

Filter UI spec

👤 Created by	👤 Vibhor Meshram
🕒 Created time	@May 6, 2025 11:10 AM
🏷️ Tags	

UI Filter Builder Component Specification

Purpose

A visual component that allows users to build complex logical filters using **rules** and **groups of rules**. It reflects the structure of the following schema (inspired by Hex's filter UI).

Data Schema Overview

The filter UI works on the following fields:

Column Name	Description
brand_names	Brand name text field
manufacturer_d_names	Manufacturer descriptive names
product_codes	Product codes
event_type	Type of the event
event_text	Raw event text
event_text_description	Description of the event
event_text_additional	Any additional event info
patient_problems	Patient-related problems
product_problems	Product-related problems
web_address	Associated web address

	All columns (special option)
--	------------------------------

UI Elements

1. Filter Rule Block

Each rule consists of:

- **Column Selector** (dropdown: includes all columns +)
- **Operator Selector** (context-aware, options based on column selection)
- **Value Field** (text input or multiselect based on operator)

Supported Operators:

- `contains_substring`
- `contains_word`
- `not_contains`
- `not_contains_word`
- `in`
- `not_in`

Example:

```
Column: brand_names
Operator: contains_word
Value: Promark
```

2. Filter Group Block

- A **group** can contain any number of filters or nested groups
- Logical operator: `AND` / `OR` toggle
- UI: Indented block for visual nesting

Interactions

- **Add Filter**: Adds a rule inside the current group
 - **Add Filter Group**: Adds a nested group
 - **Delete**: Deletes a rule or a group
 - **Toggle Operator**: Switch between **AND** / **OR**
-

Layout Behavior

- Indented display for nested groups
 - Visual clarity through borders and light separators
 - Collapsible nested groups (optional enhancement)
-

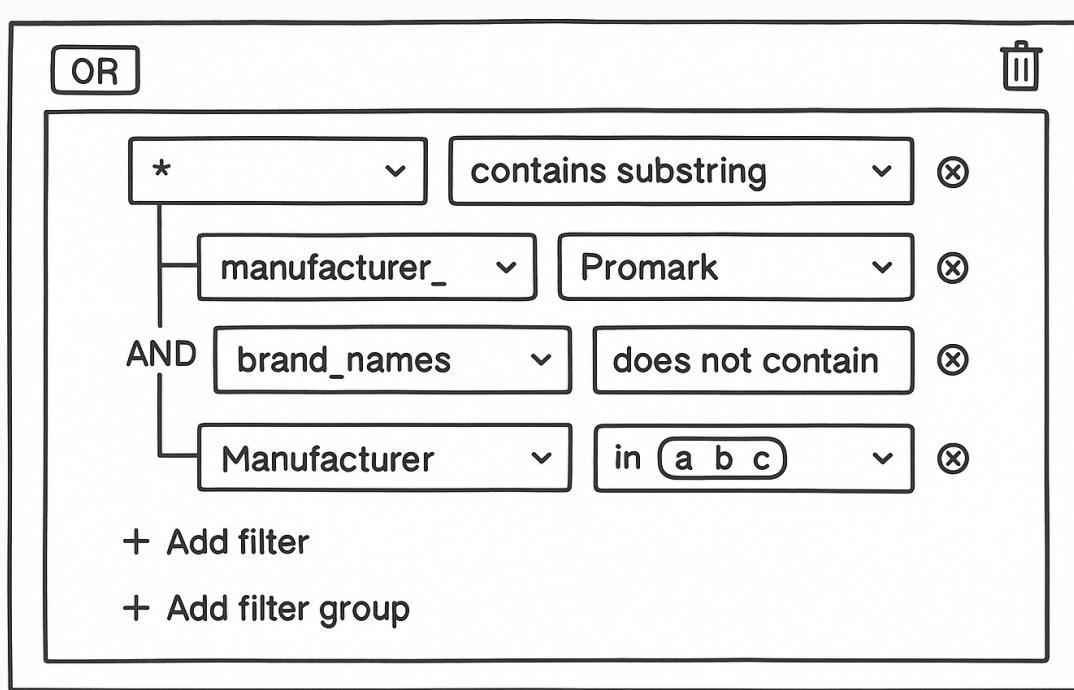
Wireframe Mockup

Below is a wireframe-style mockup illustrating the component UI based on your schema and an example filter configuration.

Filter Configuration Modeled:

```
{
  "logical_operator": "or",
  "rules": [
    {
      "column": "*",
      "operator": "contains_substring",
      "value": "Aseptico"
    },
    {
      "logical_operator": "and",
      "rules": [
        {
          "column": "brand_names",
          "operator": "contains_word",
          "value": "Promark"
        },
        {
          "column": "brand_names",
          "operator": "contains_word",
          "value": "Aseptico"
        }
      ]
    }
  ]
}
```

```
{
  "column": "manufacturer_d_names",
  "operator": "not_contains",
  "value": "Hifi"
}
]
},
{
  "column": "Manufacturer",
  "operator": "in",
  "value": ["a", "b", "c"]
}
]
```



🧠 Validation Rules (UI-Only)

- If column is , only allow `contains_substring` or `contains_word`
 - Operators like `in`, `not_in` require multi-select input
 - Display user-friendly messages for invalid combinations
-

Example Use Cases

1. Simple Filter

Find rows where `brand_names` contains word "Promark".

2. Nested Filter

Find rows where either:

- Any column contains "Aseptico"
- OR `brand_names` contains "Promark" **AND** `manufacturer_d_names` does **not** contain "Hifi"

Deliverables

- Dynamic rule creation
- Nested group support
- Operator- and column-aware UI logic
- Schema-limited column list