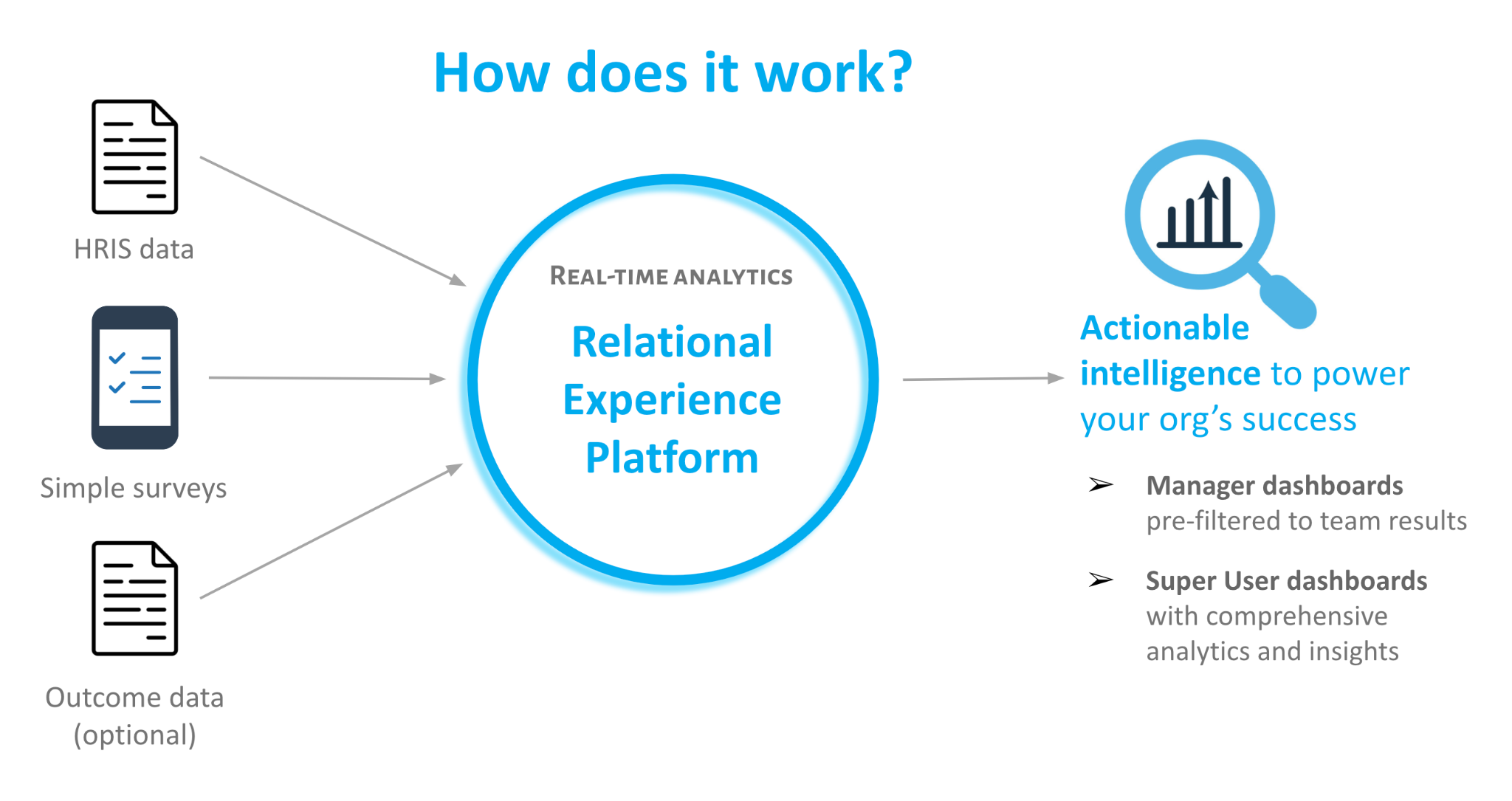
**Welcome to OrgAcuity, Dindo!**

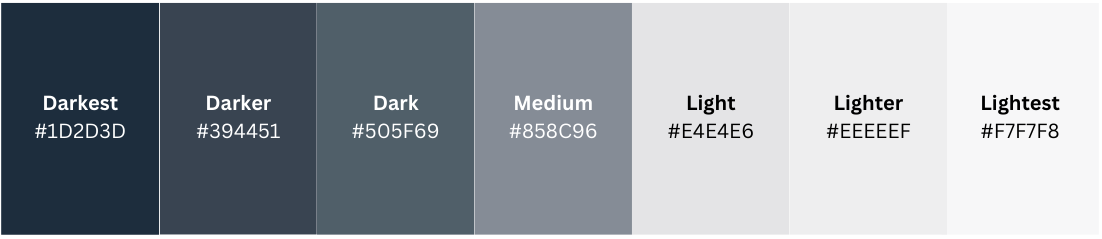
**Overview**

This section outlines important high-level information that should accelerate your onboarding. We highly recommend reading through this document in its entirety to understand the full scope of the requirements before getting started.



The overall vision for this front end is a single dashboard with a **clean**, **decluttered**, **simple**, and **elegant design** aesthetic that allows the data to take center stage. This is achieved via ample white space and strategic use of colors (e.g., muting less important elements and highlighting key pieces of information on which users should focus their attention). This UI should be inviting and engaging for even the most non-technical of users.

Across the dashboard, blue (#00ACEE) indicates a favorable result, red (#FA5551) indicates an unfavorable result, and gray (#E4E4E6) indicates a neutral result. The colors below should be used for the various shades of gray…



We have drawn a lot of inspiration from Glint’s UI, a competitor acquired by LinkedIn in 2018. Here is a helpful reference containing a style guide (e.g., icons) written by their former product designer: <https://www.jonathanruiz.net/glint> Glint has recently been integrated into Microsoft’s suite of enterprise software since acquiring LinkedIn, so this UI looks very different today.

*Environments*

We have three environments (projects) within Google BigQuery (GBQ): o*rgacuity-dev-316714* (development), *orgacuity-prod-299705* (production), and *orgacuity-backup* (production copy). All data for actual customers (employer\_id >= 1) and demos (employer\_id = 0) needed for this app reside in the *curated, surveys, and reference* schemas within the GBQ project.

*End Users and their Access Controls*

Our analytics platform needs to service three types of users: (1) Admin, (2) Super Users, and (3) Managers. Super Users have elevated permissions that allow them to view results for the entire company, while Managers are those with 1 or more direct reports who should see results only for respondents within their reporting line (e.g., direct reports only or everyone who rolls up to them, if they have multiple layers of people beneath them).

When users authenticate to the dashboard via SSO, their email should be evaluated against their access role(s) to determine how data should be filtered, what pages should be displayed, and what attributes should be available to them. The requirements are detailed in the “Access Management” section of this document.

Super User access supersedes Manager access; therefore, if a user is listed as both a Super User and Manager, they should receive Super User access (i.e., they can view all data for their company). Managers without Super User access should only see data filtered to their manager\_id (curated.viz\_mgr\_vw\_setup drives this), no other filters such as Attributes, and only the following dashboard pages:

* Home >> Summary
* Home >> Action
* Participation >> Trends
* Participation >> Representation
* Participation >> Teams
* Questions >> Ratings
* Questions >> Comments
* Questions >> Multiple Choice
* Analytics >> Heatmap

Dashboard pages default to Manager = ‘All’, which represents the entire population of records the user can see (i.e., entire company for Super Users or entire reporting line rollup for Managers). At the global level, users can filter to a specified Manager’s entire rollup (all survey respondents who report up through the manager) or a specified Manager’s direct reports only (by checking the *Direct Reports Only* box). Note that the only page that the Manager and Direct Reports Only filters do NOT apply to is Home >> Alerts.

Beyond the users of our platform, it is important to understand three groups of people for which we process and store data: (1) employers provide data on their entire population of workers (**“active workers”**); (2) employers invite a sample (or all) of these workers to participate in surveys (**“survey recipients”**); and (3) a subset (or all) of the invited workers respond to surveys (**“survey respondents”** submit **“survey responses”**). Note that we use worker, employee, and person/people interchangeably as well as employer, company, and organization.

There are several key fields that are important to note:

* **employer\_id**: unique numeric identifier for each employer (customer). This column exists in every GBQ table, and it is a partitioned column to support performant queries.
  + **Note:** This is how we ensure that each employer’s data are presented only to that employer and no one else. This is a VERY important field for our application.
* **snapshot\_date**: the effective date of the worker data provided by the employer (note that for each survey, we retrieve and integrate the most recent worker data with a snapshot\_date that is on or before survey\_start\_datetime)
* **survey\_id**: unique string identifier for each survey
* **employee\_id**: unique string identifier for each employee
  + It is possible for an employee\_id to be shared across employers, as this is only unique within a given employer’s records. Therefore, employer\_id + employee\_id ensures uniqueness for a particular person.
  + Since employee\_id and email may change over time (e.g., company acquisition, employee name change), we apply the following to employer-provided data to create an immutable unique identifier for each person: TO\_HEX(SHA256(CONCAT(COALESCE(stg.employee\_id), COALESCE(stg.email)))). This hashed id is used to populate employee\_id and manager\_id columns downstream from staging tables for a common reference. The history of employee\_id and email changes is tracked at the worker level in people.dim\_worker\_ids\_scd.
  + **manager\_id**: the employee\_id for an employee’s manager

Survey items are organized and grouped under factors – a high-level categorization (e.g., Engagement, Manager Effectiveness). Customers can utilize our standard survey items or create their own custom survey items. Our standard survey items have designated reference\_id values beginning with “s” (*standard*) that are stored in reference.ref\_survey\_config; these reference\_id values do not change from employer-to-employer or survey-to-survey. Custom survey items do not have fixed reference\_id values. The keys for trending survey items and overall scores for multi-item factors (i.e., multiple items for a particular factor) are provided below:

* **Overall multi-item factors:** When item is null, join on factor + reference\_id
* **Standard items:** When category <> ‘Custom’, join on reference\_id
* **Custom items:** When category = ‘Custom’, join on item

A critical table that drives global Manager filtering is curated.viz\_mgr\_vw\_setup. For each snapshot\_date, this table contains every manager (user\_id column) and each manager’s managers (manager\_id column). The snapshot\_date that should be used for each survey is defined in surveys.def\_survey\_parameters.snapshot\_date. The user\_layer field indicates how many layers a manager is from the senior-most leader in the company (e.g., CEO), while the manager\_layer field indicates the *relative* layer of a manager to the respective user\_id. For example, if John is a manager who reports to the CEO and Jane is a manager who reports to John, John will have a user\_layer value of 1 and Jane will have a manager\_layer value of 1 for rows in this table where the user is John.

* This table is intended to build the global manager hierarchy for filtering results and does NOT contain employees who are not people managers – only people with 1 or more direct reports. This table must be joined with the manager\_id column in our viz tables to capture all employees – managers and individual contributors.
* There will be tables that support some dashboard pages that have no match to certain manager\_id values in viz\_mgr\_vw\_setup. This is expected due to small samples that require us to suppress the data (e.g., Analytics >> Drivers). Therefore, the global Manager filter must function like a LEFT JOIN, displaying all values regardless of whether there is a match to various base data sets.
* For “Experience” survey types, the global Manager filter should reflect curated.viz\_mgr\_vw\_setup records for the snapshot date defined in surveys.def\_survey\_parameters. For “Lifecycle” survey types, the global Manager filter should reflect curated.viz\_mgr\_vw\_setup records for the max(snapshot\_date), as there is not a snapshot\_date value defined in surveys.def\_survey\_parameters for “Lifecycle” survey types.

*Surveys*

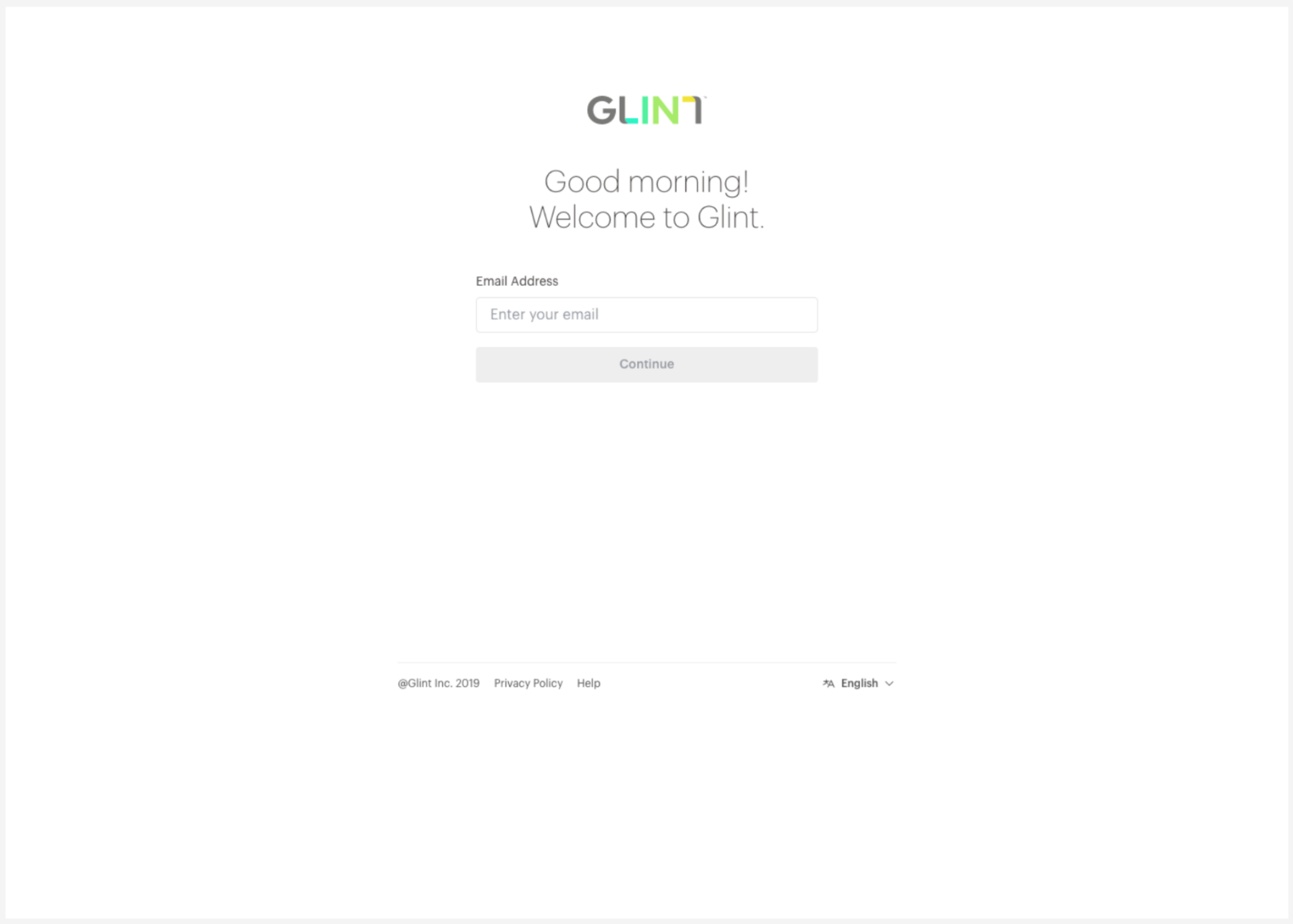
We support two types of surveys: (1) Experience and (2) Lifecycle. Every customer will have Experience surveys, but not all will have Lifecycle surveys. Experience surveys are point-in-time surveys with a distinct start and end datetime (usually a two-week window), while Lifecycle surveys are “always on” and triggered by events such as a hire (e.g., onboarding surveys) or termination (e.g., exit surveys).

**General Design**

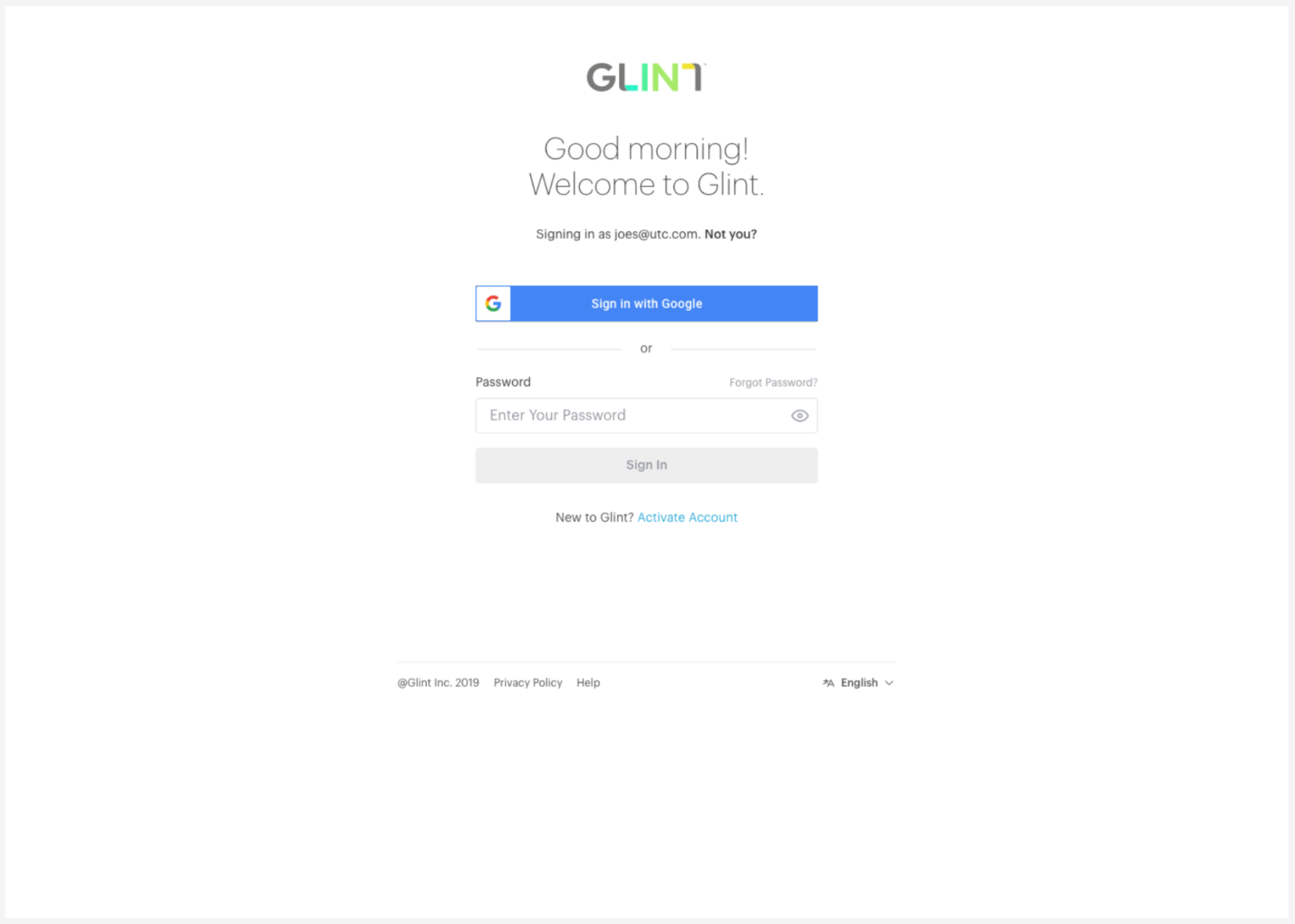
**Our existing NiceGUI dashboard plus the changes outlined in this document constitute success for our new frontend**.

* AG Enterprise (Grid + Charts), available for all major Javascript frameworks, should be used for the UI. Custom CSS styles will be required to achieve the desired formatting and aesthetics.
  + Data table design: <https://www.ag-grid.com/example-hr/>. All data tables should look like this and have sorting, filtering, search, and Excel export capabilities. Advanced features like pivots and aggregate functions should be disabled. For tables that display hierarchical data (e.g. Participation >> Teams) as well as the tables on the Influencers pages, names should be formatted like [this example](https://www.ag-grid.com/example-hr/) (i.e., name in darker font, job title beneath with lighter font).
  + Bubble Charts: <https://www.ag-grid.com/charts/gallery/multiple-bubble-series/>
  + Heatmap: <https://www.ag-grid.com/charts/gallery/heatmap-with-labels/>
  + Line Chart: <https://www.ag-grid.com/charts/gallery/simple-line/>
  + Normalised Bar Chart: <https://www.ag-grid.com/charts/javascript/bar-series/>
* The same font types, colors and sizes, tooltips, dialog boxes (info icons), animations (page transition and chart loading effects), and overall design aesthetics in our NiceGUI dashboard should be retained. The exception is that NiceGUI grays should be replaced with the grayscale shown above.
* Note: Do not build the Influencers >> Teams page that exists in our NiceGUI app, as this is being deprecated.
* Our current NiceGUI dashboard is not reactive; we have fixed/static dimensions on the dashboard elements (e.g., pages, tiles). We’d like a more flexible UI to accommodate differences in users’ screen sizes and amount of text that needs to be displayed on screen.
* The global Manager filter in NiceGUI (left hand pane) is somewhat janky (e.g., when searching, the tree doesn’t expand to the searched value), so we would love to see a new design for this.
* The Survey Name filter should always default to the survey with an Experience survey type that has the most recent survey\_start\_datetime (surveys.def\_survey\_parameters table) that is on or before the current date. Surveys with a survey\_start\_date after the current date should not show in the dashboard at all. The Survey Name options in the dropdown list should be sorted from most recent (top) to earliest (bottom).
* The requirements in this doc are specific to surveys with an “Experience” survey type. For surveys with a “Lifecycle” survey type (the only other type), the dashboard differences are outlined [here](https://docs.google.com/document/d/16K3BD7OkHUwIvDWEm_X6PNPbxHnk7t5jnNhKNL0tyeg/edit?usp=sharing).
* Our current dashboard is mostly reading pre-aggregated metrics. Going forward, we would like the following dashboard pages to read row-level data, with the calculations and aggregations that are currently performed in GBQ being performed in the UI. This will allow dynamic date filtering for “always on” Lifecycle survey types, which function differently from the Experience and Pulse survey types that have defined start and end dates.
  + Questions >> Ratings
  + Questions >> Comments
  + Questions >> Multiple Choice (new page… requirements are below)
  + Analytics >> Heatmap (page is currently named Distributions)
* The front end should have an OrgAcuity-branded URL.
* We need a user login page where users can authenticate to the dashboard (with the proper level of access… Super User vs. Manager). Shown below is the design we want to achieve for the first and second login screens. OrgAcuity’s logo (svg file) is [here](https://drive.google.com/file/d/1LUObSrQCN4qv_LrwkxumLBhF87-bFKbf/view?usp=drive_link). You can ignore “Good Morning!” and include “Welcome to OrgAcuity.” beneath the logo.

**Screen 1**



**Screen 2**



Each employer will have a different number of attributes based on the optional data dimensions they choose to provide on their active employees file. Therefore, we should only surface non-null attributes as options to choose in the UI.

Provided in the reference.ref\_attributes table are the display names and *derived* flags for each standard filter/attribute. Provided in [this table](https://docs.google.com/spreadsheets/d/1yltVG2heDr1bElR2Kbd9llNHeuUF3IckcYzqnY6vVs4/edit?usp=sharing) are the display names (queries) and *derived* flags for each custom filter/attribute.

* *Filters* are referred to here as a mechanism for limiting/filtering the underlying data. These drill down to the attribute value level (Attribute Type >> Attribute Name >> Attribute Value).
* *Attributes* refer to a grouping of people (e.g., respondent segments represented by bubbles in a bubble chart or columns in a heatmap). These drill down to the attribute name level (Attribute Type >> Attribute Name), as the attribute values display as groups on the respective page.

Attributes should be made available on the Participation >> Representation and Analytics >> Heatmap pages.

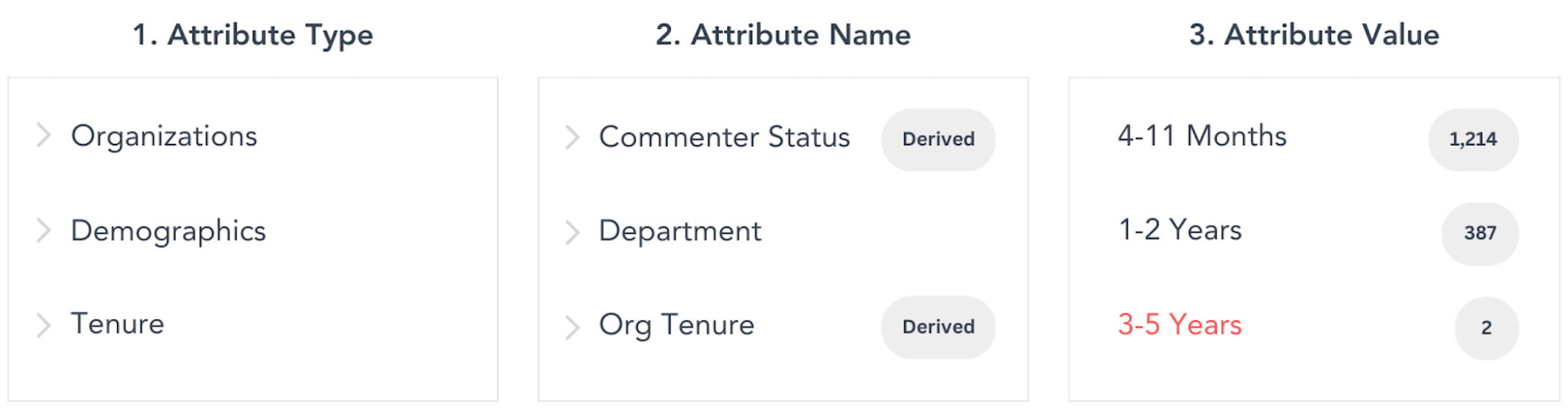
Filters should appear on the left-hand pane (beneath global Manager filter) for the relevant pages. All standard and custom attributes should be available as filters on the following dashboard pages, and these should be *multi-select* filters:

* Questions >> Ratings
* Questions >> Multiple Choice
* Analytics >> Heatmap

[Here](https://drive.google.com/file/d/1DCmCimdu0IrUjHR0GJzJOq8wQFRm8Etw/view?usp=drive_link) is a short video demonstrating the desired *functionality* of the multi-select “Add Filter” option on the left-hand pane. As shown in the image below, we need tags on Attribute Names and Attribute Values.

* Since some attributes are “derived”, meaning we use data provided by employers and respondents to create additional data dimensions, we need to indicate these in the Attribute Name drop-down list. The table linked above indicates which attributes are derived.
* The dynamic metric tag next to each attribute value in the dropdown shows how many distinct employee\_ids there are for each option, with red attribute values indicating an insufficient n-count for filtering results (i.e., COUNT DISTINCT (employee\_id) < reference.ref\_survey\_parameters.quantitative\_threshold for the selected employer\_id and survey\_id). Attribute values should be sorted per [this logic](https://drive.google.com/file/d/1WROw2vDR810SVOX3pKQUrILB_XJGFNel/view?usp=drive_link), within two groups: (1) attributes with sufficient employee count show at the top; and (2) attributes with insufficient employee count show at the bottom. The above sorting logic should be applied *within* each group.

**Attribute Hierarchy Filter Design:**



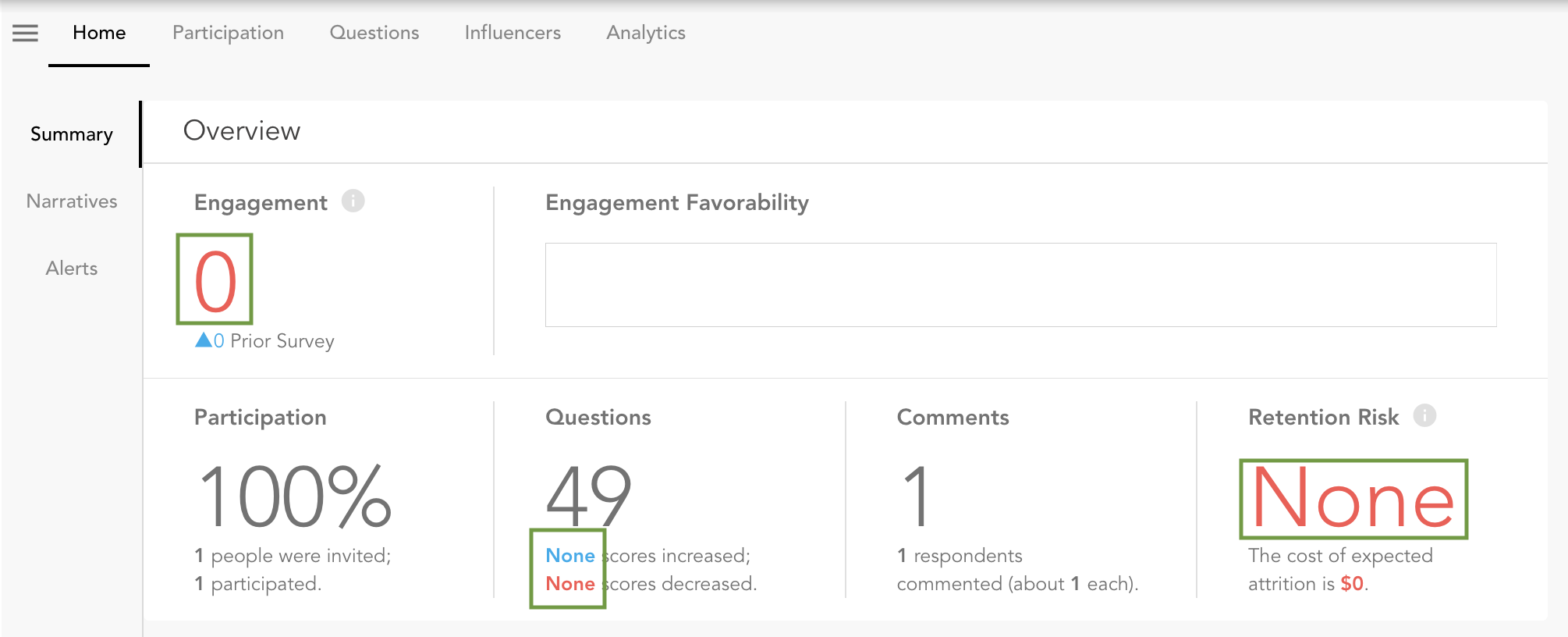
Notes:

* Only show the attributes that are not null for the user-selected survey\_id. For example, custom\_grouping\_dimension\_10 should not be a filtering option if there are only null values in this field.
* Users should not be able to select a filter option that results in COUNT DISTINCT (employee\_id) < reference.ref\_survey\_parameters.quantitative\_threshold for the respective employer\_id and survey\_id.
* For attributes where COUNT DISTINCT (employee\_id) < reference.ref\_survey\_parameters.quantitative\_threshold for the respective employer\_id and survey\_id, metrics should be suppressed.
* [Here](https://drive.google.com/file/d/1vtgpPHXfHiIQ4wYos1Vx5ZlDS6MfVLUu/view?usp=drive_link) is the logic for retrieving attribute types and names.

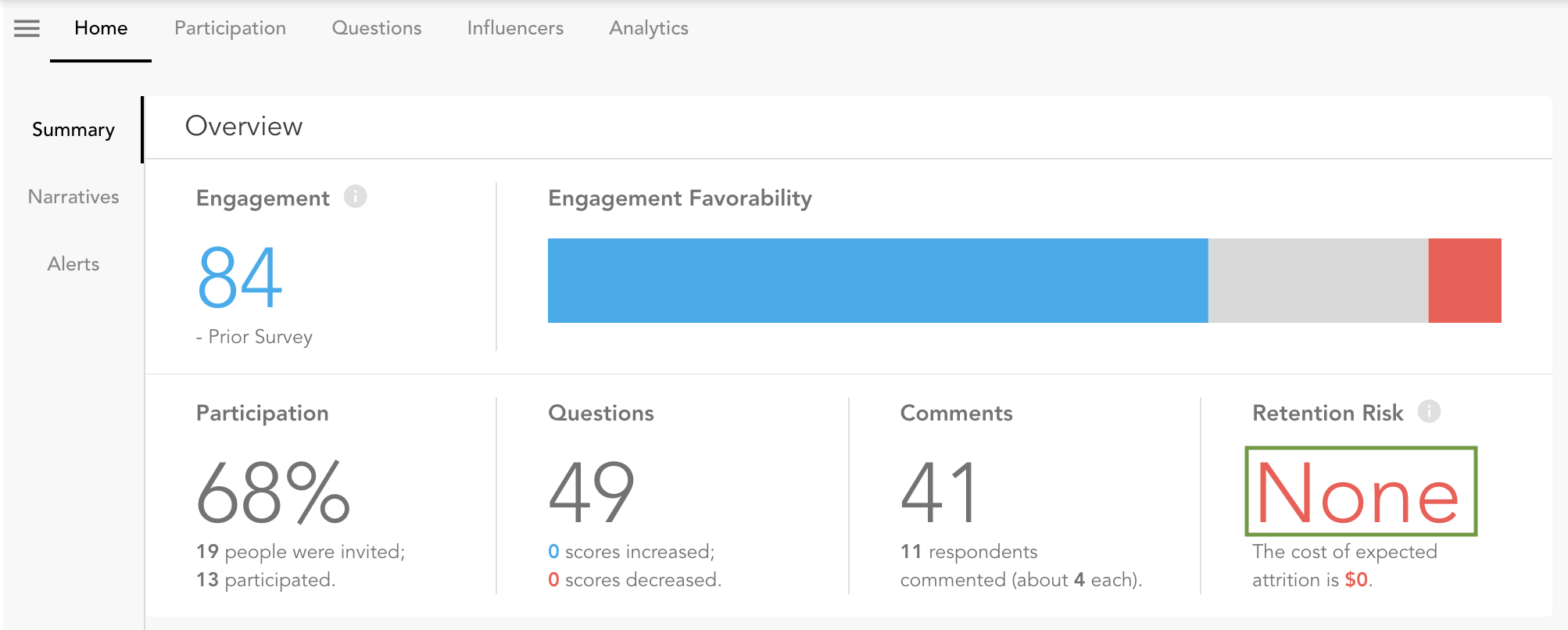
The dashboard page order, base GBQ tables, and selection criteria for every dashboard page (existing and new) are defined [here](https://docs.google.com/spreadsheets/d/1420xuf1v8yTNA7zTBzKyZHEilVJfpp2Ic1FiJKg1Ey0/edit?usp=sharing).

**Page: Home >> Summary**

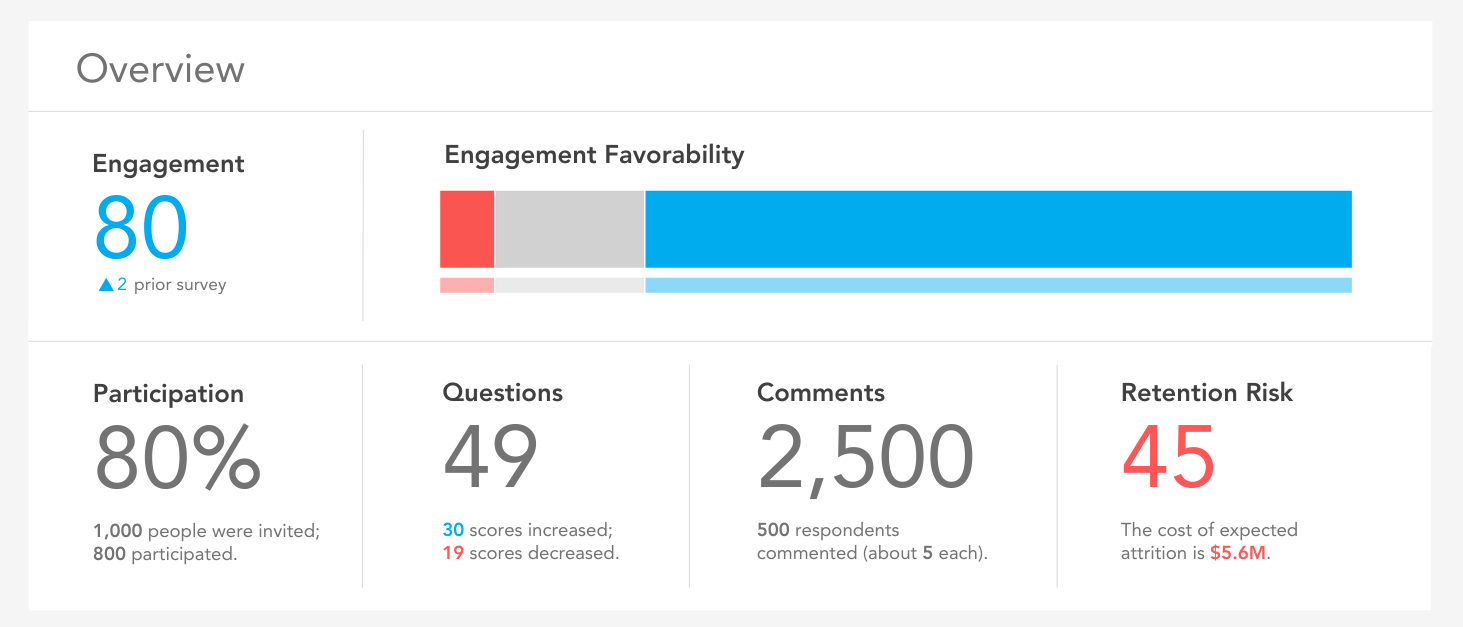
* We need to implement handling for two scenarios that are currently not being addressed.
  + **Scenario 1:** When the minimum n-count threshold **is not met** for a particular manager, (a) the Engagement score needs to be “--” (this is currently 0, which incorrectly implies a highly disengaged team); (b) “None” should be replaced with “--” in the “scores increased” and “scores decreased” sentence under Questions; and (c) the Retention Risk needs to be “--” rather than “None”.



* + **Scenario 2:** When the minimum n-count threshold **is met** for a particular manager but there are no respondents flagged for Retention Risk, we should display 0 instead of “None”.



* Point to viz\_strengths\_opportunities table to surface Top Strengths and Top Opportunities on this page. Top Strengths should be sorted from highest score to lowest score. Top Opportunities should be sorted from highest neutral\_pct to lowest neutral\_pct.
  + Add an info icon with this [help text](https://docs.google.com/document/d/1j6ZGdB8O6hxOeKC8QsgjPXc5DUtoJRyOQzkpVJ_8L1U/edit?usp=sharing).
* Add a thin secondary favorability stacked bar (with 45% transparency) so users can compare current favorability (bold) to prior favorability (faint) for engagement and comment sentiment. Both current and prior survey favorability metrics are provided in the curated.viz\_summary\_metrics (Summary page). It would be ideal to include both *n* and % in the tooltips for both current and prior survey metrics. See mock up below…

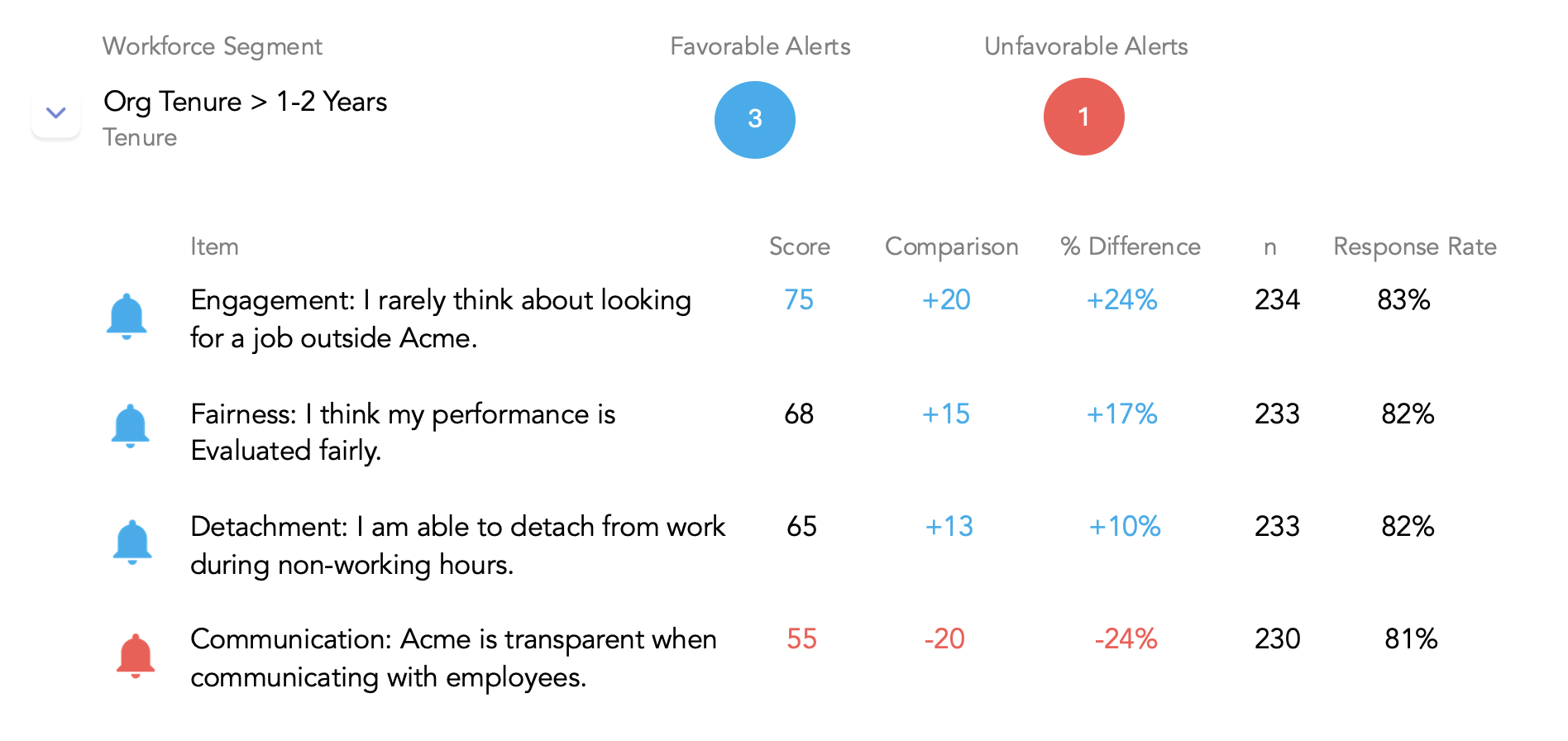
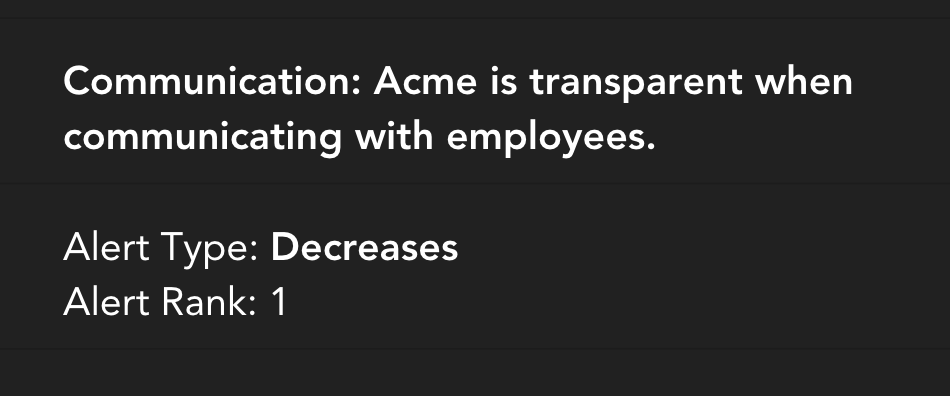
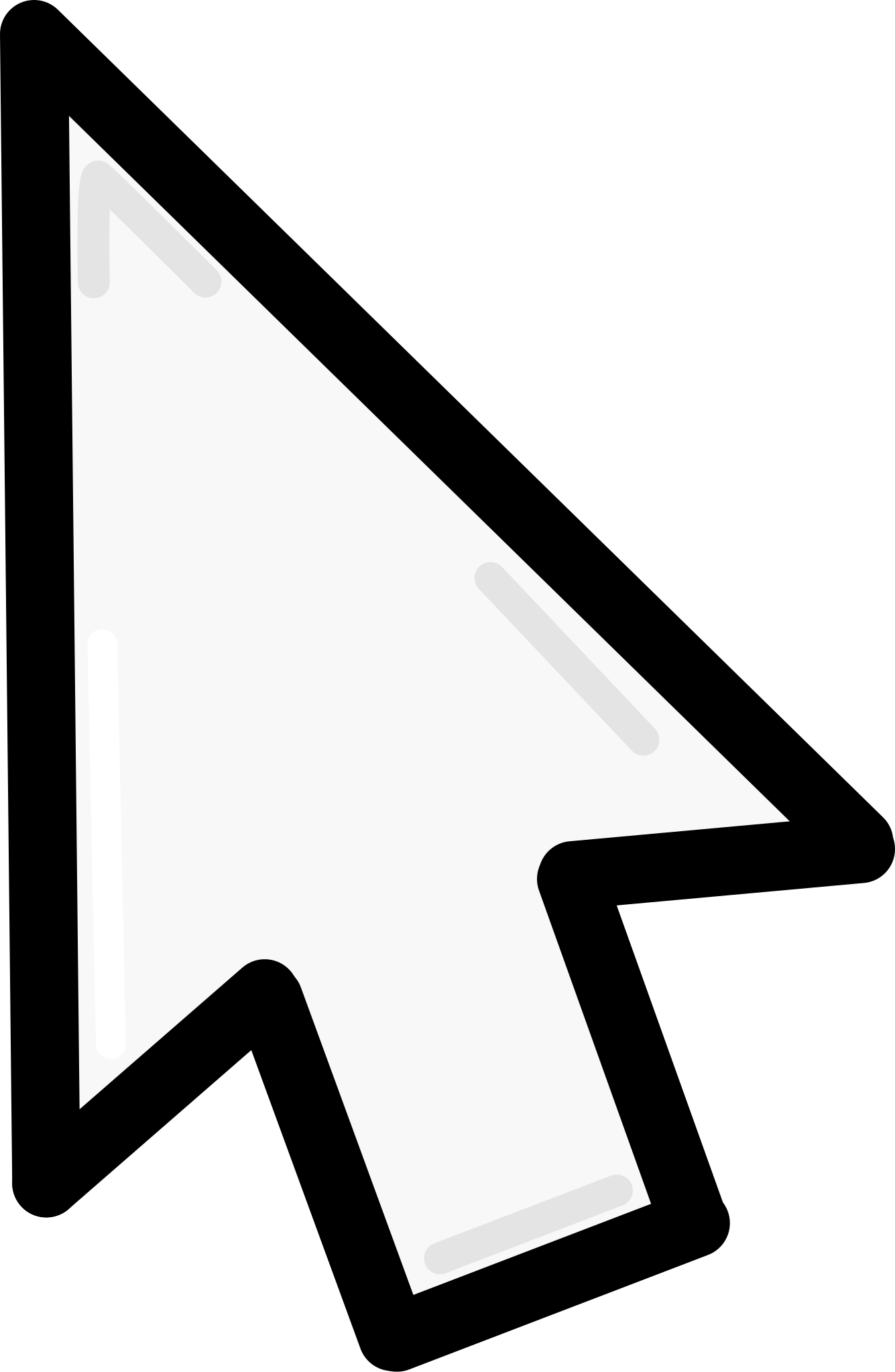


* Color the Participation Rate per the following thresholds:
  + 0 - 25% = #FA5551
  + 26% - 50% = #E4E4E6
  + 51% - 100% = #00ACEE
* Add this [help text](https://docs.google.com/document/d/1j6ZGdB8O6hxOeKC8QsgjPXc5DUtoJRyOQzkpVJ_8L1U/edit?usp=drive_link) to the Strengths and Opportunities section.
* Add scores to the Strengths and Opportunities sections to compliment the favorability distributions reflected by the horizontal stacked bar charts. The source data is curated.viz\_strengths\_opportunities, and the items should be sorted by the “rank” field in this table. The mockup below reflects the desired design…



**Page: Alerts**

Since workforce segments for alerts have a hierarchical structure (Attribute Type >> Attribute Name >> Attribute Value >> Factor/Item), it would be ideal to create expandable *Workforce Segments* like the mock up provided below. This will help users quickly identify which workforce segments have the most areas to celebrate vs. opportunities to improve.

****

We should display only three columns when the table’s records are collapsed:

* **Workforce Segment (**Attribute Name > Attribute Value, with Attribute Type displayed in smaller, lighter font beneath)
* **Favorable Alerts** (display the count of level 2 records where Alert Type is ‘Increases’ or ‘High Scores’)
* **Unfavorable Alerts** (display the count of level 2 records where Alert Type is ‘Decreases’ or ‘Low Scores’)

We need a way to display the following metrics on the level 2 rows:

* Score
* Comparison
* % Difference
* n
* Response Rate

We need two page-specific filters:

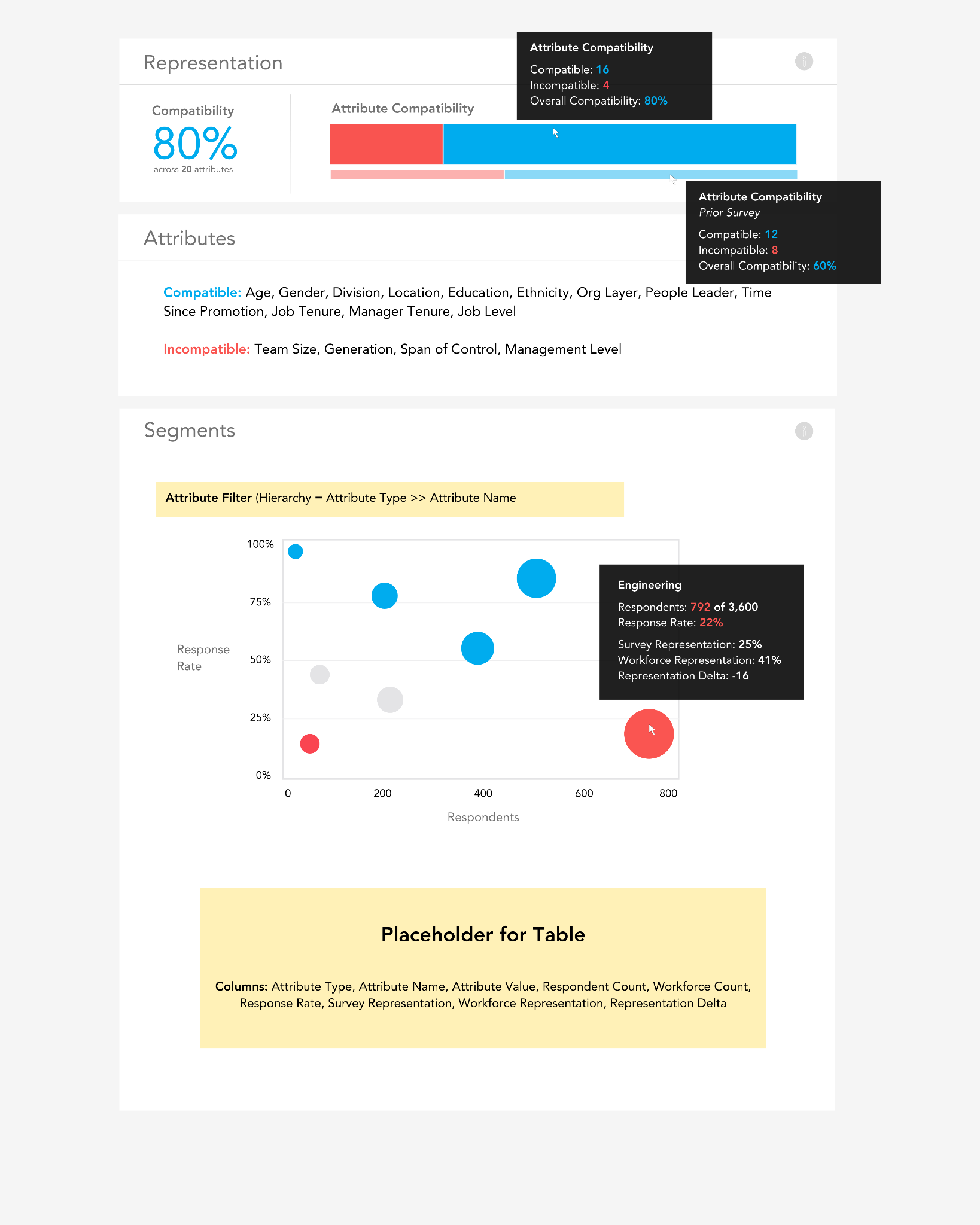
* **Alert Type** (multi-select)
* **Show** (single-select… display top X alerts based on Rank field)
  + Options
    - All
    - Top 10 *(default)*
    - Top 25
    - Top 50
    - Top 100

**Page: Participation >> Teams**

* Replace the existing help text available via the info icon with [this text](https://docs.google.com/document/d/1oUwoGzsDqpdLjZ22MG28cMhTmvGd7IQV06zDOnwXeZs/edit?usp=sharing).
* We should display managers (teams) in a drillable tree structure (like [this example](https://www.ag-grid.com/example-hr/)) rather than displaying the Layer value before the manager name. Note that these layer values are dynamic based on global Manager filter selections. For example, Layer 0 in this Team field represents people who report into **the specific Manager selected via the global filter**.

**Page: Participation >> Representation (NEW PAGE)**

Shown below is the desired design for this new dashboard page:



The help text that should be made available via the info icons are provided below:

* [Representation](https://docs.google.com/document/d/1cFOwpSX1SjazWZQA2iAQT7Jaw2g1g5cKW_D1R1YGbtY/edit?usp=sharing)
* [Segments](https://docs.google.com/document/d/1oUwoGzsDqpdLjZ22MG28cMhTmvGd7IQV06zDOnwXeZs/edit?usp=sharing)

**Required tables:**

* viz\_participation\_representation (“Representation” and “Attributes” sections)
  + [SQL](https://drive.google.com/file/d/1qMrOSJBGlUs6wP3GoUGhhhT7lm0Tyc3U/view?usp=drive_link) for metrics and detail
* viz\_participation\_distributions (“Segments” bubble chart and table)
  + Only surface records where show = 1.

**Overall Compatibility Metric Color:**

* 0 - 49% = #FA5551
* 50% - 74% = #E4E4E6
* 75% - 100% = #00ACEE

**Bubble Chart**

**Axes:**

* **X-axis:** The x-axis (respondent count) should be scaled *relative* to the minimum (far left) and maximum (far right) number of respondents for the user’s filter selections.
* **Y-axis:** The y-axis (response rate) should be fixed to 0% (bottom) and 100% (top).

**Bubble Size:**

* Bubble size should be based on *relative* respondent count. We should predetermine a minimum and maximum size for bubbles in this chart. If the largest respondent count for a user-selected attribute segment is 400, this segment should receive the maximum bubble size. If the largest respondent count for a *different* user-selected attribute segment is only 150, this segment should also receive the maximum bubble size.

**Bubble Colors:**

* Bubble colors should be based on response rate ranges outlined below, and both the respondent count and response rate in the tooltip should match the bubble color.
  + 0 - 25% = #FA5551
  + 26% - 50% = #E4E4E6
  + 51% - 100% = #00ACEE

**Metric definitions:**

* Respondent Count = sample\_n
* Workforce Count = population\_n
* Response Rate = response\_rate
* Survey Representation = sample\_pct
* Workforce Representation = population\_pct
* Representation Delta = sample\_pct - population\_pct

**Page: Questions >> Ratings**

Currently, our Ratings page provides the ability to view scores by Manager + Direct Reports Only (via the global filters on the left hand pane). In addition to these Manager + Direct Reports Only cuts, we need the ability to look at scores by intersections of attributes (e.g., scores by manager + location + job level + tenure band values). All scores and attributes for filtering reside in the curated.viz\_survey\_responses table.

Note: With the current pre-aggregated approach, we enforce the minimum quantitative n-count within GBQ SQL. With this new approach, we will need to read individual-level records and enforce the minimum n-count threshold within the UI (per **reference.ref\_survey\_parameters.quantitative\_threshold**), and suppress the metrics when a user selects a filter -- or combination of filters -- that result in the distinct employee\_id count falling beneath the threshold. This threshold can vary employer to employer, and survey to survey.

Data will be needed from three tables: (1) curated.viz\_survey\_responses (base data), (2) reference.ref\_display\_names (dashboard display names for custom dimensions), and (3) reference.ref\_survey\_parameters (minimum n-count thresholds). You can view [this script](https://drive.google.com/file/d/1YC4JP28BoGPoAHhCl9gt0-L2TB3eFvLW/view?usp=drive_link) to see how factor/item trends are calculated, as calculations differ between standard vs. custom survey items. This script will also demonstrate the logic to match an individual's survey response to the Manager hierarchy for global filtering.

Other:

* Change “What are our **strengths** and **opportunities**?” to “What are our **highest** and **lowest** scores?”
* Change “Top Strengths” to “Highest Scores”
* Change “Top Opportunities” to “Lowest Scores”

The first column in the table should be a new field named “Driver Strength”. This column should be populated by joining the user-selected manager (manager\_id + direct\_reports\_only) to the output of [this query](https://drive.google.com/file/d/1H63UxtxRrTYOK6got_zu3zOpnyNGpxN8/view?usp=drive_link). This [help text](https://docs.google.com/document/d/1y3wRCDbCguWh6LqjCHpiRF3wjxLdEOi-/edit?usp=drive_link&ouid=111450598158587379038&rtpof=true&sd=true) should be added to the Dialog for this table to educate users on what this field represents.

**Page: Influencers >> Influencers**

Add the following help text to this page, rather than the help text from NiceGUI:

* [How are influencers identified?](https://docs.google.com/document/d/1SrF4wqksrbTsQ4Q18qBlh4zP9wcEgtyYz5YYjoN3YfU/edit?usp=drive_link)
* [What is a Relative Influence Score?](https://docs.google.com/document/d/1q917HzwEAMQsA9fqxQXfJgZhEJXXhpSEMHUMvU0EzG0/edit?usp=drive_link)

**Page: Analytics >> Heatmap (rename… currently named “Distributions”)**

Data will be needed from three tables: (1) curated.viz\_survey\_responses filtered to question\_type = ‘Opinion Scale’ (respondent scores), (2) curated.viz\_survey\_recipients (population data for response rate calculations), (3) reference.ref\_display\_names (dashboard display names for custom dimensions), and (4) reference.ref\_survey\_parameters (minimum n-count thresholds).

We need to replace the table on the Distributions page with a dynamic heatmap containing every Factor + Item down the y-axis and each value of the selected Attribute across the x-axis (as shown in the image below). By default, only factor-level rows/metrics should display in the heatmap, and users should be able to drill into the factors to see rows/metrics for the individual survey items related to each factor. Both horizontal and vertical scrolling will need to be enabled given the number of display values is not static across surveys and dimensions. We need to display the following metrics within tooltips when users hover over each cell/metric:

* Favorable
* Neutral
* Unfavorable
* Response Rate

**Notes:**

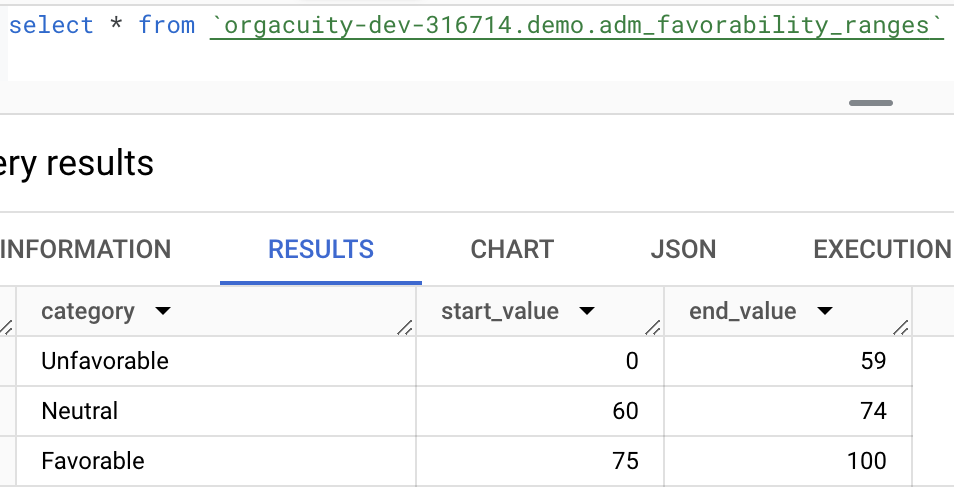
* There should be a toggle with two options: “Scores” and “Changes”...



* + When a user selects “Scores” (the default option), the metrics in the heatmap should represent average scores: AVG(CAST(response AS INT64)). We should display the metric rounded to the nearest whole number.
  + When a user selects “Changes”, the metrics in the heatmap should display the difference between the average score for the user-selected survey and the average score for the prior survey. Use [this logic](https://drive.google.com/file/d/1rsjAbp3SdZOhcRP9KCgGY5N7zbKBCqWd/view?usp=drive_link) to identify the survey\_id for the prior survey.
* There should be an additional toggle that only displays when the “Scores” option above is selected; cell colors and density for “Changes” should always be *relative*. This toggle should have two options: “Relative” and “Absolute”...



* + When a user selects “Relative” (the default option), cell coloring is *relative* to the median value (50th percentile) based on all changes or scores shown in the heatmap (excluding respondent counts) across all Factors/Items for the user-selected Attribute.
  + When a user selects “Absolute”, cell colors and density should be based on the *absolute* ranges defined in reference.ref\_favorability\_ranges…

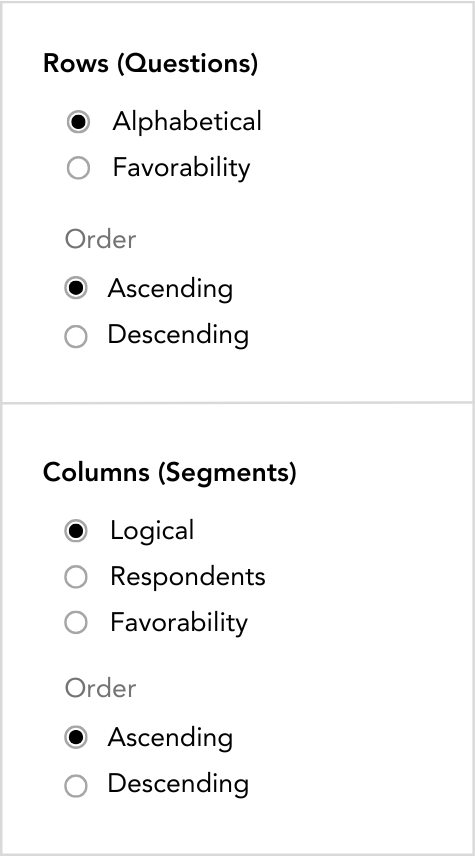


* The Linkage Method filter impacts the change metrics only.
  + If “Respondent”, join current and prior survey data by employee\_id (i.e., show changes based on people who participated in both surveys).
  + If “Attribute”, join current and prior survey data by the selected attribute (i.e., survey score changes may be based on different people who share the same attributes at each point in time).
* **Sorting.** We should be able to sort heatmap rows and columns via the “Sort” option pictured in the heatmap example below.
  + We need the ability to sort the heatmap *rows* (“Questions”) in both ascending and descending order by the following methods:
    - **Alphabetical**… this is the default
    - **Favorability** (based on average score across all columns for each row)
  + We need the ability to sort the heatmap *columns* (“Segments”) in both ascending and descending order by the following methods:
    - **Logical** (e.g., alphabetical, lowest to highest tenure band)... this is the default sort (see [this logic](https://drive.google.com/file/d/1WROw2vDR810SVOX3pKQUrILB_XJGFNel/view?usp=drive_link))
    - **Respondents** (respondent count)
    - **Favorability** (based on average score across all rows for each column)
* The respondent count (row 1 in heatmap) for each attribute value can be retrieved via COUNT(DISTINCT employee\_id) FROM viz\_survey\_responses table.
* If a cell has a missing value, or the respondent count is less than def\_survey\_parameters.quantitative\_threshold, default to “--”.
* The Factor, Item, and All columns should be fixed when users horizontally scroll to view columns that may not fit on the page. By default, only the Factor field should show; the Item record(s) should appear when drilling into Factor. This preserves real estate on the screen and is a logical design choice since most factors are not multi-item factors (i.e., they have only 1 item associated with them).
* 10 is the maximum number of attribute values that should be displayed along the x-axis. If there are more than 10 attribute values, the remainder (the attribute values with the lowest respondent counts) should be grouped into an “Other” category and the column metrics should represent this entire group.
* Add an info icon with [this help text](https://docs.google.com/document/d/1aWgVG4Ymo_w1iSxBquURjP9Vqf5eoOyBFlli_shKxwI/edit?usp=drive_link).

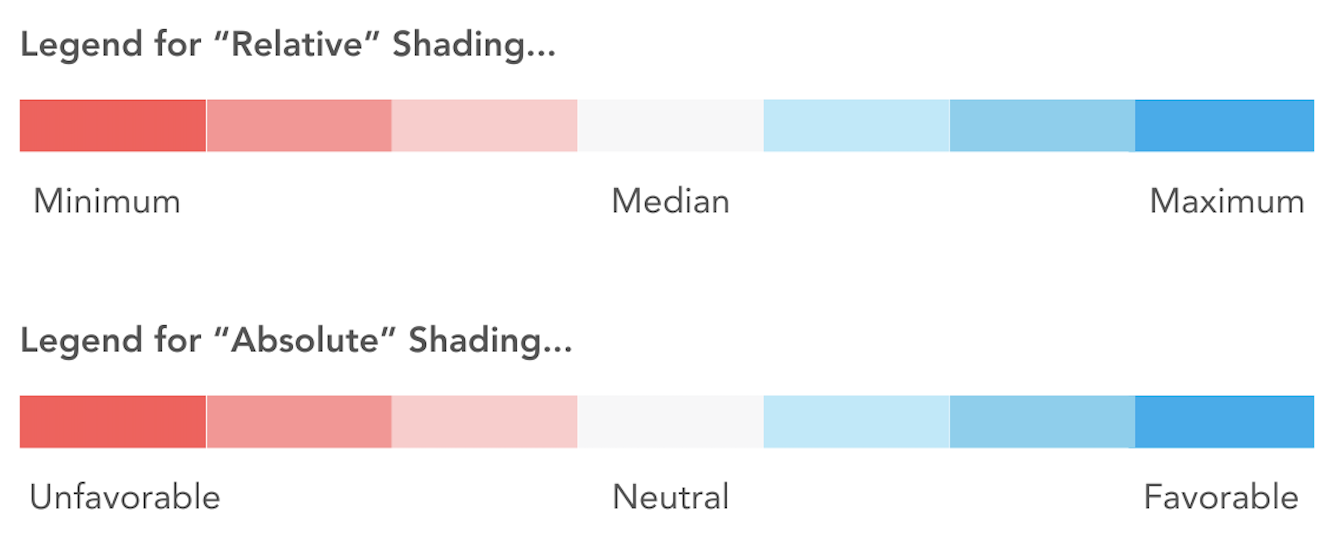
**Heatmap:**



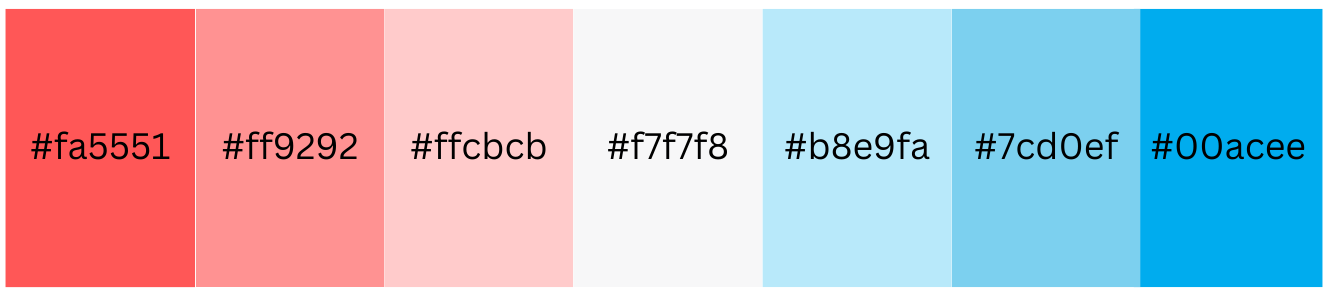
**Sort:**

****

**Legend (center beneath heatmap):**

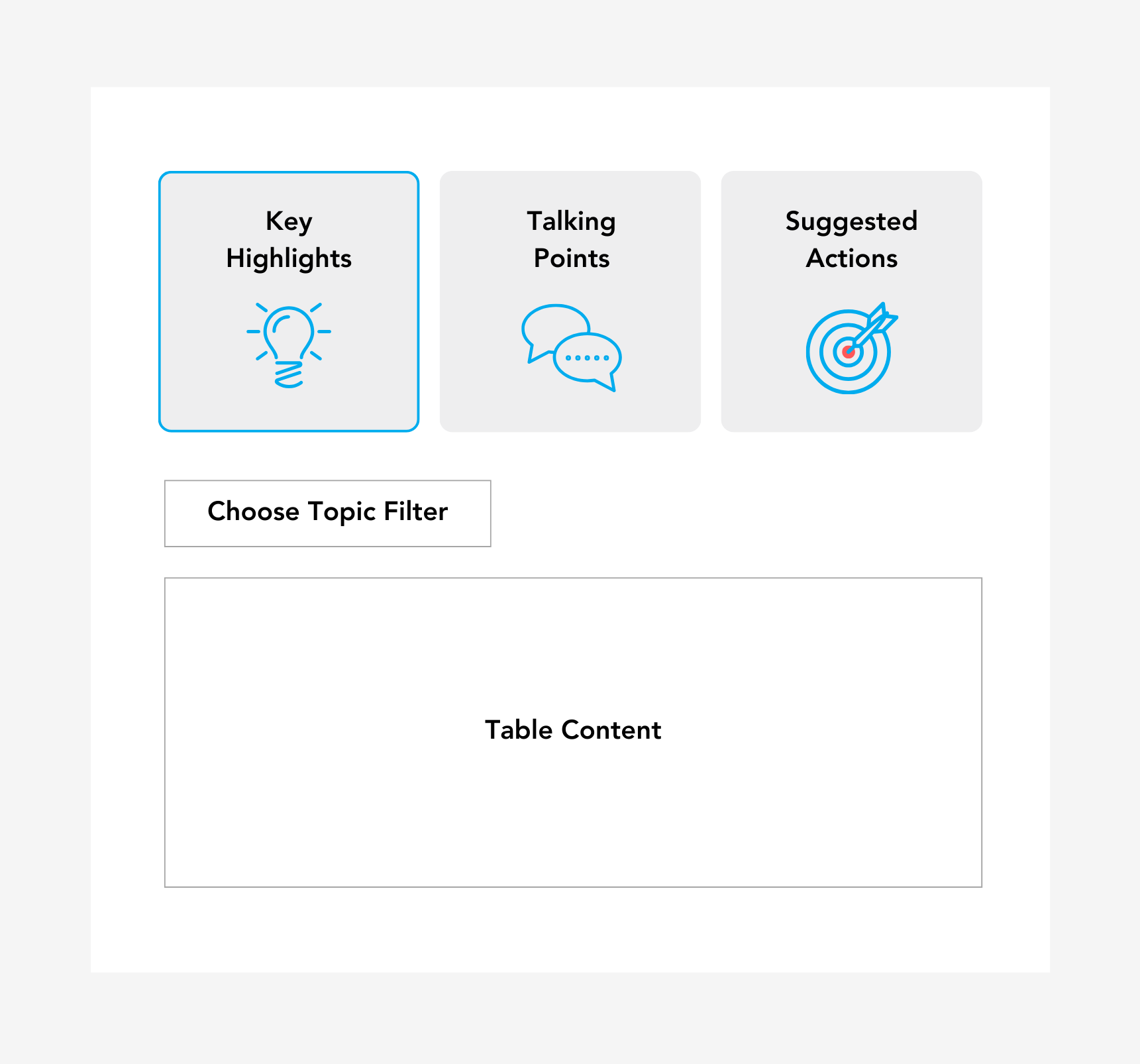


**Colors:**

****

**Page: Home >> Action (rename… currently named “Narratives”)**

* Replace “Generate **key highlights** and **insights stories**.” with “**Discover**, **share**, and **act**.”
* Regex function to bold key terms and add bullets in place of astrisks: REGEXP\_REPLACE(REGEXP\_REPLACE(narrative, r"\\*\\*(.\*?)\\*\\*", r"<b>\1</b>"), r'\b\\*\b', '●')
* Replace the “Type” filter with more engaging buttons as shown below. We should show an outline or something to indicate which button has been selected so they don’t look identical. The svg images are stored here:
  + [key\_highlights.svg](https://drive.google.com/file/d/1NIkQkukxW0JV7JPHJrQjAB0vjgWuCpDn/view?usp=drive_link)
  + [talking\_points.svg](https://drive.google.com/file/d/1gU1AchAc3x147ofuj3_kXLaIuM5F1cNS/view?usp=drive_link)
  + [suggested\_actions.svg](https://drive.google.com/file/d/1vE9rGPueX9p6e9L5Zg2Q6LLIqJeKlEVA/view?usp=drive_link)

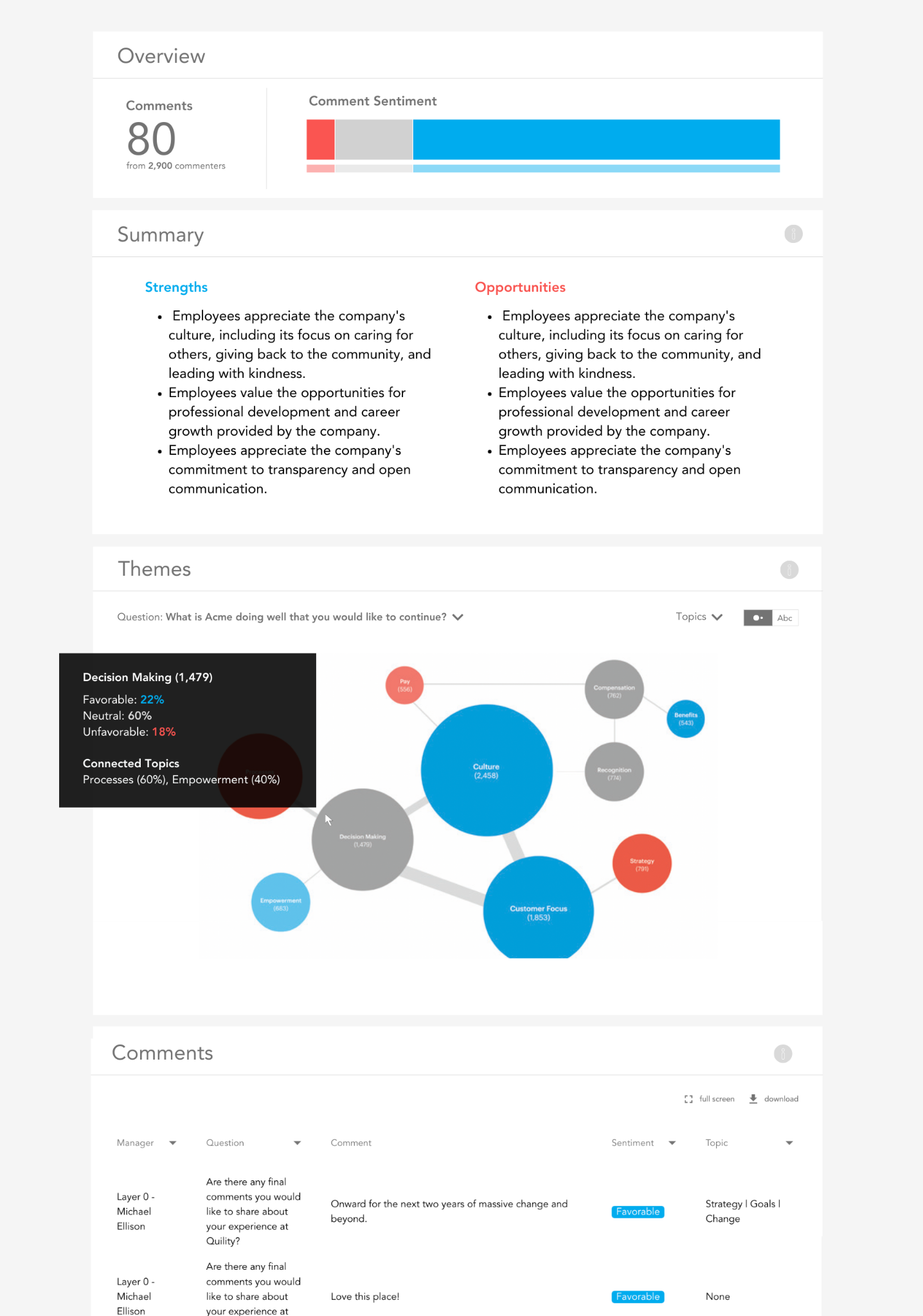


**Page: Questions >> Comments**

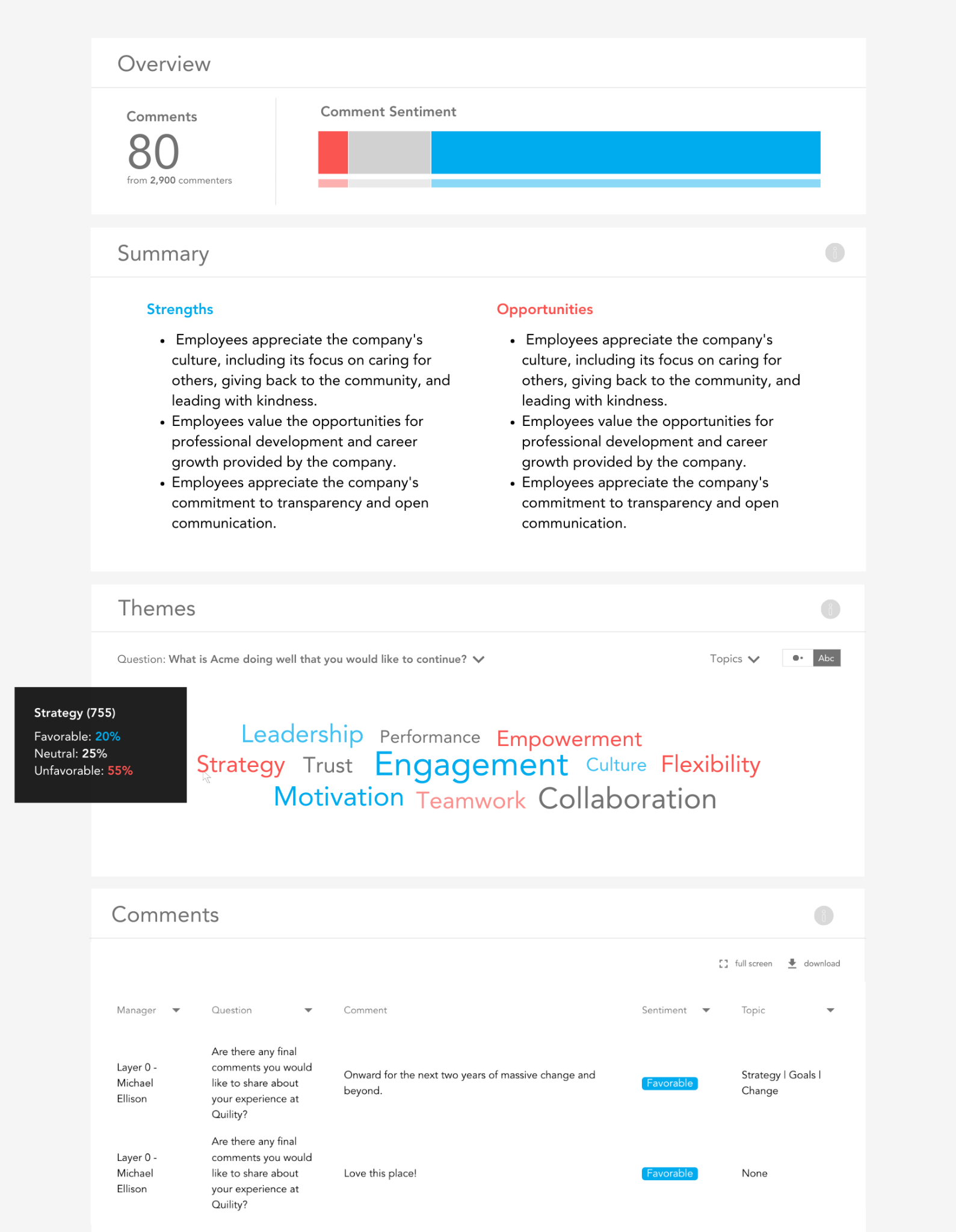
* We need to refactor the Comments page to achieve the following (UI mockup below):
  + For prior survey sentiment metrics (thin, faint stacked bar shown in mockup below), use [this logic](https://drive.google.com/file/d/1rsjAbp3SdZOhcRP9KCgGY5N7zbKBCqWd/view?usp=drive_link) to identify the survey\_id for the prior survey.
  + Users need to toggle between “Topics” and “Keywords” by Question. The Question filter should be multi-select (users can select one question up to all questions). We also need the ability to display these topics/keywords within both a network graph (e.g., When people mention “culture” what other keywords/topics do they tend to mention?) and a word cloud (e.g., What is the relative frequency and sentiment by keyword/topic?).
  + For the network graph, we can adapt and join this [edge\_list.sql](https://drive.google.com/file/d/1FCNDIgA317UBOUgKilxgk0RVDBOy51oL/view?usp=drive_link) to [overall\_node\_statistics.sql](https://drive.google.com/file/d/1wIuKfAGxpDzAYpLMfpLUynA8tQj6Ii0e/view?usp=drive_link) on “selection\_type” and “word”.
  + For the word cloud, [overall\_node\_statistics.sql](https://drive.google.com/file/d/1wIuKfAGxpDzAYpLMfpLUynA8tQj6Ii0e/view?usp=drive_link) should be used.
  + Since AG Charts doesn’t appear to support network diagrams or word clouds, visx may be a good library for these visuals:
    - <https://airbnb.io/visx/network>
    - <https://airbnb.io/visx/wordcloud>
  + For word coloring, we can use a basic algorithm starting from a uniform distribution (33.3% favorable, neutral, and unfavorable):
    - Step 1: If a word has 75% or more of a given sentiment category, it should be assigned the respective color for that sentiment (e.g., #00ACEE for words that are at least 75% favorable).
    - Step 2: If a word does not have 75% or more of a given sentiment category, the category with the highest percentage is the base color we should use for the word, and the shade of that color should be determined by the range from 33.4% (lightest) to 75% (darkest).
    - Step 3: In the event of a tie for the highest sentiment category (e.g., favorable\_pct = 45%, unfavorable\_pct = 45%), assign the color gray.
  + Network Graph
    - Only display the top 10 keywords or topics (based on frequency) when the network graph toggle option is selected to avoid congestion.
    - Node size is based on the *relative* topic or word frequency.
    - The thickness of lines connecting nodes is based on the *relative* number of connections between nodes.
  + Add text search to the table so users can search for comments with specific words.
  + Our current Comments table contains a pipe-delimited string of topics when comments are tagged with more than one topic. We’d like to replace this string with a set of non-editable tags (no “x”) that look something like [this](https://drive.google.com/file/d/1knfgHPiGiF7Lh1_ZBIsMlv1rj8wGxHzM/view?usp=drive_link).
* **Note:** curated.viz\_comment\_detail.manager\_id\_show may be null for “Lifecycle” survey types, in which case the respective comment should only show when users filter to manager\_id = ‘All’. Users can then filter further by other dimensions (as long as distinct(employee\_id) >= reference.ref\_survey\_parameters.qualitative\_threshold).

**Mockup of New Comments Page**

*1. Page with Comments toggle set to ‘network graph’*



*2. Page with Comments toggle set to ‘word cloud’*



**Note:** The base tables for this page are viz\_comment\_summary (Summary section) and viz\_comment\_detail (everything else). Overall sentiment logic for the current and prior surveys is available [here](https://drive.google.com/file/d/1TpSr0jmaFeEJpYvbk0cyEp0w1VzsdYRj/view?usp=drive_link). The viz\_comment\_detail logic for the Super User dashboard is [here](https://drive.google.com/file/d/1bTxWI6ydfF7u09vG3N6c-e0XW7ukVrCa/view?usp=drive_link), and the viz\_comment\_detail logic for the Manager dashboard is [here](https://drive.google.com/file/d/1gDWbPIP6ZfqUzd1SubpwcqPWOmMWHgOj/view?usp=drive_link).

**Page: Questions >> Multiple Choice**

We need an additional dashboard page (Questions >> Multiple Choice) to show distributions of survey respondents’ selections. You can reference [this script](https://drive.google.com/file/d/1eIJQw83tQNdef_AsyAmVDlxgUcIt0-Ho/view?usp=drive_link) for the general logic.

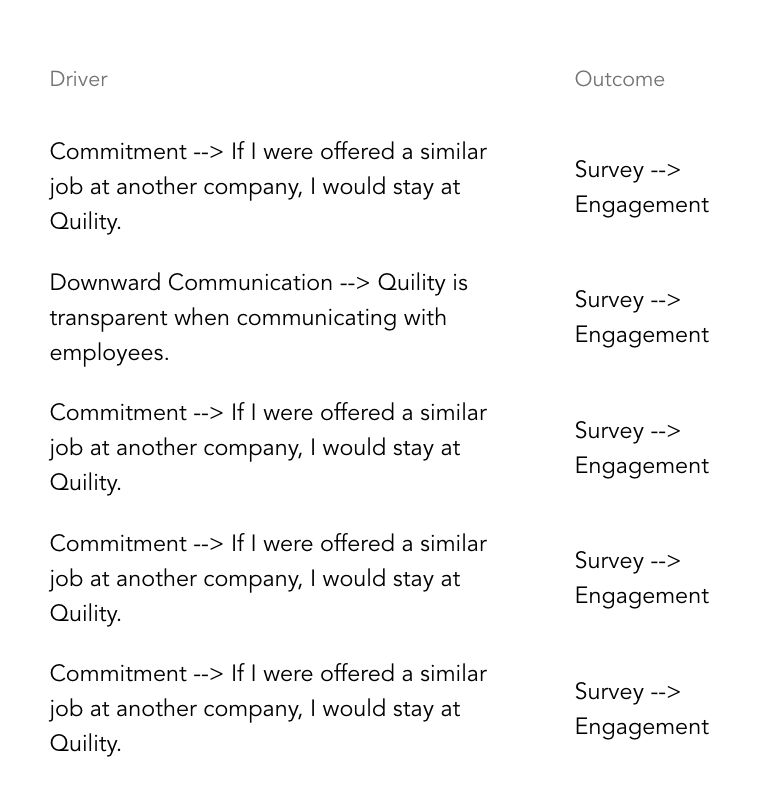
We need to surface the selection stats for each multiple choice question. Each question’s selections should be sorted from highest to lowest %. Ideally, we should dynamically expand the page vertically to display all questions, rather than selecting each question one-by-one.

[Here](https://drive.google.com/file/d/1x0CzKF_rinGrHNaOhJnwBUcDLZFl2TMz/view?usp=drive_link) is an example of a design for this page that displays the 3 categories with highest frequency for each question, by default. Note that the green bars should be #00ACEE – consistent with the other pages. The size and format of bars should match the stacked bar charts on other pages, and the “Comparison” column is not needed.

Note: The same “Add Filter” option on the Questions >> Ratings page should be available on this page as well.

**Page: Analytics >> Drivers**

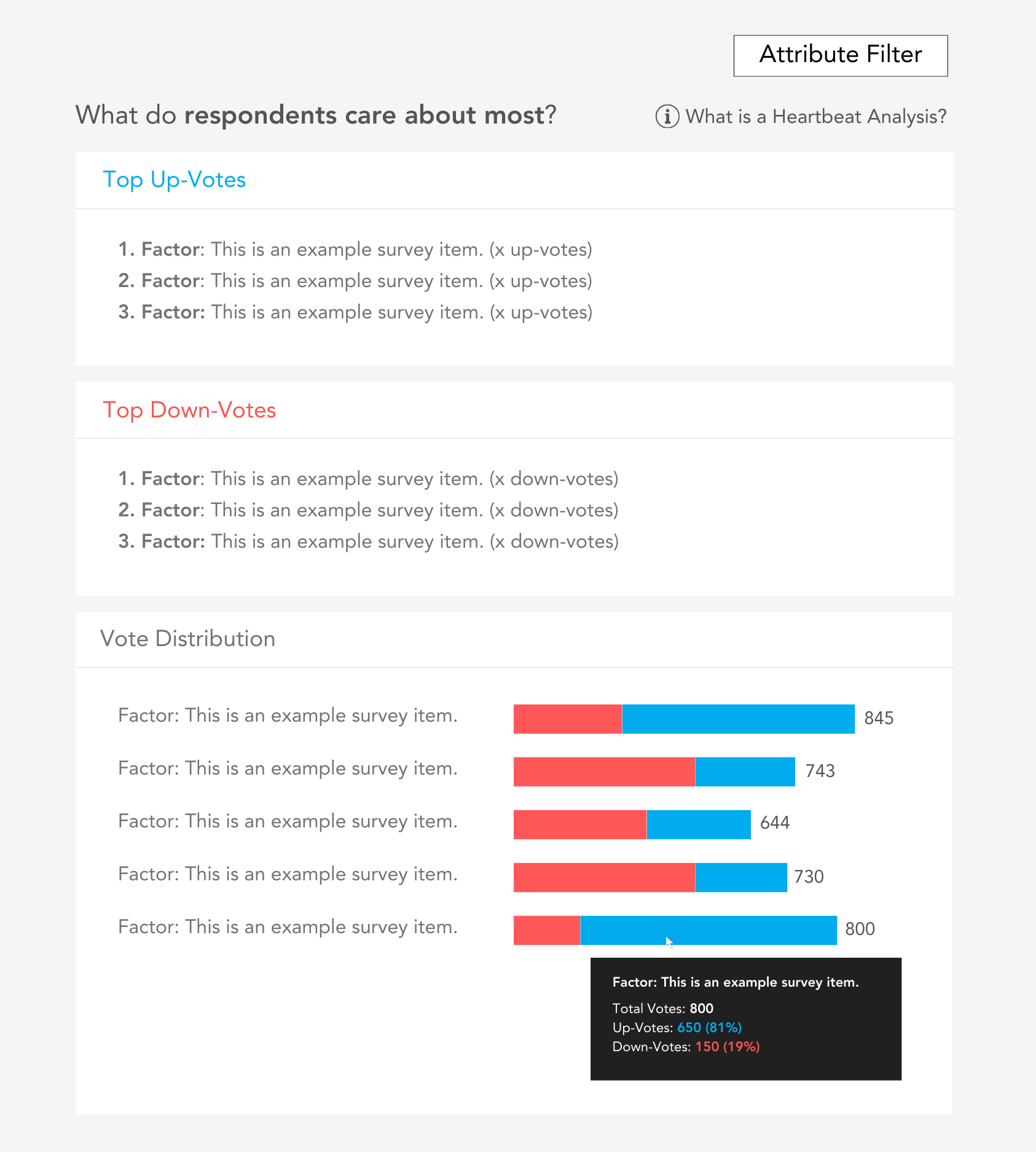
* Change the formatting of the following table columns (including what displays in the tooltip in the corresponding bubble chart):
  + The Driver field should be formatted as “Factor: Item” rather than “Factor –> Item”.
  + The Outcome field should only display the Factor value (e.g., “Engagement”) – not the Outcome Type value.



**Page: Analytics >> Heartbeat Analysis (NEW)**

This [SQL script](https://drive.google.com/file/d/1K-fb41xWHxDLtQhyN0QTMdC2Pul1IJMh/view?usp=drive_link) provides the logic for the three sections on this new page. This page should include the Attribute filter available on several of the other pages, which requires enforcement of the minimum n-count threshold within the app (def\_survey\_parameters.quantitative\_threshold).

The icon for the lefthand pane is indicated [here](https://docs.google.com/document/d/1OHVsTNeGxsF2_veoIU5adEBmYWOl1Ano04fySBbOOgU/edit?usp=drive_link).

****

**Page: Admin**

We need the ability to assign users to groups as well as assign both users and group members to roles (e.g., Admin, Super User, Manager).

There should be three pages under a new **User Management** section on the left nav, which can be accessed via a new **Admin** option along the top nav:

1. **Users**
2. **Groups**
3. **Roles**

[Here](https://docs.google.com/spreadsheets/d/1EKfNHjTY6-wCNDv1IZnaKw374jvPn_oS/edit?usp=sharing&ouid=111450598158587379038&rtpof=true&sd=true) are the recommended data structures to support this. In short, users are assigned to groups; users and group members are assigned roles; and roles receive access to surveys, pages, attributes, and records. The data sets and logic are defined below. *Lists* represent the population of values a user can select in the UI. *Assignments* represent the user-level, group-level, and role-level access entitlements.

**Lists**

* **user\_list:** Dynamic list of every person sent on worker files – whether currently active or not – by each employer over time. This list could change daily for an employer. [Here](https://drive.google.com/file/d/1iWp2QFXAyhB7a4c64WrIiilkO7sk9L77/view?usp=drive_link) is the logic for producing this list for each employer\_id. Additionally, users can be manually added and appended to the employer-provided list (e.g., a consulting firm supporting the employer who isn’t included in the employer’s active employee data).
* **group\_list:** Employer-created user groups for organizing users into logical groupings.
* **survey\_list:** This is the reference.ref\_survey\_parameters table in GBQ.
* **role\_list:** Default list of access roles (defined [here](https://docs.google.com/spreadsheets/d/1EKfNHjTY6-wCNDv1IZnaKw374jvPn_oS/edit?usp=sharing&ouid=111450598158587379038&rtpof=true&sd=true)). We should have the ability to add and assign new roles in the future.
* **page\_list:** List of dashboard pages (defined [here](https://drive.google.com/file/d/12dmR98e_cG_dcx25KrBv8G_fik9-00F3/view?usp=drive_link)). We will need to modify this list over time as our product evolves and pages are added, removed, or changed.
* **attribute\_list:** Dynamic list of every attribute relevant to each employer over time. This list could change daily for an employer. [Here](https://drive.google.com/file/d/1kgCONs7tTKB9U_PNHU7ePEUi_Duc2tky/view?usp=drive_link) is the logic for producing this list for each employer\_id. Since new custom attributes can be added from one employer data file to another, the list of attributes a user can be permissioned to view should always represent all attribute types and names across all snapshot dates.
* **results\_list:** There are only two options: “All” and “Team”

**Assignments**

* **user\_assignments**
  + By default, every person with at least one direct report should receive the Manager role. This could change daily for an employer. [Here](https://drive.google.com/file/d/1m5ArUY23_q2RW8EJ-c0G5w0UsvgJf_T3/view?usp=drive_link) is the logic for identifying every manager for a given employer.
  + Super User role assignments will be handled manually via the UI, as there is no logic to identify them; they are employer-defined.
* **group\_assignments**
  + No defaults. Users are manually assigned to groups, if/as needed.
* **survey\_assignments**
  + Defaults
    - The Admin role should receive access to every survey.
    - The Super User role should receive access to every survey.
    - The Manager role should receive access to every survey with an Experience type.
* **page\_assignments**
  + Defaults
    - The Admin role should receive access to every page.
    - The Super User role should receive access to every page except Admin.
    - The Manager role should receive access to view the following page\_ids: 1, 3, 4, 5, 6, 7, 8, 9, and 13
* **attribute\_assignments**
  + Defaults
    - Admin and Super User roles receive access to every attribute.
    - Manager role receives access to no attributes; this role should only explore results via the Manager hierarchy (for those within their reporting line).
* **result\_assignments**
  + Defaults
    - Admin and Super User roles should default to “All”.
    - The Manager role should default to “Team”.

**Note:** By default, [admin@orgacuity.com](mailto:admin@orgacuity.com) should have the Admin role for every employer’s dashboard.

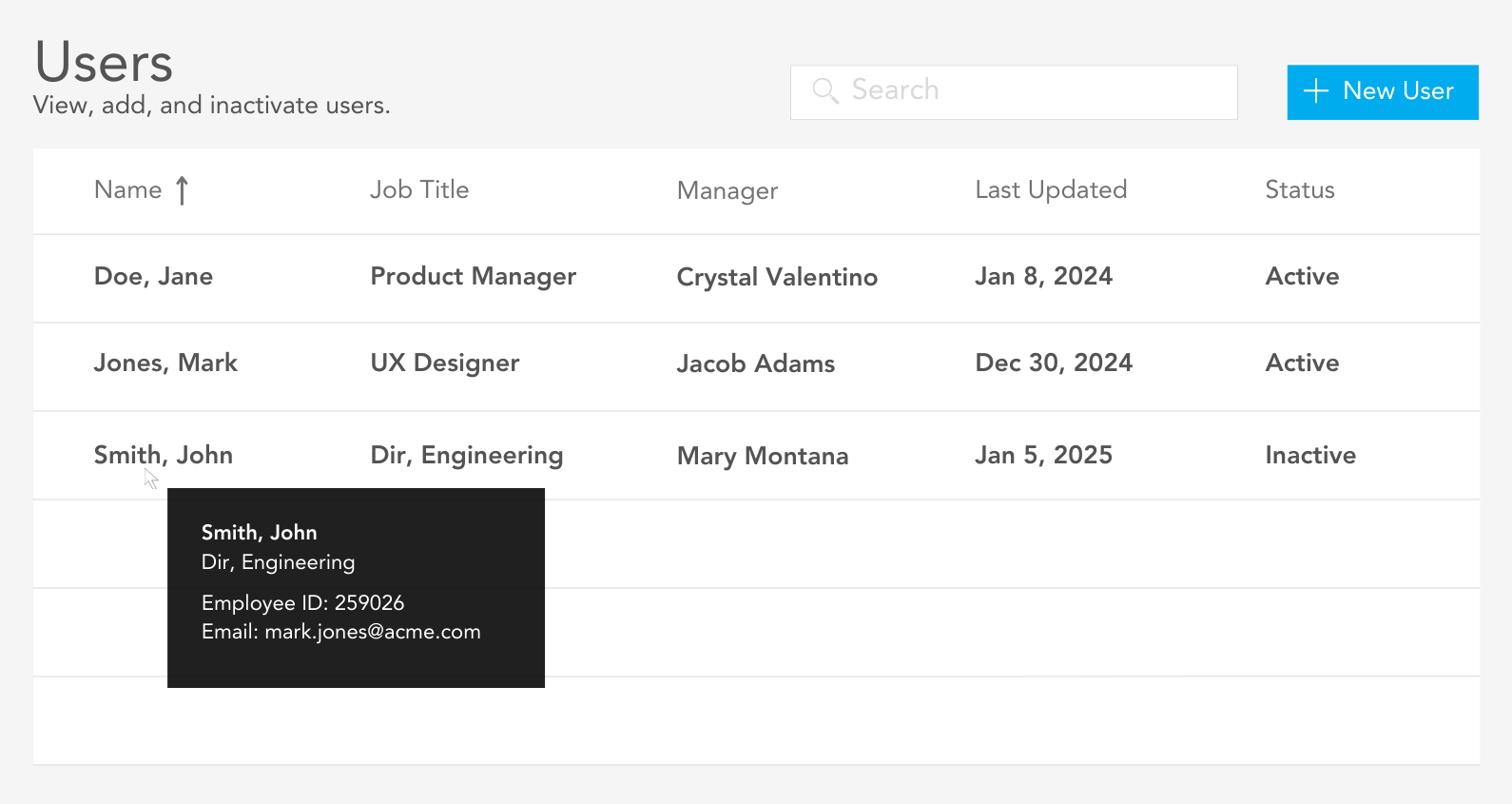
Mockups of the screens are provided below. Only people with the Admin role should see this Admin page in the dashboard.

**Users**

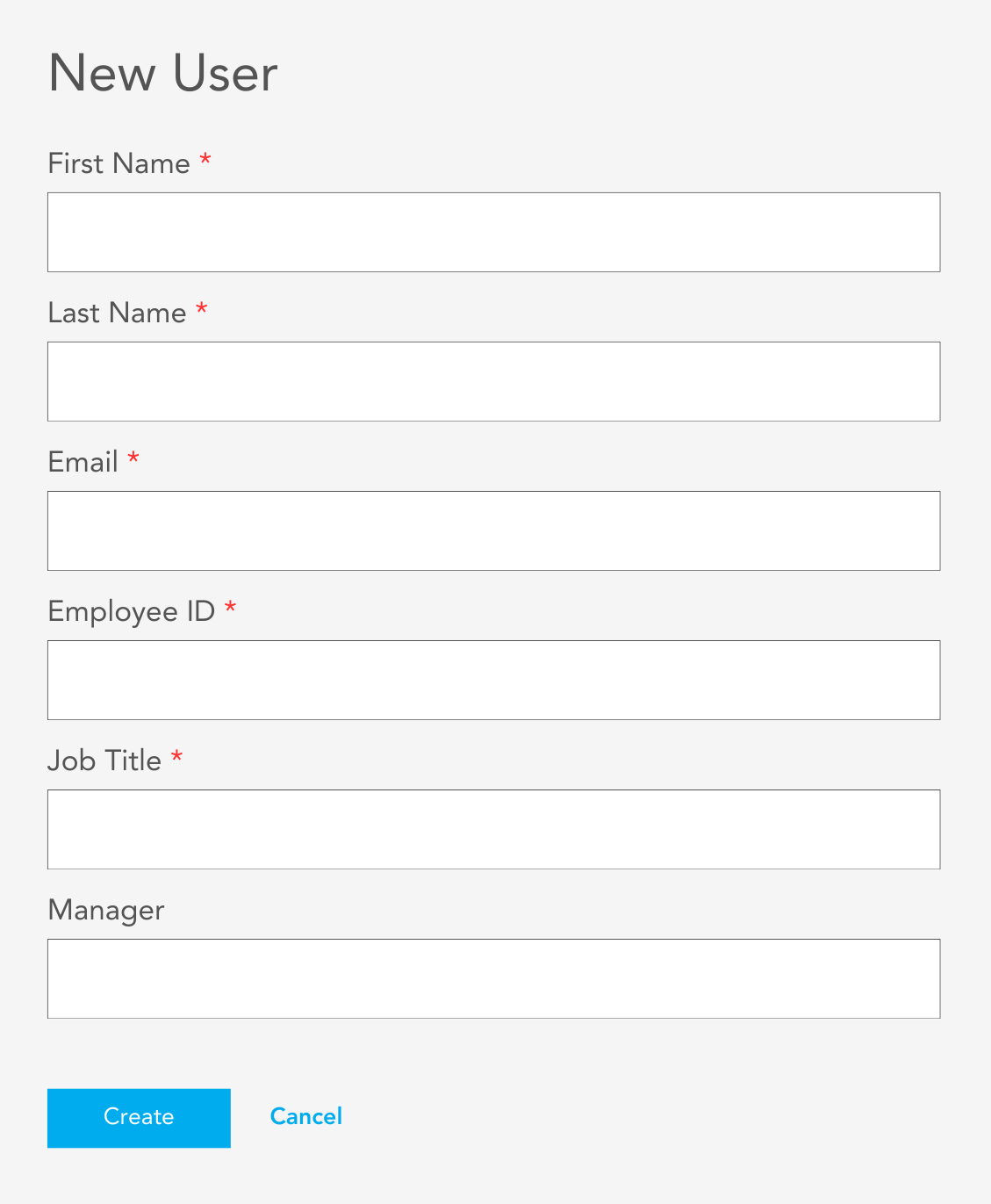
We need the ability to create new users apart from an employer’s uploaded worker data, as there will be additional support resources who need to service a particular account who are not members of the employer’s organization (e.g., an external consulting firm supporting the employer who isn’t included in the employer’s worker data).

The users list retrieved from GBQ (people.dim\_users) will always contain the full population of all people provided across an employer’s worker data uploads – whether currently active or inactive. We should use the WorkOS authorization API to automatically set users to inactive who are not authorized to access the app. Inactive users (a) should not show up as Member options on the Groups and Roles pages and (b) should be removed from any Groups and Roles they were previously assigned.

**Note:** This process should not inactivate manually added users, as support resources who are not members of the employer’s organization will never come through the authorization API. For this reason, we also need the ability to manually inactivate users independent of the automated authorization API comparison (e.g., an external consulting firm that is no longer servicing the account).

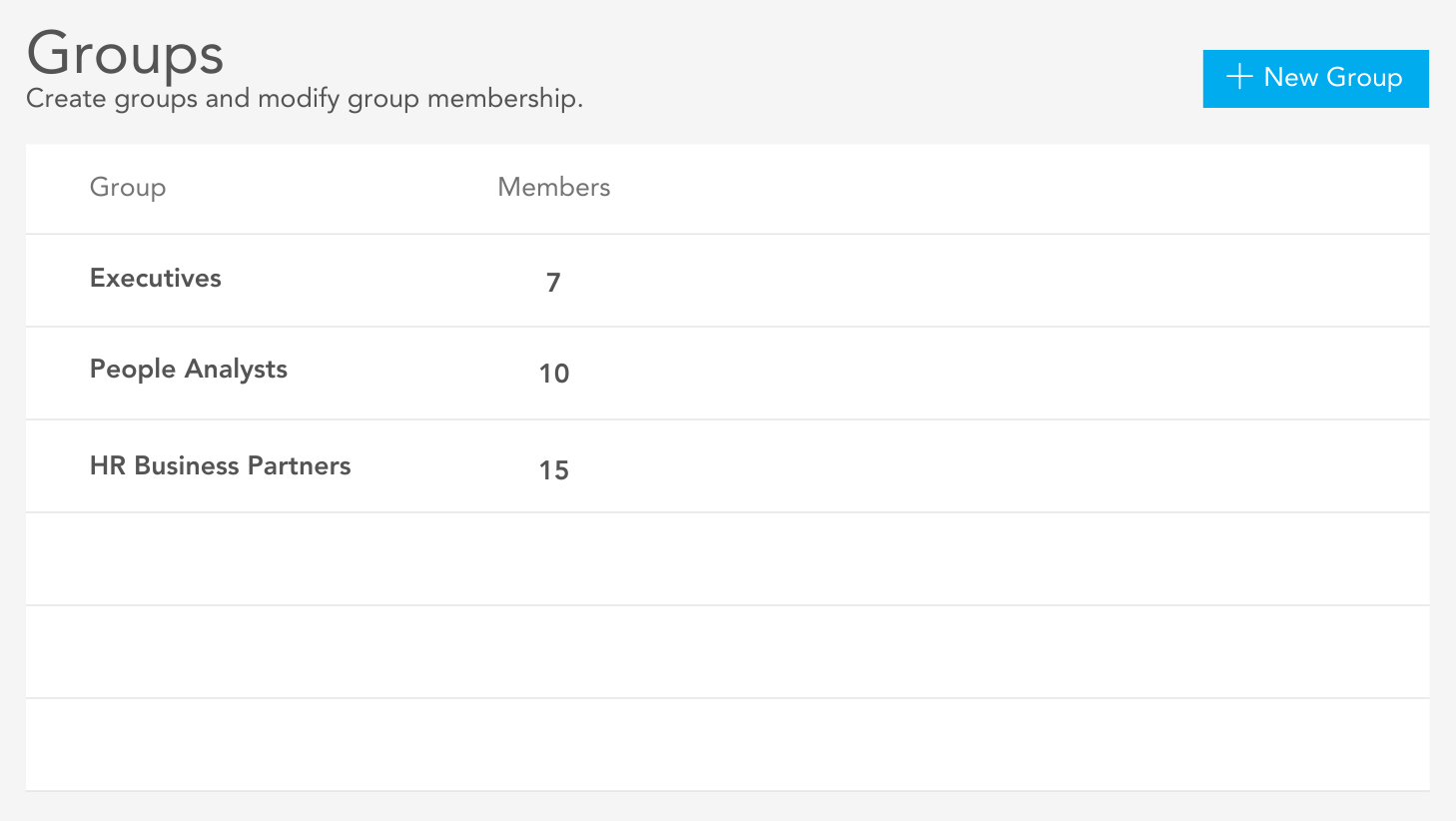
****

**Note:** It should not be possible to create a new user with an email or employee id that already exists.

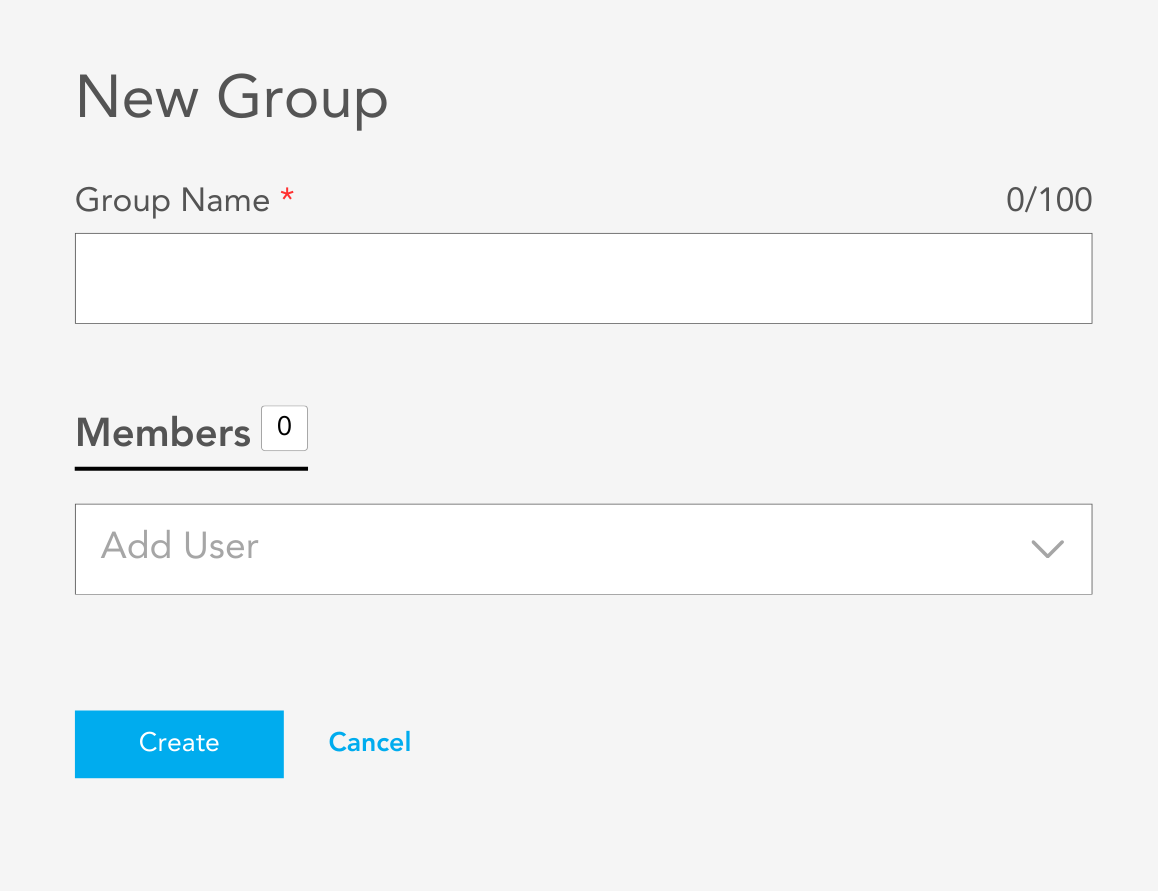
****

**Groups**

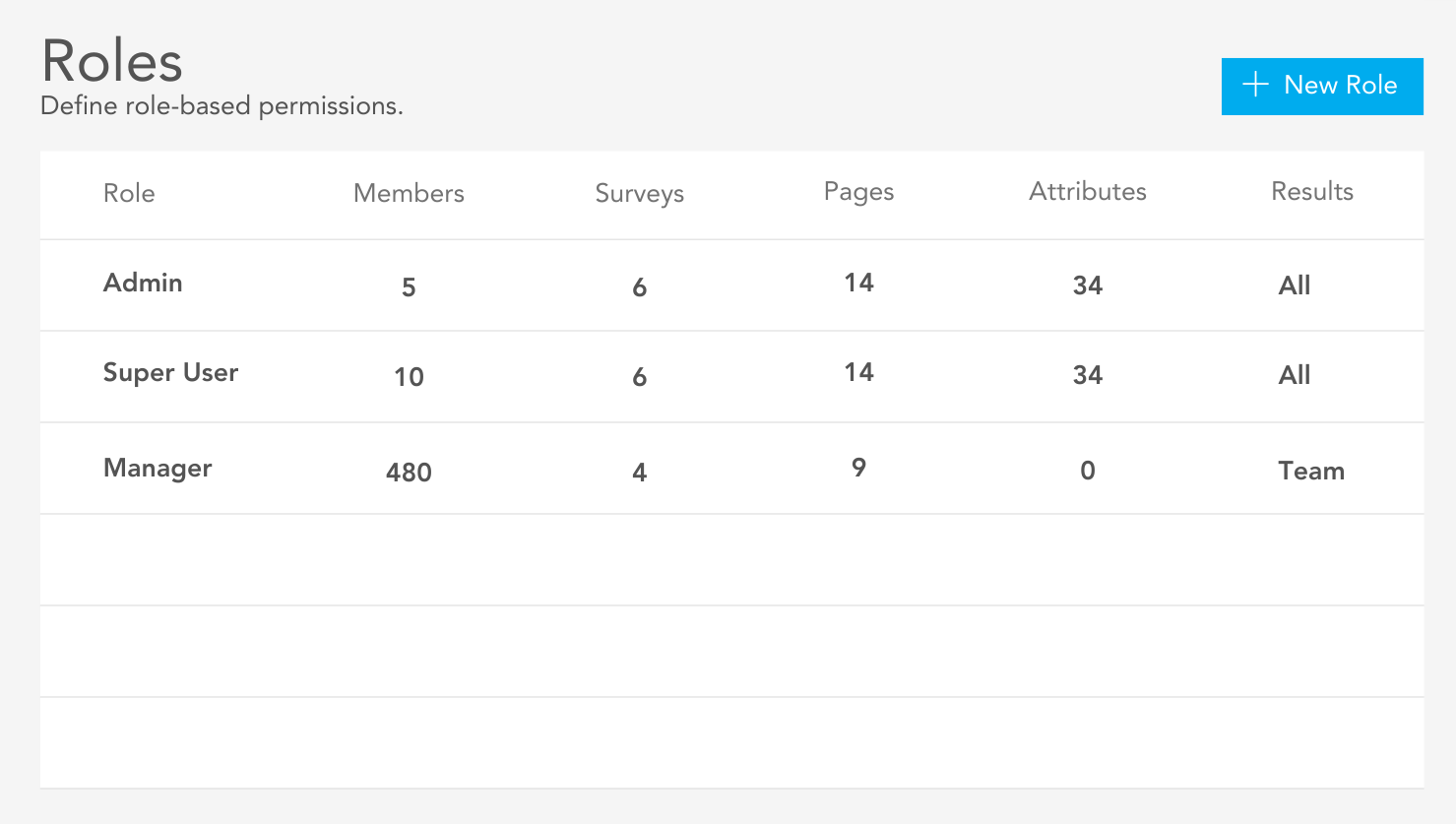
**Note:** In addition to creating new groups, users should be able to click an existing group on the Groups screen and select “View Group” or “Edit Group” to view/edit group membership.

****

**Note:** It should not be possible to create a new group with a group name that already exists.

****

**Roles**

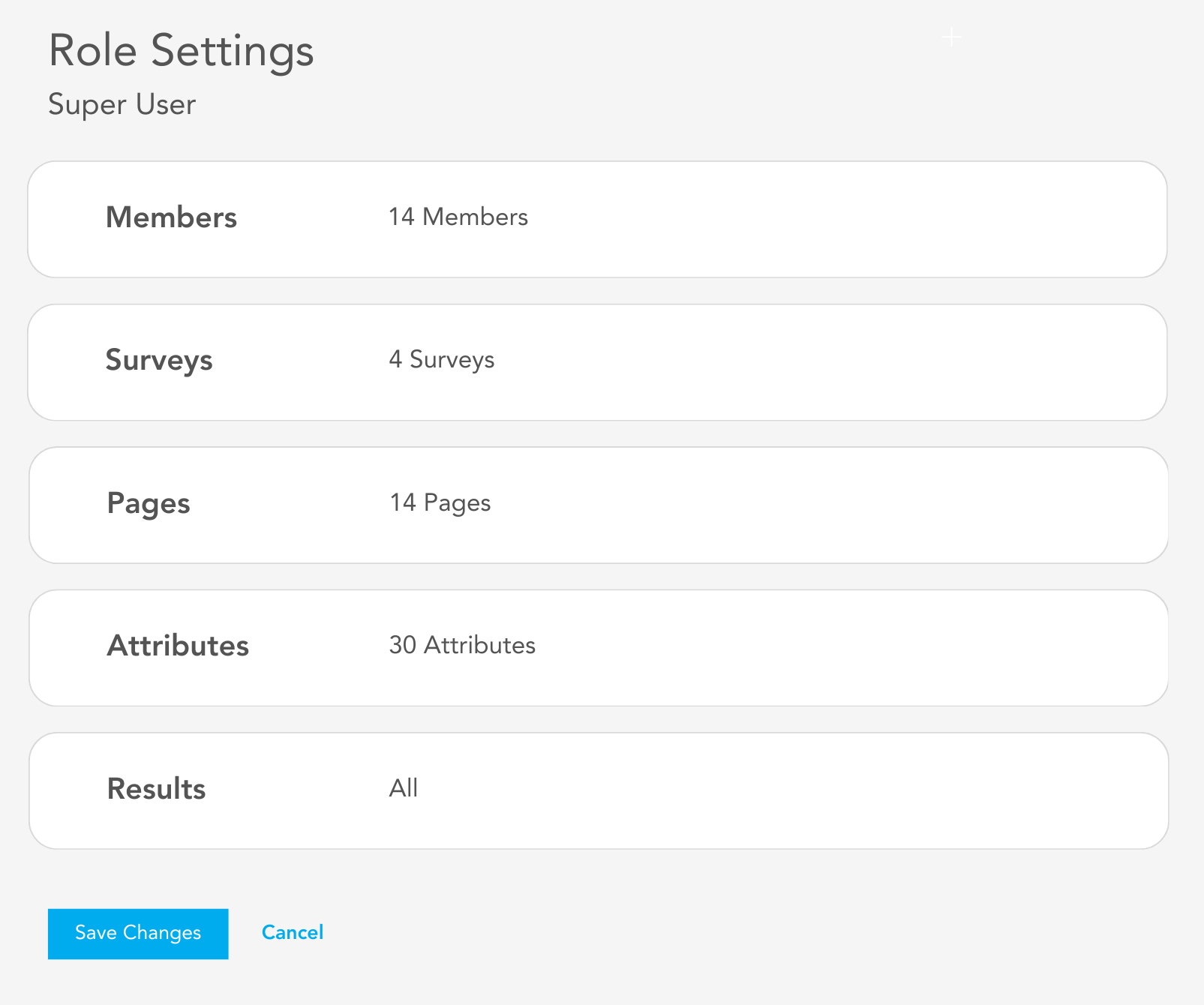


**“Edit Role” Window**

**Note:** In addition to creating new roles, users should be able to click a role on the Roles screen and select “View Role” or “Edit Role” to view/edit roles. The “Edit Role” mock up provided below can be adjusted for the “New Role” and “View Role” use cases.

There should be a checkbox for each of the available options when expanding each section under Role Settings. These sections should contain a search field, as there may be many options (e.g., Members may contain thousands of people) and it should be easy for users to find a specific individual without significant scrolling. The collapsed view should provide summary information on what is selected.

When expanding the *Members* section, users should be able to select individual users as well as groups. When groups are selected, the group members should be reflected in this overall membership count. As group membership changes, roles and permissions should be updated dynamically for the respective members.

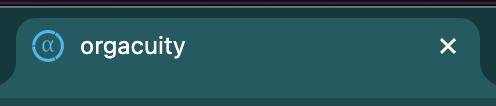


**Other**

* There are several pages in which data dimensions are concatenated. We need to break these concatenated values into separate filters.
  + **Page: Influencers >> Networks**
    - Factor and Item should be separate filters. The Factor filter should be populated with all possible values regardless of which Item value is selected, while the Item drop-down should only populate with the values relevant to the selected Factor value.
    - Filter names displayed to the user should be: “Choose Factor” and “Choose Item”.
  + **Page: Analytics >> Heatmap**
    - Factor and Item should be separate filters. The Factor filter should be populated with all possible values regardless of which Item value is selected, while the Item drop-down should only populate with the values relevant to the selected Factor value.
    - Attribute Type and Attribute Name should be separate filters. The Attribute Type filter should populate with all possible values regardless of which Attribute Name value is selected, while the Attribute Name drop-down should only populate with the values relevant to the selected Attribute Type value.
    - Filter names displayed to the user should be: “Choose Attribute Type” and “Choose Attribute Name”.
  + **Page: Analytics >> Drivers**
    - Outcome Type, Outcome Factor, and Outcome Item should be separate filters. Outcome Type should populate with all possible values regardless of the selected Outcome Factor or Outcome Item values. Outcome Factor values should populate based on the selected Outcome Type value(s), and Outcome Item values should display based on the selected Outcome Factor values(s).
    - Filter names displayed to the user should be: “Choose Outcome Type”, “Choose Factor”, and “Choose Item”.
* There should be two icons in the upper right that look like those provided below. The lighter grey color assigned to the existing icons in NiceGUI should be applied to these so that they do not detract from the data on which users should be focused.



* When the user clicks the question mark icon, there should be two options:
  + **Product Feedback** (link: <https://www.orgacuity.com/product-feedback>)
  + **Technical Support** (link: <https://www.orgacuity.com/technical-support>)
* The title that displays on browser tabs should be mixed case: “OrgAcuity”. The favicon (svg) is [here](https://drive.google.com/file/d/10Lq87-YIc-cSi-JqRi527oJTuEbWFz7U/view?usp=drive_link).



* Every data visualization should be exportable (e.g., .png file).
* In our NiceGUI app, when exporting tables to Excel only the subset of data stored in memory (user’s current page) exports. We need to export the *entire* underlying data set for the user’s filter selections.

## **Add Filter** functionality per page

#### Home

* Summary: only global manager
* Alerts: no filters
* Action: only global manager

#### Participation

* Trends: only global manager
* Representation: only global manager
* Teams: only global manager

#### Questions

* Ratings:
  + global manger
  + attribute type
  + attribute name
  + attribute value
* Multiple Choice:
  + global manger
  + attribute type
  + attribute name
  + attribute value
* Comments: only global manager

#### Influencers

* Influencers: only global manager
* Networks: only global manager
* Leaders: only global manager

#### Analytics

* Heatmap:
  + global manger
  + attribute type
  + attribute name
  + attribute value
* Drivers: only global manager
* Heartbeat analysis:
  + global manger
  + attribute type
  + attribute name
  + attribute value