# AI Powered Search Bar

| | |
|---|---|
| Target release | |
| Epic | Dwell time |
| Document status | DRAFT |
| Document owner | @Jared Poole |
| Designer | @Jared Poole |
| Tech lead | |
| Technical writers | @Jared Poole |
| QA | |

### Objective

Enable users to find and filter orders using a single natural-language input instead of manually configuring multiple dropdowns and date filters.

The feature aims to:

- Reduce cognitive load and time spent searching for orders
- Improve usability for infrequent or non-technical users
- Maintain accuracy, transparency, and trust in filtered results

### Success metrics

| Metric | Target |
|---|---|
| Average time to find a specific order set | <50% vs. dropdown filtering |
| User-reported “search difficulty” | Down significantly (use qualitative) |
| Search response time | < 2 seconds |

### Assumptions

- Users are familiar with logistics concepts (order status, location, courier, dates)
- Users may not know exact system terminology (status names, field labels)
- Natural-language input may be incomplete, vague, or ambiguous
- The system must explain what filters were applied
- Existing dropdown filters will remain available as a fallback (not replaced)

### Requirements

| | Requirement | User Story and Specifications | Importance | Notes |
|---|---|---|---|---|
| 1 | Natural language search input | **User story**<br>As an operations user, I want to search for orders using natural language so that I don't need to manually configure multiple filters.<br>**Specification**<br>Provide a single-line text input above the orders table. Users can submit via the Enter key or the search button. Placeholder text should include an example query.<br>**Acceptance criteria**<br>• Search input is visible on page load<br>• User can submit via Enter and button click<br>• Empty input cannot be submitted | HIGH | Primary entry point |
| 2 | Natural language query parsing | **User story**<br>As a system, I want to interpret user queries and extract structured filters so that search results are deterministic and explainable.<br>**Specification** | HIGH | LLM output must be schema-validated |

| | | | | |
|---|---|---|---|---|
| | | Convert NL input into a validated JSON filter object (e.g. status, location, date range). The AI must not generate executable SQL.<br><br>**Acceptance criteria**<br><br>• Valid NL queries produce structured filter JSON<br>• Invalid or unsupported queries return a controlled error<br>• AI output is schema-validated before use | | |
| 3 | Orders table refresh | **User story**<br><br>As a user, I want the orders table to update in response to my query so I can see the results immediately.<br><br>**Specification**<br><br>Results must be server-side filtered, paginated, and sorted consistently with existing table behaviour.<br><br>**Acceptance criteria**<br><br>• Table updates after successful search<br>• Pagination and sorting continue to work<br>• Total result count reflects applied filters | HIGH | |
| 4 | Applied filters visibility | **User story**<br><br>As a user, I want to clearly see which filters were applied so that I can trust the search results.<br><br>**Specification**<br><br>Display a human-readable summary or filter chips above the table that reflect the parsed filters and exclusions.<br><br>**Acceptance criteria** | HIGH | |

| | | | | |
|---|---|---|---|---|
| | | • Applied filters are displayed after every search<br>• Included and excluded filters are distinguishable<br>• Display matches actual backend filters | | |
| 5 | Date range interpretation | **User story**<br>As a user, I want to use natural date expressions so that I don't need exact timestamps.<br><br>**Specification**<br>Support phrases such as "today", "last 7 days", "from Jan 1 to today". Default to `time_created` unless another date field is explicitly stated.<br>**Acceptance criteria**<br>• Relative and absolute dates are parsed correctly<br>• Default date field is applied when unspecified<br>• Date assumptions are shown to the user | HIGH | |
| 6 | Explicit date field support | **User story**<br>As a user, I want to specify whether the date refers to the created, stored, or collected time.<br>**Specification**<br>Recognise keywords such as "created", "stored", "collected" and map them to the corresponding timestamp fields.<br>**Acceptance criteria**<br>• Correct date field is applied when explicitly stated | HIGH | |

| | | | | |
|---|---|---|---|---|
| | | • Conflicting date fields trigger an error or warning | | |
| 7 | Status include and exclude | **User story**<br>As a user, I want to include or exclude order statuses so I can narrow results accurately.<br>**Specification**<br>Support multiple statuses and negation (e.g. “exclude expired”). Status mapping must align with backend enums.<br>**Acceptance criteria**<br>• Included statuses appear in results<br>• Excluded statuses do not appear in results<br>• Multiple statuses are handled correctly | HIGH | |
| 8 | Location-based filtering | **User story**<br>As a user, I want to search by location name without knowing system IDs.<br>**Specification**<br>Support fuzzy matching on location names. If multiple matches exist, return a clarification or warning.<br>**Acceptance criteria**<br>• Single matching location filters correctly<br>• Multiple matches trigger clarification or warning<br>• Selected location is shown in applied filters | HIGH | |
| 9 | Courier / carrier filtering | **User story**<br>As a user, I want to filter orders by courier so I can isolate specific delivery partners.<br>**Specification** | MEDIUM | |

| | | | | |
|---|---|---|---|---|
| | | Recognise known carrier names (e.g. DHL, UPS) and apply them as filters.<br><br>**Acceptance criteria**<br><br>• Courier name in query filters results correctly<br>• Unknown courier names return a warning | | |
| 10 | Service type filtering | **User story**<br><br>As a user, I want to filter between delivery and returns services.<br><br>**Specification**<br><br>Support explicit keywords (“delivery”, “returns”). If omitted, include all services by default.<br><br>**Acceptance criteria**<br><br>• Delivery-only queries exclude returns<br>• Returns-only queries exclude delivery orders | MEDIUM | |
| 11 | Collected-by filtering | **User story**<br><br>As a user, I want to include or exclude parcels collected by courier, customer, or operator.<br><br>**Specification**<br><br>Parse phrases such as “collected by courier” or “exclude operator collected”.<br><br>**Acceptance criteria**<br><br>• Included collection types appear in results<br>• Excluded collection types are removed | MEDIUM | |
| 12 | Flag-based filtering | **User story**<br><br>As a user, I want to filter or exclude parcels with specific flags so that I can | MEDIUM | |

| | | | | |
|---|---|---|---|---|
| | | handle exceptions.<br><br>**Specification**<br><br>Support flags such as EXPIRED. Both include and exclude conditions must be supported.<br><br>**Acceptance criteria**<br><br>• Flagged orders are filtered correctly<br>• Multiple flags are handled together | | |
| 13 | Ambiguity handling | **User story**<br><br>As a user, I want the system to handle unclear queries gracefully so that I understand how the results were derived.<br><br>**Specification**<br><br>When ambiguity exists, either ask for clarification or proceed with the best assumption and display a warning.<br><br>**Acceptance criteria**<br><br>• Ambiguous queries do not silently fail<br>• User is informed of assumptions made | HIGH | |
| 14 | Error handling | **User story**<br><br>As a user, I want clear feedback if my query cannot be processed.<br><br>**Specification**<br><br>Display user-friendly error messages for invalid, empty, or unsupported queries. Preserve the last valid results.<br><br>**Acceptance criteria**<br><br>• User-friendly error message is shown<br>• Previous valid results remain visible | HIGH | |
| 15 | Security and validation | **User story** | HIGH | For the task, maybe just tell |

| | | | | |
|---|---|---|---|---|
| | | As a system owner, I want to ensure the feature is secure and resistant to misuse.<br>**Specification**<br>Validate all AI outputs against a whitelist. Use parameterised queries only. Reject unknown fields or operators.<br>**Acceptance criteria**<br>• Prompt injection attempts are rejected<br>• SQL injection attempts do not affect the database | | us how you would address it. |
| 16 | Performance | **User story**<br>As a user, I want search results to load quickly, even with large datasets.<br>**Specification**<br>Server-side filtering and pagination required. Target sub-2s response time for typical queries.<br>**Acceptance criteria**<br>• Typical queries return within 2 seconds<br>• Pagination works with filtered results | HIGH | |

## User interaction and design

#### Primary Interaction Flow

1. User enters a natural-language query into the search bar
   Example:

   “Show me all parcels delivered to Location A from 1st January to today. Exclude expired parcels collected by the courier.”
2. User submits query (Enter or Search button)
3. System:
   - Parses query into structured filters
   - Applies filters to orders dataset
   - Updates the table results

4. UI displays:
    - Filtered order list
    - **Applied Filters Summary**
    - Optional warnings or assumptions

#### Applied Filters Summary (Mandatory)

Displayed above the table as readable text or filter chips.

Example:

> **Applied filters:**
> Location: Location A
> Status: Delivered
> Date (Created): 1 Jan 2026 → Today
> Excluded: Expired, Courier-collected

This ensures:

- Transparency
- Debuggability
- User trust

#### Ambiguity & Assumptions

If the query is ambiguous, the system must:

- Either request clarification
  *"Did you mean Location A (Office Locker) or Location A (PUDO)?"*

**OR**

- Make the best assumption and clearly state it
  *"Assuming 'date' refers to order creation time."*

#### Error States

| Scenario | Expected Behavior |
|---|---|
| Empty query | Show validation message |
| Unrecognized terms | Show friendly error + examples |
| Conflicting filters | Explain conflict |

| | |
|---|---|
| System / AI failure | Show error, keep previous results |

### Open Questions

| Question | Answer | Date Answered |
|---|---|---|
| | | |

### Out of Scope

The following are **explicitly excluded** from this feature:

- Authentication and role-based access control
- Editing or mutating orders
- Replacing existing dropdown filters entirely
- Analytics, dashboards, or saved searches
- Multi-language query support (English only)
- Voice input
- Full-text search over comments or notes