Time spent writing this document: 30 minutes
Process:

For this task I had experimented with 3 possible approaches based on the requirements. Here they are:

1. Full Contrib module
2. Completely Custom Entities
3. Custom Content Type and Custom Entity

Full Contrib Module (Time spent: 1.20 hours)
Even though this approach was the one that I invested most of the time, I most admit that it was a failed attempt. The idea was to leverage the contrib webform module. Now, this contrib module met most of our acceptance criteria, like:

- generating forms
- permitting user with correct access to access and modify form
- gather results and statistics of form submission
- ability to send email when form is submitted
- graphql has knowledge of the webform module, which webforms and submissions can be accessed by query

Yet with all the positives features given by this contrib module, it had a few flaws which made me not use it, this are:

- to complex for non technical people, which I believe this is an important point for good editorial experience.
- No webform bundle, which means that if a new survey form is created and wants to be exposed
- No field for weighted items.
- No field weight aggregation.

This cons in my opinion, outweigh the pros. For this reasons, I didn't continue working with the webform contrib module.

Completely Custom Entities (Time spent: 30 minutes)
The second approach taken for tackling the task was creating two custom entity, one called forms and the other one called form submissions, and a custom field type that would be a composition of a question (string), weight for yes (int) and weight for no (int). None of the custom entities were going to have bundle, just simple entities that were interconnected by an entity reference. This way everything was custom, we control everything, yet it is much more work, but a good approach. What kept me away from this approach is the fact that one of the requirements is the usability and simplicity for an editor, in that moment I decided to take the next approach.

Custom Content Type and Custom Entity (Time spent: 1 hour)
In this approach the idea is simple, have a custom Content Type called survey, this custom content type will have a one to many mapping to a custom field type called Polar Weight Question, is a simple field that consist of a textbox, and two range, were each range represents the weigh of the yes or no option. Then we will have a custom entity called FormSubmissionEntity. This entity will be in charge of saving some results, for this case, it will be the Survey, also it will contain an entity reference back to the survey, this entity reference will help us query and filter by form. This approach I believe is the correct one.

General Idea:

The final general idea: have a custom content type called survey. This custom entity type will be pulled and managed by the react application. In other words, the react application is going to query and mutate the results data.