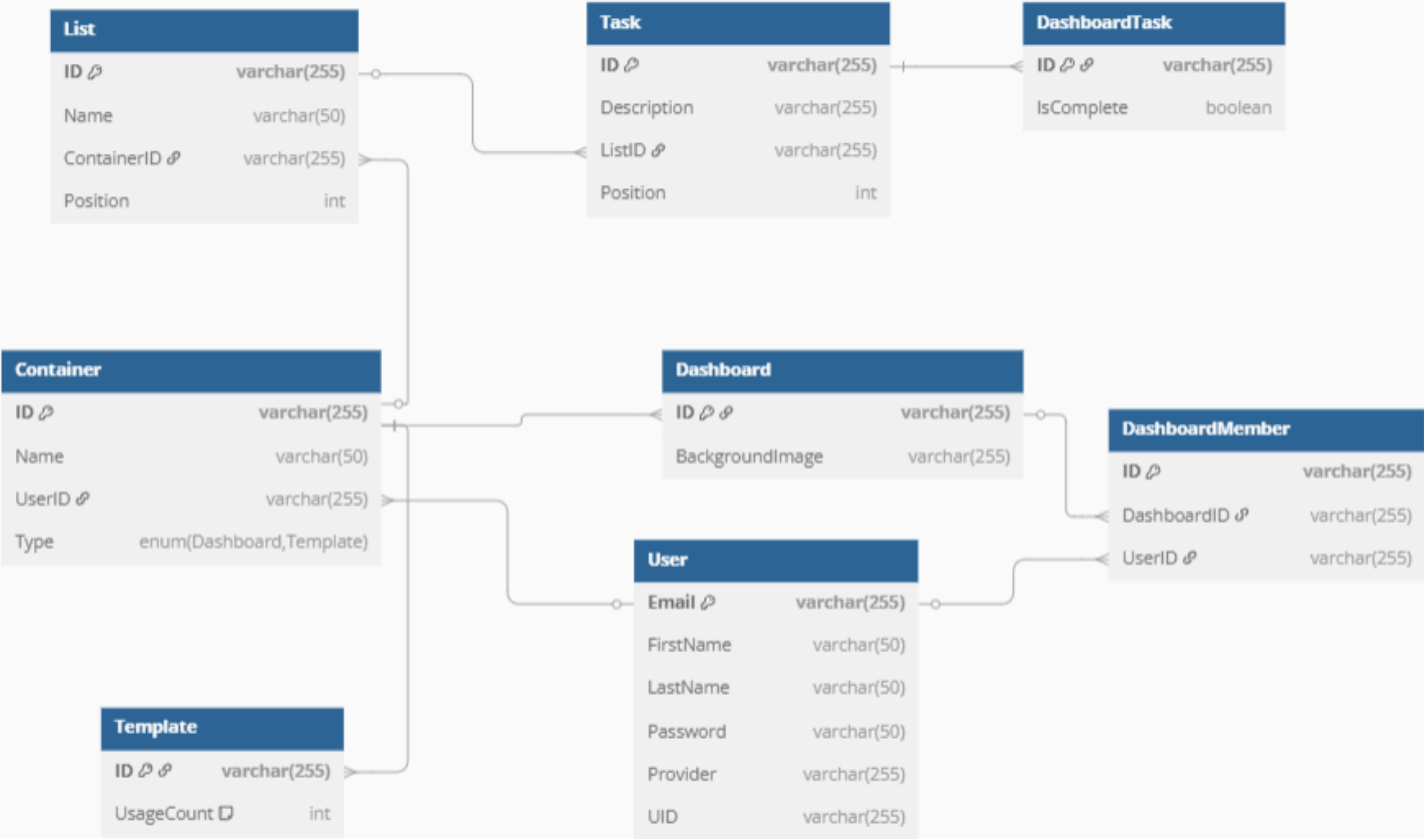


Functional Requirements	MoSCoW Priority
1. The user must be able to create/delete/rename a dashboard.	M
2. The user must be able to create/delete/rename a list inside a dashboard or template.	M
3. The user must be able to create/delete/rename tasks inside a list.	M
4. The user must be able to mark/unmark a task as complete if it belongs to a dashboard.	M
5. The user can create/delete/rename a template.	M
6. The user can create a dashboard out of a template.	M
7. The user can see the most popular templates.	M
8. The user must be able to authorize the platform using an email address and password.	S
9. The user must be able to authorize the platform using Google Authentication .	S
10. The user can reset his password during authorization.	S
11. The dashboard creator could delete other members from the dashboard.	S

12. The user can leave the dashboard that he is a member of.	S
13. The dashboard creator can add other users by email to a dashboard.	S
14. The user can sort the lists and tasks in the dashboard.	S
15. The user can drag tasks between lists.	S
16. The user can search for a template by its name.	S
17. The user can see the last 3 recently viewed dashboards.	S
18. The user can select a background image for his dashboard.	C
19. The system can sanitize/validate user input.	C
20. The user can add a template to their favorites .	C
21. The user can create a workgroup for his teammates to have shared dashboards.	W
22. The user can assign members to a specific task/list.	W
23. The user could copy/paste tasks	W

24.The user can edit his name.	W
25.The user could generate a PDF file from the dashboard.	W

DB Schema - Class Diagram



Dependencies List

Used Gems

1. **Tailwind Css-rails** - Tailwind for styling/grid.
2. **Sassc-rails** - CSS alternative for more comfortable styling.
3. **Hamli** - HTML abstraction alternative for ERB.
4. **html2haml** - converter from HTML to HAML.
5. **Devise** - Authentication with pre-built controllers, views, and routes for common authentication features like login, registration, password reset.
6. **pg** - postgres adapter for database.
7. **dotenv-rails** - loads the .env file automatically.
8. **AASM** - state management inside the models, actually not that needed in my project :)
9. **Omniauth-google-oauth2** - allows authentication via google OAuth.
10. **letter_opener_web** - preview email in the browser instead of having a mail server.
11. **Carrierwave** - file upload handling for images and documents.
12. **mini_magick** - file processing, in my case resizing the images.
13. **Sidekiq** - background job processing, I used for background image resizing.
14. **Redis** - dependency of Sidekiq.
15. **Stimulus-rails** - JS framework designed to work alongside traditional server-side rendering.

Other Used Technologies

1. **TypeScript** - JS with types.
2. **Sortablejs** - library used for grab&sort in JS.
3. **Axios** - library used for simpler requests.

Unused Gems

1. **Capybara** - used for testing user interactions.
2. **Prawn** - used for generating PDF files.
3. **Nokogiri** - web parsing / scraping to access data or to test user interface.
4. **Turbo-rails** - replacement for AJAX, intercepts form-submission etc.

Unused Technologies

1. **AWS** - cloud services for hosting, storage.
2. **Terraform** - tool for managing cloud resources.
3. **ElasticSearch** - search and analytics engine for handling large datasets.
4. **SAML** - single sign-on (SSO) across multiple applications.