is then fired via POST to the VAPI API, with the assistant ID, phone number ID, client name as a variable override, and metadata containing the row number and sheet name. The row number and sheet name in metadata are critical

— they allow the post-call webhook to write results back to exactly the right row in exactly the right sheet.

After firing calls, the workflow waits 2.5 minutes, then polls VAPI to check call status. If all calls in the batch have ended, the next batch begins. If any are still in progress, it waits and re-checks. This loop continues until the entire campaign is complete.

Post-Call Logging

When a call ends, VAPI fires a webhook to the `postcall-abu-taxi` endpoint. The workflow checks the event type — it handles both `end-of-call-report` (for logging) and `assistant.started` (for tracking). On end-of-call, it extracts the row number, sheet name, ended reason, and phone number from the call metadata, then uses a Switch node to route the update to the correct Google Sheet tab and writes the call timestamp back to the "Data rozmowy" (call date) column, matched by phone number.

Incoming Call Handler

A separate webhook (`incoming-abu-taxi`) handles cases where a driver calls back after receiving a notification. When an inbound call arrives, the workflow checks which sheet the driver belongs to by sequentially querying all four sheets and matching their phone number. Once matched, it responds to VAPI with the assistant ID and the relevant sheet metadata so Elliot handles the callback with the correct context. If the number isn't found in any sheet, a fallback response is still returned.

7-Day Re-trigger System

A weekly Schedule Trigger fires every 7 days. It reads the Outstanding Invoices sheet, filters for rows where the last call was at least 7 days ago AND the "Arrived" checkbox has not been ticked, then resets the activator column back to "yes" — which causes the campaign triggers to pick those drivers up again in the next polling cycle. This makes the follow-up loop fully automatic: if a driver doesn't show up, they get called again the following week without anyone on the team having to remember or manually re-trigger it.

Tech Stack

Component	Tool
Voice AI Agent	VAPI
Automation / Orchestration	n8n
Data Source & Tracking	Google Sheets (4 sheets, 1 spreadsheet)
Outbound Calling	VAPI Outbound Call API
Post-Call Logging	n8n Webhook + Google Sheets Write
Scheduling & Re-trigger	n8n Schedule Trigger (weekly)
Language Support	Polish, Ukrainian, English (runtime variable)

Key Design Decisions Worth Highlighting

Single agent, four document types: Rather than building four separate AI agents, we built one agent (Elliot) with a tightly scoped system prompt and injected the document type, language, client name, and office details as variables at call time. This keeps maintenance simple — changes to Elliot's behavior apply across all four workflows instantly.

Activator checkbox as campaign gate: The office staff controls who gets called and when by simply checking a box in the spreadsheet. This gives the team full control without needing to access n8n or VAPI directly. The activator column being the filter means calls only fire when explicitly authorized.

Row number + sheet name in call metadata: Every VAPI call carries the Google Sheets row number and sheet tab name in its metadata. This is what allows the post-call webhook to write results back to precisely the right place in the spreadsheet — even when multiple campaigns are running simultaneously across different sheets.

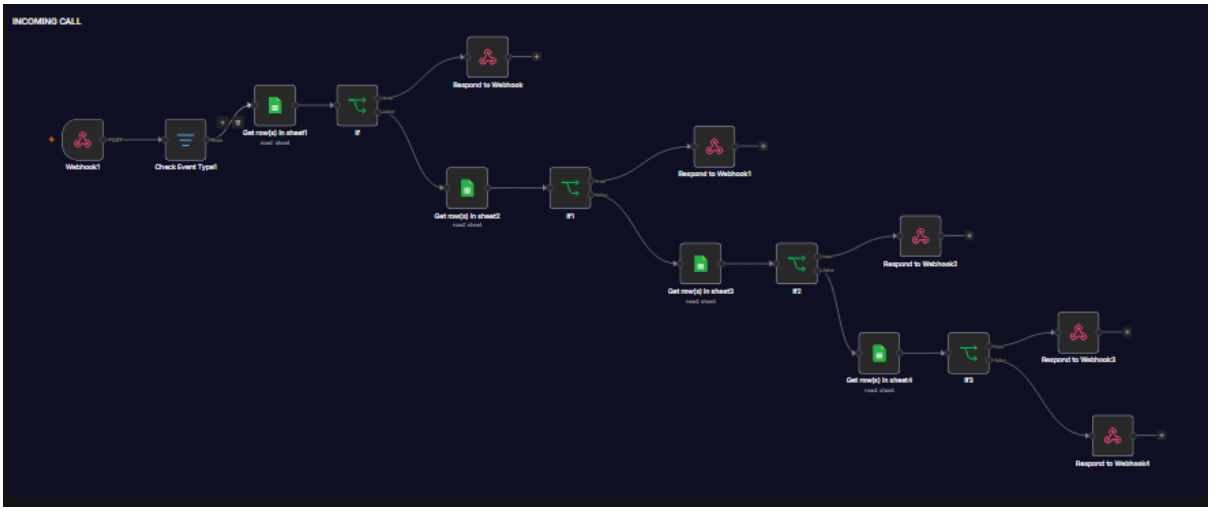
Identity verification before notification: Elliot never delivers document information without first confirming he's speaking to the right person. If someone else answers, he stops and requests a callback. This was a deliberate privacy protection for sensitive financial and legal documents.

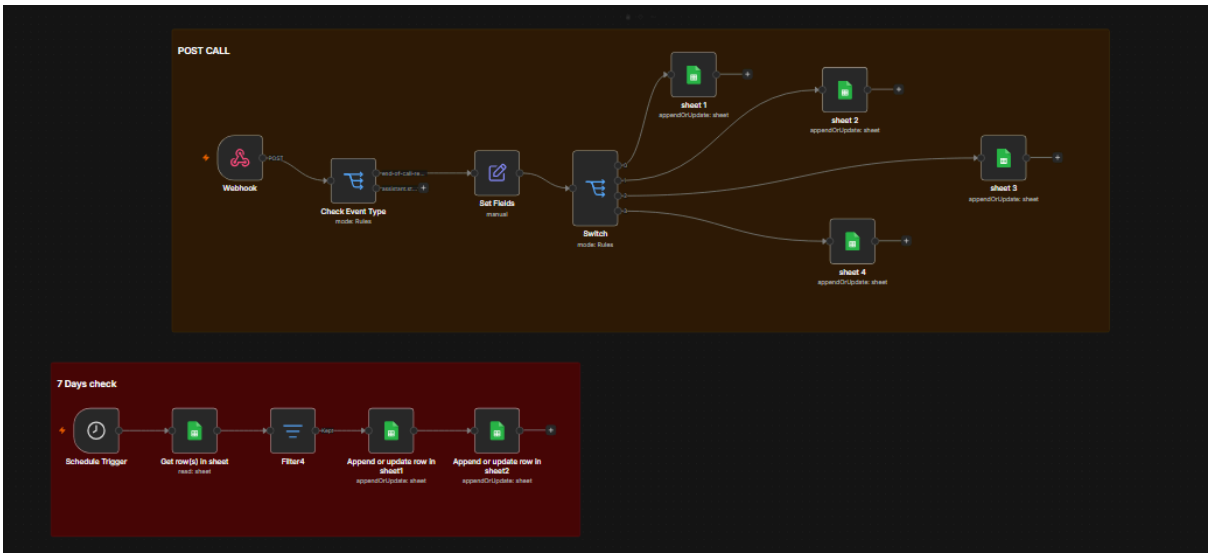
7-day automatic restart: The re-trigger logic doesn't require any human action. The scheduler checks the data, evaluates the conditions, and flips the activator back on automatically. This ensures no driver falls through the cracks after the initial campaign.

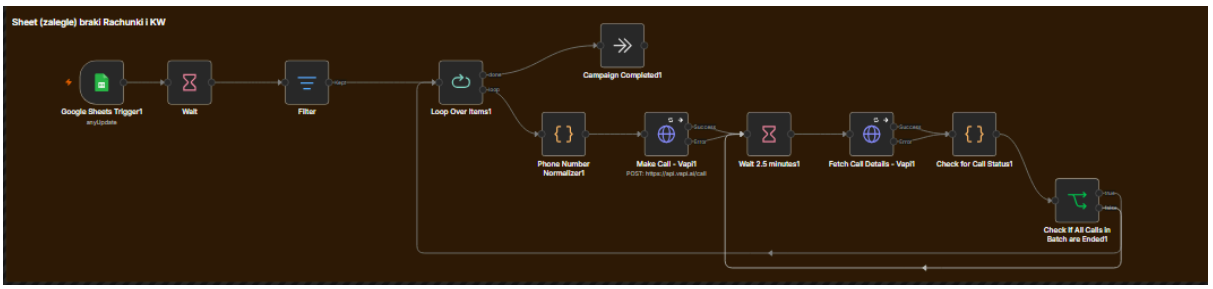
Multilingual from the ground up: Language is injected as a runtime variable, meaning the same assistant handles Polish, Ukrainian, and English calls without branching logic. The voice model and language setting adjust per call.

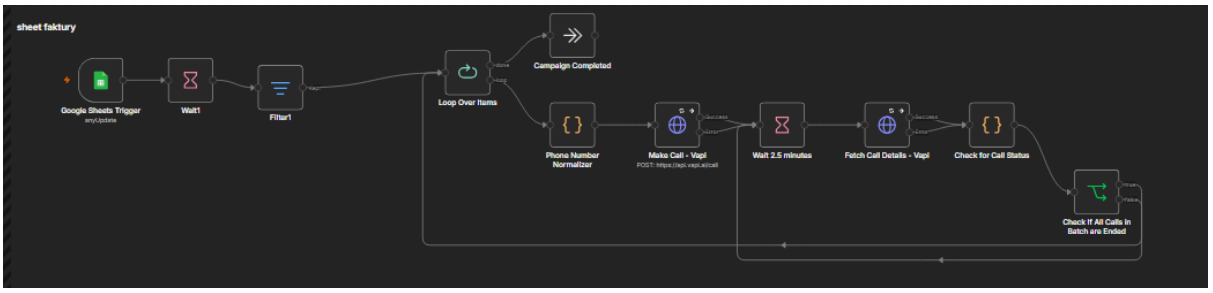
Results

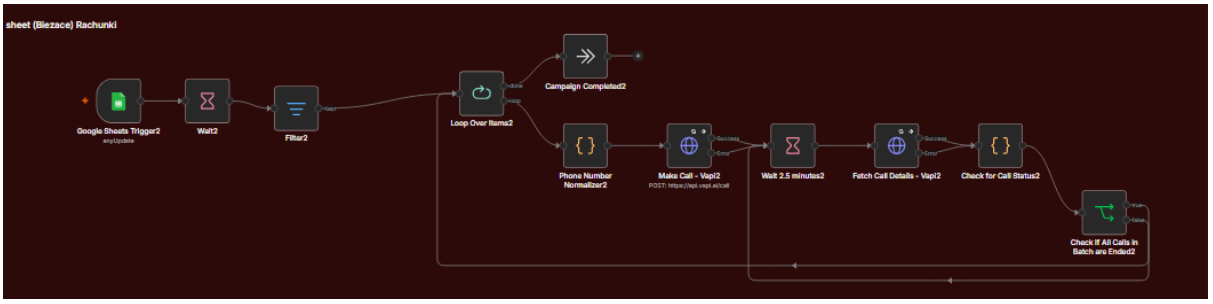
- Four separate document reminder campaigns fully automated with a single shared system
- Zero manual dialing required from the Abu Taxi office team
- Calls handled in three languages with automatic language matching per driver
- Post-call data (call date, outcome, pickup intent) written back to Google Sheets automatically after every call
- Drivers who don't show up within 7 days are automatically re-queued for the next weekly round — no human memory required
- Incoming callbacks from drivers are handled by the same AI agent with the correct context loaded from the sheet
- Complete call outcome tracking per driver across up to 6–9 contact attempts

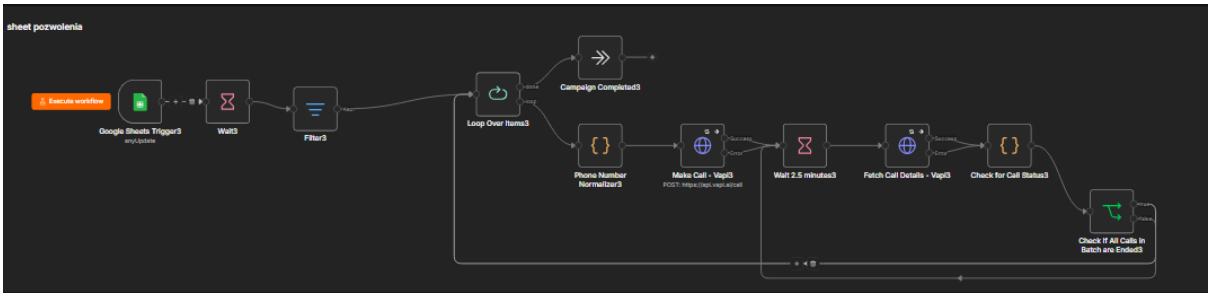





Sheet (zalegle) braki Rachunki i KW
Google Sheets Trigger1
anyUpdate
Wait
Filter
Loop Over Items1
Campaign Completed1
Phone Number Normalizer1
Make Call - Vapi1
POST: https://api.vapi.ai/call
Wait 2.5 minutes1
Fetch Call Details - Vapi1
Check for Call Status1
Check If All Calls in Batch are Ended1


sheet faktury
Google Sheets Trigger
anyUpdate
Wait1
Filter1
Loop Over Items
Campaign Completed
Phone Number Normalizer
Make Call - Vapi
POST: https://api.vapi.ai/call
Wait 2.5 minutes
Fetch Call Details - Vapi
Check for Call Status
Check If All Calls in Batch are Ended


sheet (Biezace) Rachunki
Google Sheets Trigger2
anyUpdate
Wait2
Filter2
Loop Over Items2
Campaign Completed2
Phone Number Normalizer2
Make Call - Vapi2
POST: https://api.vapi.ai/call
Wait 2.5 minutes2
Fetch Call Details - Vapi2
Check for Call Status2
Check If All Calls in Batch are Ended2


sheet pozwolenia
Execute workflow
Google Sheets Trigger3
anyUpdate
Wait3
Filter3
Loop Over Items3
Campaign Completed3
Phone Number Normalizer3
Make Call - Vapi3
POST: https://api.vapi.ai/call
Wait 2.5 minutes3
Fetch Call Details - Vapi3
Check for Call Status3
Check If All Calls in Batch are Ended3