DevNotes Pitch:

User stories that helped us decide what problem we were trying to solve:

- As an experienced system administrator I want to log and catalog technical problems and solutions from my experience for future reference.
- As a beginner coder, I don't want the app to be complicated so that I can hit the ground running.
- As a 10x developer, I want to see all outstanding issues on all branches so that I can do all the work.
- As a beginner frontend designer, I want to have an archive of my project' designs stored in one place so I can see and show my progress to my boss
- As a software developer, I want all features to work as expected
- As a tester, I want to make sure that there will no bugs within the application
- As a UI/UX designer, I want a built-in figma tool as well as a chat gpt based assistant to suggest ideas, color schemes, brands, etc.
- As a backend/frontend developer, a large chunk of whose work involves code, I want to be able to put code snippets that are formatted correctly in the journal and maybe be able to link some line of code so that I can keep track of what bugs I've made in code I'm working on or what I'm doing right
- As as old person named David, I want to have accessibility features (e.g. bigger text) so that I can see.
- As a beginner programmer, I would like some simple documentation for the journal that explains different features that could be useful for me as I am new to tools like this
- As a no-life code monkey who codes at 3 am named Bobby, I want dark/light mode options so that I can protect my eyes.
- As a 5 year old kid who has some experience in front-end development, I think it would be nice to be able to visualize my figma designs in my journal and possibly any FigJam boards that I work on with my dad on the weekends
- As a Database manager, I want a high level view of my SQL connections and the data being posted and requested
- As a busy software engineer, I want keyboard shortcuts so I can save time by not using my mouse.

Scope & Problem being solved

From these User Stories, we narrowed down to our scope of a developer working on multiple projects at a company. We figured that if a developer is at a company and wants to show their boss how much work they accomplished, they need to be able to create organized information based on 4 categories that act as objects/views inside of our app: **meetings, general freeform markdown notes, designs and image artifacts, and tasks worked on**.

How a developer would need the journal organized to show their boss based on a developer's workflow

We decided a developer would like this information to be organized into projects. This organization by project is essential for proper User-Centered Design because the application mimics how a developer works within their job. Organizing the journal in this way leads to easily identifiable work the developer did that they can show their boss. The boss will be familiar with what projects the company did and would most likely have access to what projects the developer worked on.

Keeping the journal organized and easily searchable

From this, the notes and information a developer can input into our app is organized and made more accessible with the introduction of "tags", which are common to the 4 categories of information. There are tags for date, importance, difficulty, and name. The end user should be able to search for their journal entries by these tags, ensuring an easy-to-use app flow.

We also recognized that a lot of what a developer does and produces is connected to other things, and cannot exist alone in isolation for the full picture to be shown. For example, tasks and GitHub issues that a developer might work on could be connected to a meeting! To handle this, we decided a developer on our app should be able to link a task to a specific meeting – showing their boss how they attended meetings and what tasks got done as a result of that meeting.

Design of the app:

Potential risks

We identified the risk of our app being too hard to use and too unfamiliar. We knew there was a major risk in this. If our app was too unfamiliar and abstract to use, nobody would bother using it because it would have a big learning curve! To remedy this risk, we decided to use the 99% percent rule for creating layouts and designs

Making the app familiar UI-Wise

Since most people spend most of their time using other UIs 99% of the time, we knew that our app needed to have a familiar layout and design. And since developers use common tools such as Slack, Discord, and Obsidian, we decided to take inspiration from how their UI works and is organized. For example, we think developers would be familiar with how Discord organizes channels based on servers on the left bars. So to tap into this "ui memory" a developer would likely already have, we decided to organize journal entries based on projects on the left bars of our app.

Layout of app and the content in each view:

Main portion of app has 5 different views it can take on:

-The 4 "objects" a dev can make
-Search result on tags that show objects with the searched tags

-We may reduce these 4 "views" to a single view of markdown notes with a template for each "view" instead.

How the journal entries is organized by views and layout:

Group journal by projects with following categories/objects:

-Name, date, title, tag, connection/link for each journal entry (all)

- **Meetings with notes**
  - Name for meeting
  - Meeting Minute/standup notes sent out to team for meeting in markdown
  - At the bottom there's a section for developer to add their own notes in markdown

- **General markdown notes**
  - Used mainly for code snippets and bug fixes for major bugs that they fix
  - Assign a date to each note
  - Name for note
  - Markdown editor
  - Save
  - Edit
- **Design artifacts worked on / design notes**
  - They can add images that you can view
  - embeds of figma (if possible)
- **Tasks (tasks assigned, and a section to say who you worked on it with)**
  - Meeting Notes:
  - link a task with multiple meeting notes from other section above
  -Technical aspects of task:
  - Issue/task name
  - Issue description/spec
  - Anyone you worked on this with
  - IF POSSIBLE, USE GITHUB REST API TO JUST EMBED A GITHUB ISSUE INSTEAD OF ADDING ALL OF THE ABOVE (https://docs.github.com/en/rest/about-the-rest-api/about-the-rest-api?apiVersion=2022-11-28)
  - Image of delivered task to show it was completed

When a user searches by tag, show a list of the objects associated with that tag, but organize them based on what kind of object they are (like task, meeting note, etc)

-Search bar at top utilizes the wide main portion of our app, the multi-use page

Tag system:

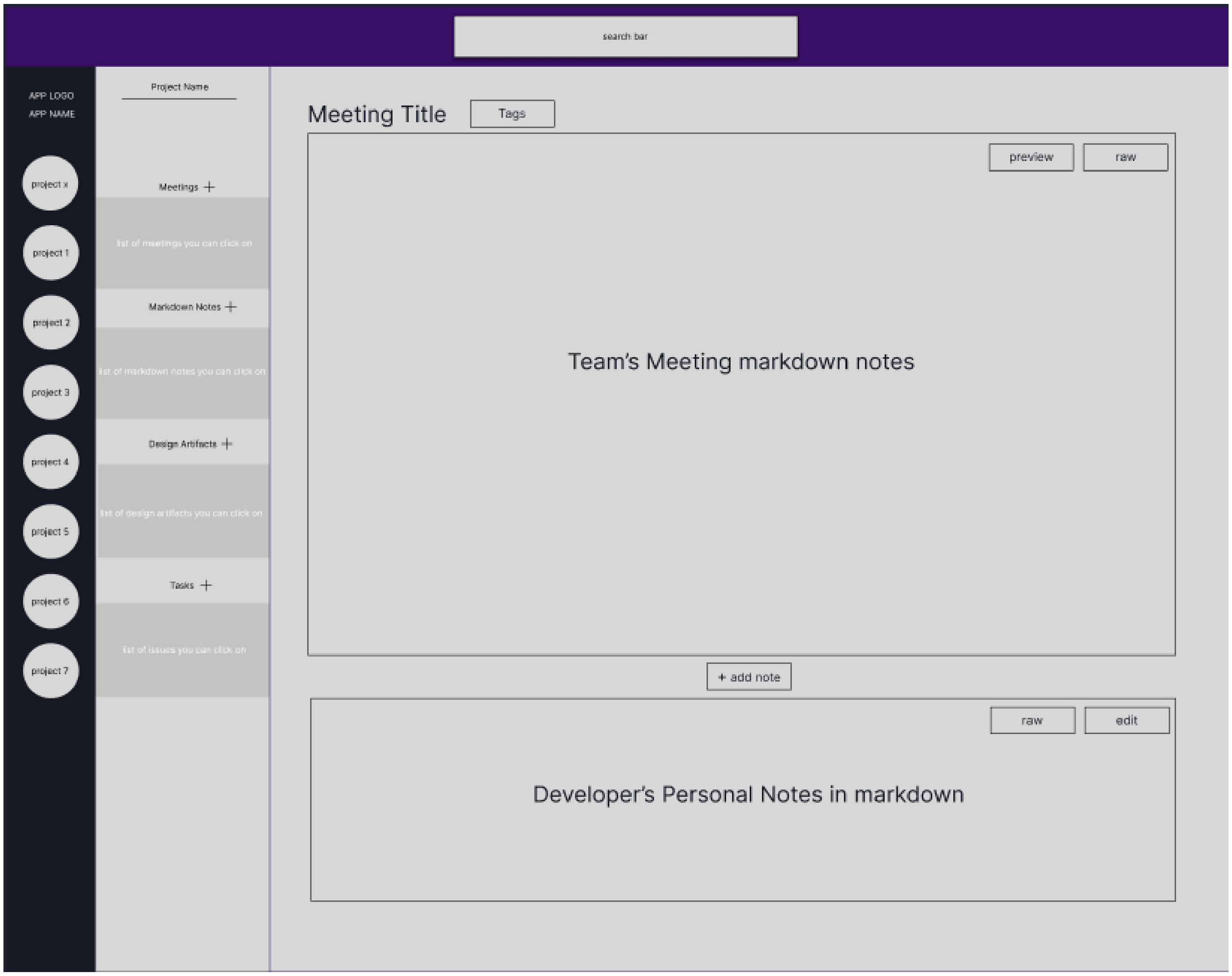
Common tags: date, importance, difficulty, feature, bug, title

All searchable

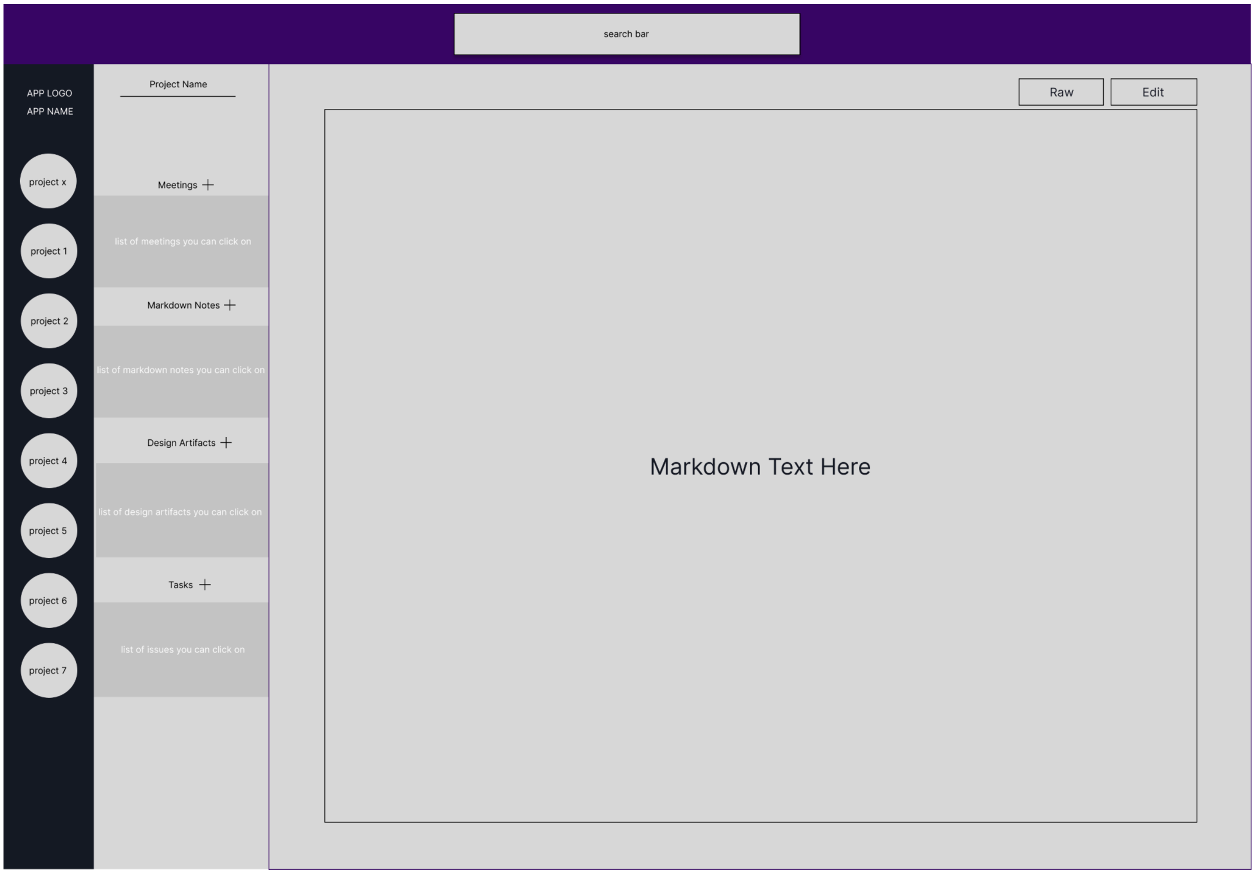
Wireframes

Search

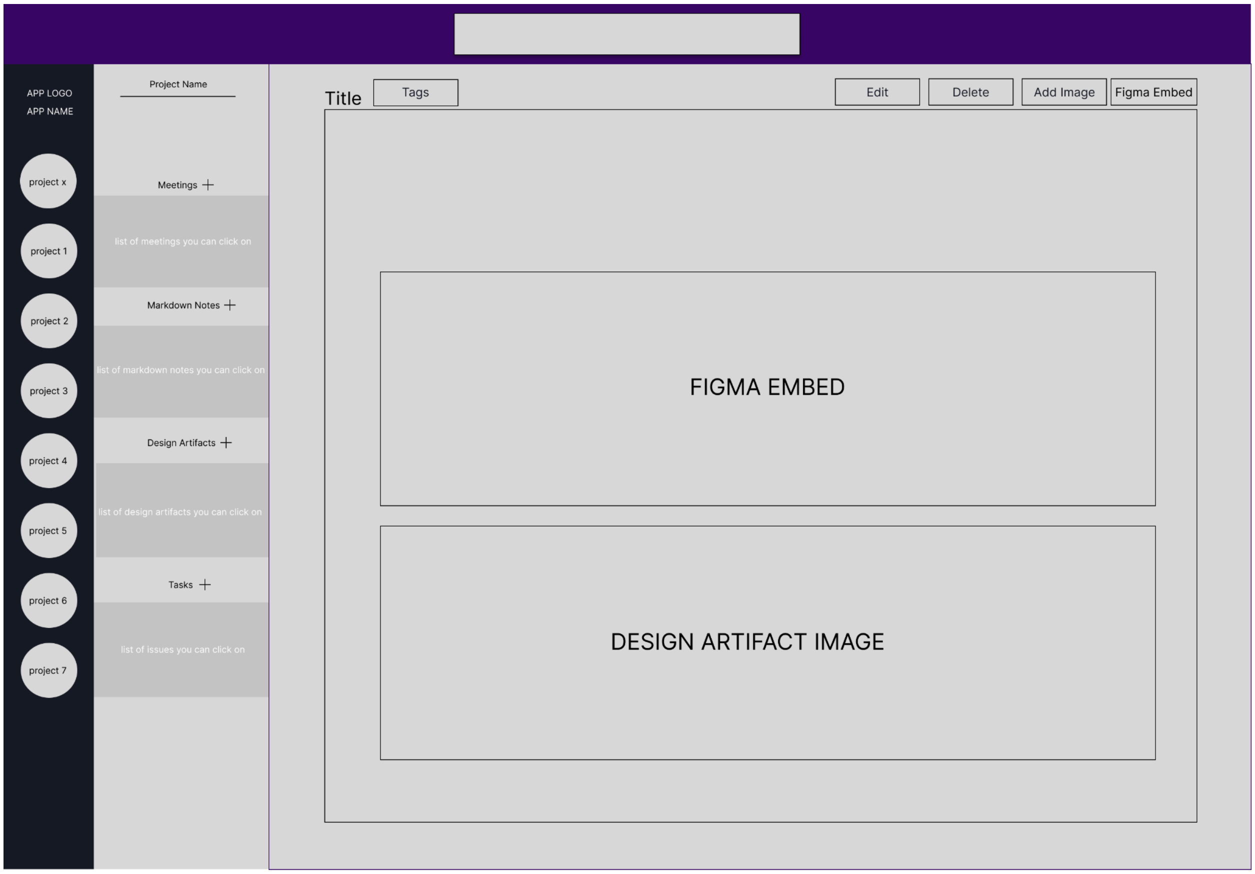
Meetings


search bar
APP LOGO
APP NAME
project x
project 1
project 2
project 3
project 4
project 5
project 6
project 7
Project Name
Meetings +
list of meetings you can click on
Markdown Notes +
list of markdown notes you can click on
Design Artifacts +
list of design artifacts you can click on
Tasks +
list of issues you can click on
Meeting Title
Tags
preview
raw
Team's Meeting markdown notes
+ add note
raw
edit
Developer's Personal Notes in markdown

Markdown Notes



Design Artifacts



Tasks

Design choices for:

“Meeting Entry”

“Markdown Entry”

“Design Artifact Entry”

“Task Entry”