Symba

Progress Report I

March 7, 2020

Authors

Vivian Fan

Sarrah Ali

Christopher Elliot

Dylan Tsai

Kehao Yao

Sudeshna Pontula

Zhihao Tang

Contents

1 Main Objective

This report serves to outline the preliminary progress made on the dashboard project for the Symba Internship Management Platform. During the first few weeks of contact with our Client, Symba, most of our team's effort was initially focused upon establishing communication with the Client and discussing what they wanted most out of this product. This resulted in our first set of requirements which is detailed below for reference. Following that, our team got a start in the process of creating low to mid-fidelity mockups to roughly visualize these requirements. Sometime after this stage, however, discussions with the Client revealed that they were shifting their company's focus and restructuring their priorities. As a result, we were asked to accommodate for these changes by modifying the responsibilities of the dashboard. Currently, we are in the process of re-evaluating the dashboard requirements and getting to a point where we can once again agree with the Client on a feasible and clear project scope.

2 Initial Project Requirements

At first, the dashboard was intended to function as a tool for admins and HR coordinators, where they could monitor the statuses of working remote interns in their companies. Specifically, it was expected to do the following:

- Be viewed by company administrators and HR coordinators but not interns.
- Allow users to collect information on interns.
- Display data visualizations concerning the interns.
- Allow users to see all interns' basic information.
- Allow easy navigation to intern profiles.

- Summarize progress of different interns / intern projects.
- Display interns' completion of paperwork and onboarding materials.

2.1 Feedback on Initial Requirements

In the comments to our initial feasibility report, Professor Arms suggested that we should consider having full tests covering the entire system for the sake of using spiral development. In this case, that would include having tests for functionality over both the main Symba platform and our proposed dashboard. This would help with integration testing and making sure the dashboard works as expected with the rest of the Symba platform. We have not yet discussed how to go about this with the Client but plan to bring it up to them in the near future to better understand how exactly integration will look. Additionally, he suggested setting up weekly full-team meetings to hold reviews of project progress with everyone. While this may not be feasible for our entire team specifically, we were able to maintain a schedule of communicating with the Client at least once a week as of now with as many team members present as possible.

3 Response to Client Restructuring

A few weeks after certifying the first set of requirements, we were notified by our Client that their focus recently shifted so that, now, connecting companies with remote interns has more immediate importance. As a result, the Development Team is in the process of redefining the project scope and working with the Client to clarify and reform project requirements.

After discussing with the Client about several potential directions with which to now move the project in, our current proposal is to develop a rudimentary "Discover" page through which administrators can search through a pool of qualified and available remote interns.

However, in relation to this new feature, we have yet to fully define the idea of an intern who is "matched" to a company. Broadly, a matched intern is an intern who has been determined to be compatible with a company in some way. The matching system has not been developed by the Client, and so the precise definition of a "matched" intern is currently vague and subject to change — it is possible that the idea of "matching" could even be extended to include all interns, but each intern is matched to a different degree. The Development Team has decided to respond to this challenge by planning for flexibility in the matching algorithm.

3.1 Feedback on New Direction

During the Milestone 2 presentation, our Client expressed eager interest in having an 'Discover' page. They also did not necessarily have expectations for us to implement a matching algorithm for interns and companies. The Client gave various helpful suggestions with which to improve our initial ideas, such as displaying how soon remote interns would be able to work and accommodating for the fact that some interns with different seniority levels may describe their skills differently.

The Client was happy that the new design utilized space more effectively by making the search the focus and removed profile pictures, which could also promote bias. The Client hopes other elements of the intern profile, along with skills, will be able to be filtered via free form typing such as names.

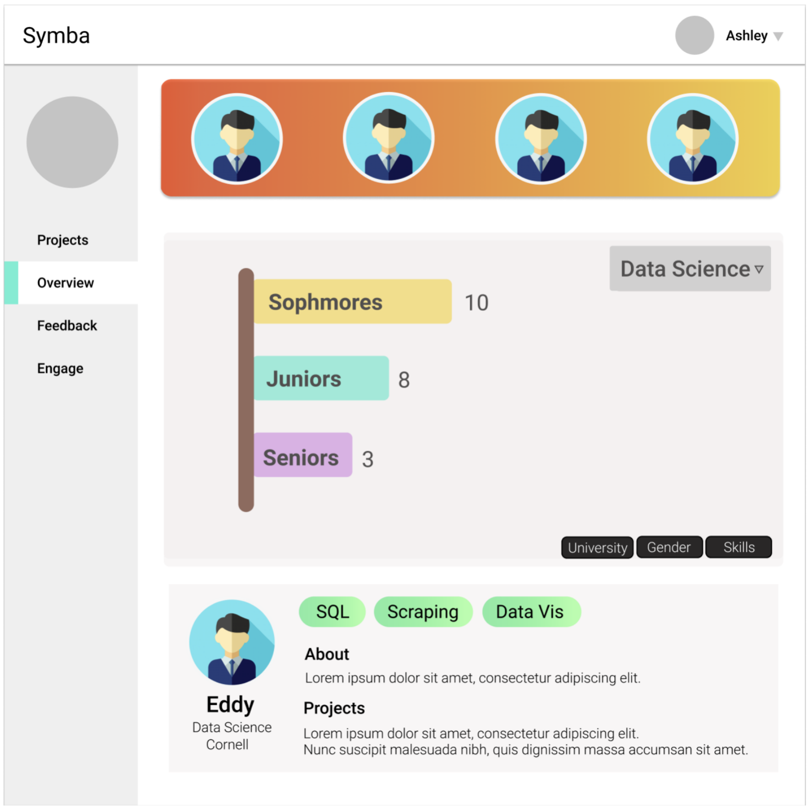
We plan to discuss further with the Client other ways the intern profile can reduce bias and what other short-form information the profile can include that can best assist HR in making decisions and lessen their burden.

4 Proposed Project Requirements

So far, we have decided to propose the aforementioned 'Explore' page to the Client as well as add any other potential features they may find useful to our final project scope. The current product now would include functionality for company admins to discover suitable matches for their internship programs. Detailed below is a compilation of updated requirements for the intern search functionality:

- Allow administrators to quickly navigate varying sizes of matched interns
- Allow administrators to quickly search for matched interns who are available for their specified work terms
- Give access to relevant matched intern information
- Display information about:
  - Flexibility for available information on matched interns / information chosen to be displayed
  - Soonest start date of interns
- Flexibility for changes in definition of "matched" interns
- Allow administrators to select or contact interns in some capacity

5 Frontend Design Decisions



About

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Projects

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc suscipit malesuada nibh, quis dignissim massa accumsan sit amet.

Figure 1: First iteration of the original dashboard project

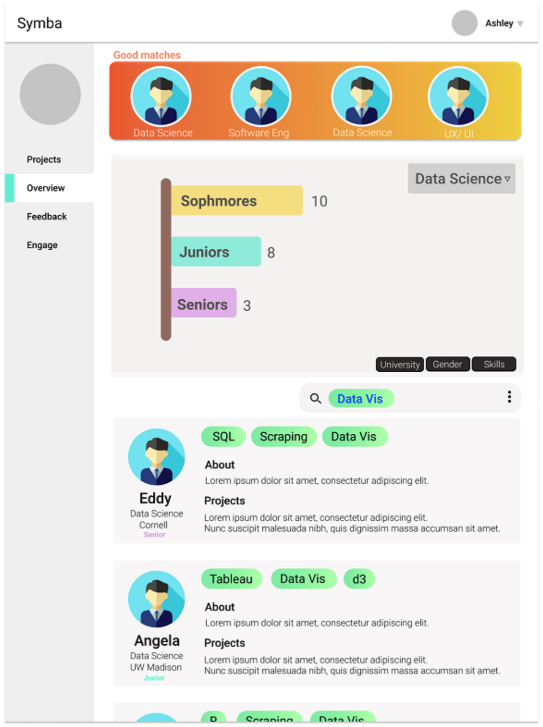
This design was formulated off the initial plan with the Client to build a dashboard to provide an HR administrator an overview of the cohort. This dashboard design attempted to satisfy Client requirements by:

- Having visualizations that give an overview of the intern cohort gender, skills, university, and seniority (important for hiring post-graduation) by department and overall.
- Giving a brief overview of interns and their projects by filter value selected on the visualizations (over all interns or those in a specific department).

- A hot zone that floated relevant interns based on whatever metric the Client desired. This was designed with the intention to compromise with the Client on the matching system that was newly pitched to us.

One part of the Client's feedback to these designs was that the 'hot zone' was not intuitive at all and they wanted a way for the administrator to simply filter intern profiles based upon skills.

The second iteration (shown in Figure 2) included a search bar to allow administrators to filter by skill by typing them in as comma separated values that would turn into the green icons on the profile. We also marginally improved the hot zone by providing more detail about the department the intern would be good for and placing "Good matches" above the zone to remove some ambiguity.



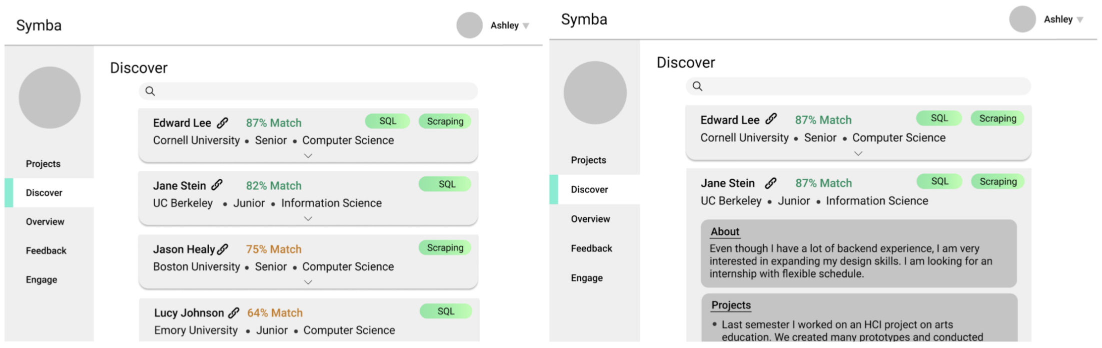
About
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Projects
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc suscipit malesuada nibh, quis dignissim massa accumsan sit amet.

About
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Projects
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc suscipit malesuada nibh, quis dignissim massa accumsan sit amet.

Figure 2: First iteration of the intern search page

The Client feedback for this design was that the hot zone remained unintuitive, and the red color gradient made the hot zone seem negative.

Accommodating for these design problems and feedback from the Client, we decided to cater toward the Client's shift. The following design presents a more streamlined intern search result. Screenshots of the prototype are given below, and the Figma prototype itself can be found here.


Symba
Ashley
Projects
Discover
Overview
Feedback
Engage
Discover

Edward Lee 87% Match SQL Scraping
Cornell University • Senior • Computer Science

Jane Stein 82% Match SQL
UC Berkeley • Junior • Information Science

Jason Healy 75% Match Scraping
Boston University • Senior • Computer Science

Lucy Johnson 64% Match SQL
Emory University • Junior • Computer Science

Symba
Ashley
Projects
Discover
Overview
Feedback
Engage
Discover

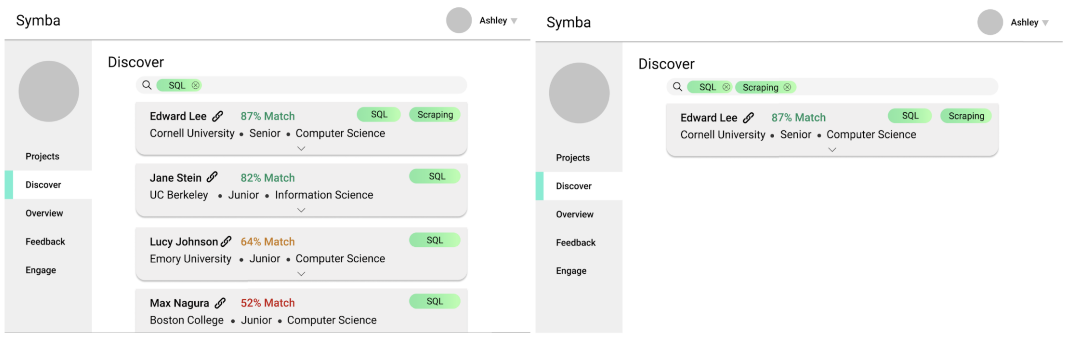
Edward Lee 87% Match SQL Scraping
Cornell University • Senior • Computer Science

Jane Stein 87% Match SQL Scraping
UC Berkeley • Junior • Information Science

About
Even though I have a lot of backend experience, I am very interested in expanding my design skills. I am looking for an internship with flexible schedule.

Projects
• Last semester I worked on an HCI project on arts education. We created many prototypes and conducted

Figure 3: Left, the intern search list with profiles collapsed.
Right, same list with an expanded profile, shown by clicking a collapsed profile.


Symba
Ashley
Projects
Discover
Overview
Feedback
Engage
Discover
SQL

Edward Lee 87% Match SQL Scraping
Cornell University • Senior • Computer Science

Jane Stein 82% Match SQL
UC Berkeley • Junior • Information Science

Lucy Johnson 64% Match SQL
Emory University • Junior • Computer Science

Max Nagura 52% Match SQL
Boston College • Junior • Computer Science

Symba
Ashley
Projects
Discover
Overview
Feedback
Engage
Discover
SQL Scraping

Edward Lee 87% Match SQL Scraping
Cornell University • Senior • Computer Science

Figure 4: Left, the intern search result when filtered on "SQL".
Right, the intern search result when filtered on "SQL" and "Scraping".

This is designed to streamline the intern search results by only previewing important information about prospective interns on the webpage. This was done so that users would have the ability to quickly go through all of the interns that matched a company's requirements. Moreover, if a user is interested in a specific client, this iteration of the design allows for users to click on the preview profile of a user to fully expand the profile. In addition to this, this iteration also adds a skill search feature in which users can click on specific skills they are looking for and filter out prospective interns that do not have that skill or combination of skills listed on their profile. We hope to determine how intuitive this feature is with user testing rounds in the near future. As well as determine what the search bar should accommodate (e.g. autocomplete feature, pre-approved skills etc.).

Furthermore, this iteration of the design also had minor aesthetic changes from previous designs. Specifically, the "hot zone" bar was removed from the design a separate overview table was created for data visualizations. Additionally the webpage was renamed as the "discover" page and photos were removed from the intern search results.

6 Backend Design Decisions

6.1 Current Progress

The Development Team had planned to develop a Python-centered system for database interactions in order to increase team efficiency later on and adapt to potential changes in the database schema. However, after planning out a few of the necessary components, the Development Team decided that the cost of development was greater than the benefit would be to the Client. The Development Team reallocated that time to learning how to use React, Flask, and PostgreSQL. While no individual is entirely comfortable with all the technologies, most team members have enough basic knowledge to contribute to a part that uses any of the project's major technologies.

During the first iteration of development, we set up all the requirements needed for data retrieval and data visualization. Because within the scope of our project, we do not need to keep track of new data and update databases, our Development Team can focus on relations between SQL tables and work on clean and efficient code. As per the milestone on presentation 2 yesterday, we have managed to retrieve data from our newly set-up AWS cloud server by using psycopg2, a python library dedicated to PostgreSQL databases. Our current major working technologies are:

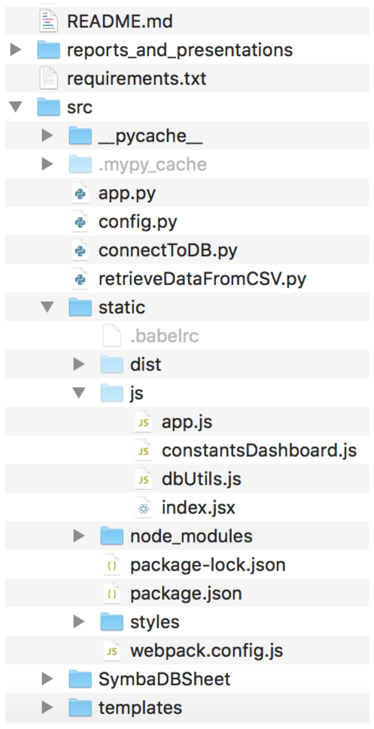
- **Psycopg2**: A popular PostgreSQL database adapter for the Python programming language. Its main features are the complete implementation of the Python DB specification and the thread safety. It is also open-source and free to use. We have 2 members leading this part of the technology
- **Flask**: Flask is a lightweight web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It is also one of the most popular Python web application frameworks. The strengths of Flask includes the Development Team choosing the tools and libraries they want to use. It is also open-source and free to use. The Development Team has 1 member leading this part of the technology, with 3 other members being familiar.
- **React**: The Development Team has 2 members leading this part of the technology. Pair programming will be discussed more in the next section.
- **Babel with Webpack**: Babel is used to transpile ES6 Javascript to ES5

Javascript. Webpack is used with Babel to watch files for changes (for real-time transpilation), and captures transpiled code into a smaller set of files (to the Development Team's understanding, there will be one transpiled file for Javascript and one transpiled file for CSS). The current setup does not use minification. We have 1 member leading this part of the technology.

The current source directory is in Figure 5, and we explain it in detail below:

- **src** has most of the source code. Files outside `src` pertain to documentation and developer tools.
  - All **Python** files are directly in **src**, but some may move into other directories later.
    - **app.py** is the entrypoint to the main application. It runs Flask to deploy the frontend and set up HTTP endpoints to help interact with the database, and connects to the AWS database. Frontend modules that need data access will call those HTTP endpoints to retrieve data.
    - **config.py** is needed for Flask. The current setup keeps `config.py` blank.
    - **connectToDB.py** and **retrieveDataFromCSV** are for local testing. They are aimed at putting information into a local database and connecting to it. It uses data stored locally in **SymbaDBSheet**.
  - **static** is the nodeJS folder. `package.json`, `package-lock.json`, `.babelrc`, and `webpack.config.js` help configure NodeJS, Babel, and Webpack.
    - **dist** contains the transpiled Javascript and CSS files.
    - **js** contains all Javascript files.
      - **app.js** is the entrypoint for all Javascript files — it handles all direct changes to the webpage, including inserting React Components. All Javascript files will be (transitively) imported into app.js.
    - **styles** includes CSS files.

– **templates** includes HTML files.



```
README.md
▶ reports_and_presentations
requirements.txt
▼ src
    ▶ __pycache__
    ▶ .mypy_cache
    app.py
    config.py
    connectToDB.py
    retrieveDataFromCSV.py
    ▼ static
        .babelrc
        ▶ dist
        ▼ js
            app.js
            constantsDashboard.js
            dbUtils.js
            index.jsx
        ▶ node_modules
        package-lock.json
        package.json
        ▶ styles
        webpack.config.js
    ▶ SymbaDBSheet
    ▶ templates
```

Figure 5: Current file system.

*Brief note about using AWS Postgres: Our Client has offered to provide us with their AWS account, but we would not want to risk contaminating the Client's database, or creating unnecessary trouble. We currently are using a free-tier AWS private cloud and will switch to the Client's cloud when we are ready for the testing stage.

During this time that designs are going through iterative review, the Development Team has spent some time thinking about how to increase future coding efficiency. The Development Team has done this by creating a command-line script to help with working on the different technologies in the code stack.

This script is not portable and the team has no intention of fully testing it — it is solely for the Development Team's efficiency. The script currently does some environment setup and then polls the appropriate project files, running MyPy when any Python files are changed, running/restarting Flask when prompted by the user, and using Webpack to watch Javascript and CSS files for Babel transpilation. The script manages synchronization of messages from Babel, MyPy, and Flask's messages. The primary benefits of this script are:

- Quick environment setup
- Insurance that MyPy is used to check all Python files
- Allowing developers to use one Terminal instead of three Terminals
- Cleaning of potential background processes spawned by Babel and Flask that can interfere with development
- Truncating the unnecessary portions of Babel's longer error messages
- Quick inspection of Babel, MyPy, and Flask logs, which are saved to files for persistence

Additionally, the Development Team has begun working on several dummy components that implement features that are similar to the new features. The most complex component integrates basic uses of all currently used technologies, and consists of the following functionalities:

- Displaying column names of a table
- Retrieving column values from a table with user-defined column name
- Displaying messages when there is no data and during loading time (that is, while data is being retrieved from the database)
- Crudely visualizing tables of user chosen dataset in comma separated values

6.2 Next steps

The one technology that has not been set up is testing technology. Some members of the Development Team have experience in Selenium and Jasmine, but further research and discussion with the Client is needed. Setup and learning for testing will be done in parallel with the development of the first high-fidelity Discover page prototype. Since this stage of the development is done using Iterative Refinement, as discussed in the Feasibility Report, the Development Team plans to ship the first high-fidelity Discover prototype with a set of tests. However, the Development Team plans for this set of tests to be somewhat minimal. Aside from considering the time constraints, the Development Team has chosen to focus less on testing in this prototype due to the inflexibility of web testing frameworks; in particular, small changes to the page's structure can result in many tests needing to be rewritten.

In lieu of a full test suite for this prototype, the Development Team intends to focus on shipping this iteration with as complete documentation as possible. The Development Team has begun moving towards this documentation goal by paying attention to explaining the codebase's setup and drafting instructions for developers using the codebase. During this next stage of the codebase's growth, the team intends on paying attention to concisely detailing the codebase's overall structure, as well as each new component that is made.

There is still much work to be done in the way of teaching the members how to use technologies. Aside from the typical teaching sessions and individual learning, the Development Team will use Pair Programming and code review to facilitate this learning.

In the beginning, Pair Programming will consist of one member who is skilled in a technology being paired with one member who is not. This will allow members who are not familiar with a technology to get hands-on experience working with it early on. Additionally, since no member is extraordinarily skilled, we believe this method to be especially effective — even the experienced member will likely learn a great deal by explaining their thought processes and working through problems with someone who has different experiences.

These code reviews will consist of two parts: presentation and discussion. The presentation will consist of a summary of the code's purpose, important factors that should be taken into consideration (for example, flexibility of a section that is prone to change with future design iterations), rationale for the code's structure, and a quick walkthrough of each part of the code. The presentation portion will also serve as practice for presenting technical details to the Client and the Professor. The discussion portion serves as feedback to help the presenter improve and as a time for team members to learn from how others approach problems or use the technologies available. The discussions are also expected to increase consistency throughout the team's work. Also, since documentation is a large focus between this progress report and the next, the Development Team intends to also review documentation during these code reviews.

7 Future Plans

7.1 Changes in communication

One prominent challenge for our team has been in clarifying project requirements. Especially with the change in the project's focus, members of the Development Team have been conflicted and unclear about project requirements. This has hindered the Development Team's ability to fully understand the Client's needs. To try to fix this, at the Client's suggestion, calls between the Development Team and the Client will shift from being scheduled separately each week to being a regular weekly call. Some times have been tentatively proposed, and will likely be finalized on March 9. During the same meeting, the Development Team will also speak with the Client about potential structures for design reviews. The Development Team has also scheduled a weekly full-team meeting to help maintain team cohesion. Lastly, the Development Team will begin assigning managers for different tasks. This will ease communication both within the Development Team and with the Client. The details of these changes will be discussed further on March 9.

7.2 Upcoming Schedule

Week of March 9th:

- Wrap up initial design discussions with Client, finalizing project scope and settling on a tentative design for Explore
- Discuss conventions/structure for database querying with the Client
- Begin initial development on frontend for 'Explore' page, setting up skeleton for basic required components and functions. Pair programming will be used
- Plan out React components that will be used in the 'Discover' page. Write rough documentation for each planned React component
- Add functionality to database to reset to the original dummy data every time a request is made, to ensure information isn't lost or altered while testing
- Research testing technologies and discuss with Client. Decide on one

Week of March 16th:

- Create documentation of ‘Discover’ page functionality
- Discuss how retrieving results from a matching algorithm will look. As this may take several days, this must be done by March 20th
- Start code reviews this week
- Begin adding CSS and formatting to the ‘Discover’ page skeleton
- Begin implementing the React components for the ‘Discover’ page that were planned in the previous week. Add to documentation of each component as it is completed.
- Implement necessary databasing functions for each React component
- If necessary, generate test data to simulate matching results depending on conclusions from the discussions mentioned above
- Develop a testing plan for the ‘Discover’ page
- One or two members begin experimenting with testing

Week of March 22nd:

- Execute testing plan that was created in previous week
- Code reviews, with a focus on documentation
- Add more CSS and formatting to the ‘Discover’ page components
- Add features to increase scalability with size of matched intern pool, such as data caching, restricted query size, and dynamic rendering on scrolling