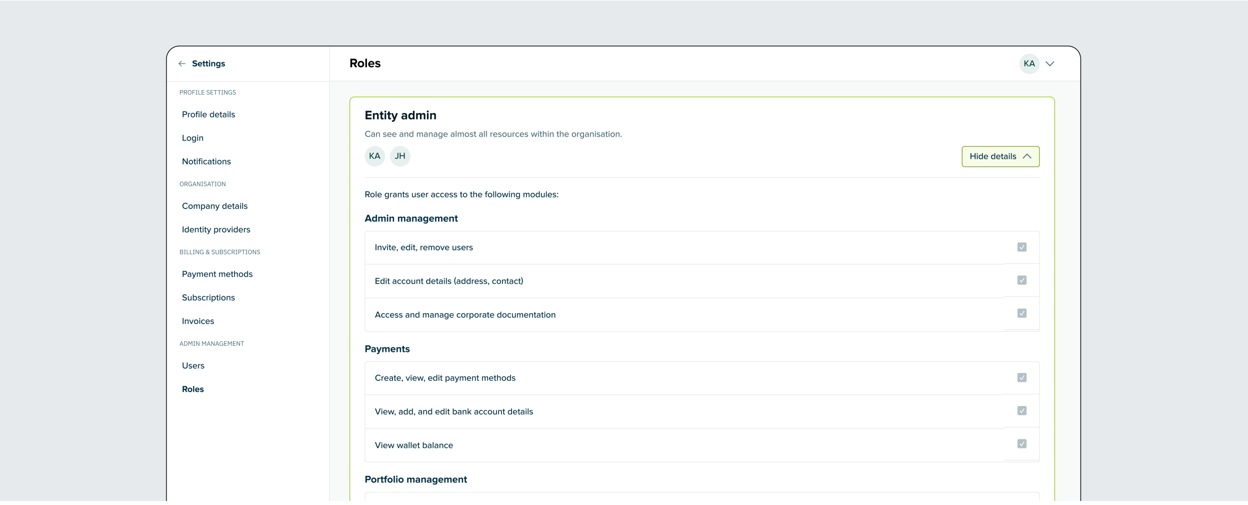


Roles

Entity admin

Can see and manage almost all resources within the organisation.

Role grants user access to the following modules:

Admin management

- Invite, edit, remove users ☑
- Edit account details (address, contact) ☑
- Access and manage corporate documentation ☑

Payments

- Create, view, edit payment methods ☑
- View, add, and edit bank account details ☑
- View wallet balance ☑

Portfolio management

Corporate accounts

Role	Ideating on the feature aimed at the high-value org clients
Skills	Content mapping Feature discovery
Team	1 Business analyst, 1 Product strategies, 4 developers, 2 architects, 2 Compliance officers
Project type	Client project

In this case study I will talk you thought the process of building a new account feature for Private marketplace platform, an asset tokenization and investing platform designed for asset issuers and investors.

Discovering gaps

The initial version of our platform offered users the option to open either a corporate or personal account. Personal account holders were onboarded as individual investors, while corporate account holders could choose to be either asset issuers or corporate investors during registration.

But there **were some gaps**.

Regardless of which account type users selected, the **platform only catered to one account user**. There was no way to invite other users to a corporate account (in a way that doesn’t involve sharing your login credentials, passing identity and AML verifications). Additionally, **there was only one role available: system admin**, also known as the super user, who had access to all platform features.

User testing confirmed that this was a **serious limitation for our highest-value clients**, the corporate users. They needed to be able to invite different types of users to the platform and assign them access rights based on their org structure.

> 💡 *“If we don't build user management functionality, our highest-value clients will become frustrated with the lack of flexibility to manage users who are engaged with the platform”*
> - ***Insight from the stakeholder interview***

Our team set out to build a user management feature, but the scope of the project changed several times. The initial feature spec included:

- Inviting new users
- Creating user groups/departments
- Setting different permission levels (page, component, API levels)
- Switching contexts between personal and corporate accounts

Understanding subject domain

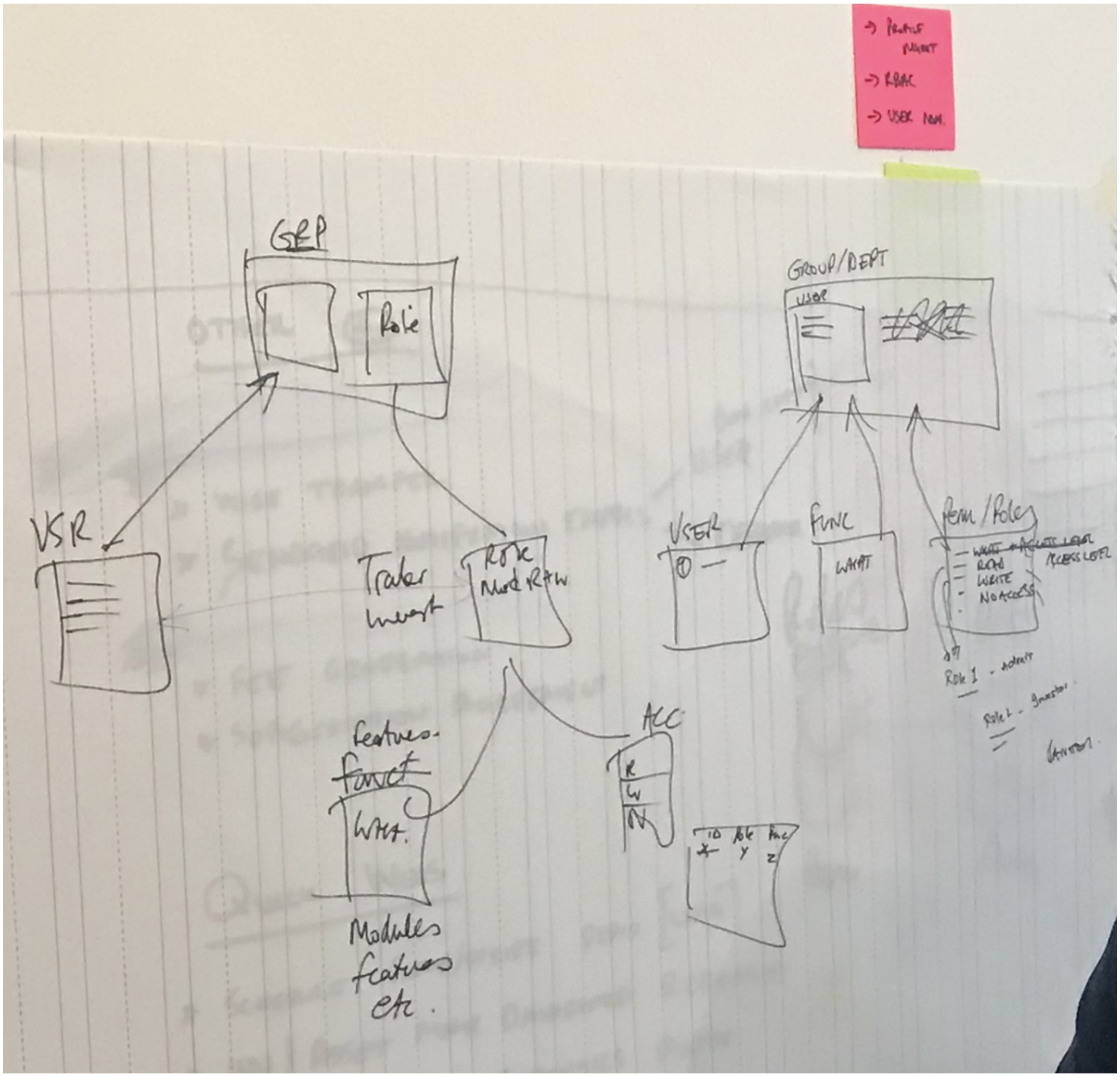
Armed with business and user requirements, I embarked on a journey to understand the domain we're dealing with. This included identifying all the nouns used in the permission structure, which some people call domain or content modeling, and others call object-oriented design.

I also collaborated with my team to break down our main concepts (roles, departments, functions, and account types) into separate data tables to better understand their connections and see what made sense.

There were many technical concepts we were struggling to comprehend (such as "function"), and I wanted to preserve our end users' mental health by sparing them the effort.

The following comments from stakeholders illustrate "the struggle":

> *"We need to get the ontology right. We must describe what the major nouns are. Otherwise, we would never reach an alignment"*
>
> *“A number of business functions and user roles are unclear to me…”*



Permission model exploration: roles, groups, departments, functions,...have we covered everything?

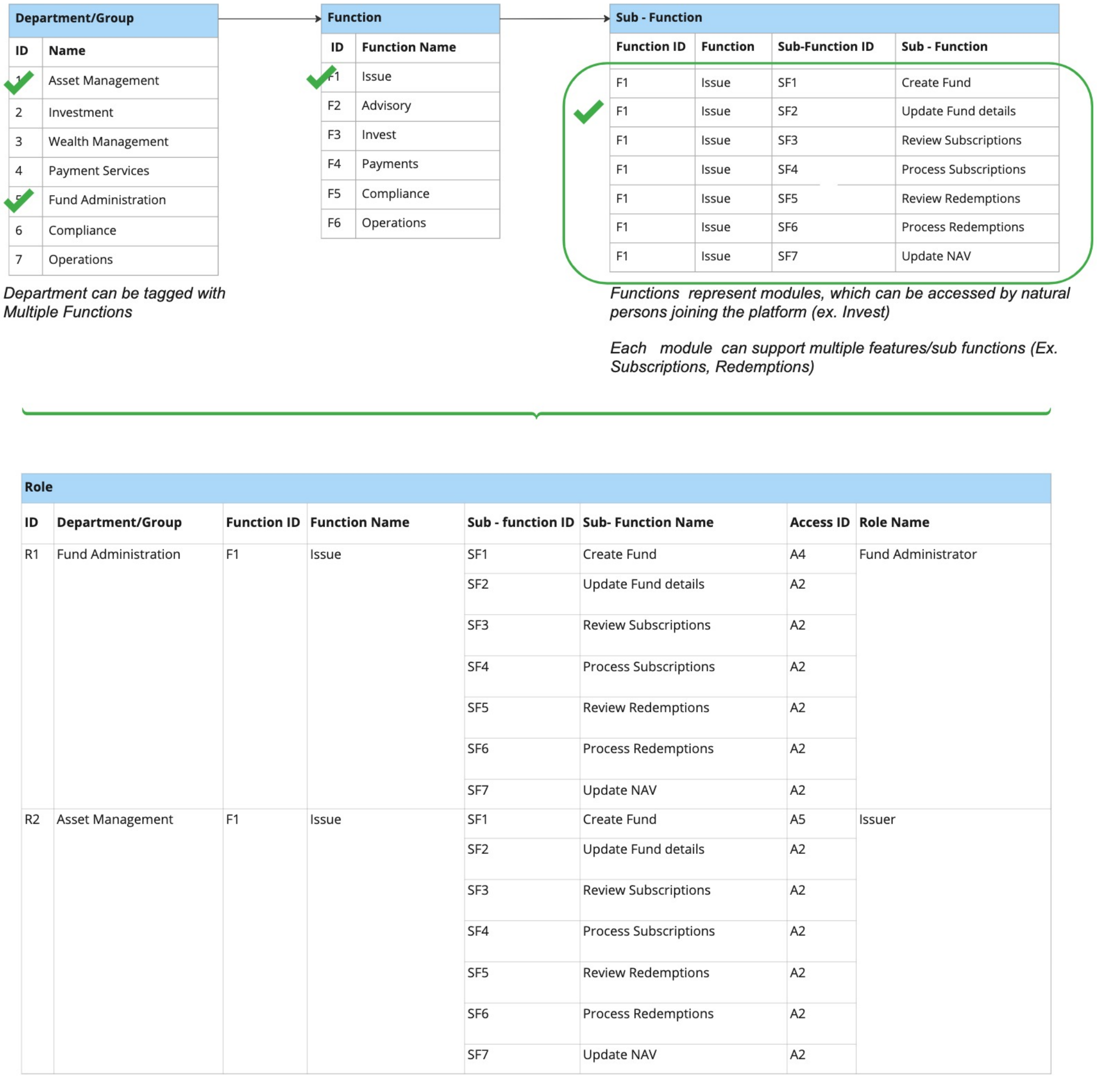
Roles - functions relationship

Scenario:

Institutional client "Acme Inc" is looking to onboard and manage their private funds on our platform.

Questions:

- Which function we will assign to this entity?
- How functions related to user roles? What are the connection between?



Department/Group

ID	Name
✓ 1	Asset Management
2	Investment
3	Wealth Management
4	Payment Services
✓ 5	Fund Administration
6	Compliance
7	Operations

Department can be tagged with Multiple Functions

Function

ID	Function Name
✓ F1	Issue
F2	Advisory
F3	Invest
F4	Payments
F5	Compliance
F6	Operations

Sub - Function

Function ID	Function	Sub-Function ID	Sub - Function
F1	Issue	SF1	Create Fund
F1	Issue	SF2	Update Fund details
F1	Issue	SF3	Review Subscriptions
F1	Issue	SF4	Process Subscriptions
F1	Issue	SF5	Review Redemptions
F1	Issue	SF6	Process Redemptions
F1	Issue	SF7	Update NAV

Functions represent modules, which can be accessed by natural persons joining the platform (ex. Invest)

Each module can support multiple features/sub functions (Ex. Subscriptions, Redemptions)

Role

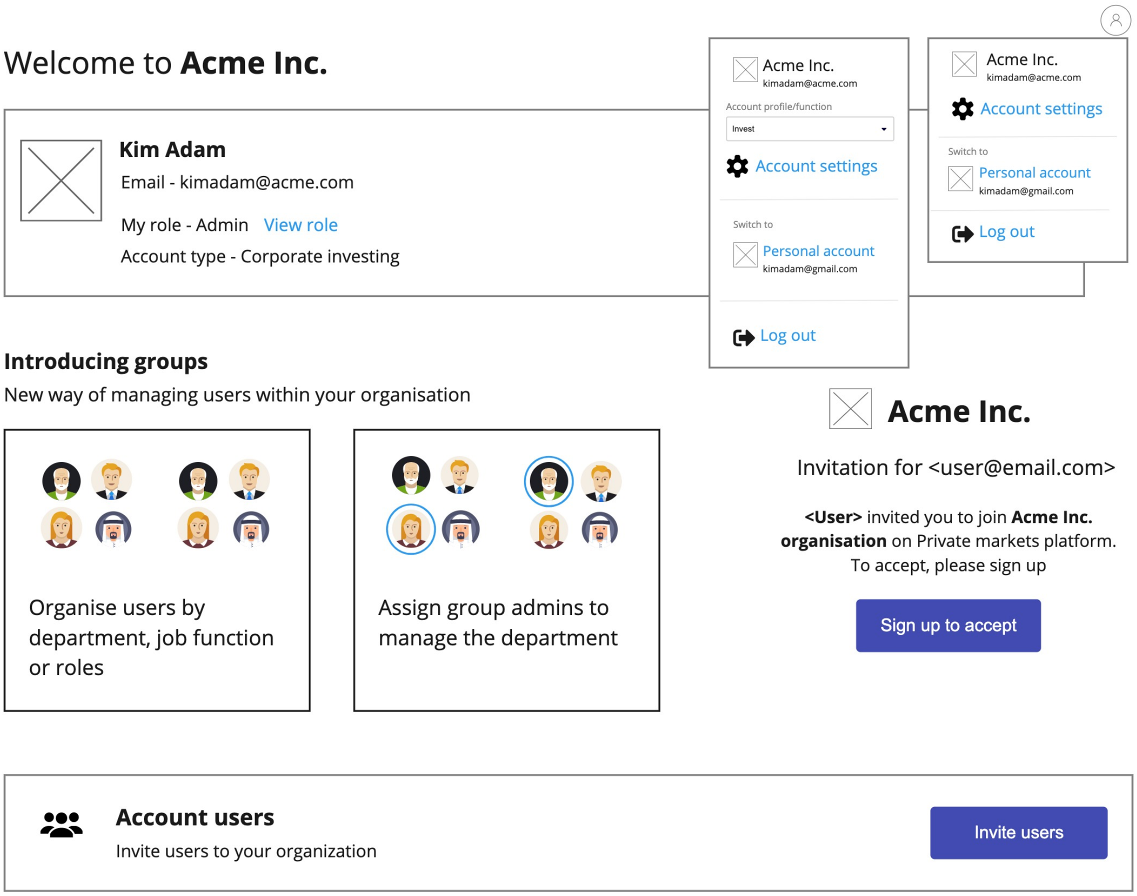
ID	Department/Group	Function ID	Function Name	Sub - function ID	Sub- Function Name	Access ID	Role Name
R1	Fund Administration	F1	Issue	SF1	Create Fund	A4	Fund Administrator
				SF2	Update Fund details	A2	
				SF3	Review Subscriptions	A2	
				SF4	Process Subscriptions	A2	
				SF5	Review Redemptions	A2	
				SF6	Process Redemptions	A2	
				SF7	Update NAV	A2	
R2	Asset Management	F1	Issue	SF1	Create Fund	A5	Issuer
				SF2	Update Fund details	A2	
				SF3	Review Subscriptions	A2	
				SF4	Process Subscriptions	A2	
				SF5	Review Redemptions	A2	
				SF6	Process Redemptions	A2	
				SF7	Update NAV	A2	

Thinking about main objects in a database format - great way to converge & see connections between. Our main architect loved it!

Visualizing these connections worked really well in helping our team align on the initial permissions model setup 😊.

User Testing

Once I have a good idea of the feature set, user journey and IA for a given project, my next step is to validate it. So I created a quick and dirty prototype to test with users.



Welcome to **Acme Inc.**

Kim Adam

Email - kimadam@acme.com

My role - Admin View role

Account type - Corporate investing

Acme Inc.
kimadam@acme.com

Account profile/function

Invest

Account settings

Switch to

Personal account
kimadam@gmail.com

Log out

Acme Inc.
kimadam@acme.com

Account settings

Switch to

Personal account
kimadam@gmail.com

Log out

Introducing groups

New way of managing users within your organisation

Organise users by department, job function or roles

Assign group admins to manage the department

Acme Inc.

Invitation for <user@email.com>

<User> invited you to join **Acme Inc. organisation** on Private markets platform. To accept, please sign up

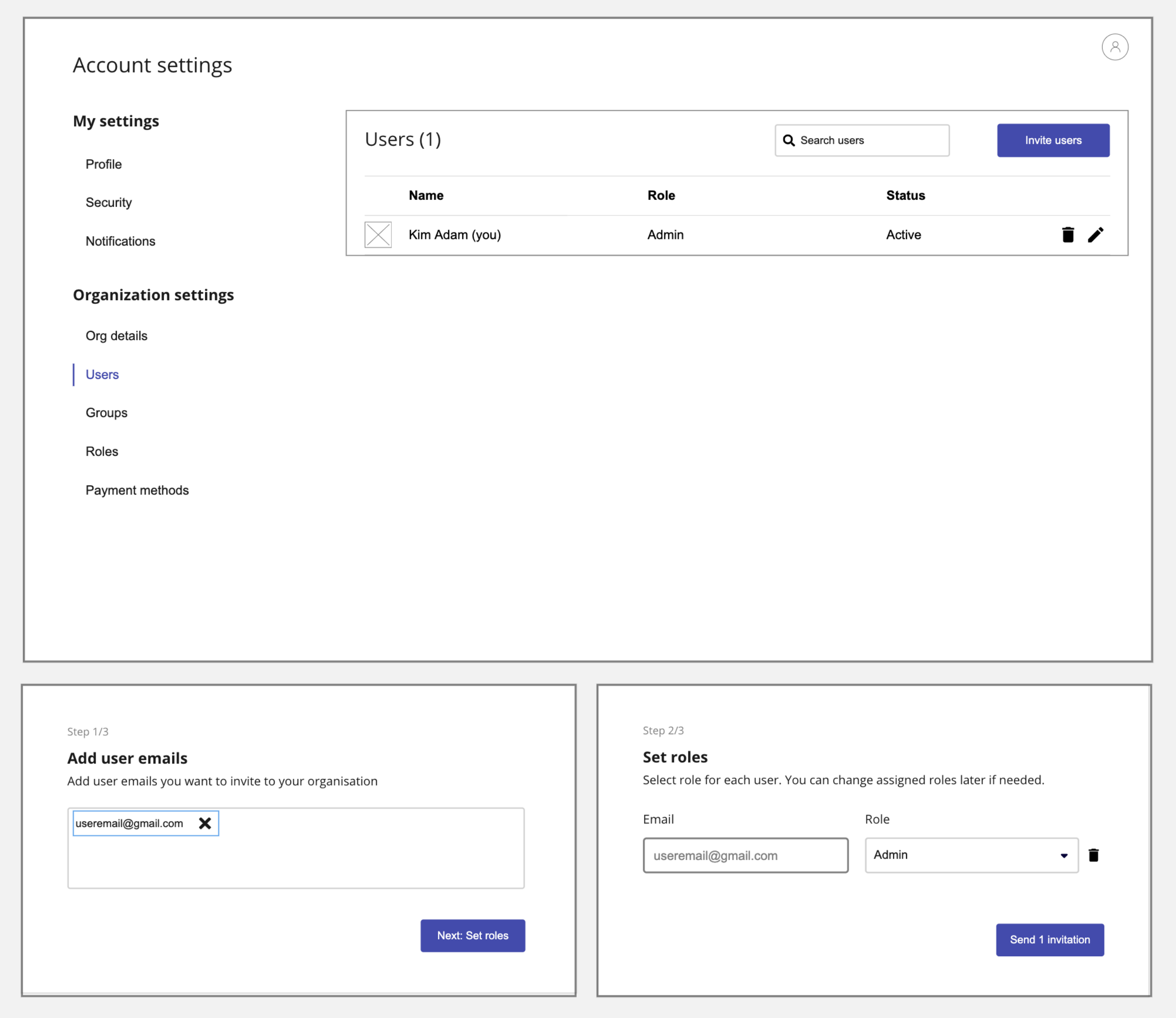
Sign up to accept

Account users

Invite users to your organization

Invite users

Mix of components used in the prototype (made in Miro)



Name	Role	Status
Kim Adam (you)	Admin	Active

Step 1/3

Add user emails

Add user emails you want to invite to your organisation

useremail@gmail.com

Step 2/3

Set roles

Select role for each user. You can change assigned roles later if needed.

Email	Role
useremail@gmail.com	Admin

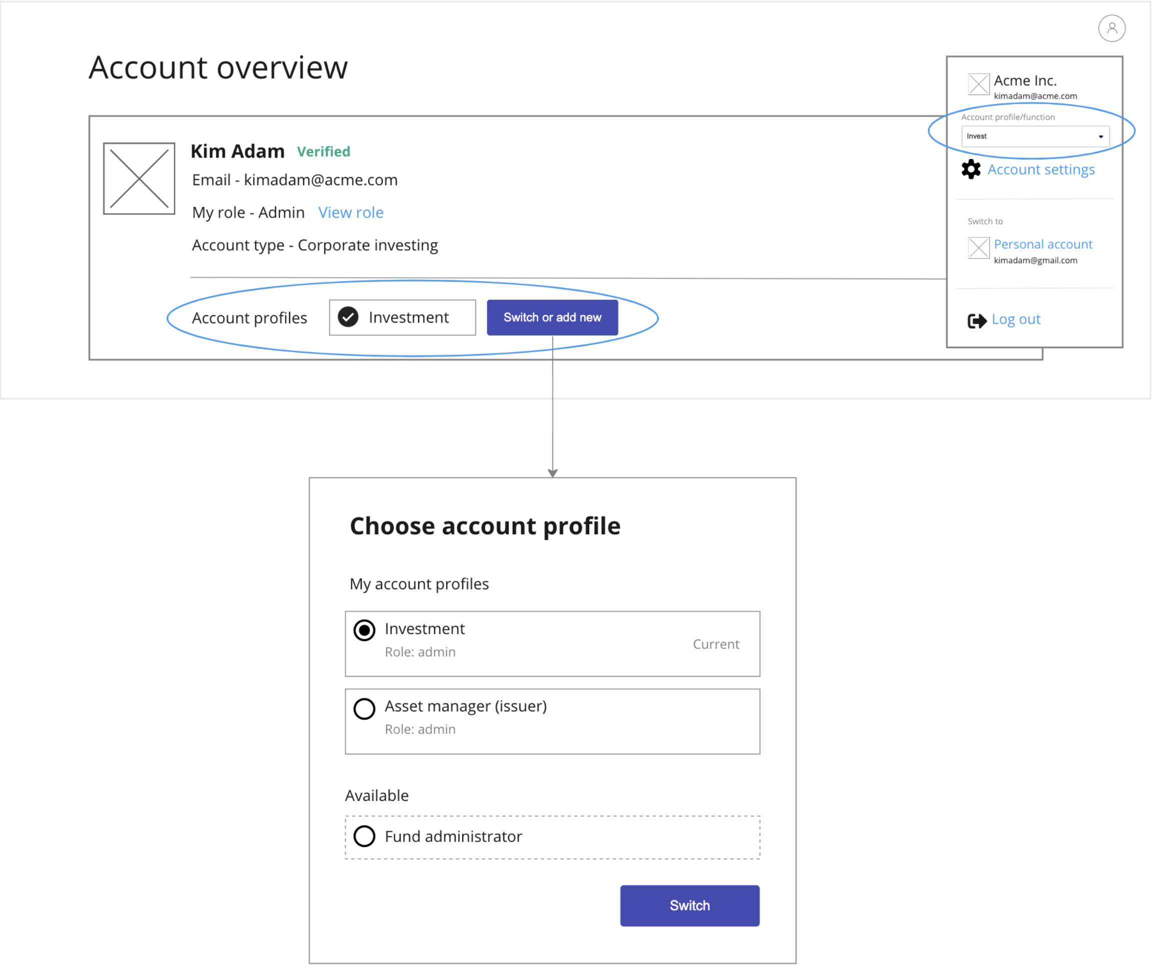
Account settings & invite users flow

I scheduled 4 interviews with users from the client side who had requested the feature. Within a week I gained a ton of valuable insights, and by the end of the 2d week, the iterated design was ready for another round of testing.

Most of the positive insights related to the ease of inviting users, a set of initial roles (admin, entity admin, investor, issuer, fund admin) and the UX around navigating between resources for individual and corporate account users.

Negative ones included the messaging regarding the permissions of different roles, the conceptual separation of personal and organisational contexts, and the meaning of account profiles (which were another name for the confusing “functions” mentioned earlier).

> 📝 **Extracts from the usability report:**
>
> - Why would I want to change my investing account to an issuer one if my intent is just to invest?
> - Almost all users asked me to explain the use of account profile/function switcher inside the same dropdown where they create new corporate account
> - Usually fund admins are corporate entities, not individual users. There should be an opportunity to nominate an entity
> - Fund administration is a type of business activity that requires a licence. How does the platform handle that?
> - Assign group to the user when inviting user to join the account
> - We should include basic description of permissions for each role
> - Invited users don’t need to deal with identity verification. Their corporate email is an identifier.
> - What’s the difference between groups & roles?

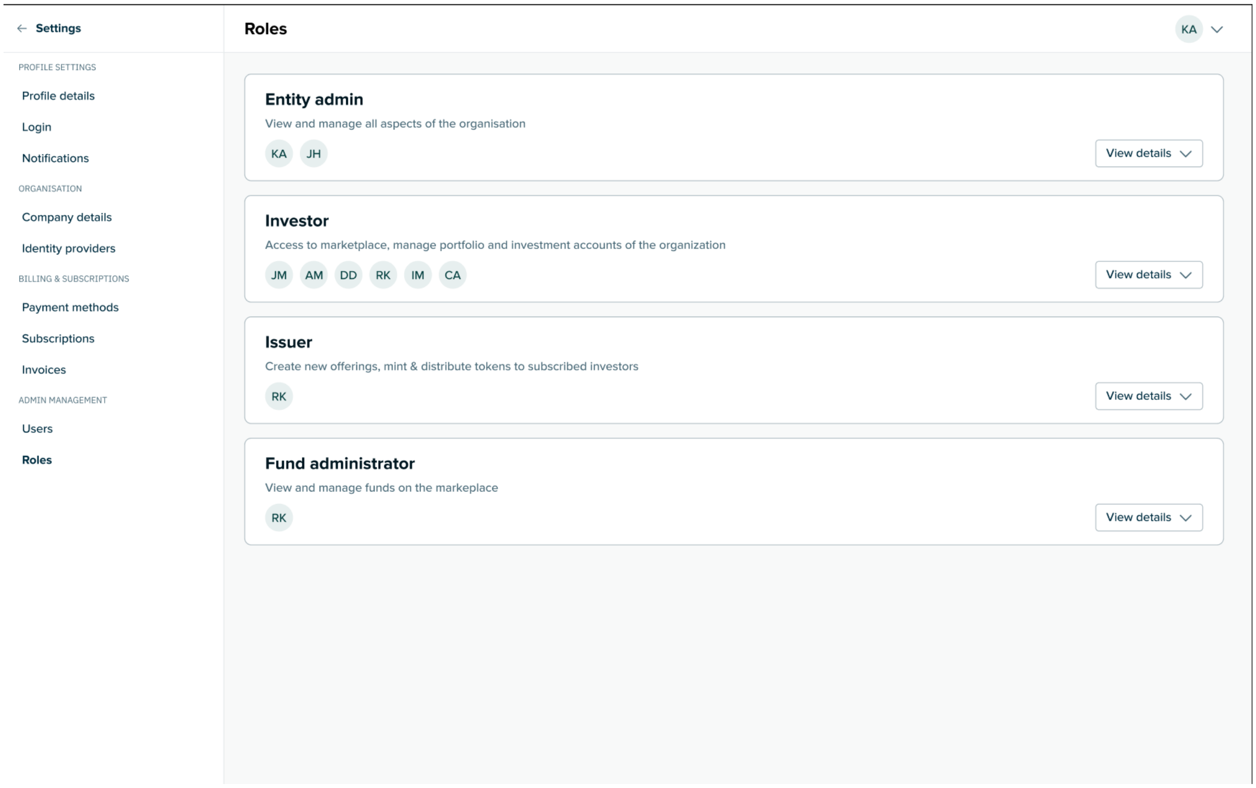


Context switcher turned out to have the most confusing UX

During our testing, we found that **switching between functions** such as invest, asset management, and fund management **was frustrating for everyone.** I knew there had to be a way to optimize this experience, so I investigated further.

First, **I challenged our assumption** that groups were the best way to manage user access rights to different platform modules (admin, portfolio management, orders, transaction modules, marketplace, offering modules). Technically, it seemed like there was no difference between them and roles. In fact, assigning a user to a role essentially makes them a "role member", much like joining a group.

As a result, we questioned the need for an additional permission layer - "groups" - and instead proposed utilizing "roles" to provide more precise control over user access rights. After all, meeting the key user requirement for granular access control was paramount, and we saw no reason to overcomplicate the UI with an unnecessary feature.



Roles

Entity admin
View and manage all aspects of the organisation
KA JH

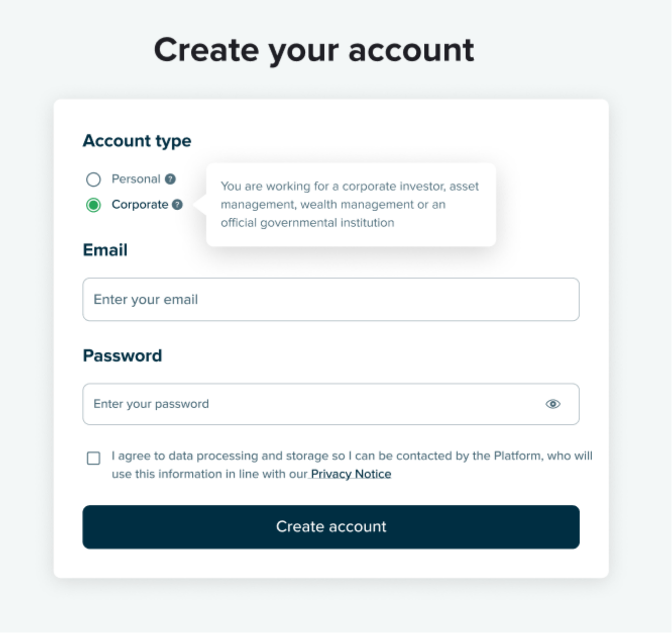
Investor
Access to marketplace, manage portfolio and investment accounts of the organization
JM AM DD RK IM CA

Issuer
Create new offerings, mint & distribute tokens to subscribed investors
RK

Fund administrator
View and manage funds on the markeplace
RK

Initial set if roles available for our corporate account users

To make things even easier, I **made the decision to remove the concept of "function" from the user's view.** Instead, all they had to do during registration was select their account type (personal or corporate), go through the verification process, and invite others to join the organization.



Create your account

Account type

○ Personal
● Corporate

You are working for a corporate investor, asset management, wealth management or an official governmental institution

Email

Enter your email

Password

Enter your password

□ I agree to data processing and storage so I can be contacted by the Platform, who will use this information in line with our Privacy Notice

This decision had a significant impact. Users were pleased with the new flow, and I was very proud of the result.

Another tweak was made to the Settings page. I split the list of tabs into 3 categories: profile settings (which belongs to the individual user), organization-wide settings (which apply to all members of the organization), and user administration settings (which are only visible to users with an entity admin role).

This way, users could easily locate the settings that are relevant to them, and those with administrative roles can access the necessary controls without any confusion.

Unexpected change

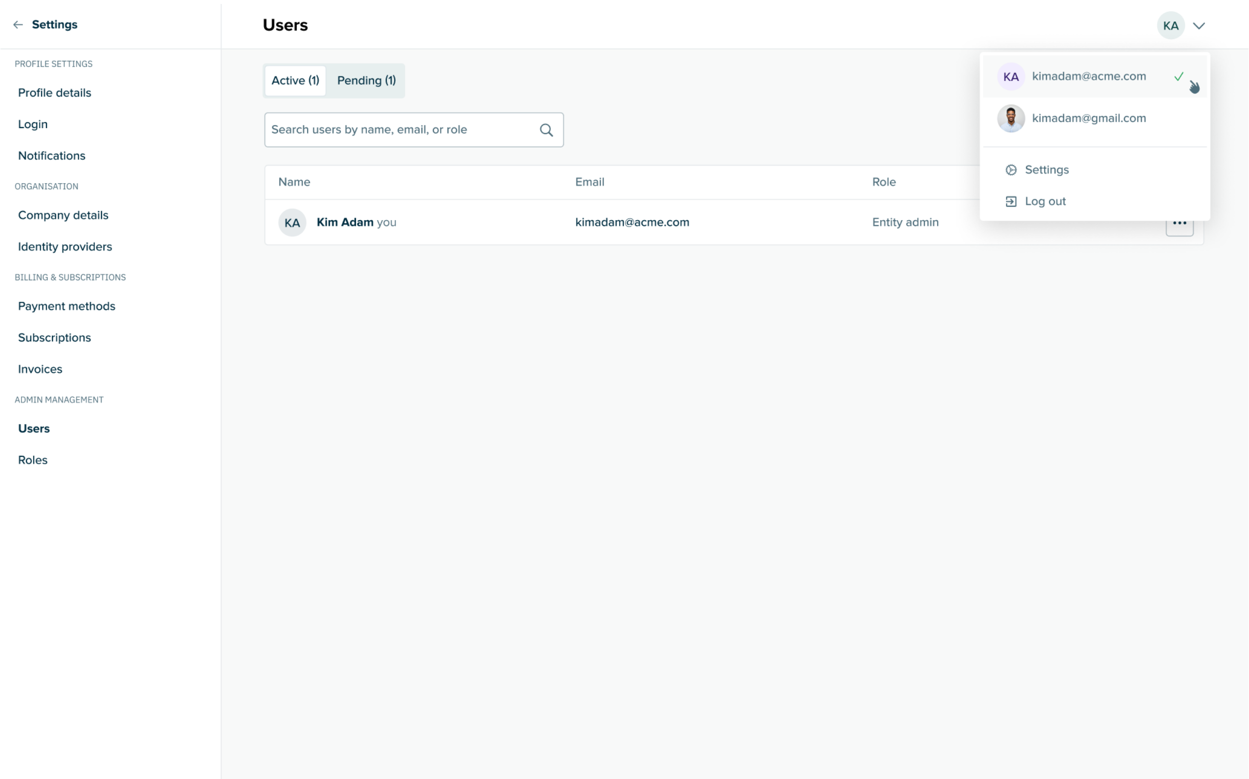
As it often happens in startups, we soon **discovered new tech constraints**: it turned out that the way our backend was set up made it **impossible for invited users to just join existing accounts** without creating their personal account 1st. Otherwise, users won’t exist in our system.

This meant they are required to have several logins - one for personal use, and the others to access their organisation.

This wasn’t ideal UX. But the deadline was set and there was nothing we could do but to require all invited users to create their personal accounts using corporate email address and go thought identity verification flow. For this release at least before the backend is ready to support our design.

Fortunately, the rest of the design was approved, and I was relieved. One of the key features we left was a context switcher, which due to tech constraints was a mandatory for all corporate account users.

Here is the final UI we released for MVP:

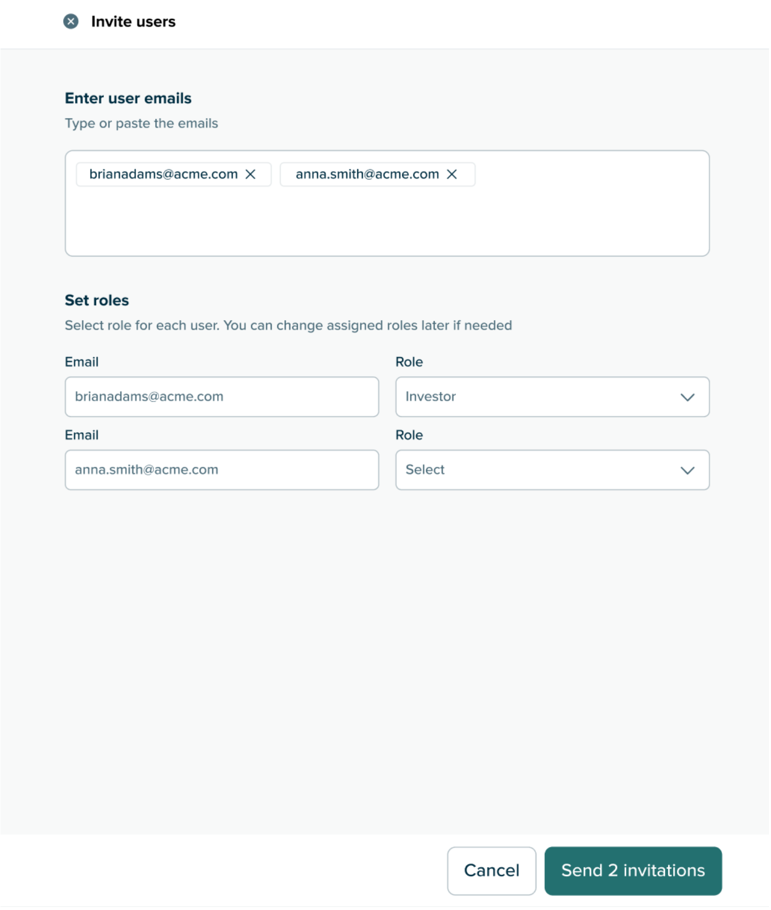


Name	Email	Role
KA Kim Adam you	kimadam@acme.com	Entity admin

Finalised settings page with a context switcher

Even though the initial MVP was far from perfect, corporate users found a lot of utility in User management feature, and for the most part didn’t mind to go over verification process to join corporate account.

Inviting users was a breeze for them, taking only 10 seconds at most.



Invite users

Enter user emails

Type or paste the emails

brianadams@acme.com × anna.smith@acme.com ×

Set roles

Select role for each user. You can change assigned roles later if needed

Email	Role
brianadams@acme.com	Investor
anna.smith@acme.com	Select

Inviting users was a breeze for entity admins

Conclusion

Corporate accounts and user management capability was our 1st big step in our roadmap towards a more professional user base. Plus, it was the perfect opportunity for our team to finally set up role-based permission model which had been a longstanding item on our backlog.

While we received positive feedback from our stakeholders and users, our team had to make difficult decisions due to resource constraints and unexpected priorities. As a result, we shipped only basic profile settings without the full user management feature.

This experience proved how crucial it is to really understand your current capabilities and constraints to avoid any surprises down the line.

All in all, collaborating with our brilliant architects, learning about the foundation of permission systems, helping my team to define the requirements, design, and iterate on this feature was an incredibly rewarding experience.

Looking ahead, we remain committed to materializing the complete user management feature in future releases, as we continue to enhance our platform's functionality and deliver value to our users.