

# Development Specification Sheet

Heffernan Insurance Brokers  
WC Prospects Lead Generation System  
Seamless AI API Integration

---

<b>Prepared By</b>	Anthony Pinto, Veteran Vectors
<b>Prepared For</b>	Nadia Messiah, Director of Innovation/Development
<b>Client</b>	Heffernan Insurance Brokers
<b>Date</b>	February 12, 2026
<b>Version</b>	1.0 — Initial Specification
<b>Workflow ID</b>	D0tSzFV5AZh0qgcZ
<b>Status</b>	Pending IT Review & API Credentials

# Table of Contents

1. System Overview
2. Third-Party Service Dependencies
3. API Configuration — Seamless AI (New)
4. API Configuration — AnyMailFinder (Being Replaced)
5. Seamless AI API — Complete Inputs & Outputs
6. POC Search Rules & Seniority Logic
7. Request / Response Payloads
8. Field Mapping — Seamless AI → Google Sheets
9. Google Sheets Schema
10. Workflow Step-by-Step (All 6 Steps)
11. Google Apps Script — Orchestration Layer
12. Error Handling & Retry Logic
13. Rate Limits & Throttling
14. Security & Data Flow
15. Rollback Plan
16. Action Items & Prerequisites

## 1 System Overview

The WC Prospects Lead Generation System is an n8n automation workflow that processes Workers' Compensation prospect data from California ECU reports, enriches it with AI-powered research, discovers decision-maker contacts, and prepares qualified prospects for outreach via the Outreach CRM platform.

This specification covers the replacement of the AnyMailFinder API with the Seamless AI API for the POC (Point of Contact) discovery step (Step 4), and the updated row logic moving from 3 fixed rows per company to 5 fixed rows.

### Scope of Change

- **Replacing:** 3x AnyMailFinder HTTP Request nodes with 1x Seamless AI HTTP Request node
- **Expanding:** Seamless AI also takes over revenue, industry, and SIC/NAICS lookups (previously Perplexity in Step 2)
- **Updating:** Build POC Updates code node to support 5-row output with seniority ranking
- **Updating:** POC category structure from 3 categories to 4 categories + 1 extra slot
- **Updating:** Step 2 Perplexity scope reduced to domain and FP/NP classification only
- **Removing:** Form 5500 URL, EE Reasoning, and Revenue Reasoning fields
- **No changes to:** Steps 1, 5, or 6 of the workflow

## 2 Third-Party Service Dependencies

Service	Purpose	Auth Method	Step	Status
<b>Seamless AI</b>	POC discovery, email enrichment, revenue, industry, SIC/NAICS	Persistent API Key (Bearer token)	Step 4	<b>NEW — Pending</b>
<b>AnyMailFinder</b>	POC discovery (decision-maker search)	Static API Key (header)	Step 4	<b>REMOVING</b>
<b>Perplexity AI</b>	EE count, domain, FP/NP classification	API Key	Steps 1–2	<b>No change</b>
<b>Apify</b>	LinkedIn profile URL scraping	API Token	Step 5	<b>No change</b>
<b>Google Sheets API</b>	Primary data store (read/write)	OAuth 2.0	All steps	<b>No change</b>
<b>Google Docs API</b>	Address label document generation	OAuth 2.0	Step 6	<b>No change</b>
<b>Google Apps Script</b>	Sheet formatting, bold labels	Webhook (public URL)	Steps 2, 6	<b>No change</b>

### 3 API Configuration — Seamless AI (New)

Parameter	Value
Base URL	<a href="https://api.seamless.ai/v1/">https://api.seamless.ai/v1/</a>
Contact Search Endpoint	POST /contacts/search
Authentication	Bearer token via Persistent API Key
Header	Authorization: Bearer {SEAMLESS_API_KEY}
Content-Type	application/json
Rate Limit	~100 requests per 60 seconds per endpoint
Rate Limit Response	HTTP 429 (remaining credits in response headers)
Response Format	JSON (REST API)
Compliance	SOC2, GDPR, HIPAA compliant — 99.9% uptime SLA
n8n Webhook URL	<a href="https://timbheffins.app.n8n.cloud/webhook/a3575277-4e2f-4845-9296-a235360e3b81">https://timbheffins.app.n8n.cloud/webhook/a3575277-4e2f-4845-9296-a235360e3b81</a>
n8n Instance	timbheffins.app.n8n.cloud (Heffernan's n8n Cloud)
n8n Node Type	HTTP Request (generic)
Credential Storage	n8n encrypted credential vault (not in workflow JSON)

## 4 API Configuration — AnyMailFinder (Being Replaced)

Documented here for reference during transition. These 3 nodes will be removed.

Parameter	Value
Endpoint	POST <a href="https://api.anymailfinder.com/v5.1/find-email/decision-maker">https://api.anymailfinder.com/v5.1/find-email/decision-maker</a>
Auth Header	Authorization: {API_KEY} (static, unencrypted in workflow)
Nodes Being Removed	Find POCs (AnyMail) (HR), Find POCs (AnyMail) (OPS), Find POCs (AnyMail) (CEO)
Calls Per Company	3 (one per category: hr, operations, ceo)
Request Body	{ "domain": "{Domain}", "decision_maker_category": ["{category}"], "company_name": "{Primary Name}" }

## 5 Seamless AI API — Complete Inputs & Outputs

This section documents the exact request parameters (inputs) and response fields (outputs) for the Seamless AI API endpoints relevant to this integration. Source: docs.seamless.ai API Reference and Seamless.AI Data Dictionary.

### 5A. Contact Search — Request Parameters (Inputs)

Endpoint: POST /contacts/search. All filter parameters accept arrays of strings unless noted otherwise.

Parameter	Type	Description	Our Use
<code>companyName</code>	String[]	Company names to search (supports multiple)	<input checked="" type="checkbox"/> Primary Name
<code>companyNameSearchType</code>	String	"default" or "exact" — controls fuzzy vs. exact match on company name	<input checked="" type="checkbox"/> "default"
<code>companyDomain</code>	String[]	Company website domains to filter by (e.g., ["acme.com"])	<input checked="" type="checkbox"/> Domain
<code>jobTitle</code>	String[]	Specific job titles to match (e.g., ["CEO", "CFO", "COO"])	<input checked="" type="checkbox"/> All 14 titles
<code>seniority</code>	String[]	Seniority levels: VP, Director, C-Level, Manager, etc.	<span style="color: orange;">⚠ Optional</span>
<code>department</code>	String[]	Department filters (e.g., Finance, Operations, HR, Executive)	<span style="color: orange;">⚠ Optional</span>
<code>industry</code>	String[]	Industry filters (Seamless.AI industry taxonomy)	<span style="color: red;">✗ Not used</span>
<code>fullname</code>	String[]	Search by specific contact name(s)	<span style="color: red;">✗ Not used</span>
<code>contactKeyword</code>	String[]	Free-text keyword search across contact profiles	<span style="color: red;">✗ Not used</span>
<code>contactState</code>	String[]	Filter by contact's state (personal location)	<span style="color: red;">✗ Not used</span>
<code>contactCountry</code>	String[]	Filter by contact's country	<span style="color: red;">✗ Not used</span>
<code>contactZipCode</code>	String[]	Filter by contact's zip code	<span style="color: red;">✗ Not used</span>
<code>locationType</code>	String	"bothOR" — match contact OR company location	<span style="color: red;">✗ Not used</span>
<code>companySize</code>	String[]	Employee size ranges (e.g., ["51-200", "201-500"])	<span style="color: red;">✗ Not used</span>
<code>companyRevenue</code>	String[]	Revenue ranges for filtering	<span style="color: red;">✗ Not used</span>
<code>companyFoundedOn</code>	String[]	Company age filters (e.g., "Less than 1 Year", "Last 1-3 Years")	<span style="color: red;">✗ Not used</span>
<code>technologies</code>	String[]	Web technologies the company uses	<span style="color: red;">✗ Not used</span>
<code>technologiesIsOr</code>	Boolean	true = match ANY tech; false = match ALL tech	<span style="color: red;">✗ Not used</span>
<code>lastModifiedAfter</code>	DateTime	ISO 8601 date to filter contacts modified after this date	<span style="color: red;">✗ Not used</span>
<code>lastModifiedBefore</code>	DateTime	ISO 8601 date to filter contacts modified before this date	<span style="color: red;">✗ Not used</span>

### 5B. Contact Search — Response Fields (Outputs)

Each result in the `data[]` array returns the following fields. These are the fields available in the API response before research/enrichment.

Response Field	Type	Description	Maps To
<code>searchResultId</code>	String	Unique ID for the search result (used for research/enrichment calls)	Internal ref
<code>name</code>	String	Full name of the contact	<input checked="" type="checkbox"/> Prospect
<code>company</code>	String	Company name (cleaned by Seamless AI)	Verify match
<code>title</code>	String	Job title of the contact	<input checked="" type="checkbox"/> Title
<code>department</code>	String	Department (e.g., Finance, Operations, HR,	POC Reasoning

		Executive)	
<b>seniority</b>	String	Seniority level (e.g., VP Level, Director, C-Level, Manager)	Seniority sort
<b>domain</b>	String	Company domain/website	Verify match
<b>city</b>	String	Contact's city	✗ Not mapped
<b>state</b>	String	Contact's state	✗ Not mapped
<b>country</b>	String	Contact's country	✗ Not mapped
<b>companyCity</b>	String	Company headquarters city	✗ Not mapped
<b>companyState</b>	String	Company headquarters state	✗ Not mapped
<b>companyCountry</b>	String	Company headquarters country	✗ Not mapped
<b>liurl</b>	String	Contact's LinkedIn profile URL	✓ LI Profile
<b>companyLIProfileUrl</b>	String	Company's LinkedIn page URL	✗ Not mapped
<b>sicCode</b>	String	Standard Industrial Classification code	✓ SIC Code
<b>industries</b>	String[]	Array of industries the company belongs to	✓ Industry

## 5C. Contact Research / Enrichment — Additional Output Fields

After using the searchResultId to trigger a research/enrichment call, Seamless AI returns the full contact data dictionary. Key additional fields not in the search response:

Enriched Field	Type	Description	Maps To
<b>Business Email (1–10)</b>	String	Up to 10 verified business emails with validation status (valid/invalid/accept all)	✓ Contact Email
<b>Email Validation Status (1–10)</b>	String	Validation result: valid, invalid, accept all, unknown	Filter logic
<b>Email Total AI Score (1–10)</b>	Number	AI confidence score for email accuracy	Filter logic
<b>Contact Phone (1–10)</b>	String	Up to 10 phone numbers — Main and Mobile types	✓ Contact Phone
<b>Company Phone (1–10)</b>	String	Company switchboard/main phone numbers	✗ Not mapped
<b>Personal Email (1–3)</b>	String	Personal email addresses (non-business)	✗ Not mapped
<b>First Name / Last Name</b>	String	Parsed name components	Available
<b>LinkedIn Profile URL</b>	String	Contact's LinkedIn URL (enriched/verified)	✓ LI Profile
<b>Contact Location</b>	Object	Region, Country, City, State, Zip Code	✗ Not mapped
<b>Company Employee Size / Range</b>	String	Exact count and range bracket	✗ Not mapped
<b>Company Annual Revenue / Range</b>	String	Revenue figure and bracket	✓ Revenue (\$M)
<b>Company Industry / SIC / NAICS</b>	String	Industry classification with SIC and NAICS codes	✓ Industry / SIC
<b>Company Description</b>	String	Brief company description/about text	✗ Not mapped
<b>Company Website Domain</b>	String	Cleaned website domain	✗ Not mapped

## 5D. Company Search — Key Fields (Reference)

The Company Search endpoint returns organizational data. While this integration primarily uses Contact Search, these fields are available if Heffernan wants to expand the workflow later:

- **Company Name / Name Clean:** Official and AI-cleaned company name
- **Website / Domain:** Company website URL
- **Industry:** Basic and advanced industry classification, SIC Code, NAICS Code

- **Staff Count / Range:** Employee headcount (exact and range)
- **Revenue / Range:** Annual revenue (exact and range)
- **Location:** Full address — Street, City, State, Zip, Country
- **Founded Date:** Year company was established
- **Company Type:** Public, Private, etc.
- **Company LinkedIn:** Company LinkedIn profile URL and ID
- **Company Specialties:** Areas of expertise/services
- **Company Intelligence:** Links to CRM/Social, News, Employee Reviews, Google Finance, Job Postings, SEC Filings, Web Technologies, Whois
- **Paid Search Intelligence:** Advertising and keyword data

## 5E. Key Implementation Notes

- **Two-step process:** Contact Search returns basic fields. To get emails and phones, the searchResultId from each result must be passed to a Research/Enrich endpoint. This means 1 search call + N research calls per company.
- **Credit impact:** Each search call and each research/enrich call may consume credits. Confirm credit model with Nadia during onboarding.
- **Expanded enrichment:** Seamless AI Contact Search response includes company-level fields (industries, sicCode, revenue) in every result. Revenue, Industry, and SIC Code will be extracted from the first POC result and written to the Filtered Accounts sheet — eliminating these from the Perplexity Step 2 scope.
- **LinkedIn URL bonus:** Seamless AI returns liUrl in the search response, which may reduce or eliminate the need for the current Apify LinkedIn scraper (Step 5). To be validated in sandbox.
- **Email validation:** Seamless AI provides built-in email validation scores (Total AI). We can filter for "valid" emails only before writing to sheets.
- **Removed fields:** Form 5500 URL, EE Reasoning, and Revenue Reasoning are no longer tracked. Corresponding columns can be removed or left blank on the sheets.

## 6 POC Search Rules & Seniority Logic

Each company produces exactly 5 rows on the “1. Filtered Accounts & POCs” sheet. Each row is assigned to one POC slot as defined below. Seniority is determined by title priority order (position 1 = most senior).

Row	Category	Target Titles (Priority Order)	Qty	Selection Rule
1	Org Head	CEO → President → Owner	1	Most senior found
2	Finance Head	CFO → VP Finance → Finance Director → Head of Finance	1	Most senior found
3	Operations Head	COO → VP Operations → Director Operations	1	Most senior found
4	HR Head	Chief People Officer → Chief HR Officer → VP Human Resources → Human Resources Director	1	Most senior found
5	Extra / Overflow	Next best match from any category not already assigned (e.g., 2nd Org Head, or best available from unfilled slot)	1	Best remaining

### Seniority Ranking Algorithm

For categories with multiple title matches, the system selects the most senior contact using this logic:

1. Query Seamless AI with all target titles for the company (single API call)
2. Sort returned contacts into 4 category buckets by title keyword matching
3. Within each bucket, rank by title priority order (listed in table above, left = most senior)
4. Select top 1 per category for rows 1–4
5. Row 5: Best remaining contact not already assigned (next Org Head, or backfill for empty category)
6. If fewer than 5 contacts found, remaining rows are left blank (row still created)

## 7 Request / Response Payloads

### Seamless AI — Contact Search Request

Single request per company, replacing 3 separate AnyMailFinder calls:

```
POST https://api.seamless.ai/v1/contacts/search
Headers:
  Authorization: Bearer {SEAMLESS_API_KEY}
  Content-Type: application/json

Body:
{
  "domain": "{{ Domain from Sheets }}",
  "company_name": "{{ Primary Name from Sheets }}",
  "title": [
    "CEO", "President", "Owner",
    "CFO", "VP Finance", "Finance Director",
    "Head of Finance",
    "COO", "VP Operations", "Director Operations",
    "Chief People Officer", "Chief HR Officer",
    "VP Human Resources", "Human Resources Director"
  ],
  "limit": 20
}
```

### Seamless AI — Expected Response Structure

Note: Exact field names must be verified against Seamless AI's sandbox environment. The below is based on published documentation and may differ.

```
{
  "data": [
    {
      "name": "Jane Smith",
      "first_name": "Jane",
      "last_name": "Smith",
      "title": "Chief Financial Officer",
      "email": "jsmith@company.com",
      "phone": "+1-555-123-4567",
      "department": "Finance",
      "seniority": "VP Level",
      "company": "Company Name",
      "company_domain": "company.com"
    }
  ]
}
```

## 8 Field Mapping — Seamless AI → Google Sheets

### POCs Sheet (Tab: “2. POCs”)

Sheets Column	Seamless AI Field	Type	Notes
Primary Name	company	String	From Sheets (not API)
Prospect	name	String	Full name of contact
Title	title	String	Job title
Contact Email	email	String	Verified email
Contact Phone Number	phone	String	NEW — not in AnyMailFinder
POC Reasoning	(generated)	String	Category + rank logic
POC Row #	(computed)	Integer	Row ref on Filtered sheet
Contact Last Updated	(timestamp)	Date	Execution date
Done	(flag)	String	Set to “Done” on completion

### Filtered Accounts & POCs Sheet (Tab: “1. Filtered Accounts & POCs”)

Sheets Column	Source	Notes
Prospect	Seamless AI → name	Mapped from POC selection logic
Title	Seamless AI → title	Job title of selected POC
POC #	Computed	References row on POCs sheet
POC Done	Flag	Set to “Done” when all 5 rows populated
LI Profile	Apify (Step 5)	No change — populated downstream
Revenue (\$M)	Seamless AI → Company Annual Revenue	NEW source — previously Perplexity
Industry	Seamless AI → industries	NEW source — previously Perplexity
SIC Code	Seamless AI → sicCode	NEW source — previously Perplexity
Domain	Perplexity (Step 2)	No change
For-Profit/Non-Profit	Perplexity (Step 2)	No change
Form 5500 URL	REMOVED	No longer tracked
EE Reasoning	REMOVED	No longer tracked
Revenue Reasoning	REMOVED	No longer tracked

## 9 Google Sheets Schema

Spreadsheet: WC Prospects Import & Filtering (via Tech Stack)

Document ID: 1mno3Gy9qotBKO36Qx28p8ft5krR3sncZis8lrBVC5Cw

Tab Name	Purpose	Rows	Trigger
<b>0. NEW HERE</b>	Intake — raw ECU WC prospect data lands here	~198	rowAdded
<b>1. Filtered Accounts &amp; POCs</b>	Qualified prospects with 5 POC rows each, enrichment data, Outreach upload tracking	~35	rowAdded, rowUpdate
<b>2. POCs</b>	Raw POC results from Seamless AI (all contacts found)	~45	None
<b>Prompts</b>	AI prompt templates for Perplexity calls	2	None
<b>Sources</b>	Research source URLs for EE and revenue calculations	Varies	None
<b>Address Labels (General)</b>	For-profit prospect address labels for mail outreach	Varies	rowUpdate
<b>Address Labels (Non-Profit)</b>	Non-profit prospect address labels for mail outreach	Varies	rowUpdate

## 10 Workflow Step-by-Step (All 6 Steps)

### Step 1: Process New WC Prospects & Find Employee Count

- **Trigger:** Google Sheets Trigger on “0. NEW HERE” tab (rowAdded)
- **Process:** Filter new rows → Merge with Prompts → Perplexity AI (sonar-pro) calculates employee count → Parse response → Update EE’s and Sources columns
- **Output:** Enriched rows with employee count
- **Change:** None

### Step 2: Move to Filtered & Find Domain, FP/NP Status

- **Trigger:** Google Sheets Trigger on “1. Filtered Accounts & POCs” tab (rowAdded)
- **Process:** Dedupe → Merge with Prompts → Perplexity AI (sonar-pro) finds domain and for-profit/non-profit status → Parse response → Update Renewal sheet + Sources → Format rows via Google Apps Script webhook
- **Output:** Company records with domain (required for Step 4) and FP/NP classification
- **Change:** Reduced scope — revenue, industry, and SIC/NAICS now handled by Seamless AI in Step 4. Form 5500 URL, EE Reasoning, and Revenue Reasoning removed.

### Step 4: Find POCs for Company

#### **⚠ THIS IS THE STEP BEING MODIFIED**

- **Trigger:** Manual execution (When clicking ‘Execute workflow’)
- **Current flow:** Read Renewal3 → Dedupe1 → Loop Over Items4 (batch 1) → 3 parallel AnyMailFinder calls (HR/OPS/CEO) → Write POC sheet → Build POC Updates code → Update Filtered sheet → Wait5
- **New flow:** Read Renewal3 → Dedupe1 → Loop Over Items4 (batch 1) → 1 Seamless AI call (all titles) → Seniority Sort code node → Write 5 POC rows + Revenue/Industry/SIC to Filtered sheet → Wait
- **Key change:** 3 parallel nodes → 1 node; 3 rows per company → 5 rows per company; category-based search → title-based search with seniority ranking; Seamless AI also writes Revenue (\$M), Industry, and SIC Code (previously Perplexity)

### Step 5: Update Renewal Sheet with POCs and LI Profile

- **Trigger:** Google Sheets Trigger on “1. Filtered Accounts & POCs” (rowUpdate)
- **Process:** Trigger LI Profile Search → Filter → Loop → Apify LinkedIn Profile Search scraper → Check result → Update LI Profile column
- **Change:** None (will process 5 rows per company instead of 3 automatically)

### Step 6: Move to Address Labels (For-Profit & Non-Profit)

- **Trigger:** Google Sheets Trigger on “1. Filtered Accounts & POCs” (rowUpdate) and Address Labels tabs (rowUpdate)
- **Process:** Move qualified rows → Corrected Name (Perplexity) → Parse → Generate labels → Write to Google Docs → Bold formatting via Apps Script
- **Change:** None (handles variable row counts already)

## 11 Google Apps Script — Orchestration Layer

A bound Google Apps Script is attached to the WC Prospects spreadsheet. It acts as the orchestration layer between Google Sheets and n8n, handling data formatting, filtering, formula injection, trigger polling, and webhook dispatch. It is also deployed as a Web App so n8n can call it via HTTP POST.

### 11A. Deployment & Execution Model

Property	Value
Script type	Container-bound (attached to WC Prospects spreadsheet)
Web App deployment	Deployed as doPost() Web App — accepts POST requests from n8n
Time-driven trigger	mainScheduledRunner() executes on a timer trigger (~every 60 seconds)
Runner guard	45-second minimum gap between runs (RUNNER_MIN_GAP_SEC). Uses LockService + ScriptProperties timestamp.
onEdit trigger	Simple trigger — fires on manual user edits only (NOT on API/script writes from n8n)

### 11B. CONFIG Constants

All configurable values are centralized in a CONFIG object at the top of the script. Key entries:

Key	Value	Purpose
RENEWAL_SHEET	1. Filtered Accounts & POCs	Primary working sheet
NEW_HERE_SHEET	0. NEW HERE	Intake sheet
POCS_SHEET	2. POCs	POC results sheet
AUDIT_LOG_SHEET	Audit Log	Change tracking (auto-created)
EMPLOYEE_WEBHOOK_URL	https://timbheffins.app.n8n.cloud/webhook/47ddd94f...	n8n EE check webhook
SEAMLESS_WEBHOOK_URL	https://timbheffins.app.n8n.cloud/webhook/a3575277-...	n8n Seamless AI webhook
ADDRESS_LABELS_DOC_IDS	GENERAL: 1Abb062Y... / NON_PROFIT: 1nbCmlo...	Google Docs for label output
RUNNER_MIN_GAP_SEC	45	Minimum seconds between runner executions

User-configurable thresholds are stored in ScriptProperties (not hardcoded):

Property Key	Default	Purpose
vv_wc_premium_min	25,000	WC Premium minimum for filter-to-Filtered logic
vv_employee_count_min	0	Employee count minimum for filter-to-Filtered logic
vv_sort_column	wc_premium	Sort preference for NEW HERE sheet (wc_premium or employees)

### 11C. Scheduled Runner — Execution Sequence

The mainScheduledRunner() function executes on a timer trigger. It acquires a script lock (25s timeout), checks the 45-second gap guard, then runs the following functions in order:

#	Function	What It Does
1	bucketizeEmployeeCounts()	Rounds EE counts ≤ 50 down to 25 in NEW HERE
2	normalizePocsPrimaryNames()	Uppercases Primary Name column A in POCs sheet
3	fillNewHereFormulasForK_()	Inserts VLOOKUP formulas in columns A–D, G (WCIRB rates) and column K (filter formula) for any row with a Bureau Number

4	applyWCPremiumConditionalFormat_( )	Green highlight on EE's and WC Premium cells meeting thresholds
5	applyNewHereDropdownsAndFormatting_( )	Adds Yes/No dropdowns to columns H, I, J and Yes-only to L, M. Green/red conditional formatting.
6	sortNewHereBySelectedColumn_( )	Sorts NEW HERE by WC Premium or EE's (user preference), highest to lowest
7	fillAddressLabelsFormulas_( )	Inserts L and M formulas in Address Labels sheets when column J is populated
8	applyAddressLabelsColumnQDropdown_( )	Adds "Yes" dropdown + green formatting to column Q when Prospect exists
9	fillPocNumbers_( )	Sets POC # = row number in Filtered sheet when Domain is populated
10	fillPocRowNumbers_( )	Sets POC Row # = row number in POCs sheet when Prospect or Email exists
11	checkSeamlessTrigger_( )	★ Polls "Ready for Seamless.AI" column. If "Yes": fires webhook, marks "Sent". (See 11D below.)
12	deleteTrailingBlankRows_( )	Removes trailing blank rows from Filtered Accounts sheet

## 11D. Seamless AI Trigger Mechanism (checkSeamlessTrigger\_)

This is the bridge between n8n and the Seamless AI workflow. Because n8n writes values to the sheet via API (not manual edits), the onEdit() trigger cannot detect them. Instead, the scheduled runner polls for changes on every cycle.

- **Trigger column:** "Ready for Seamless.AI" (header-based lookup, not a hardcoded column letter)
- **Trigger value:** "Yes" — set programmatically by n8n after Step 2 completes
- **After firing:** Cell value changes from "Yes" to "Sent" to prevent re-firing
- **Block detection:** Function walks backward/forward from trigger row to find the 5-row company block boundaries (startRow → endRow)

Webhook payload sent to n8n:

Field	Type	Description
spreadsheetId	String	Google Sheets document ID
sheetName	String	"1. Filtered Accounts & POCs"
pocsSheetName	String	"2. POCs"
startRow	Number	First row of the 5-row company block
endRow	Number	Last row of the 5-row company block
rowCount	Number	Number of rows in block (always 5)
triggerRow	Number	Row where "Yes" was found
companyName	String	Primary Name from sheet
bureauNumber	String	Bureau Number from sheet
domain	String	Company domain (from Perplexity Step 2)
streetAddress	String	Street address
city	String	City
state	String	State
timestamp	String	ISO 8601 timestamp of webhook dispatch
source	String	Always "seamless_trigger"

## 11E. Data Flow Functions

`copyNewHereToRenewalByWcPremium_()` — Copies qualifying rows from 0. NEW HERE to 1. Filtered Accounts & POCs.

- **Trigger:** Runs when column L (“Manual Override to Add?”) or column M (“Final Okay to Add?”) is set to “Yes” via onEdit()
- **Filter logic:** Row passes if: (a) Manual Override = “Yes”, OR (b) Final Okay = “Yes” AND (WC Premium  $\geq$  threshold OR EE count  $\geq$  threshold)
- **Deduplication:** Checks Bureau Number and Primary Name + Street Address against existing Filtered rows. Skips duplicates.
- **Column mapping:** 39 columns mapped by header name (not position). Includes Bureau Number, Primary Name, address fields, EE’s, WC Premium, ExMod data, rates, Governing Class, SubClass data, Domain, Revenue, Prospect, Title, LI Profile.

`reformatARenewalRows_()` — Generates 5 duplicate rows per company block on the Filtered sheet.

- **Trigger:** Called via doPost() Web App when n8n sends trigger = “reformat\_a\_renewal\_rows”
- **Logic:** Groups rows by Bureau Number (or Primary Name + Street Address). If a company has exactly 5 rows, it’s preserved. Otherwise, the first row is duplicated 5 times. A blank separator row is inserted after each block.
- **Formatting:** All data rows reset to white background, black text, Arial 10pt, normal weight.

## 11F. onEdit Triggers

The simple onEdit(e) trigger fires only on manual user edits. It handles:

Sheet	Column	Action
0. NEW HERE	L or M	Runs copyNewHereToRenewalByWcPremium_() to check if row qualifies for Filtered sheet
2. POCs	B (row $\geq$ 2)	Auto-upercases the value
Address Labels (*)	J (row $\geq$ 2)	Inserts L and M formulas (address concat + TEXTJOIN label)
All sheets (except Audit Log)	Any	Logs change to Audit Log sheet (timestamp, user, old/new value, change type)

Important: onEdit() does NOT fire when n8n writes to the sheet via API. This is why the Seamless AI trigger uses the polling approach in the scheduled runner instead.

## 11G. doPost Web App Endpoints

n8n can call the Apps Script Web App URL via HTTP POST with a JSON body containing a trigger field. Supported triggers:

trigger Value	Action
reformat_a_renewal_rows	Runs reformatARenewalRows_() — generates 5-row blocks on Filtered sheet
clear_all_data_formatting	Runs clearAllDataFormatting() — strips all formatting from Filtered data rows
main_scheduled_runner	Runs mainScheduledRunner() — full automation cycle on demand

All responses return JSON: { status: “ok”, action: “...” } or { status: “error”, message: “...” }.

## 11H. Custom Menu (Automation Tools)

The onOpen() function adds an “Automation Tools” menu to the spreadsheet UI for the Heffernan team:

- **Re-check EE:** Fires the Employee webhook to re-run Perplexity EE count check
- **Set Thresholds (WC + EE):** Prompt to set WC Premium minimum and EE count minimum for the copy-to-Filtered filter logic
- **Show Current Settings:** Displays current threshold values
- **Column to Sort 0. NEW HERE Sheet:** Toggle between sorting by WC Premium or EE’s
- **Audit Log submenu:** View, Search (with highlight), or Clear the audit log
- **Delete All Rows:** Nuclear option — clears all data rows from all sheets and resets both Address Labels Google Docs templates (402 label placeholders)

## 11I. Audit Logging

Every manual edit across all sheets (except the Audit Log itself) is logged automatically via `onEdit()`. Each log entry captures: timestamp, user email, user name, sheet name, cell reference, change type (INSERT / EDIT / DELETE / FORMULA), old value, new value, and column header. The log is capped at 10,000 entries (oldest are trimmed). Entries are inserted at row 2 (newest first).

---

## 11J. End-to-End Flow Summary

The complete data flow from WC report upload to Seamless AI POC discovery:

1. WC Report data lands in 0. NEW HERE (via upload or n8n processing)
2. Scheduled runner inserts WCIRB rate formulas, calculates WC Premium, applies dropdowns and formatting
3. User (or n8n) sets column L or M to "Yes" → `onEdit` fires → `copyNewHereToRenewalByWcPremium_()` copies qualifying row to 1. Filtered Accounts & POCs
4. n8n calls `doPost()` with trigger = "reformat\_a\_renewal\_rows" → `reformatARenewalRows_()` creates 5-row block
5. n8n runs Perplexity for domain + FP/NP, then sets "Ready for Seamless.AI" = "Yes"
6. Next scheduled runner cycle detects "Yes" → `checkSeamlessTrigger_()` fires webhook with company details → marks cell "Sent"
7. n8n receives webhook → calls Seamless AI API → writes POC results to 2. POCs sheet + Revenue/Industry/SIC to Filtered sheet
8. Scheduled runner fills POC # and POC Row # values, cleans up trailing blank rows

## 12 Error Handling & Retry Logic

Scenario	Current Behavior	Expected Behavior (Seamless AI)
<b>API returns 429 (rate limit)</b>	No explicit handling	n8n retry on error (3 attempts, exponential backoff). Wait nodes between batches prevent hitting limits.
<b>API returns 401/403</b>	Workflow errors	Workflow stops. Alert via n8n error workflow. Check API key validity in Seamless AI Settings.
<b>API returns empty results</b>	Blank rows written to sheet	Same — blank rows created for unfilled POC slots (5 rows always created).
<b>API timeout</b>	n8n default timeout	HTTP Request node timeout set to 30s. Retry on fail enabled.
<b>Duplicate contacts returned</b>	Dedupe1 code node	Same Dedupe logic + seniority sort code ensures no duplicate POCs across the 5 rows.
<b>Domain not found / invalid</b>	AnyMailFinder returns empty	Seamless AI returns empty data array. Row still created as blank. Company name used as fallback search.

## 13 Rate Limits & Throttling

Metric	AnyMailFinder (Old)	Seamless AI (New)
API calls per company	3	1
API calls per 100 companies	300	100
Rate limit	Undocumented	~100 req / 60 sec
n8n batch size	1 (Loop Over Items4)	1 (same — one company per batch)
Wait between batches	Wait5 node (configurable)	Same Wait node (recommend 2–5 sec)
Credit consumption	3 credits per company	1 credit per company (verify with Seamless AI)

## 14 Security & Data Flow

### Data Flow Direction

All data flows in one direction. No customer PII is sent outbound to Seamless AI.

- **Outbound to Seamless AI:** Company name + domain only (public business information)
- **Inbound from Seamless AI:** Contact name, title, email, phone, seniority, department
- **Data storage:** Google Sheets (Heffernan's Google Workspace) → encrypted at rest by Google
- **Credential storage:** n8n encrypted credential vault (AES-256). API keys never appear in workflow JSON.
- **Network:** All API calls over HTTPS/TLS 1.2+

### Access Control

- **n8n instance:** Self-hosted with role-based access. Only authorized users can view/edit workflows.
- **Seamless AI key:** Persistent key scoped to Heffernan's org. Can be revoked at any time from Settings → API Key.
- **Google OAuth:** Scoped to Sheets and Docs read/write only. Managed under VV Google Sheets account.

## 15 Rollback Plan

1. The existing AnyMailFinder nodes will be disabled (not deleted) during the transition.
2. A versioned copy of the current workflow JSON (v3) is archived before any modifications.
3. If Seamless AI integration fails validation, re-enable the 3 AnyMailFinder nodes and revert the Build POC Updates code node.
4. Rollback time: < 10 minutes (toggle nodes + paste previous code).
5. No sheet structure changes required for rollback — the 5-row layout is backward compatible with 3 rows.

## 16 Action Items & Prerequisites

#	Action Item	Owner	Priority	Status
1	Provision Seamless AI Persistent API Key	Nadia Messiah (Heff IT)	Critical	Pending
2	Request Seamless AI sandbox / test environment	Nadia Messiah (Heff IT)	Critical	Pending
3	Confirm API credit budget for automated searches	Nadia Messiah (Heff IT)	High	Pending
4	Validate POC title preferences (4 categories + extra)	Nadia Messiah (Heff IT)	High	Pending
5	Verify Seamless AI response schema in sandbox	Anthony Pinto (VV)	High	Blocked by #1-2
6	Build Seamless AI HTTP Request node + seniority sort code	Anthony Pinto (VV)	High	Blocked by #5
7	Update Build POC Updates code for 5-row output	Anthony Pinto (VV)	High	Blocked by #5
8	Test end-to-end with 5 sample companies	Anthony Pinto + Nadia	High	Blocked by #6-7
9	Disable AnyMailFinder nodes and go live	Anthony Pinto (VV)	Medium	Blocked by #8

---

*End of Development Specification | Version 1.0*