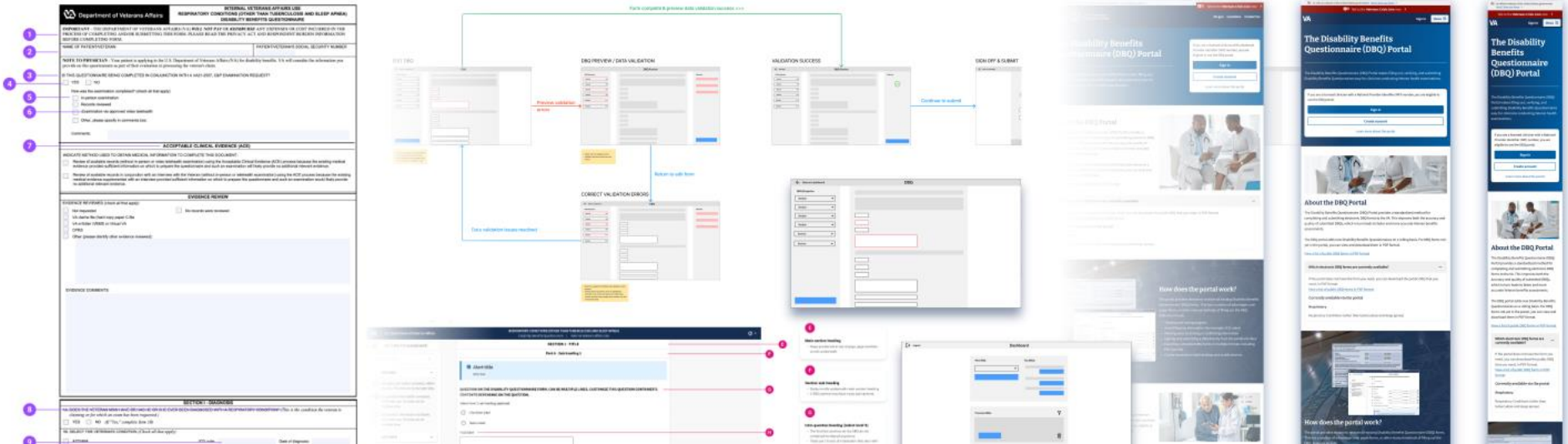
Nick Osmanski
Senior UX Designer

Case study 1

The VA DBQ Portal



The DBQ Portal | *Overview*

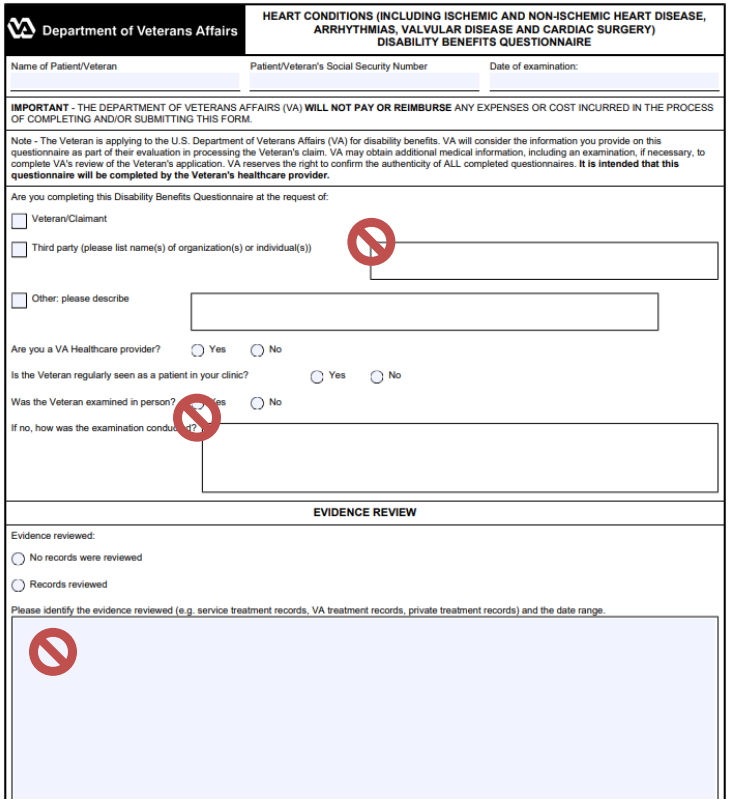
Background

- Clinicians send Disability Benefits Questionnaire (DBQ) forms to the VA
- DBQs are paper forms, electronic PDFs, or some database format (ex: csv, xml)

The problem

- VA spends a lot of time correcting errors, and confirming information accuracy
- This creates more work for Rating Veterans Service Representatives (RVSRs), and prevents Veterans from receiving their benefits in a timely manner

A DBQ form



Department of Veterans Affairs

HEART CONDITIONS (INCLUDING ISCHEMIC AND NON-ISCHEMIC HEART DISEASE, ARRHYTHMIAS, VALVULAR DISEASE AND CARDIAC SURGERY)
DISABILITY BENEFITS QUESTIONNAIRE

Name of Patient/Veteran | Patient/Veteran's Social Security Number | Date of examination:

IMPORTANT - THE DEPARTMENT OF VETERANS AFFAIRS (VA) **WILL NOT PAY OR REIMBURSE** ANY EXPENSES OR COST INCURRED IN THE PROCESS OF COMPLETING AND/OR SUBMITTING THIS FORM.

Note - The Veteran is applying to the U.S. Department of Veterans Affairs (VA) for disability benefits. VA will consider the information you provide on this questionnaire as part of their evaluation in processing the Veteran's claim. VA may obtain additional medical information, including an examination, if necessary, to complete VA's review of the Veteran's application. VA reserves the right to confirm the authenticity of ALL completed questionnaires. **It is intended that this questionnaire will be completed by the Veteran's healthcare provider.**

Are you completing this Disability Benefits Questionnaire at the request of:

- ☐ Veteran/Claimant
- ☐ Third party (please list name(s) of organization(s) or individual(s))
- ☐ Other: please describe

Are you a VA Healthcare provider? ◯ Yes ◯ No

Is the Veteran regularly seen as a patient in your clinic? ◯ Yes ◯ No

Was the Veteran examined in person? ◯ Yes ◯ No

If no, how was the examination conducted?

EVIDENCE REVIEW

Evidence reviewed:

- ◯ No records were reviewed
- ◯ Records reviewed

Please identify the evidence reviewed (e.g. service treatment records, VA treatment records, private treatment records) and the date range.

The DBQ Portal | *Overview*

My role

- The sole designer assigned to this project team, supporting:
  - Front-end developers
  - Back-end developers
  - VA PO and SMEs
  - Internal PMs

- Final deliverables included:
  - Developer-ready design system components
  - Hi-fi mockups and prototypes
  - Demos and presentations to PMs and POs
  - (5-6 months project time)

Solution & impact

- An electronic portal where users could fill out and send DBQ forms to the VA, incorporating standardized formatting and form data validation

- Consistent positive reactions from VA stakeholders and SMEs
  - 10 DBQ forms designed
  - Planned expansion of scope before project termination

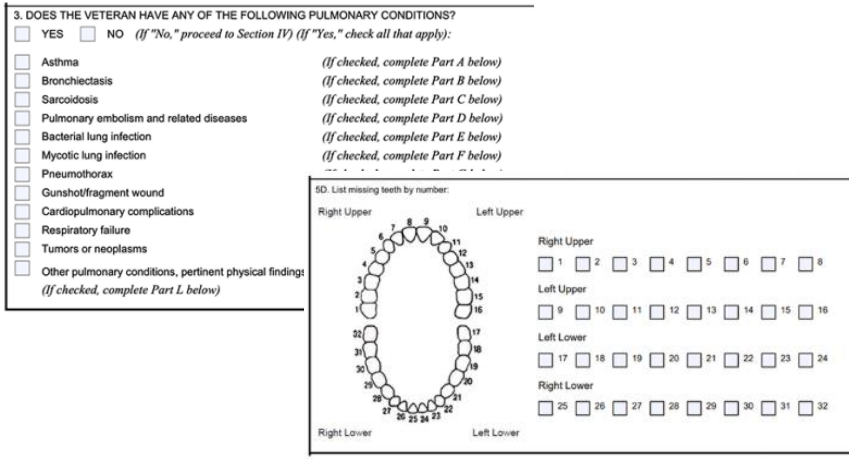
The DBQ Portal | *Discovery research*

Interviews

- VA stakeholders: POs and PMs
  - VA Product Owner feedback about the state of the DBQ form process
  - VA Project Managers provided insight to the goals and needs of the project
- VA SMEs: Clinicians and RVSRs
  - I interviewed retired and practicing VA clinicians
  - I spoke to practicing VA Rating Veterans Service Representatives (RVSRs)

DBQ form UX evaluation

- Form improvements
  - Standard sections for non-sequential navigation
  - Collapse conditional elements
  - Simplify Diagrams



The DBQ Portal | *Defining personas*

The clinician

Both private (unlicensed by the VA) and licensed clinicians see multiple patients per day, and may not complete a DBQ in one setting.

Needs

- To stop and resume a DBQ form
- To be notified of conflicting form entries and missing information

Pain points

- Ambiguous wording or unclear DBQ instructions
- Long and unnecessary sections of the DBQ form

The VA RVSR

The RVSR handles multiple DBQ forms for each Veteran. Rating is time consuming, so each error has a compounding effect.

Needs

- Complete DBQ forms
- Data consistency and accuracy (no conflicts)

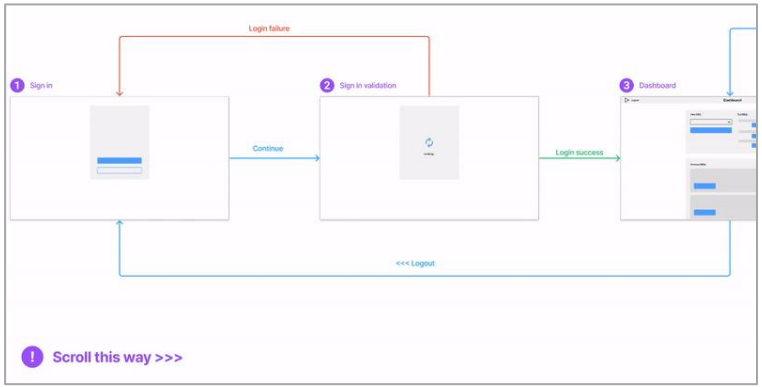
Pain points

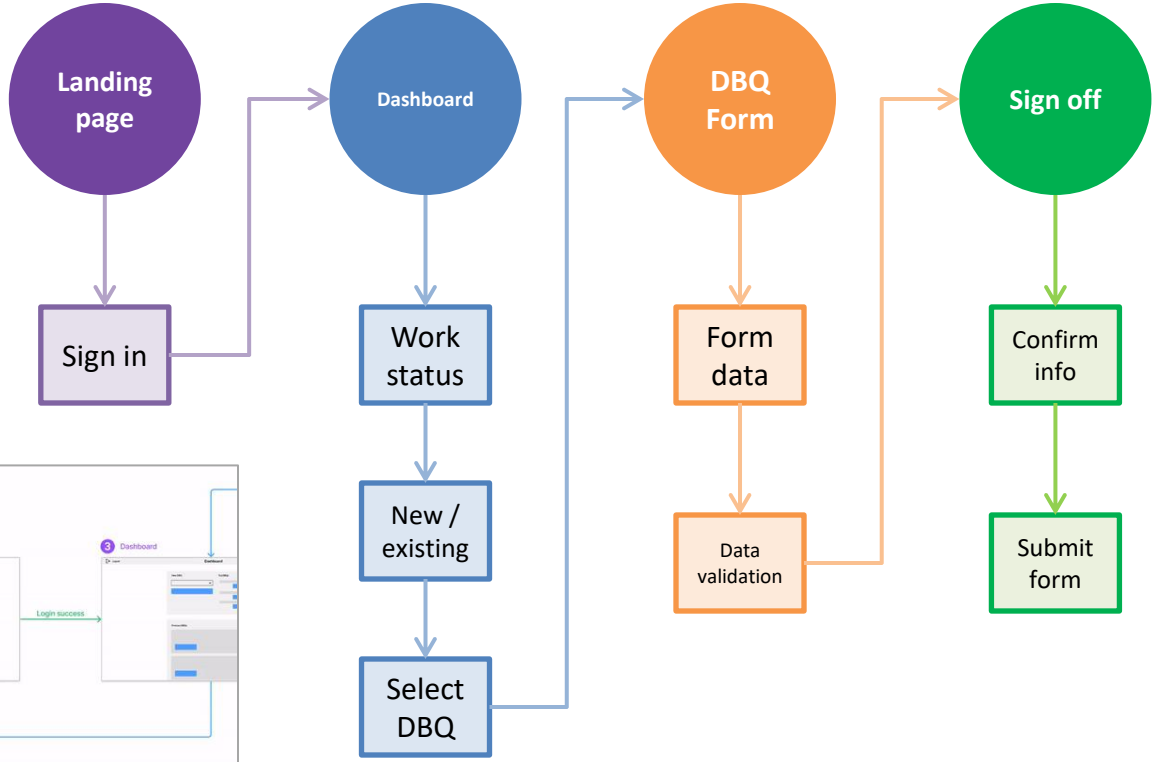
- Incomplete or conflicting form data
- Illegible handwritten form sections
- Manually searching through DBQ forms

The DBQ Portal | *User flow*

Key features

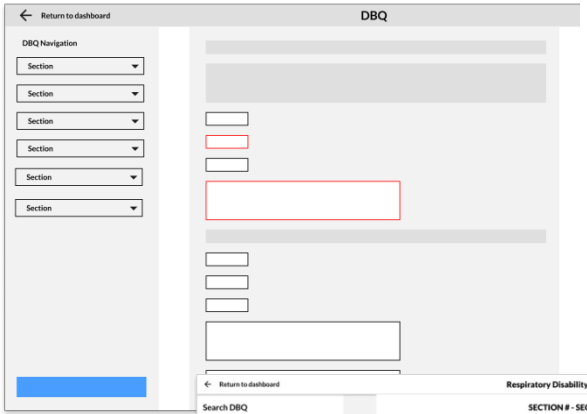
- From VA PM and POs:
  - Landing page
  - A dashboard
  - The DBQ form
  - Validating form data
  - Sign and submit



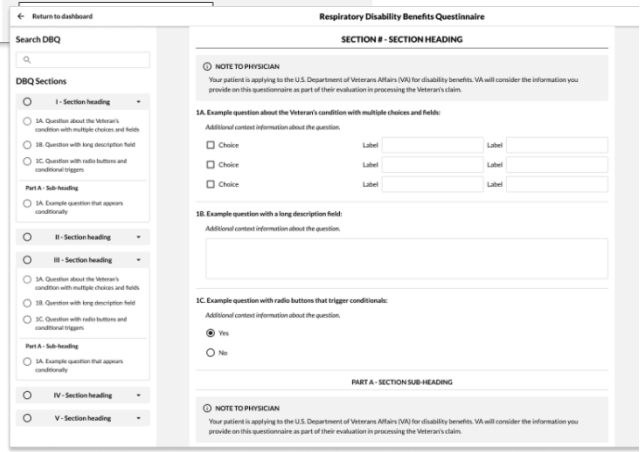


The DBQ Portal | *Design iteration*

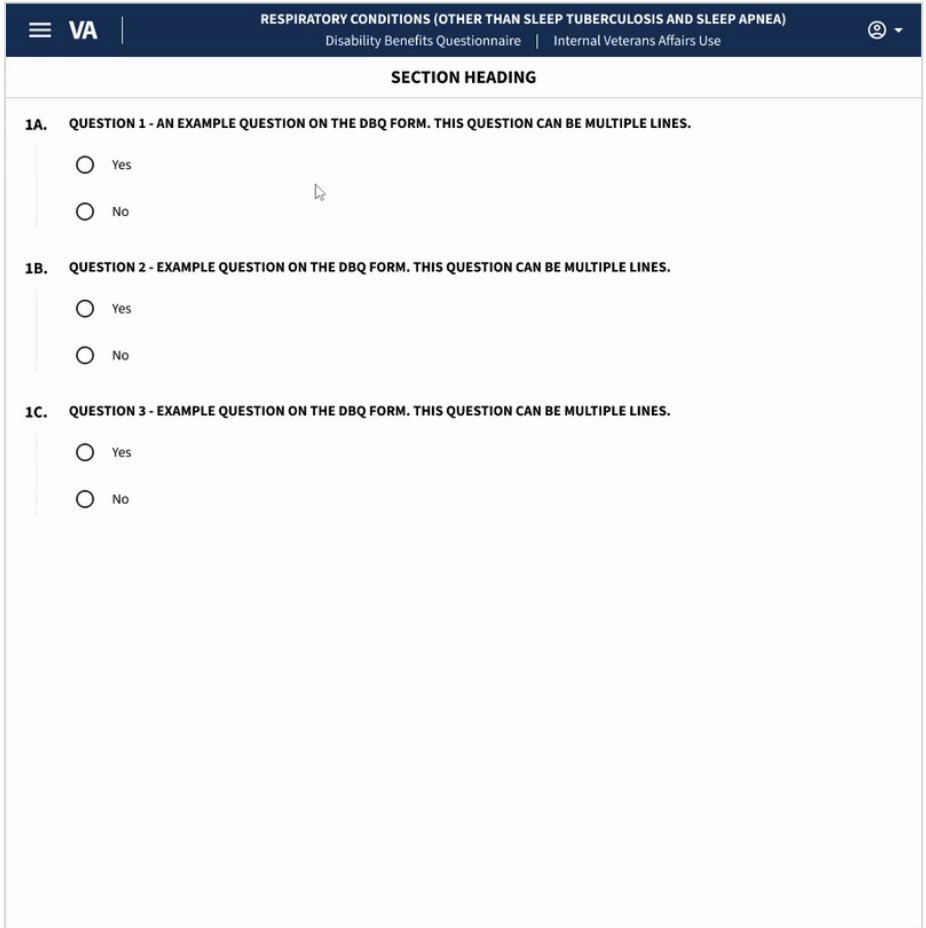
Lo-fi form



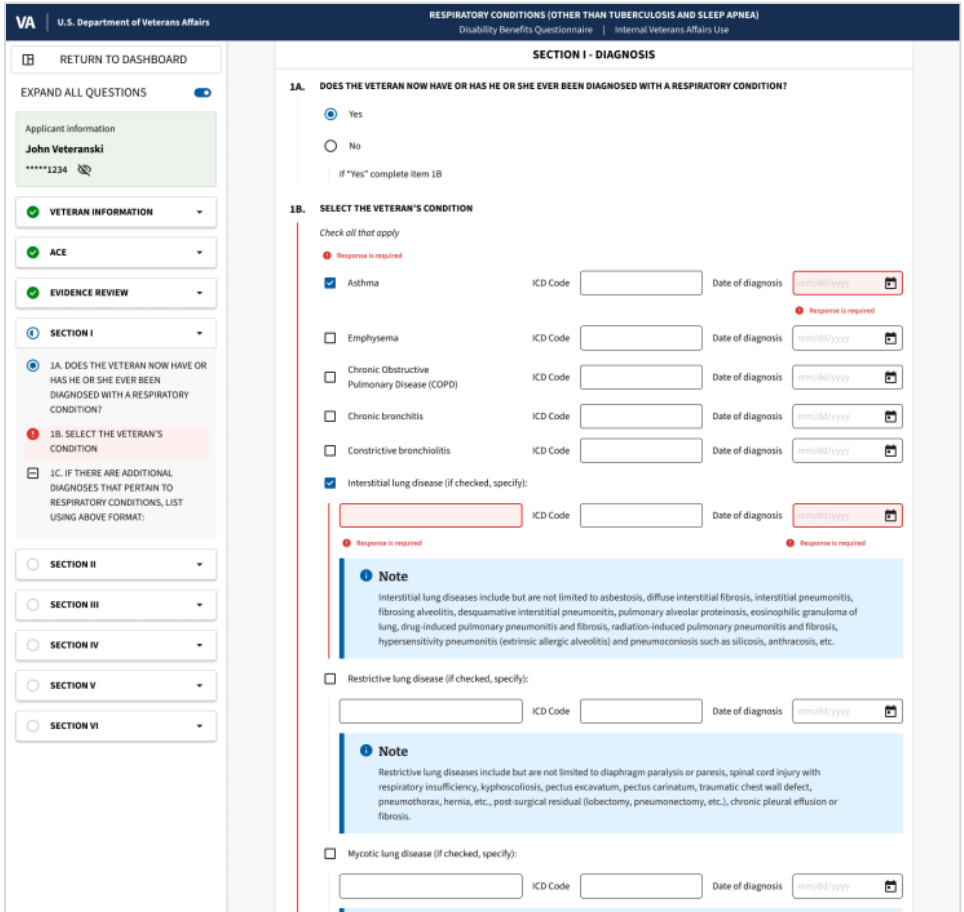
Mid-fi form



Testing prototype



The DBQ Portal | *Usability testing*



How & who

- Remote usability testing using Figma prototypes and a UAT live site
- Participants were given a general task: complete and send in a DBQ
- 3 clinicians, 1 VA PM, 1 RVSR

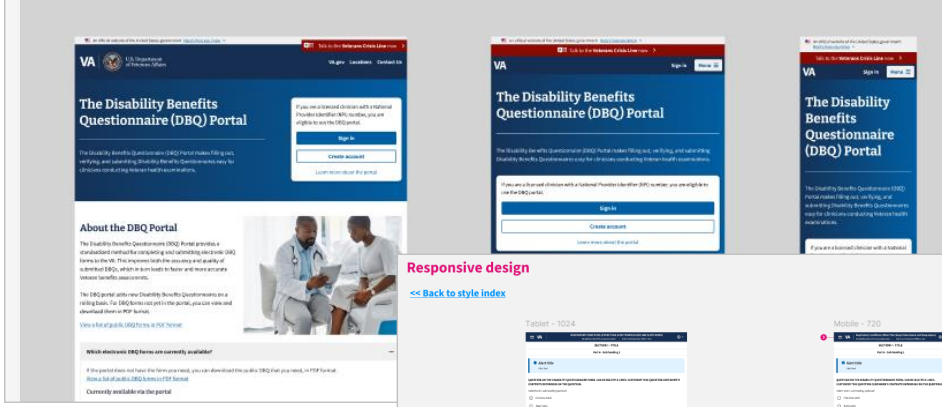
Results

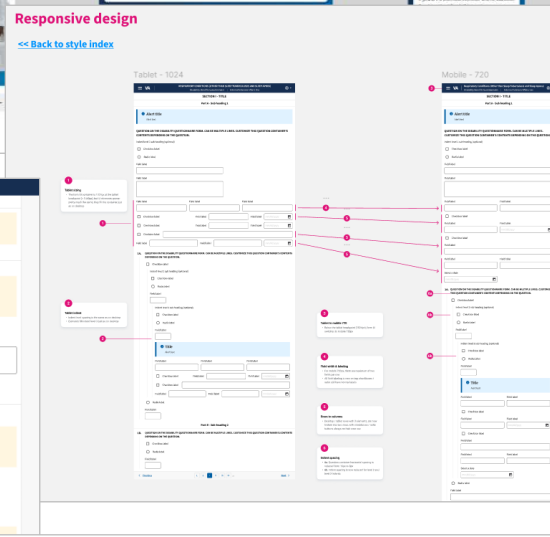
- New features included:
  - Side panel status icons
  - Veteran info context
  - Hide/show conditionals
- General positive feedback
  - More efficient
  - Participants preferred this over their current workflow

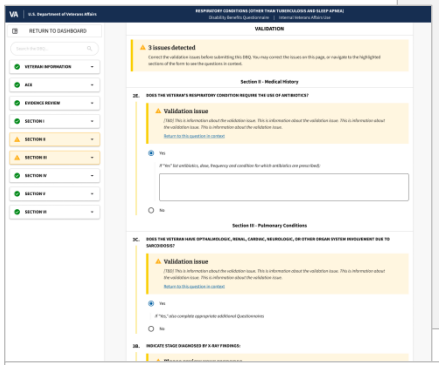
The DBQ Portal | *Final deliverables*

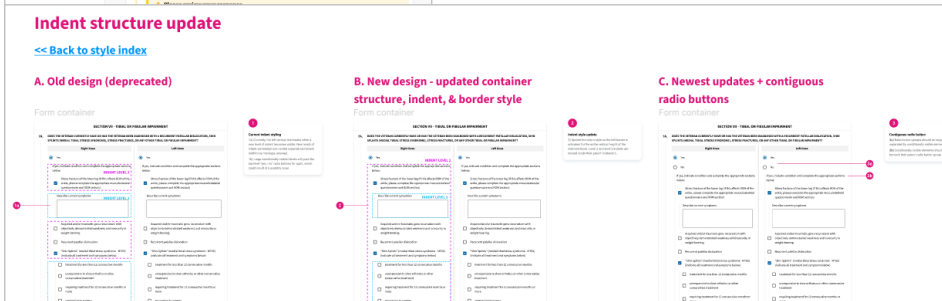
Deliverables

- Design system components
  - 508 compliant colors, grids, typography, form elements etc.
- Mockups & prototypes
  - Annotated for developers
  - Desktop / mobile breakpoints
  - 10 DBQs in total
- Demos
  - End-of-sprint presentations to PMs and POs
- Front-end QA
  - Local server in Docker
  - Notes & feedback for devs









Mini retro

- Lessons learned
  - More experience communicating design to devs (front-end QA review)
  - Making design system readable / navigable

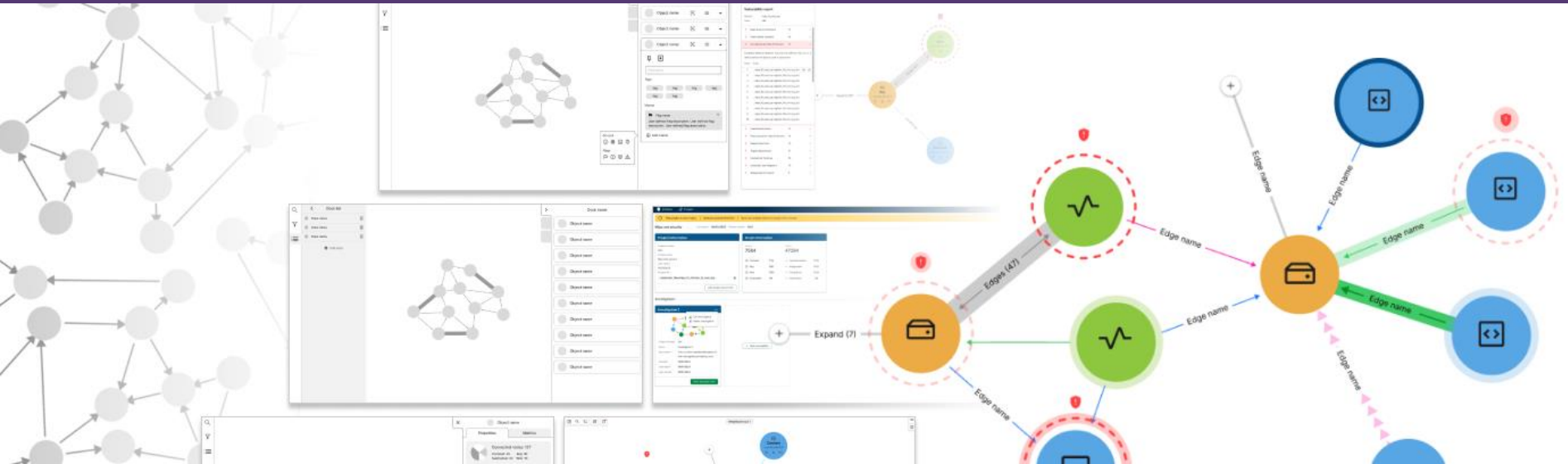
- Do anything differently?
  - More usability testing with clinicians
  - No time for read data validation testing in UAT environment

Next up…

Nick Osmanski
Senior UX Designer

Case study 2

Data visualization application



Data viz application | *Overview*

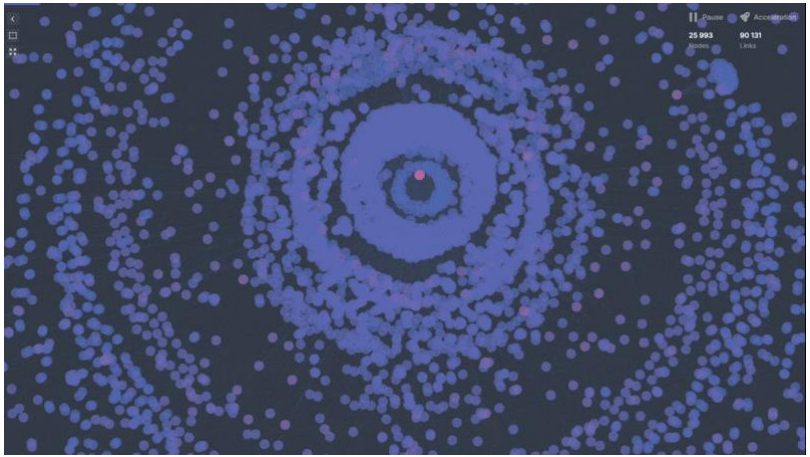
Background

- Amida's management and engineering team wanted to leverage their expertise to fill a gap in the cyber security market
- They resolved to use an existing tool that incorporated graph theory and data visualization

The problem

- The closest existing tool did not have the depth of interactivity needed to fulfill the intended use case
- Amida could not demonstrate their ability to fill the gap in the market without this functionality

Node and edge graph (Cosmograph)



(25,000 nodes; 90,000 edges)

Data viz application | *Overview*

My role

- The sole designer assigned to this project team, supporting:
  - Front-end developers
  - Back-end developers
  - Internal SMEs
  - Internal PMs and upper management

- Final deliverables included:
  - Developer-ready design system components
  - Hi-fi mockups and prototypes
  - Demos and presentations
  - (5-6 month project time)

Solution & impact

- A custom tool built in-house, incorporating graph theory in its data visualization, progressive levels interaction, and demos and marketing materials designed to pitch the solution to potential customers

- Positive reactions from internal management, SMEs, and potential customers
  - The prototype solution was pitched to a number of leaders in the cyber security field, to positive feedback, and new business leads for Amida

Data viz application | *Discovery research*

Interviews

- Team SMEs:
  - Front-end engineers
  - Back-end engineers
  - Graph theory & computer science experts

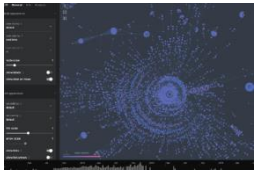
- I spoke with the team as a group, and each member individually
  - I learned the basics of node and edge graph visualization
  - They explained current solution limitations
  - Suggested features

Competitive analysis

Neo4j

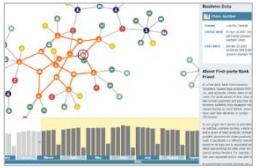
- \+ Polished UI and useful features
- − Limited depth of detail on nodes and edges, in terms of visual representation of the data

Cosmograph



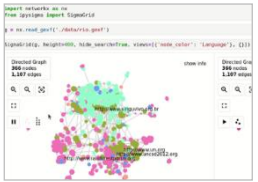
- \+ Large data set representation, real time graph movement
- − Relatively simple UI features, and limited node/edge visual detail

yFiles library



- \+ Many different options for displaying data as node-and-edge graphs
- − Limited detail and interaction for individual nodes and edges

ipySigma library



- \+ High level of customizability via library API access
- − Once again, limited out-of-the-box node and edge interaction detail

Data viz application | *Defining our persona*

John Persona, 42
Cybersecurity engineer

John is an educated cybersecurity engineer and analyst. He leads a team of supporting analysts, and he reports directly to leadership about critical business decisions.

> "I need to access low and high levels of information density, while retaining context, in order to see the bigger picture."

Goals

- John wants to make informed decisions about technical issues quickly and accurately.
- John wants to provide critical updates to leadership, who may include non-technical people.
- John wants to communicate issues to the customer, simply and effectively.

Needs

- John needs to navigate the graph quickly and intuitively, while retaining data context.
- As an SME, John needs specifically requested required features to be implemented.
- John needs the application to recommend solutions for given cybersecurity scenarios.

Frustrations

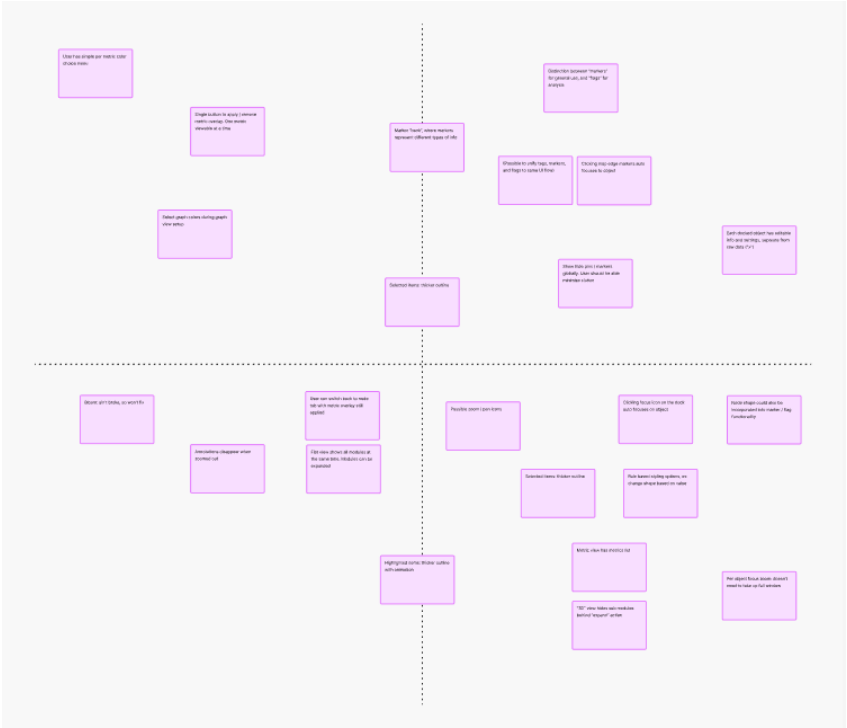
- John doesn't like the simplicity of the existing data visualization solutions.
- John doesn't like having to manually search through data that is difficult to parse.
- John doesn't want to waste time recreating the same analysis scenarios for different data sets.

Data viz application | *Scope creep*

MVP feature consolidation

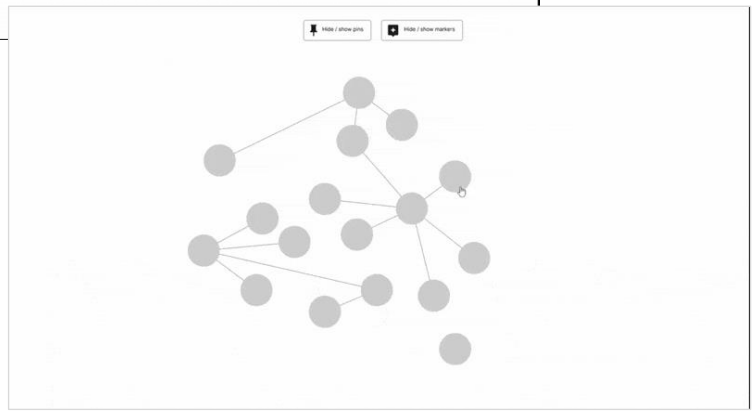
- The team consisted SMEs, and each had a list of “required” features that would suit their workflows
- I used an importance/difficulty matrix to consolidate MVP features, including:
  - A dashboard
    - Project setup
    - Multiple graph “views”
    - Analysis heuristics
  - The graph
    - Graph element interaction
    - List of elements

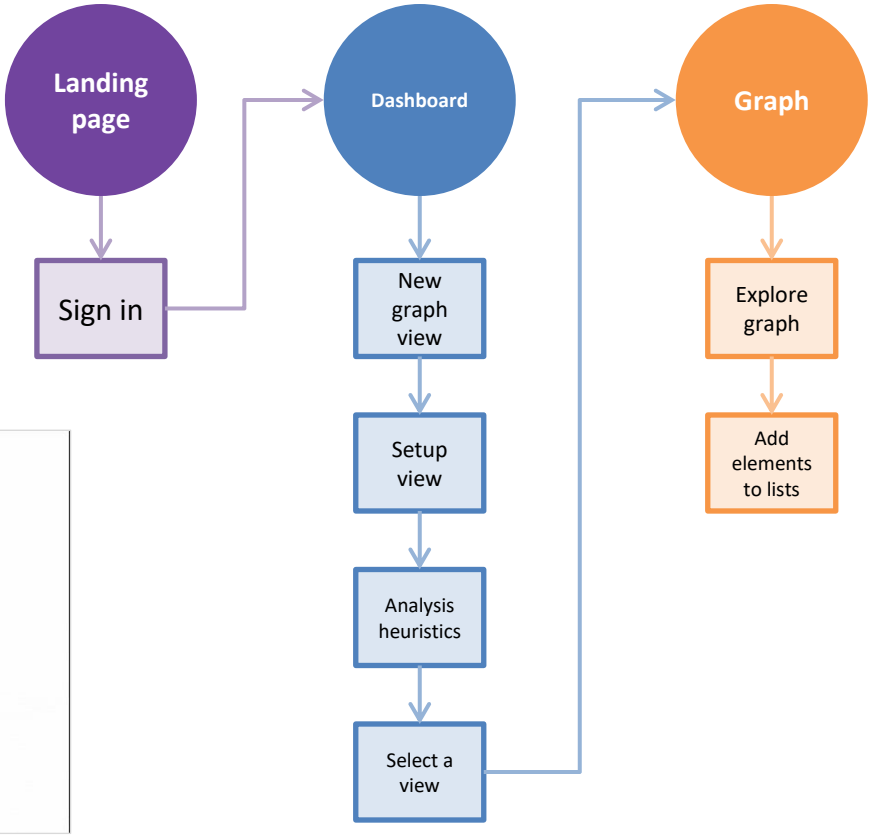
Importance (X axis) / difficulty (Y axis) matrix



Data viz application | *User flow*

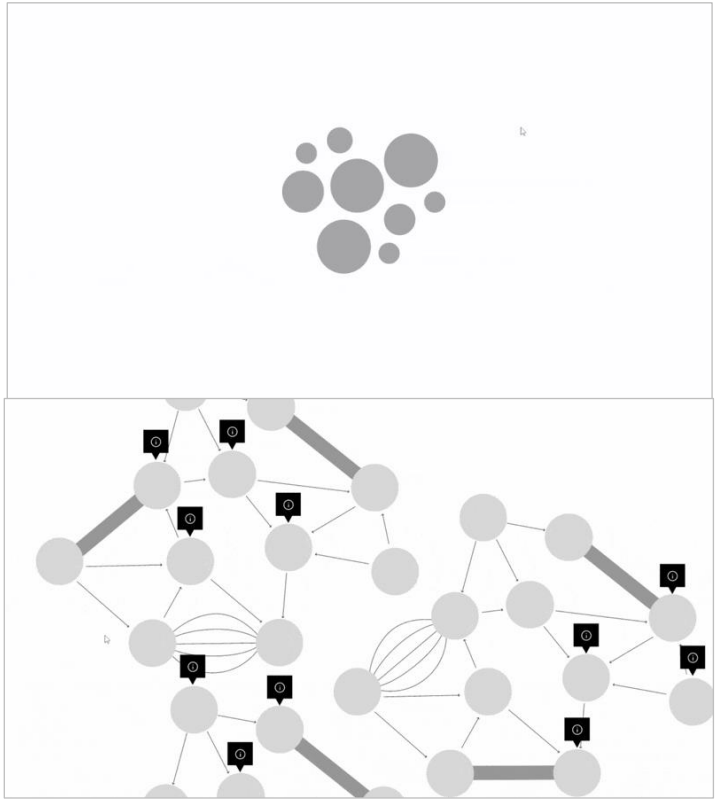
Lo-fi dashboard (top) and graph (bottom)





Data viz application | *Graph UI ideation*

Zoom/pan, edge groups, and marker clustering



6 To solve this problem, I represented large numbers of edges as a group, visually distinct from other single edges.

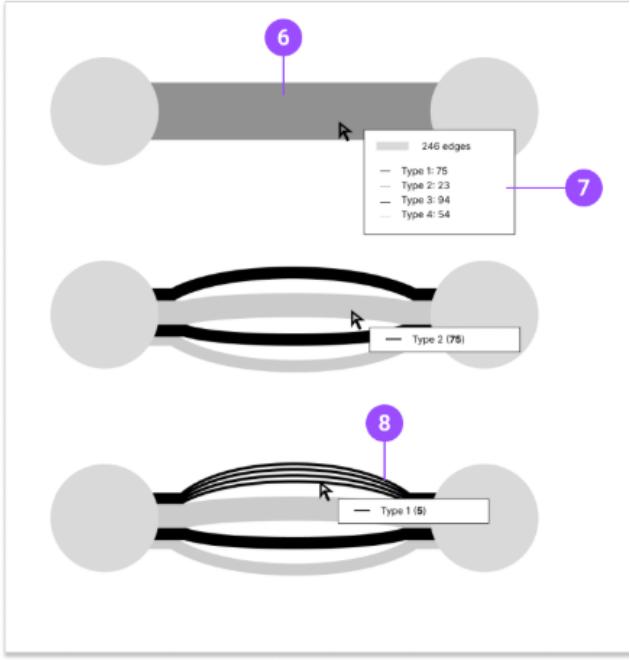
Through additional layers of interaction with the graph elements, users could reveal more information about this group.

7 Hovering or right clicking brings up a context menu, wherein users can choose to expand the edge group.

The same pattern of right clicking to access more actions, could be applied to groups of nodes as well.

8 Once expanded, users can see each edge type as part of the group. Initially we thought it made sense to allow users to continue expanding nested edge groups, but this was removed and simplified in a later design iteration.

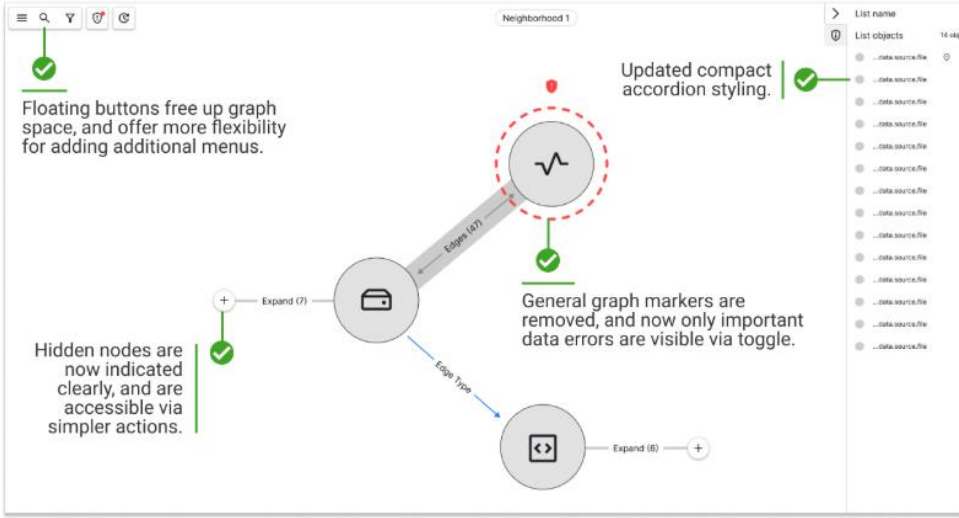
Arbitrary edge counts

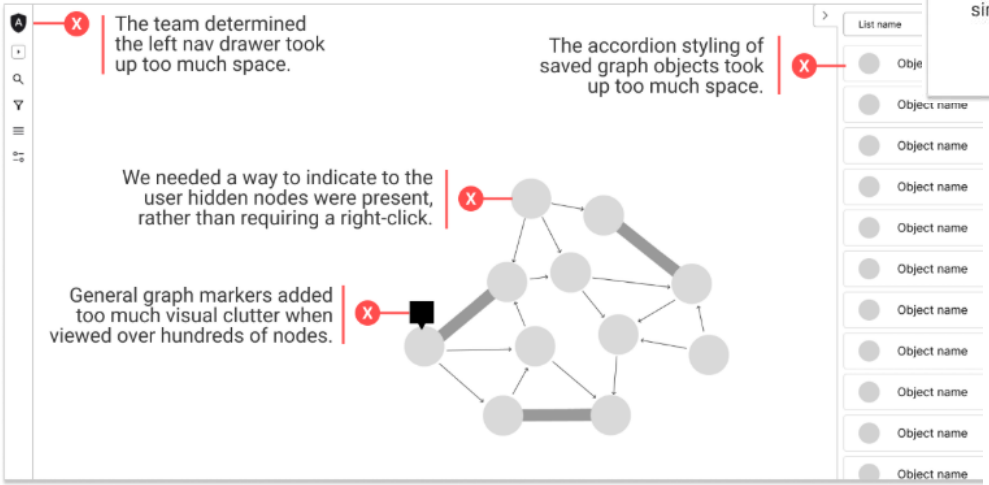


Data viz application | *Mid-fi mockups & testing*

How & who

- Remote testing with internal SMEs, using Figma prototypes
  - (Devs rarely caught up to design)
- 5 SMEs, internal PM/management



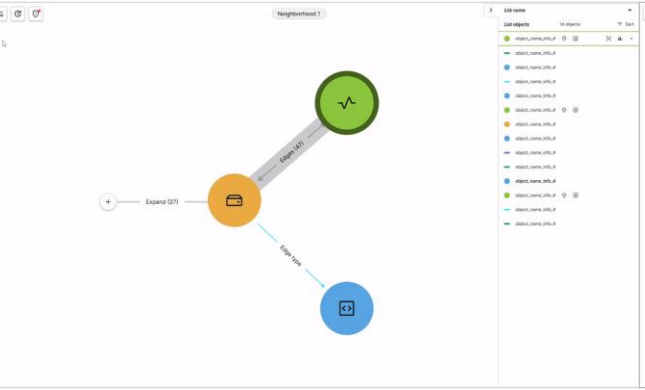


Results

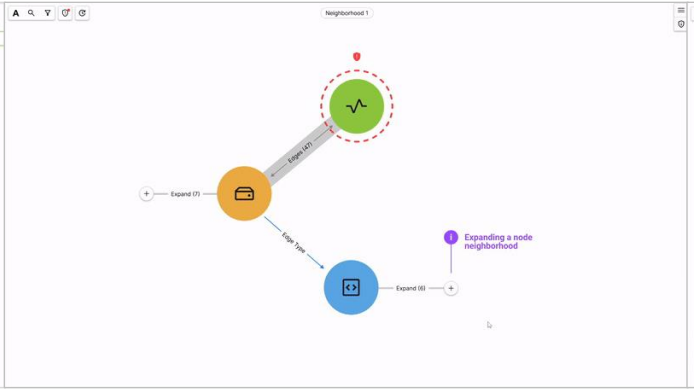
- Iterative updates to dashboard and graph UI
- Refined representations of MVP features

Data viz application | *Final deliverables*

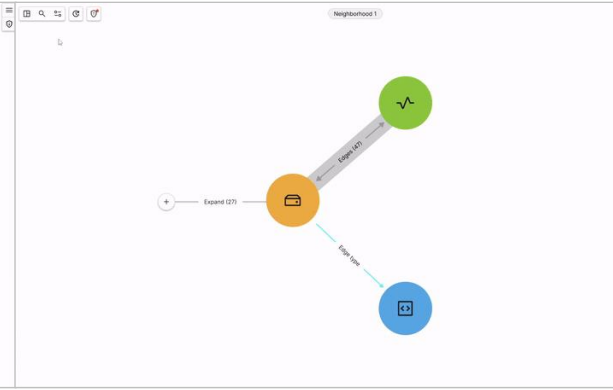
Search & lists



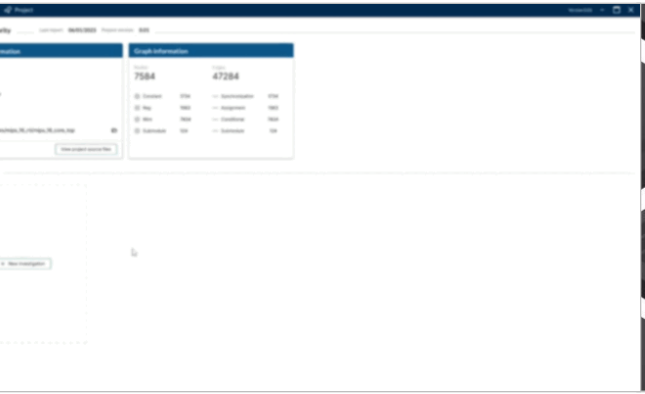
Expanding neighborhoods



Options menu & tabs

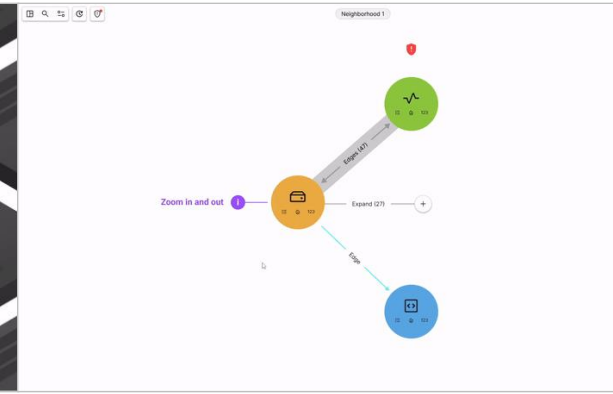


Dashboard (blurred)



Marketing animation

Icon zoom



Mini retro

- Lessons learned
  - Communicating with technical experts
  - Dealing with scope creep
  - Demo and marketing voiceovers / animation
- Do anything differently?
  - More robust testing (NDA issues?)
  - Look into 3D representations of layered data

Thank you!