

A3 – CUJ Runthrough + Demo

Group Name: TemporAI

- Mohamed Sbeinati – 1009163717
 - Zherui Zhang - 1008865122

Submission Date: 2025/09/24

TL;DR

We researched how SMBs struggle with messy inventory data, then tested OpenRefine and ChatGPT on Smell & Smile's legacy export. OpenRefine handled simple clustering/normalization; complex joins and domain conversions failed. We switched to LLM-assisted transforms via prompt iterations to finish the job. Key insight: the best system mixes automation with user input, ultimately feeding a multi-source, LLM-powered data platform.

User Goal

As a small retailer with a messy legacy inventory system, I want to clean, normalize, and categorize my product data, flagging errors and low stock, so I can reorder with confidence and prepare for an AI-assisted data platform that unifies all my sources.

Persona Description

- **Role / Background:** Owner-operator of a small fragrance retailer (Smell & Smile); responsible for inventory, purchasing, and sales.
- **Experience:** Comfortable with Excel; first time using OpenRefine or LLMs for data work.
- **Context:** Time-constrained, domain knowledge heavy, limited technical background.

Tools Used

- **OpenRefine** (clustering, faceting, regex transforms)
- **Google Sheets** (quick checks, buy list filters)
- **ChatGPT** (LLM-assisted regex, column mapping, free-text → structured rules; prompt iteration)
- **Excel** (initial preview/export)

Summary of Findings

Research & Problem Framing

Before testing tools, the user researched typical SMB inventory issues: duplicate SKUs, inconsistent units (ml/ML/mL/tola), mixed item types (raw materials, finished products, packaging), negative stock, and missing costs. The success metric was: produce a clean, categorized dataset with low-stock flags and explainable steps.

Phase 1 — OpenRefine: quick wins, clear limits

What worked well:

- **Clustering:** Normalized “Oud 6ml,” “OUD 6 ML,” and “Oud – 6mL” into one label.
- **Simple unit fixes:** Standardized ml / ML / mL to mL; pcs/PCS → pcs.

- **Facets:** Surfaced anomalies like negative stock and blank costs quickly.

Where it failed:

- **Unit conversions:** Handling tola → mL was brittle across typos and spacing.
- **Compound parsing:** Extracting size + unit from messy free-text names often broke.
- **Rule collisions:** Categorization rules (“tester bottle” vs “bottle of tester oil”) became messy chains that were hard to maintain.

Result: OpenRefine handled straightforward normalization, but struggled with domain-specific conversions and edge cases, demanding heavy manual effort.

Phase 2 — Switching to ChatGPT (LLM-assisted transforms)

To go beyond OpenRefine, the user integrated ChatGPT to generate transformations:

- **Regex parsing:** Extracted `size_value` and `size_unit` from strings like “12 ml,” “.41 oz,” “1 tola.”
- **Unit conversions:** Proposed consistent logic: map tola → 11.66 mL, handle L/cL → mL, leave pcs/kg.
- **Categorization rules:** Built a keyword table with synonyms/negatives. An `_override` column was added for human review.

Prompt iteration: Early prompts failed on decimals/hyphens. Adding **dataset-specific examples** improved accuracy. Asking the LLM to generate explanations + test cases produced more reliable patterns.

Outcome: ChatGPT accelerated regex and logic authoring, but results still required checking. Lesson: automation + domain input outperforms either alone.

Phase 3 — Cleaned Data & Actionable Outputs

Combining both tools, the business owner produced a structured, decision-ready dataset:

- **Cleaned columns:** Standardized names, normalized units (mL, pcs, kg), extracted fields like `size_value`.

- **Custom flags:** Added `NEG_QTY`, `MISSING_COST`, and `REORDER` for low stock.
- **Categorization:** Grouped items into Raw, Packaging, Finished, clarifying what was low or inconsistent.
- **Buy list:** Exported to Google Sheets, filtering on `REORDER` produced a supplier-ready restocking list.

Key Insight

SMB data cleaning is socio-technical: tools automate repetitive work, but only business owners can validate messy, domain-specific edge cases. Sustainable workflows must combine automation with lightweight human review.

Recommendations

For Products

- **Schema-Aware Import Layer:** Auto-detects common SMB schemas and maps columns for review.
- **Domain Rule Packs:** Modular packs. Stored in JSON/YAML for reuse.
- **LLM Transforms w/ Confidence:** Suggestions tagged with confidence. High-confidence (>90%) apply automatically; others flagged for override.
- **Sample-Based Previews:** Let users apply transforms on sample rows and see before/after previews to refine rules.
- **Override Columns:** Auto-generate `_override` columns to preserve human judgment.
- **Snapshots & Sandboxing:** Version data cleaning steps like Git commits; easy rollback if errors occur.
- **Provenance & Audit:** Log every transformation (who, when, confidence). Allow replay on new datasets.
- **Dashboards:** Out-of-the-box connectors for inventory health, buy lists, and error monitoring.

For Future Users

- Start with a small slice of data to test.
- Use OpenRefine for simpler tasks; switch to LLMs for messy parsing.
- Always provide **examples** to LLMs, not vague prompts.
- Keep override columns and save snapshots, mistakes are inevitable.
- Expect context switching; keep notes on steps and timing.

Highlights & Lowlights Table

Task / Moment	Severity	Notes
OpenRefine clustering/unit fixes	Great	Quick normalization; surfaced anomalies fast.
Complex parsing in OpenRefine	Severe	Struggled with edge cases; required custom logic.
LLM regex & rule generation	Great	Faster + more accurate when fed examples.
Categorization & buy list export	Moderate	Needed overrides, but produced actionable output.

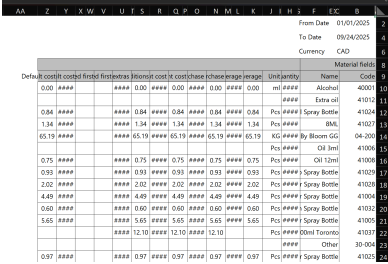
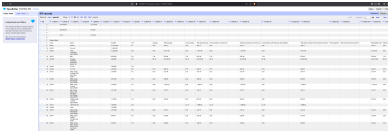
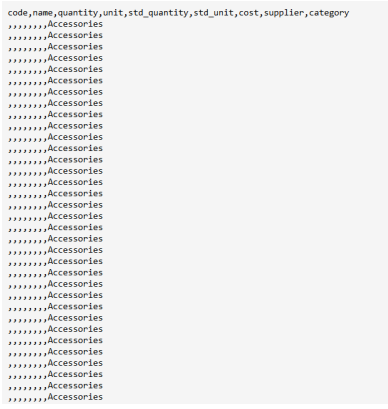
CUJ Overview Table

Task	Time	Switches
Open Excel export & inspect messy inventory	5 min	1
Load into OpenRefine, try clustering/facets	15 min	2
Attempt GREL transforms (unit normalization, regex) → docs/tutorials → back to tool	30 min	3
Switch to ChatGPT for regex/unit conversion help; iterate prompts with examples	40 min	3
Test ChatGPT-generated logic back in OpenRefine & Sheets	20 min	2
Move to Google Colab with ChatGPT-generated Pandas pipeline	50 min	2
Debug file import/export & generate clean CSVs	30 min	1
Export flagged data + buy list to Sheets/Data Studio for visualization	25 min	1

Total Time: ~3 hours 35 minutes

Total Context Switches: 15

Full CUJ Table (Step-by-Step Documentation)

Step	Notes (What + Why)	Screenshot
1	Opened messy Excel export in OpenRefine to inspect inventory. Skipped metadata rows and previewed columns. First goal: see duplicates and blanks.	
2	Used clustering and faceting in OpenRefine to merge similar product names. Also normalized simple units.	
3 (switch)	Tried to handle complex cases but OpenRefine transforms broke down. Switched to ChatGPT to draft regex + unit conversion logic. Iterated prompts with sample rows.	
4	Took the regex and categorization rules generated by ChatGPT, applied them in OpenRefine/Sheets. Created derived columns: size_value, size_unit, standardized_unit. Some mismatches still needed override.	