Yellow Pages: Simplifying Network Discovery for VC Firms by Leveraging LinkedIn Connections

Angeliki Artemis Doumeni
Technical University of Denmark
Lyngby, Denmark
s234061@dtu.dk

Georgia Tsoukala
Technical University of Denmark
Lyngby, Denmark
s233094@dtu.dk

Ioannis Tselios
Technical University of Denmark
Lyngby, Denmark
s233516@dtu.dk

Konstantina Freri
Technical University of Denmark
Lyngby, Denmark
s233022@dtu.dk

Abstract

This report presents *Yellow Pages*, a web application designed to help Venture Capital (VC) firms efficiently manage and leverage their network. The app centralizes employee LinkedIn connections into a structured database, allowing users to filter and search for relevant contacts and view key information about them. The project focused on creating an intuitive and user-centered design to improve the usability of network management tools in the investment industry. User feedback on the developed prototype indicates that it is both fast and easy to understand, with clear, well-organized data presentation and a straightforward filtering system that enhances the overall user experience.

Keywords: Venture Capital (VC), Professional Networks, Network Management, LinkedIn

ACM Reference Format:
Angeliki Artemis Doumeni, Georgia Tsoukala, Ioannis Tselios, and Konstantina Freri. 2025. Yellow Pages: Simplifying Network Discovery for VC Firms by Leveraging LinkedIn Connections. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In the fast-paced and competitive investment industry, the success of a Venture Capital (VC) firm often hinges on its ability to leverage professional networks effectively. Building strong partnerships, identifying co-investors, and finding the right expertise during the analysis phase of potential investments are essential for VC firms. To complete these requirements, investors primarily rely on their and their colleagues' professional networks. However, despite this fundamental value of networks, the process of tapping into them is often inefficient, relying on informal methods such as personal recollections, one-on-one conversations, or (rarely) basic in-house tools that fail to capture the full potential of available data.

This inefficiency can have significant consequences. Missed opportunities, delays in securing critical partnerships, and difficulty accessing the right experts can directly impact the growth and success of current and potential portfolio companies. The absence of tools for managing and utilizing employee networks limits a firm's ability to act quickly and decisively.

To address this challenge, *Yellow Pages* is being prototyped, a web application built using Django and React. This tool enables VC firms to organize the LinkedIn connections of their employees into a centralized database, allowing them to filter, search, and efficiently access key information from matching contacts.

The tool's goals are to provide users with an efficient way to search for relevant contacts by implementing advanced filtering capabilities, allowing them to find contacts based on specific characteristics that match their needs. For faster results, an AI-powered search option is available to enhance the user experience. To ensure that the tool provides accurate and useful information, it displays key details in search results, and enables users to explore full work histories. Additionally, results are ordered based on a connection strength metric, which measures how closely employees are connected to each contact, using past interactions like collaborations or LinkedIn activity to suggest a higher likelihood of response. While features like having the option of sorting results by expertise were considered, the focus for now remains on connection strength.

User testing, including comparisons with existing tools, confirmed that *Yellow Pages* successfully met its goals and improved efficiency and usability.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Background

The use of professional networks in the venture capital (VC) industry is not a novel concept. However, the tools available for managing and leveraging these networks remain limited in functionality, accessibility, and usability. Research shows that leveraging existing networks can significantly enhance the identification of potential opportunities and partnerships in investment decisions [8]. VC firms, in particular, depend heavily on their networks to gain competitive advantages by connecting with experts, co-investors, and promising startups [10].

Despite the evident value of professional networks, the reliance on informal methods persists. Studies indicate that firms often rely on memory and interpersonal communication, which are prone to inefficiencies and biases [14]. Moreover, existing tools fail to adequately address the nuanced needs of VC firms, such as prioritizing connections based on relevance and leveraging aggregated data effectively [6].

Efforts to digitize and optimize professional networking processes have gained traction in recent years. Platforms like LinkedIn provide extensive networking capabilities but lack features tailored to enterprise-level needs, such as team-wide data aggregation and advanced filtering options [13]. Enterprise tools like Customer Relationship Management (CRM) software offer some utility in network management but are often not equipped to handle unstructured relationship data from diverse professional interactions [11].

Recognizing these limitations, the *Yellow Pages* prototype was designed to bridge the gap between generic platforms like LinkedIn and specialized, enterprise-level needs.

Yellow Pages is developed from scratch, designed to address the limitations and shortcomings of another tool previously used by VC firms. The former tool, built using Streamlit, lacked a modern frontend framework and suffered from overly complex filtering options and a cluttered table format. *Yellow Pages* examines and addresses these issues by simplifying filters, focusing on connection strength, and organizing results in a clean, scrollable data table containing the main information as well as a side panel that includes the details of each result.

For the development of *Yellow Pages* many technologies were examined. The combination of Django and React is increasingly used in enterprise applications for its robust capabilities and developer efficiency [9]. Moreover, AWS services are used, since its scalability and reliability support modern web applications like Yellow Pages [15].

The user interface was designed in a human-centered approach, using iterative development which follows principles from Donald Norman's "The Design of Everyday Things" [12], focusing on simplicity and intuitiveness. By avoiding overly complex filtering mechanisms and presenting results in an easily digestible format, the interface ensures accessibility for a wide range of users.

The backend of *Yellow Pages* was built with a focus on efficient database organization, guided by principles of normalization and relational database design. By normalizing the database to reduce redundancy and maintain data integrity, the system ensures consistent and reliable access to employee network information. The PostgreSQL database schema follows Codd's relational model [7], organizing data into structured tables with well-defined relationships to enable quick filtering and querying of large datasets.

3 The Prototype

The prototype, Figure 1, is a full-stack web application, that as mentioned, is designed to aggregate and visualize an organization's professional network. It uses a Django [2] backend, a React [4] frontend, and a PostgreSQL [3] database, hosted on an AWS EC2 [1] instance. The implementation of this approach is available in the GitHub repository [5].

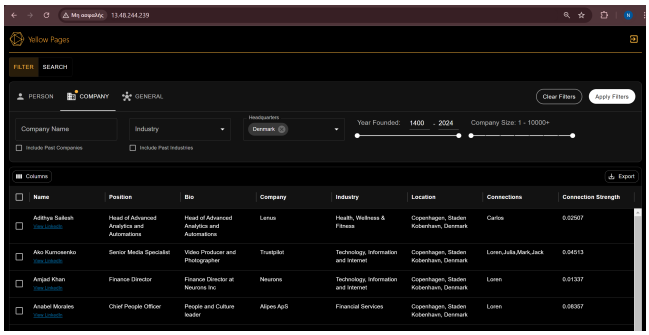


Figure 1. The Prototype

Below is a high-level description of its architecture and key components:

- **Backend**: Django serves as the API layer, implemented using RESTful endpoints to enable seamless communication with the frontend. The Django admin site is used to manage key backend functions, such as the safe bulk imports and providing an interface for admin users to calculate the "strength metric" (discussed later) or set up profiles for new employees.
- **Frontend**: The frontend, built with React, communicates with the Django backend through API calls. It uses Material-UI for its design components.
- **Database**: The application utilizes a PostgreSQL database, hosted on AWS RDS, to store the organization's connection data. A schematic presentation of the database can be found in Appendix A.
- **Hosting**: The web application is hosted on an AWS EC2 instance, ensuring consistent accessibility and reliable performance.
- **Development Environment**: Docker was used to containerize the development environment, simplifying dependency management across deployment environments.

It is important to note that the prototype relied on proprietary datasets instead of direct API integration. More details are provided in the Discussion (Part 5)

3.1 Main Use Cases

The development of the prototype was informed by user feedback, from users within a VC firm, with a focus on addressing the core challenges and limitations they faced in network mapping and analysis. Through the user interviews two primary cases that the tool needed to support were identified, along with a third secondary one that was excluded from the prototype:

- **Expert Sourcing:** The core use case, where the tool aids in identifying industry experts with relevant experience to validate potential investment opportunities.
- **Business Development and Deal Closing**: This use case involves identifying high-profile individuals—such as experts, potential customers, or key stakeholders relevant to a specific investment. By showcasing the strength and relevancy of the firm's connections, users aim to establish credibility and foster trust, ultimately facilitating deal closure.
- **Sourcing**: A secondary use case, currently outside the scope of the implemented prototype, involves tracking career changes or identifying individuals within their network, such as ex-employees or those who have transitioned to new roles.

3.2 Main Requirements

Users also outlined several key functionalities and described the shortcomings of existing processes and tools, shaping the requirements for this application.

To summarize the implemented features and their alignment with user needs, a table of core functionalities is provided (see Table 1) which address the key requirements.

The next sections further explore the main concerns of the prototype i.e. ensuring a simple and intuitive user interface and an efficient user experience, providing advanced filtering capabilities and "smart" sorting using a connection strength metric.

Feature	Description
User Authentication	Ensures secure access to the app by requiring users to log in or register, restricting usage to authenticated individuals only.
Comprehensive Access to Connections	Aggregates and displays all connections across an organization, overcoming the limitation of viewing individual profiles.
Fine-Grained Filtering	Enables narrowing down abstract concepts (e.g., "expert in fintech") to highly specific profiles (e.g., "CTO of a fintech startup with compliance expertise in the EU").
Simple and Intuitive UI	Provides a user-friendly interface designed for non-technical users.
Agility in Visualization and Information Load	Dynamically adjusts the level of detail presented, avoiding information overload while retaining full data accessibility.
"Smart" Sorting	Ranks connections based on their relevance or relationship strength, helping users identify the most effective paths to experts or leads.
Seamless LinkedIn Integration	Facilitates direct messaging through LinkedIn to streamline outreach.
Data Export	Supports exporting refined results into .xlsx files for further analysis or reporting.

Table 1. Core Functionalities of the Prototype

3.2.1 User Experience. User experience (UX) was a central focus throughout the development of *Yellow Pages*, with the primary goal of creating an intuitive and efficient tool. The design process followed a human-centered approach, incorporating feedback from real users to ensure the interface met their needs and expectations.

The prototype was designed with simplicity and clarity in mind. In particular, the filtering system was made highly intuitive. Users can access filtering options through clearly labeled dropdown menus, search bars, and range sliders, all grouped logically into tabs for easy navigation. Feedback from iterative testing showed that users appreciated the simplicity of the interface, particularly the way active filters were displayed at the top of the page for quick reference.

The lightweight integration of an AI-powered search option allowed senior users to bypass complex filtering steps and quickly retrieve relevant connections with just a few typed words.

The results are displayed in a data table that offers an intuitive experience with customizable views, allowing users to hide or unhide columns for a tailored display. Smooth scrolling and fixed column headers ensure context is not lost. Additionally, clicking on a row opens a drawer displaying

more details about the selected contact, including their full work history and information about the companies they have worked for. Users praised its simplicity and flexibility, especially for managing large datasets.

The overall user experience was validated in testing, with users expressing satisfaction with the tool's ability to save time and improve decision-making through easy-to-understand data presentation and effective search functionality.

3.2.2 Advanced Filtering and Search Capabilities. Users require not only a streamlined way to navigate the connection results but they need to narrow down large professional networks into highly specific, actionable subsets. The following filters were implemented based on user feedback to enable that:

- **Name:** Users can search for individuals by entering their first or last name to quickly locate a specific person.
- **Position (Current and Past):** Users can filter individuals by their current role or any past roles.
- **Function (Current and Past):** Users can filter by broader functional categories derived from positions, enabling a more generalized search within specific areas of expertise. Roles like React or Vue developers can be found under the broader function of frontend developers. This mapping was included in the dataset.
- **Person and Company Location:** Filters allow searches based on the geographical location of either the individual or the organization.
- **Company (Current and Past):** Users can find individuals associated with a specific company, including past employers. Early iterations of the system lacked clarity on how to specify past associations; this was refined after users requested clearer separation between current and past roles.
- **Industry (Current and Past):** Filters include the ability to search for industries tied to current or past professional roles.
- **Company Size and Founding Period:** Users can specify ranges for company size and founding period to narrow their search to organizations of interest.
- **Keywords:** Users can search for specific keywords found in the summary or bio of individuals to identify those with relevant expertise or background.
- **Connections:** Users can select to see who within their organization was directly connected to the individual in question.

Filtering was implemented exclusively on the backend to meet the users' demand for speed and efficiency; ensuring minimal data transfer and maximizing response times. More specifically, paginated API endpoints were used, in order for the system to be able to handle large datasets and avoid performance bottlenecks. This approach of course sacrifices the option for adaptive filtering, as the front end has only a subset of data available each time. However, users did not provide negative feedback regarding this.

As mentioned above, the filtering interface was designed to balance functionality and simplicity. Users interact with the filters through three tabs, each corresponding to a set of related filters. These tabs include dropdown menus, free text, range and autocomplete fields. Following user feedback, the interface displayed reminders of active filters across tabs, ensuring clarity even when switching contexts.

In addition to traditional filtering, the prototype includes a lightweight LLM integration targeted toward time-sensitive searches, enabling senior users to bypass detailed filtering processes when necessary. This feature utilizes the GPT-4o-mini model for cost efficiency and operates as an SQL generator using few-shot prompting. Users can access this functionality through a search bar, where they can input natural language queries, such as, "I am looking for an expert in oncology in Copenhagen". The model translates the query into a targeted SQL statement to retrieve the primary keys (URLs) of relevant connections from the database. These results are returned in the same format as standard filtering results.

3.2.3 Connection Strength Metric. The connection strength metric measures how well employees are connected to individuals in their professional network, prioritizing contacts more likely to respond or collaborate effectively. The goal is to identify stronger, more meaningful relationships, making the tool better for finding reliable contacts. This metric ranks search results, ensuring that connections with higher scores appear first in the table.

The metric is built using two main factors. First, LinkedIn interactions are analyzed, specifically comments made by employees on their connections' posts or profiles. With the employees' permission, comments at least six months old are gathered. Each comment between an employee and a connection adds **+1 interaction** to the connection strength, providing a straightforward measure of engagement.

Second, shared work experience is considered. By examining data stored in the database, it is checked whether an employee and a connection have worked at the same company during the same time period. If they did, the assumption is made that they may know each other better than just a standard connection. Both the interaction count and shared work history are combined in the following formula:

$$\text{Connection Strength} = w_1 \cdot I + w_2 \cdot S$$

where:
I = Number of interactions (e.g., comments or engagements)
S = Indicator for shared work experience (1 if shared, 0 otherwise).
w_1: Weight assigned to the interaction component.
w_2: Weight assigned to the shared work experience component.

The calculation process is detailed in Algorithm 1.

Algorithm 1 Connection Strength Metric Calculation

Require: Employees' comments C, Users' and Employees' work history H
Ensure: Connection strength metrics S exported to the database

for all $c \in C$ **do** ▷ Iterate over all employee comments
  Extract connections $conn(c)$ from comments and replies
  for all $u \in conn(c)$ **do** ▷ For each connection
    $I(c,u) \leftarrow I(c,u) + 1$ ▷ Increment interaction count
  end for
end for
for all $h \in H$ **do** ▷ Iterate over work histories
  Group employment periods $E(h)$ by company
end for
for all companies k **do**
  for all employees e and users u in $E(k)$ **do**
    if overlap$(e,u) > 0$ **then** ▷ Check for overlapping periods
      $W(e,u) \leftarrow W(e,u) + \text{overlap}(e,u)$ ▷ Increment worked-before count
    end if
  end for
end for
Normalize interaction counts $I(c,u)$ and worked-before counts $W(e,u)$
Combine $I(c,u)$ and $W(e,u)$ into a unified dataset D
for all $(c,u) \in D$ **do**
  $S(c,u) \leftarrow w_1 \cdot I(c,u) + w_2 \cdot W(e,u)$ ▷ Calculate connection strength
end for
Export $S(c,u)$ to the database

4 Testing and Results

The development of *Yellow Pages* followed an iterative process guided by user feedback through methods such as think-aloud protocols, interviews, and A/B testing, involving both frontend and backend improvements.

Initial testing identified key issues with the VC firm's existing tool, including confusing filters, chaotic results, and limited search capabilities. Based on this feedback, the first iteration introduced a streamlined interface with three main elements: filters, a data table, and panels for additional information.

At the same time, the backend development focused on implementing the connection strength metric. Initially, this metric only considered whether employees and connections had worked together in the past. However, testing revealed that incorporating additional data, comments, could improve the accuracy of the results by capturing more indirect connections.

Subsequent testing focused on refining the layout and interaction flow, particularly for displaying detailed connection information. This led to the implementation of a collapsible drawer for easier access to details. Further iterations fine-tuned the content and placement of information within the interface, ensuring clarity and ease of use.

After the addition of a master search field powered by AI, the final user testing was performed. This final round of testing focused on assessing the tool's usability and its ability to meet practical use cases. The results showed that *Yellow Pages* was able to provide more targeted and actionable results compared to the existing tool. Users particularly valued the filtering system for its clarity and control over the search process, confirming the importance of balancing AI features with manual filters to ensure the tool is intuitive and effective.

A detailed analysis of the iterations and their results is provided in Appendix B.

5 Discussion

A key limitation encountered during the development was LinkedIn's restricted API access. The Connections API only provides 1st-degree connections, and accessing 2nd-degree connections requires the People Search API, which doesn't offer a direct list. Furthermore, accessing these APIs requires participation in LinkedIn's Partner Program, and the approval process can be lengthy and uncertain. As a result, instead of using LinkedIn's APIs to gather connection data, a dataset of employees and their connections was sourced from a company, where each employee downloaded their own connections to create the database. This approach allowed for more controlled data access but also raised important GDPR considerations, as personal data was shared by employees rather than directly accessed from LinkedIn. In the future, integrating LinkedIn's APIs could also allow the tool to update in real time as employees make new connections, enhancing the tool's capabilities and making it more dynamic.

Additionally, the lack of advanced database features, such as indexing, is a known limitation in this prototype. Implementing indexing in future iterations could significantly improve database query performance, especially as the number of records grows, ensuring faster and more efficient searches.

On the UI side, there is always room for enhancement. For instance, future versions could highlight keywords in the results when filtering by specific terms, making it easier for users to identify relevant matches. Additional filters could also be introduced to address even more specific use cases, further increasing the tool's flexibility and usability.

In summary, while the prototype of *Yellow Pages* demonstrates significant potential, there are still a few limitations that highlight areas for future improvement.

References

[1] 2024. Amazon EC2 – Secure and Resizable Compute Capacity in the Cloud. https://aws.amazon.com/pm/ec2/ Accessed: 2024-11-30.

[2] 2024. Django - The Web Framework for Perfectionists with Deadlines. https://www.djangoproject.com/ Accessed: 2024-11-30.

[3] 2024. PostgreSQL: The World's Most Advanced Open Source Relational Database. https://www.postgresql.org/ Accessed: 2024-11-30.

[4] 2024. React – A JavaScript library for building user interfaces. https://react.dev/ Accessed: 2024-11-30.

[5] Ioannis Tselios Angeliki Artemis Doumeni, Georgia Tsoukala and Konstantina Freri. 2024. Yellow Pages: Simplifying Network Discovery for VC Firms by Leveraging LinkedIn Connections. https://github.com/artemdou/digital-media. Accessed: 2024-12-01.

[6] Ronald S Burt. 2005. Brokerage and closure: An introduction to social capital. *Oxford University Press* (2005).

[7] Edgar F. Codd. 1970. *A Relational Model of Data for Large Shared Data Banks*. Vol. 13. ACM. 377–387 pages. https://doi.org/10.1145/362384.362685

[8] Mark S Granovetter. 1973. The strength of weak ties. *Amer. J. Sociology* 78, 6 (1973), 1360–1380.

[9] James Hanson and Clara Wilson. 2021. Building scalable data-driven applications with Django and React. *Software Development Journal* 12, 3 (2021), 45–58.

[10] Yael V Hochberg, Alexander Ljungqvist, and Yang Lu. 2007. Whom you know matters: Venture capital networks and investment performance. *The Journal of Finance* 62, 1 (2007), 251–301.

[11] Sebastian Müller and Stephan Meier. 2016. The future of CRM: Trends and insights for businesses. *Journal of Business Research* 69, 4 (2016), 1390–1395.

[12] Donald A Norman. 2013. *The design of everyday things*. Basic Books.

[13] Huaxia Rui, Zhan Shi, and Andrew B Whinston. 2014. Online social networks and their impact on business performance: A study of LinkedIn usage. *Information Systems Research* 25, 3 (2014), 528–546.

[14] Brian Uzzi. 1997. Social structure and competition in interfirm networks: The paradox of embeddedness. *Administrative Science Quarterly* 42, 1 (1997), 35–67.

[15] Jinesh Varia. 2016. *Architecting for the Cloud: Best Practices*. Technical Report. Amazon Web Services.

A Appendix A: Database Schema

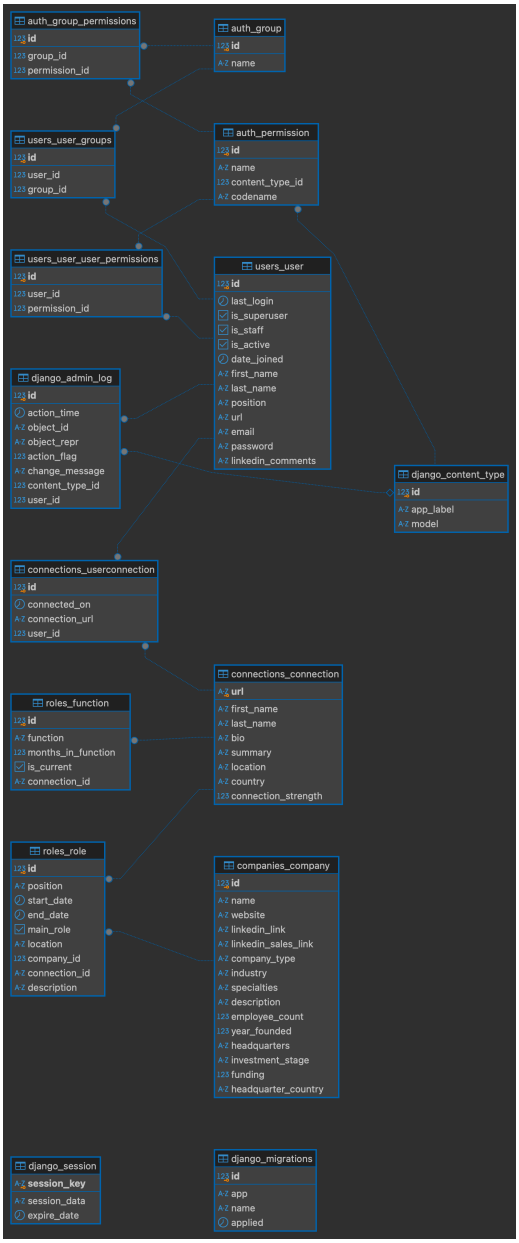


Figure 2. Database Schema

B Appendix B: Iterative Development and User Testing

B.1 Initial User Testing

At the beginning, user testing was conducted to identify the limitations of the existing VC firm tool. Key challenges included:

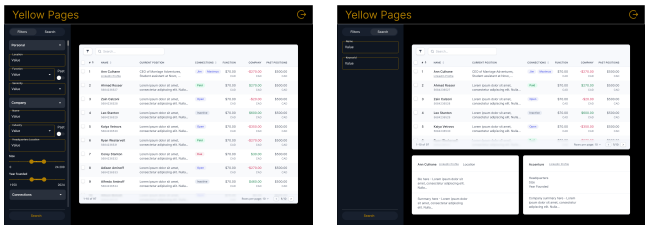
- Confusing Filters: Users found the filters overly complex, with little understanding of their function. Many filters were rarely, if ever, utilized.
- Chaotic Results: The tool often returned an overwhelming number of results or none at all, making it difficult to extract actionable insights.
- General Search Limitation: Users expressed the need for a general search option to supplement or replace the complex filter system.

These findings informed the design of the first prototype.

B.2 First Iteration

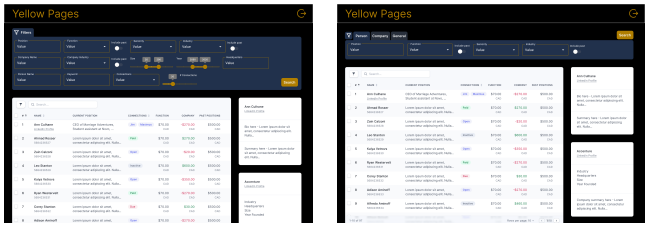
The initial prototype, designed using Figma, introduced a simple and intuitive interface with three core components:

- Filters: A redesigned filtering panel for specifying search criteria.
- Data Table: A clean, structured table to display search results.
- Panels: Panels providing additional details for selected results.



(a) Filtering. **(b)** General Search.

Figure 3. Home Page with Side Filters on Figma.



(a) Top Filters in one panel. **(b)** Top Filters in tabs.

Figure 4. Home Page with Top Filters on Figma.

B.3 Second Iteration

User feedback on the first iteration highlighted the need for better placement of detailed information. Two layouts were tested to determine the most intuitive placement for the filters and the connection details, as shown in Figure 3 and Figure 4. For the filters the Top Filters design was preferred. For the connection details, feedback suggested that neither

layout was optimal, leading to the development of a drawer-based design, where detailed information would appear in a collapsible panel upon clicking a connection, Figure 5.

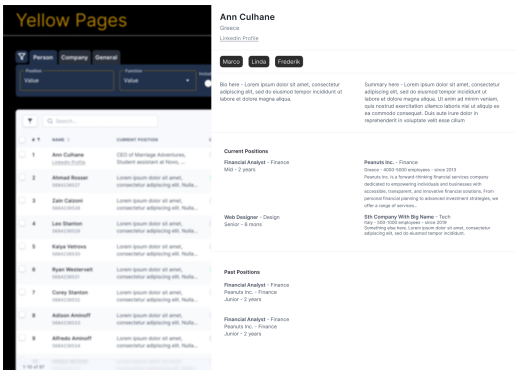
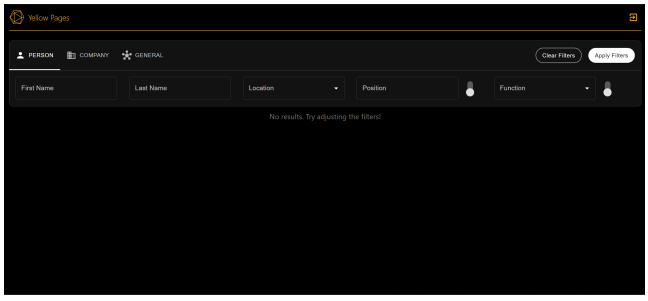


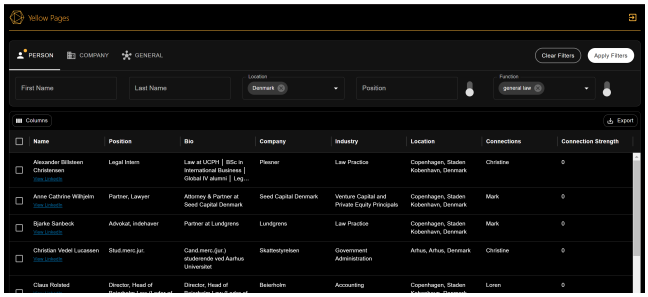
Figure 5. Home Page with Drawer on Figma.

B.4 Third Iteration

The third iteration focused on enhancing the information hierarchy and usability of the developed prototype.



(a) Home Page initial state



(b) Home Page after Search

Figure 6. Home Page - Developed Prototype

Two main things are noted:

- The level of detail necessary for each connection was established and the information was organized and displayed within a collapsible drawer to provide clarity and ease of access, Figure 7.
- Adjustments were made to define which columns should be visible by default and which should remain hidden in the initial view of the results table, improving the clarity of the presented data, Figure 9.

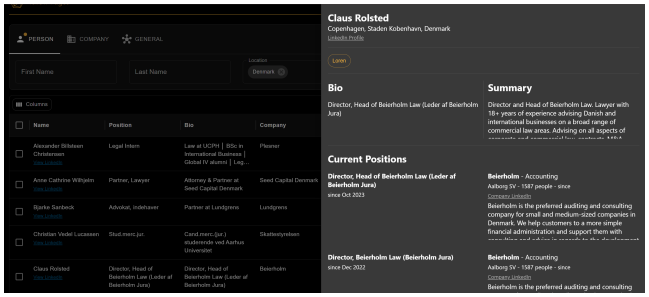
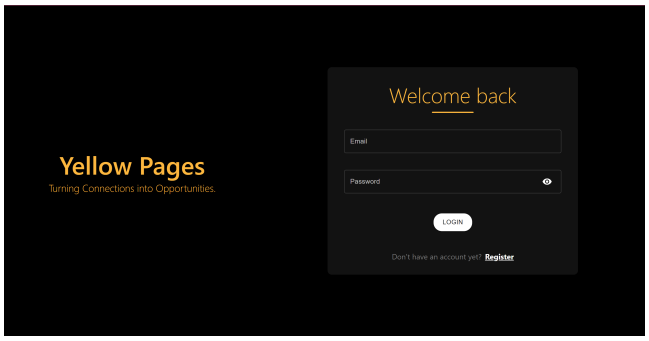


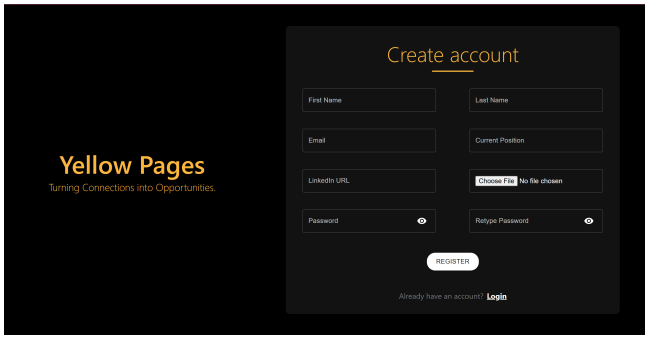
Figure 7. Home Page with Drawer.

Additionally, usability testing was conducted to evaluate the intuitiveness and functionality of the filters. Feedback highlighted issues with the toggle buttons used to enable the 'Past' search feature. Users found these buttons non-intuitive, often overlooking them, necessitating a redesign to improve visibility and engagement.

This iteration improved the tool's user experience and effectively addressed key feedback.



(a) Login Page



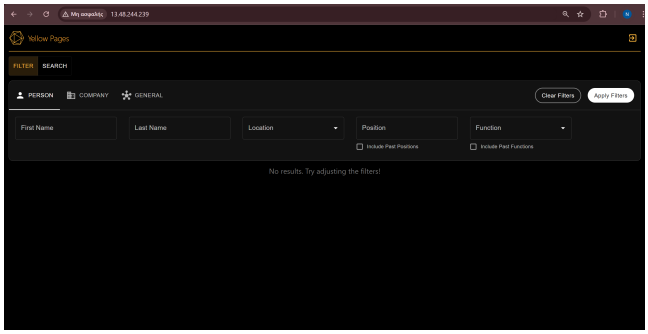
(b) Register Page

Figure 8. User Authentication

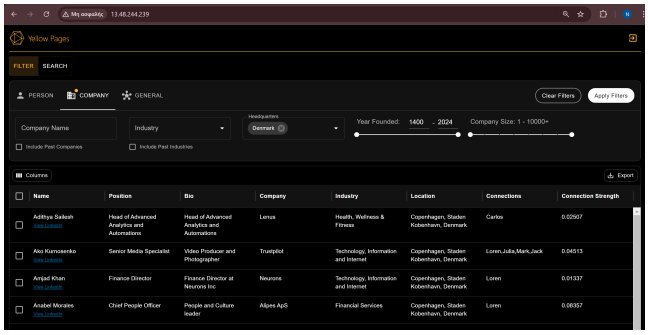
It is worth mentioning that the third iteration was conducted in the developed prototype and thus the user authentication steps (Login / Register Pages) are introduced, as shown in Figure 8

B.5 Final Iteration

The final iteration introduced an AI-powered search prompt, allowing users to quickly find relevant data through natural language queries, serving as an alternative to the traditional filter system. Additionally, the design was refined to make toggle buttons more visible, addressing earlier usability concerns.

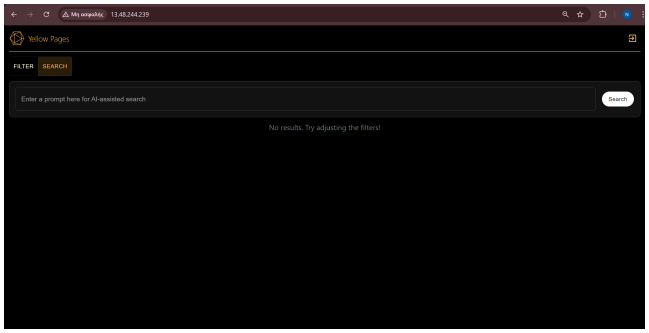


(a) Filters Input Fields

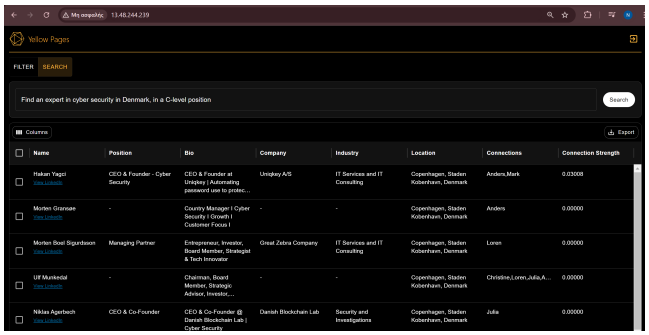


(b) Results after Applying Filters.

Figure 9. Performing search through filters



(a) AI Prompt Input Field



(b) Results after Search.

Figure 10. Performing search through AI Prompt

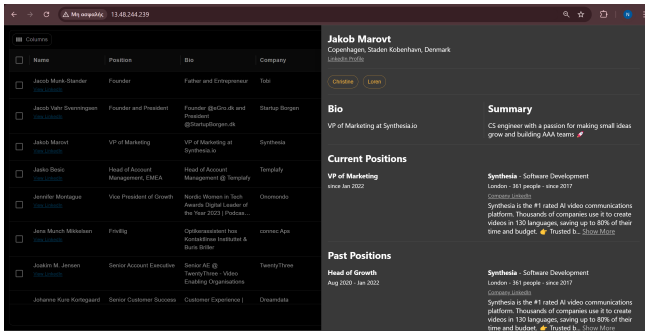


Figure 11. Home Page with Drawer

The final iteration was validated through user testing. The focus was on assessing usability, identifying remaining issues, and confirming enhancements to meet practical use cases.

The users praised the speed and simplicity of the tool. By navigating the app and trying out the filters, a few issues were identified and a few additions were noted for future work.

Our tool was tested against a practical use case: "One of the portfolio companies has an HR event we are hosting this evening, and I want to identify relevant network contacts to invite." Then, the same use case was performed on the tool previously used by VC and the reuslts were compared.

On this note, our solution delivered 61 relevant results within seconds, allowing users to confirm individual profiles through the app's drawer without leaving the platform. The previous solution required significantly more setup time, multiple manual inputs (e.g. no range input for company size), and external navigation (to LinkedIn Page) to confirm individual profiles. While it generated 270 results, many were less targeted, leading to inefficiencies.

Overall, it was clear that our solution of *Yellow Pages* significantly improved the overall user experience and delivered more precise, actionable results.

A notable thought users expressed about the AI-powered search was that it is valuable but secondary to the filtering

system. Filters were preferred since they offer more clarity and control over the search options and results. Users directly know what the available options are and re-evaluate accordingly. This feedback confirmed the necessity of balancing AI capabilities with intuitive manual filters to support diverse user preferences.

Through this iterative process, Yellow Pages evolved into a tool that effectively balances simplicity and functionality, addressing the core challenges identified in the initial testing phase.

C Appendix C: Contributions

In Table 2, the contributions of each team member are shown, divided into Report and Code sections. Each "X" indicates the areas where a member was involved.

	A.D.	**G.T.**	**I.T.**	**K.F.**
Report				
Abstract		X		
Introduction		X		
Background				X
The Prototype	X		X	
Testing and Results				X
Discussion		X		
Code				
Frontend		X		X
Backend	X		X	

Table 2. Division of Work Between Team Members