Final Design Report

Uninitialized Local Team
Kaveh Buenaventura, Angela Ferro Capera, Jason Dekema, Ian Dudder, Kez May, Hailey Schauman

Table of Contents

Team Roles

Our team managed two co-managers and individual contributors. Each team member contributed with meaningful work, regardless if they were co-manager or not.

Co-managers were responsible for meeting agendas, distributing tasks, taking meeting minutes, checking on teammates, and managing the team's kanban board.

Individual contributors were responsible for finishing tasks on time, submitting assignments, and helping other team members.

Team Member	**Role**
Ian Dudder	Co-Manager
Kez May	Individual Contributor
Hailey Schauman	Individual Contributor
Kaveh Buenaventura	Individual Contributor
Jason Dekema	Individual Contributor
Angela Ferro Capera	Co-Manager

Summary Briefing

The purpose of this report is to provide the reader with information on what inspired us to design our application, the methods we used to include the user in the design process, the resulting requirements, the information architecture, and tools used to create the prototype.

Working and studying from home has become more popular due to the pandemic. Thanks to the various online tools and software, people can be as productive as if they were working/studying at the office/campus. People have become more familiar with utilizing software like Zoom, Slack, Miro, GitHub, Google Drive, and more. As helpful as these tools can be, managing so many of them can overwhelm the user. It also can lead to exhaustion from switching from one application or tab to another. This inspired us to design Divvy, a collaborative, all-in-one, extensible tool. Divvy has everything the user needs for their projects in a single, customizable place.

Divvy is designed for any person wanting to collaborate on a team. However, due to time constraints, we focused on students at the University of Washington Bothell (UWB) as our primary audience. As current students, we understand how painful it is to manage 10+ different applications to work on assignments. We can relate to our primary audience on their main pain points. Also, it was easier for us to ask other students to test our prototype and to get a response in time. Our secondary audience was professors at UWB.

Our team gained a much better understanding of what user-centered design is by using personas, user interviews, journey maps, and usability tests through the design process. These methods helped us to empathize and get to know the user better.

For our design process, we started by defining the project. Then, we got to know the user and their needs with generative research, helping us formulate the design. Once we had the design, we were able to test it, gather user feedback, and then implement their feedback to improve the design of our product. Including the user in the design process turned out to be beneficial for the user and our team.

Hunt Statement

Understand the tools that students use for collaboration in order to identify the most popular features which can then be prioritized and included in an extensible software program.

Requirements Specification

Through our research, experience, and user interviews, we developed a long list of candidate requirements. Through discussion, they were boiled down to the following key requirements.

Functional Requirements

1. Creating Groups
    - The creator shall be able to manually assign team members into groups.
    - The creator shall be able to randomly assign team members into groups.
    - The creator shall be able to message team members and groups directly and list the specific requirements for each assignment in their course/s.
    - Creators shall be able to distribute class resources to each group.
2. Shared Cloud File Management
    - The real-time collaboration tool shall be able to show version control history.
    - The shared storage shall allow named directories to organize the files.
    - The team member shall be able to perform CRUD actions on the real-time collaboration files.
    - The owner of a file shall get to choose which team members get to edit their files.
    - Team members shall be able to live collaborate to edit files including:
        - Documents
        - Sheets

        - Drawings
        - Etc.
    - The application shall allow an annotation feature for stored images and a comment feature for stored documents.
    - The application shall show where the cursors are for every user active in a live-collaboration document.
    - The application shall highlight syntax for any program stored in the shared cloud.
3. Task Management and Visualizations
    - Team managers and team members should be able to visualize task progress in the task management section.
    - A task on the kanban board shall be linked to the assignment description or requirements so that it is easier to understand the task requirements.
    - Team managers and team members must be able to
        - Update their progress on the task.
        - Perform CRUD operations on task cards.
        - Assign tasks to team members
        - Perform CRUD operations to columns of a task board
    - Team members shall view task backlog in a configurable way as
        - To-do list
        - Kanban
        - MoSCoW
        - Etc.
    - The application shall be allow to filter task by
        - Filter by assignee
        - Filter by milestone
        - Filter by project
        - Filter by label
    - The application shall make use of graphs to represent the progress of
        - Individuals on assigned tasks
        - Milestones

        - Overall project
4. User Project Deliverables
    - The application shall allow team members to submit their assignments to the application.
    - The application shall allow assignments to be linked to group folders containing the necessary contents of the assignment.
    - The application shall allow assignment requirements to be linked to files or to highlighted text-blocks.
    - When a team member is on the submission page for an assignment
        - The application shall suggest the linked files/folder for submission.
        - The application shall display the assignment requirements on the same page.
        - When a team member clicks/taps a requirement, the application shall display a preview of the file containing the section that fulfills the requirement
5. Delighters
    - The application shall have a discussion board for all-team communications.
    - The application shall allow new functionality to be added on the extensions page.
    - The application shall allow a voice-call server for members of the group to join.

Non-Functional Requirements

1. The application shall be compatible with
    - Windows,
    - Linux,
    - MacOS,
    - Android, and
    - iOS.
2. Team members shall be able to upload and download documents of any file type to a shared cloud storage space of 15 gigabytes per shared space.
3. The shared storage shall not limit the size of an individual file upload unless it exceeds the total storage space

4. The live collaboration must be highly responsive, and reflect any changes a user makes to all other users within 5 seconds.
5. Files must not lag given more objects in a file
6. The live collaboration feature must display where active users' cursors are within 3 seconds of them moving their cursor.
7. The application shall allow the creator of the shared space to upload project requirements and resources of up to 2 gigabytes.
8. The application shall scale out as needed (depending on traffic)
9. The application shall guarantee 99.8% of availability
10. The application shall provide users with the following detailed error messages:
    - Incorrect login info.
    - Course sections that are not available.
    - Incorrect format of files trying to be submitted.
11. The UI and clickable elements on the application should be incredibly responsive with 1 second or less of delay
12. The application must provide adequate security features including log-in authentication, editing authorization, password recoverability/changing, data confidentiality, and extension moderation/screening
13. The shared cloud storage must have reliable data preservation.
14. The application must be open source to allow third parties to create and add new functionality to the extensions page.
15. The voice-call service must preserve high quality sound transmission with minimal static and no more than a quarter-second of delay.
16. The application should be easily scalable to account for new extensions and features.

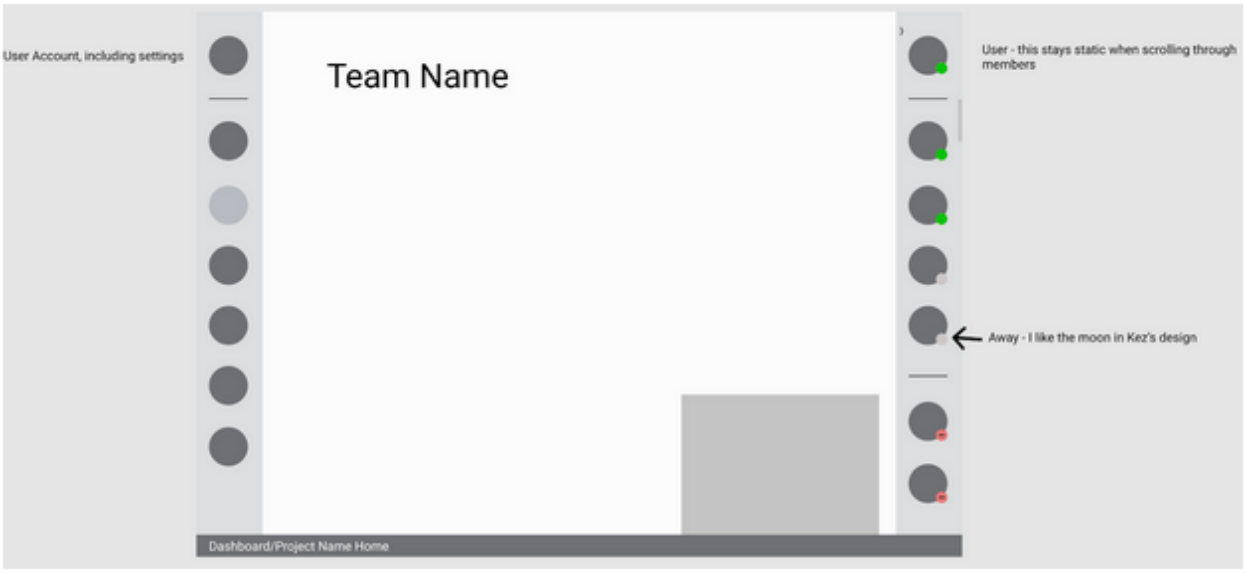
Early Sketches

As we gathered and refined our requirements, we created some sketches to help us visualize how the UI might look and behave. We sought inspiration from Discord, Canvas, Google Drive, and VSCode to make the UI intuitive to many students.

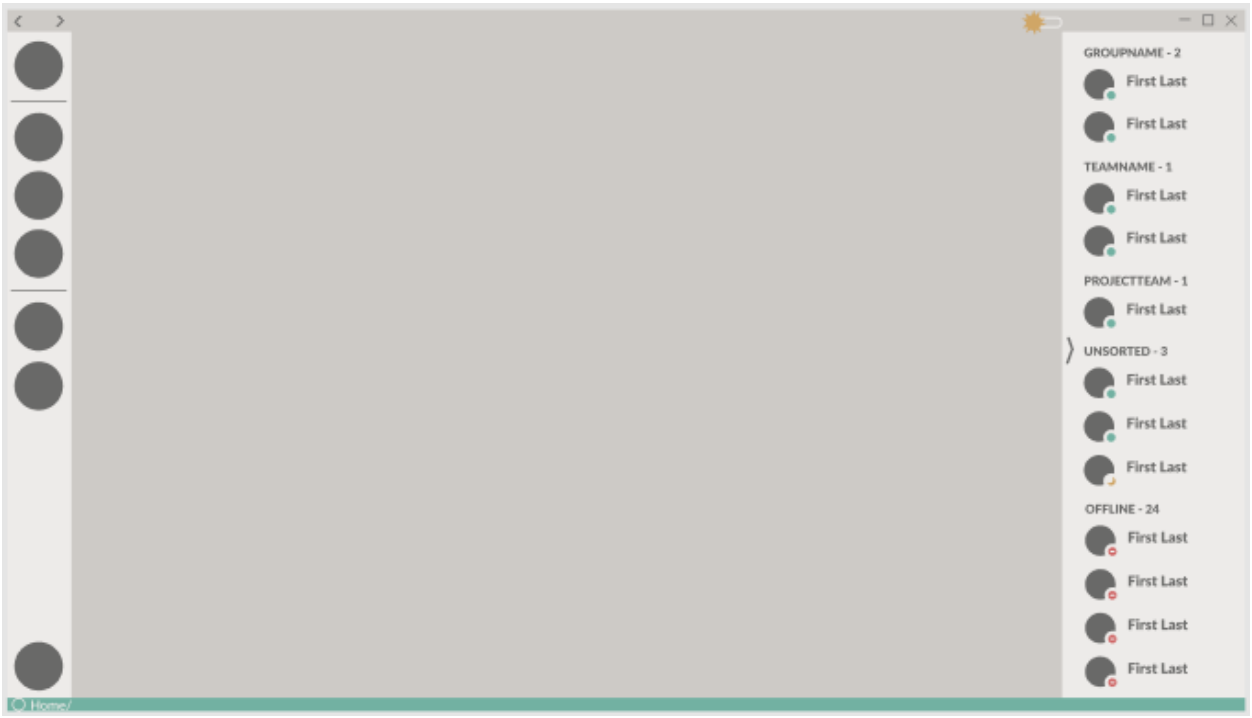
Our first iteration of this process had many of us designing our own sketches of potential layouts for the final prototype.

Some of the sketches were more focused on the base layer of the UI that would be consistent throughout the application.

Initial Sketch 1: layout focused sketch

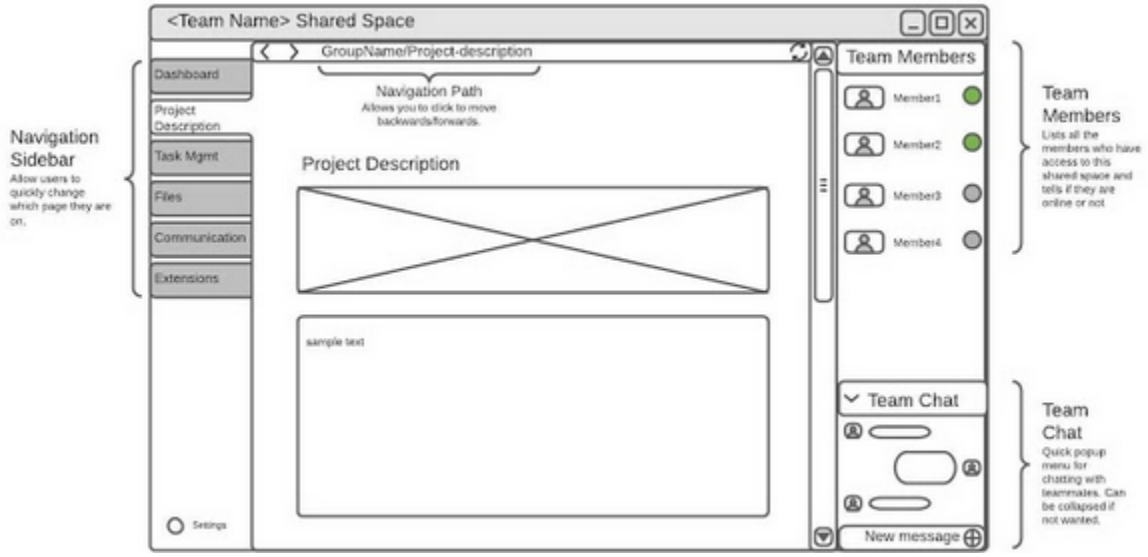


Initial Sketch 2: layout focused sketch and base for final prototype

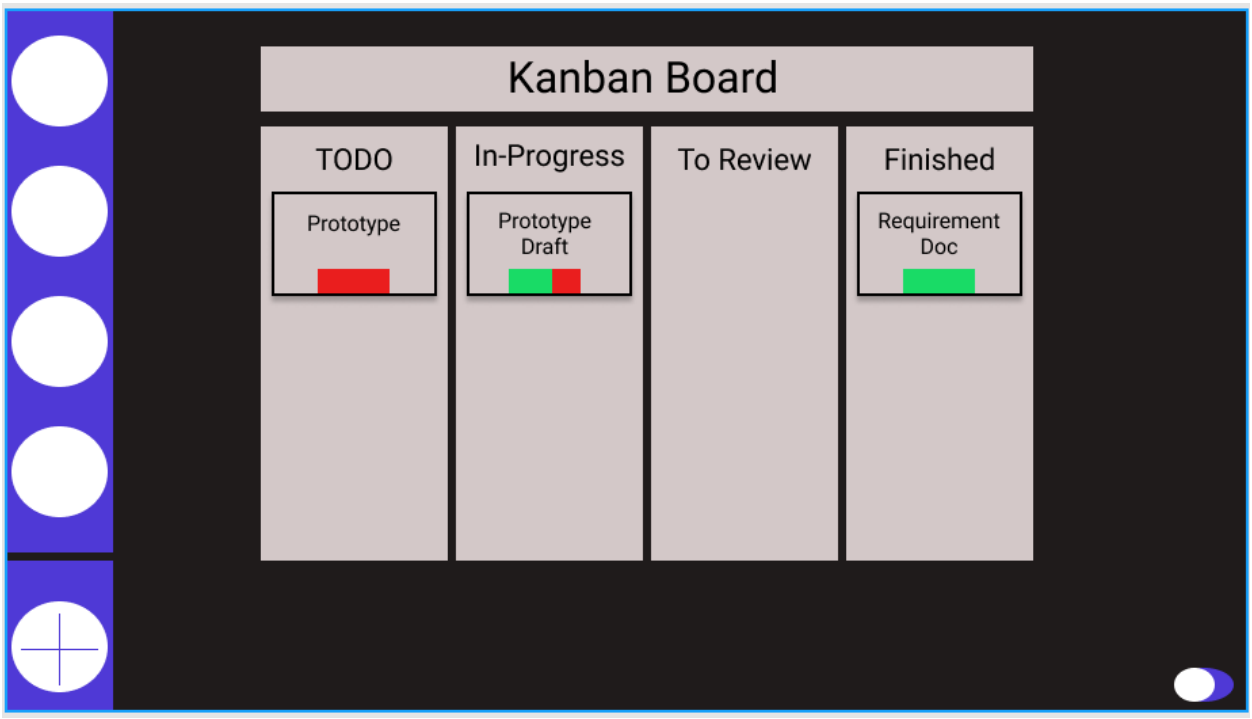


Others did the same Base layer but also tried to consider how the requirements might fit into the sketch. They looked at specific functionality and worked to show how that might be displayed in the final prototype.

Initial Sketch 3: layout and content focused

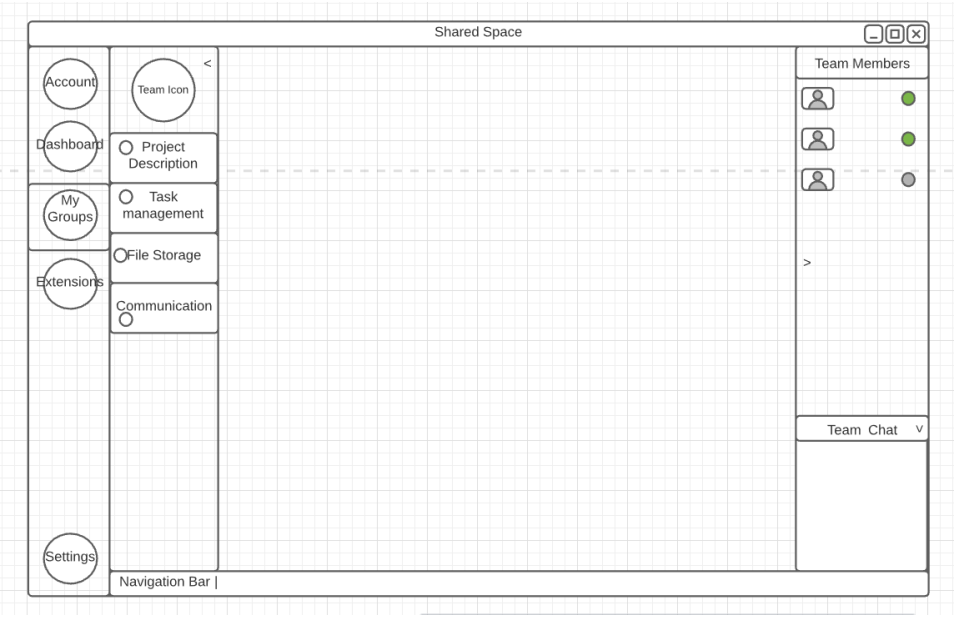


Initial Sketch 4: layout and content focused



Once we had looked through all the sketches, we broke them each down into their core elements and decided which ones were most important. Then, we decided to base our final prototype on one of the sketches that was made, and created a new sketch that combined all the elements we wanted included. This was used as the reference sketch as we developed the final prototype.

Design Layout Sketch: the synthesised sketch to be used in final prototype



After the prototype layout was determined, we turned our attention to the color scheme and fonts. We listed several fonts and color combinations and held a vote to determine the final design palette.

Design Decisions: font and color palette voting form

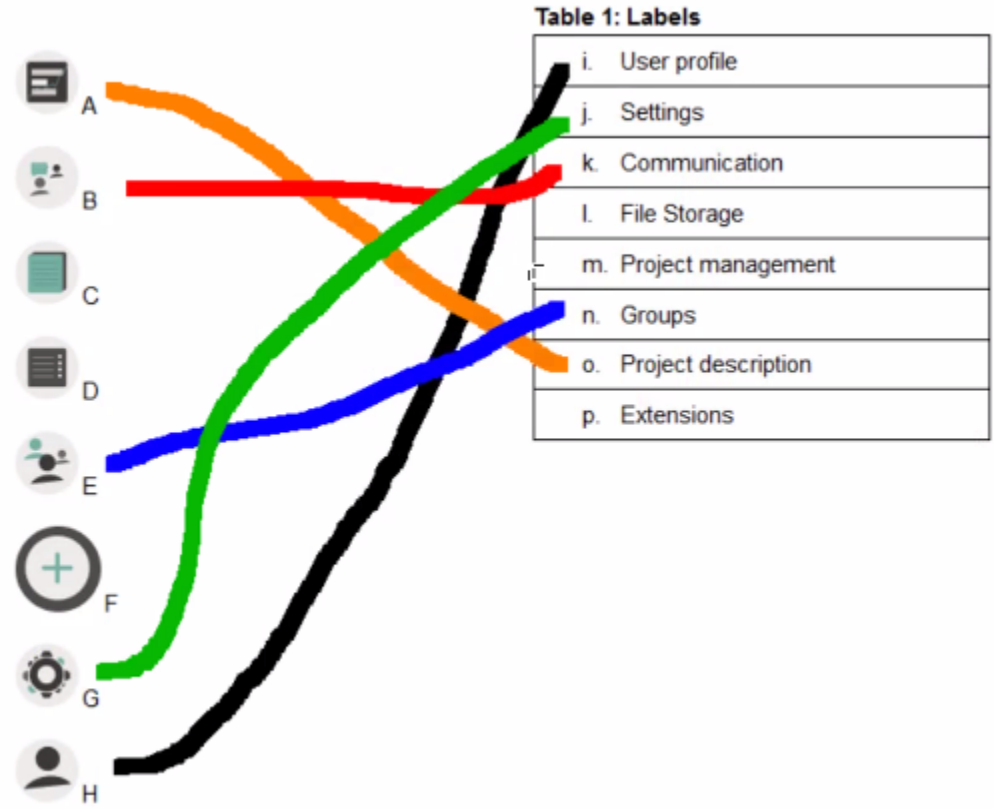
Arial	
Cambria	
Droid Sans	
Merriweather	
Montserrat	||||
Nunito	|||||
Oswald	
Roboto	||
Spectral	
Verdana	|
Alegreya	
Helvetica Neue	|||
Lato	|

#E6E6E9	Example	
Ian's second choice:		
#1C1526	**Example**	
#6C4D80	**Example**	
#BEBDC1	Example	
Kez:		
#3A3938	**Example**	
#72B1A2	**Example**	
#4281A4	**Example**	
#1E8EB1	**Example**	
#4E4C4B	**Example**	
#696968	**Example**	
#CDCAC6	Example	
#EDEBE9	Example	

Once we agreed on our design layout and color scheme, we set out to choose our icons. Initially, we planned to use royalty free icons from a website, but in the end, we decided to design them ourselves instead. We made a list of all the icons we thought we would need. Then, we created our first batch of icons.

When our first iteration of icons was ready, we had an icon usability test. We showed users the icons on their own and asked them what words came to mind when they saw the icon. After collecting this data, we showed them the same icons next to a list of descriptions. We asked them to match the icon to its best description. We told them that the relationships could be many to many.

Icon Study: the result of one of our icon usability studies



The icon study revealed some important blindspots with our icons. There were several ambiguous icons that the users were clueless on how to evaluate. This helped guide our second iteration and even helped us restructure the organization of the UI.

Storyboard

Synthesized Storyboard

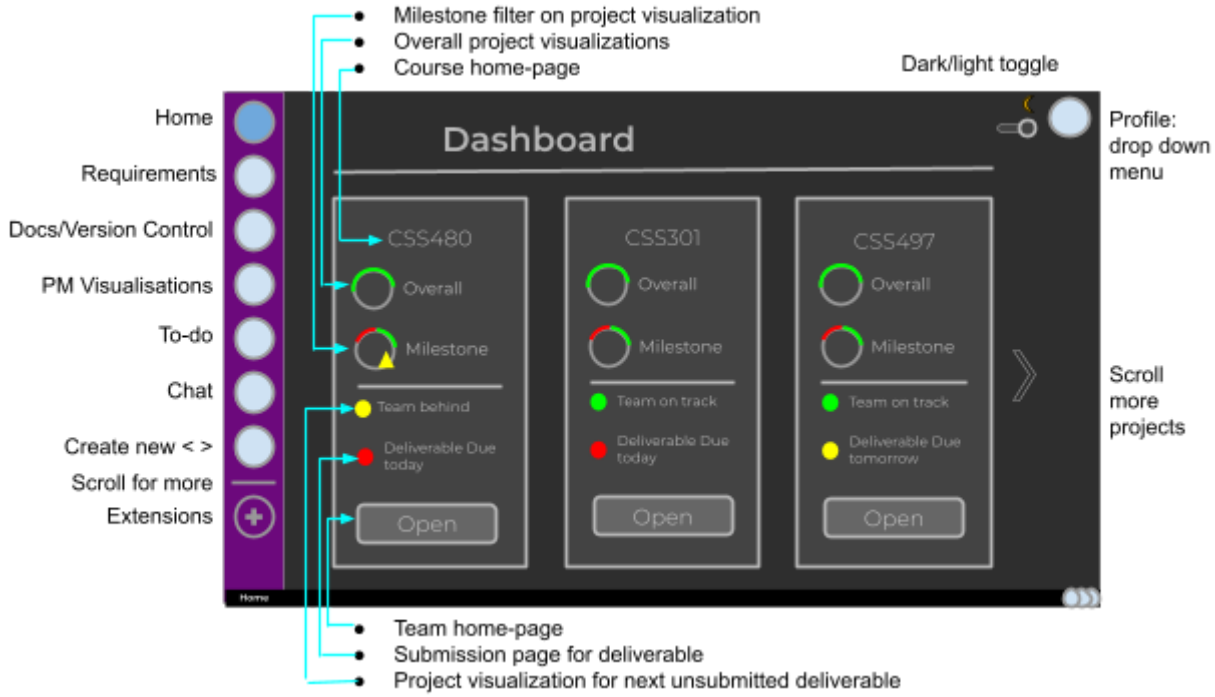
Scenario: Rylan is a sophomore in the CSSE program at the University of Washington Bothell campus. Rylan's courses often involve working on group projects, and, as a natural leader, she often ends up being the group manager.

Checking on her group projects, Rylan notices one of her group projects has a deliverable due today. After clicking on the deliverable, Rylan is taken to the team's kanban board. Once in the kanban board, she is able to see the different completed tasks which are ready to be submitted.

Having this detailed list of completed tasks helps Rylan feel less overwhelmed by the upcoming deliverable. After clicking on the “Submit” button, she is taken to the project’s page which lists all the project’s deliverables. Each deliverable has some required files, which are located in the team’s shared filed storage. Since our app has all the resources linked in one single place, it only takes Rylan a single click to submit the required files. Rylan really appreciates how easy and quick it is to upload the required files.

After submitting the files, she is taken back to the list of assignments where she can see the latest submitted deliverables.

Rylan’s student dashboard



Rylan, the team leader, looks at her dashboard for her group projects. She is able to see visual indicators of project progress, team progress (whether they are completing tasks in a timely manner), and the nearest deliverable due date.

Rylan notices that there is a deliverable due on that day

Rylan hovers over the "Deliverable Due today" element and it becomes highlighted indicating that it is clickable. She clicks and is redirected to the team's kanban board for that project.

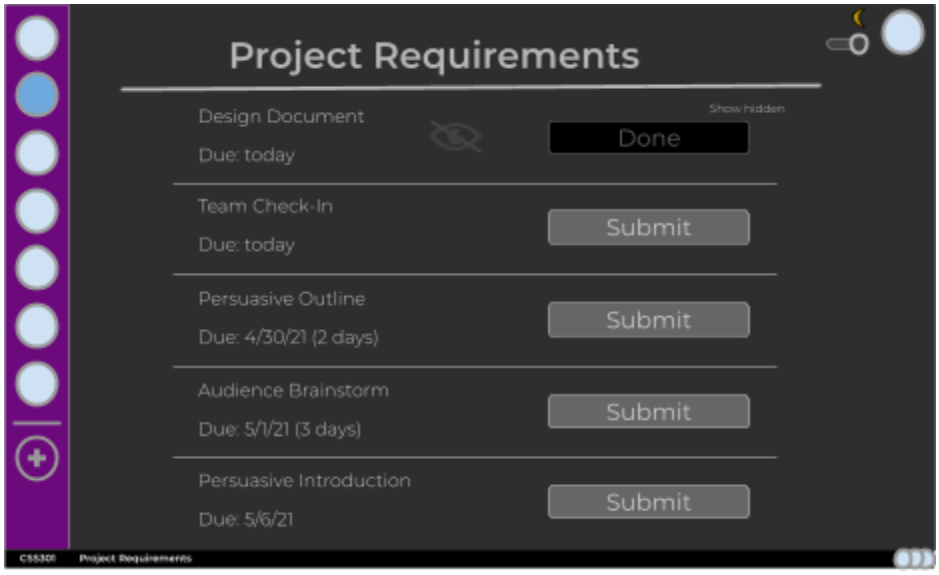
Rylan's team Kanban Board showing task completion progress

Checking her team's Kanban Board, she notices that everything has been reviewed and the deliverable is ready to submit. She clicks on the deliverable that is ready to submit and is redirected to the submission page for that deliverable.

The deliverables submission page

On the submission page, Rylan can view the specific requirements for this deliverable and because the file she needs to submit is already stored in the shared file storage, she only needs to click submit.

Project tasks/requirements page



After Rylan submits the deliverable, she is redirected to the project requirements and tasks page. Here she is able to see that the deliverable that was just submitted is marked as done. She is also able to see future task due dates and is able to submit them if they are completed.

Additional Storyboard

Persona: Sarah Diaz

Scenario: Sarah is a Junior student at the University of Washington Bothell. She enjoys team-based projects because she sees them as an opportunity to work with old and new friends

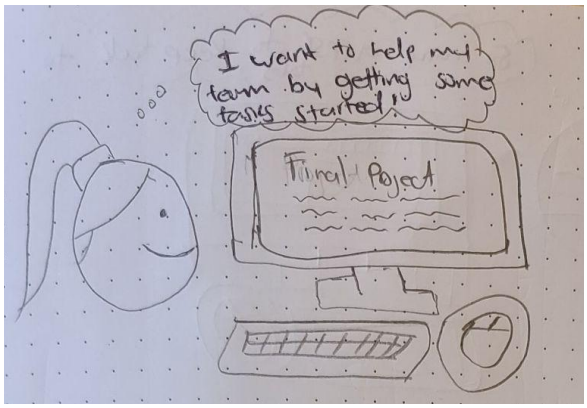
in a common interest. In previous projects, she has been an individual contributor, and she is interested in improving her leadership and task management skills.

This quarter, Sarah is taking a class that requires to work with the same team throughout the quarter on a big project. She does not know anyone in her class, but she is excited (and a little nervous) to work with new people. Trying to be more proactive, she tried breaking all the project requirements into smaller tasks, and adding them to a kanban board. She also wanted to link the kanban board to a folder containing all the team's resources. However, the current application utilized by her school does not list the project requirements clearly, causing her to feel frustrated and a little bit discouraged from trying to be proactive. Also, she feels a little overwhelmed by all the different applications she has to use for this single group project.

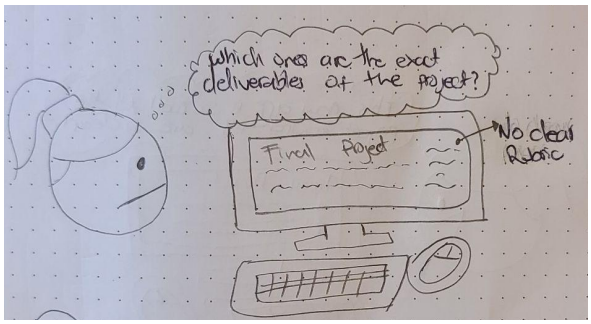
Later, Sarah found <Our App>. This application allowed her to see the project requirements in a detailed list. She also found out our app has a kanban board and file sharing feature available. After finding out about our app features, she felt inspired and encouraged to get those tasks done for her team. Now, she can relax knowing that she was able to be helpful, and she and her teammates will be able to find anything related to their group project in a single place.

Storyboard

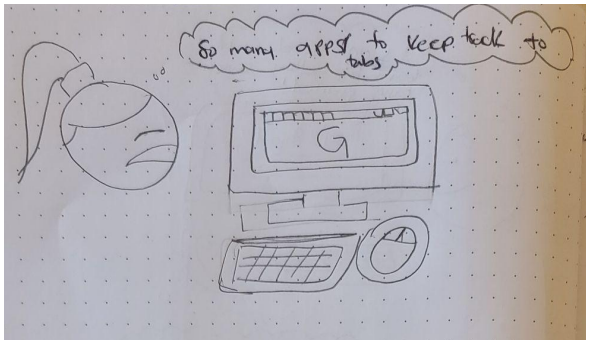
1. Sarah wants to gather the project requirements



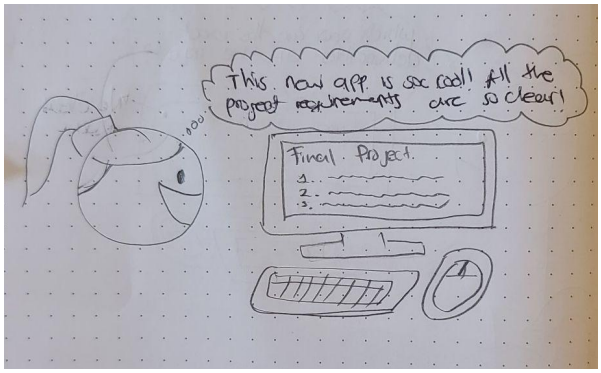
2. Sarah gets frustrated by the lack of clarity on the project requirements



3. Sarah feels overwhelmed by the number of apps she has to keep track of for a single project

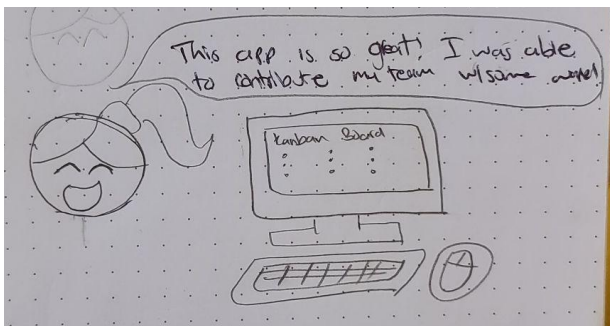


4. Sarah finds our app and see how clear the requirements for the final project are displayed



5. Sarah finds out the kanban board and share files features, she feels motivated to get her team's kanban board and shared folder started

6. Sarah feels happy now because she was able to help her team with getting all the project deliverables broken into tasks, placed into a kanban board, and linked the board with their shared folder

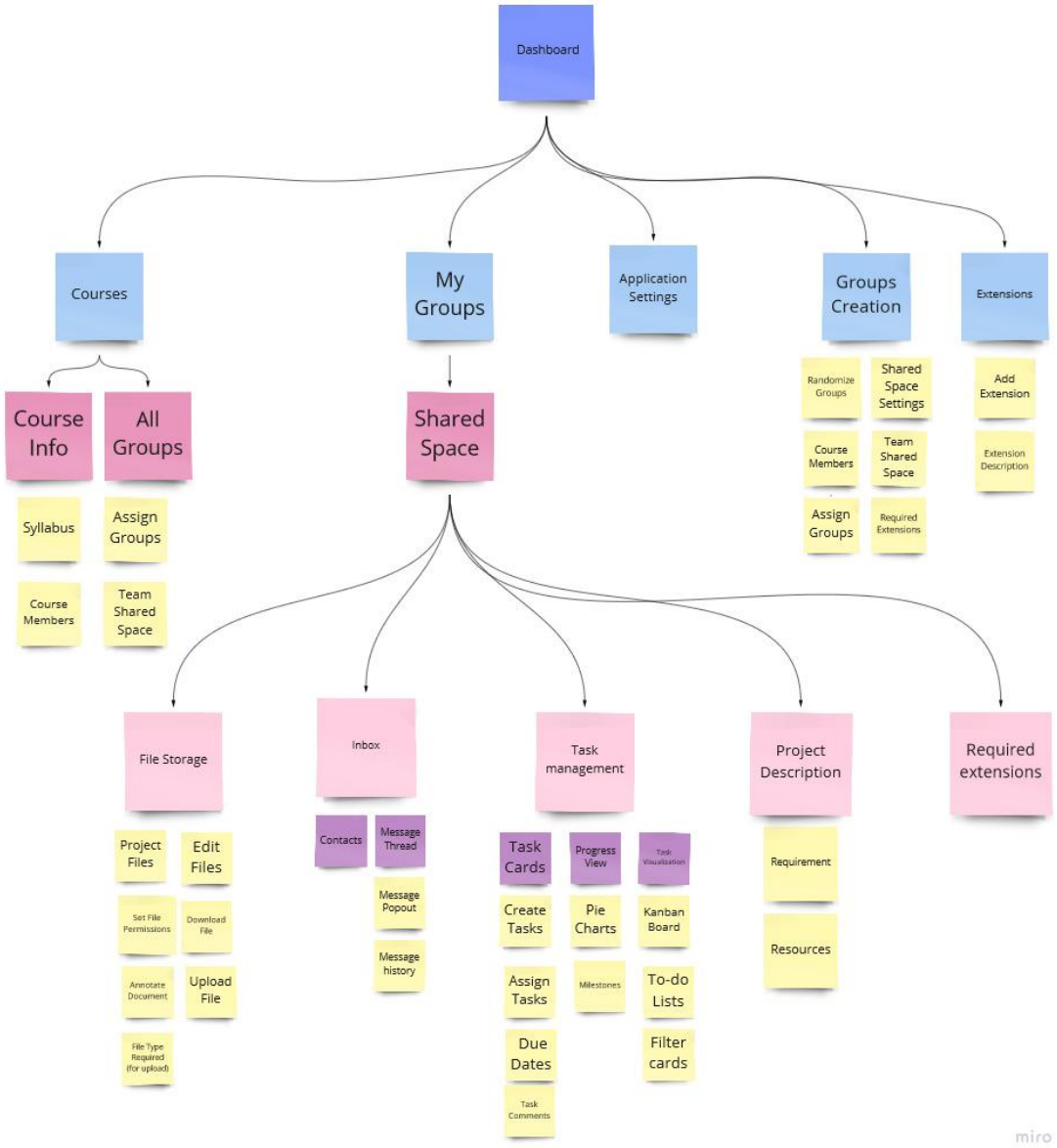


Architecture

Information Architecture

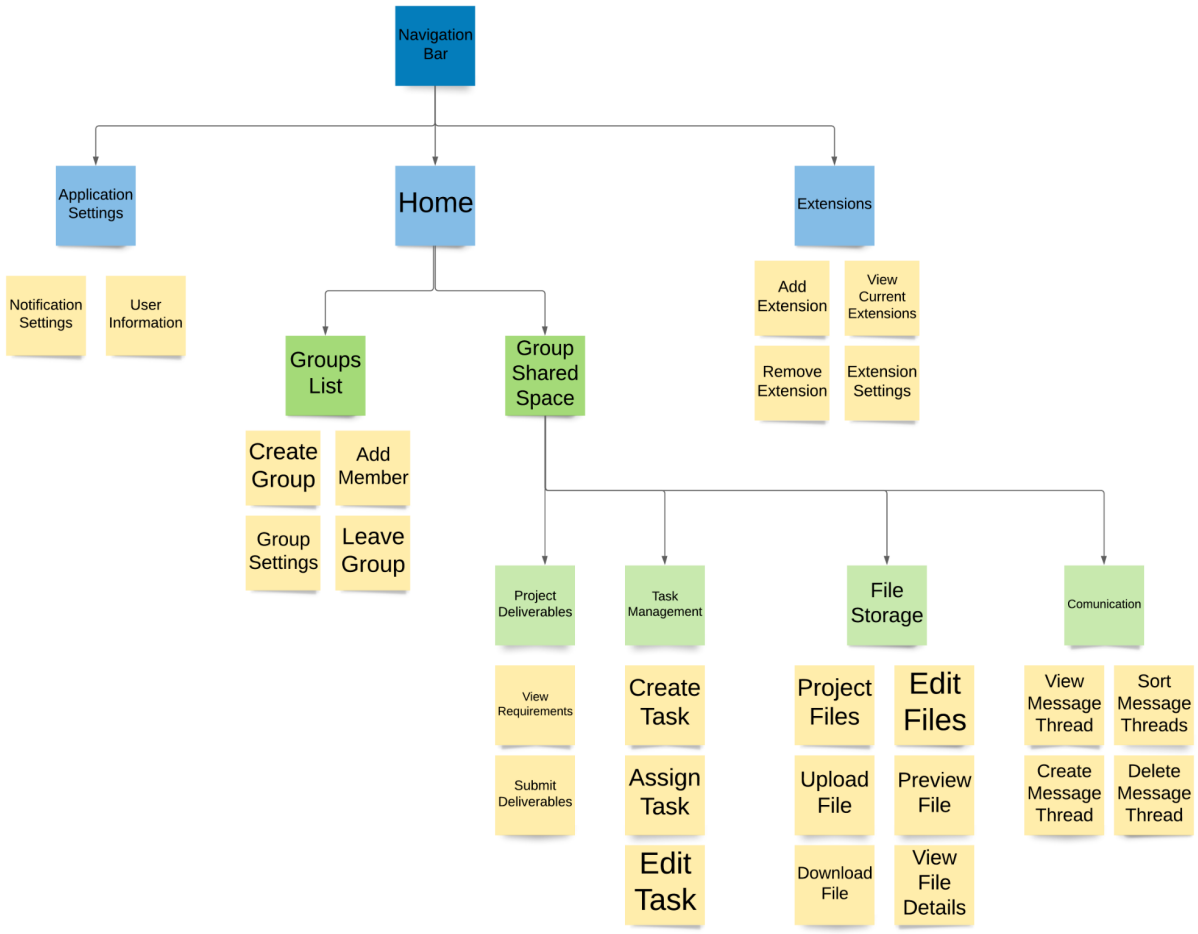
Information Architecture Version 1

This was the initial information architecture that our group designed, before we began work on our prototype. As the prototype was created, the architecture was refined to account for the challenges and insights that we came across.



Information Architecture Version 2

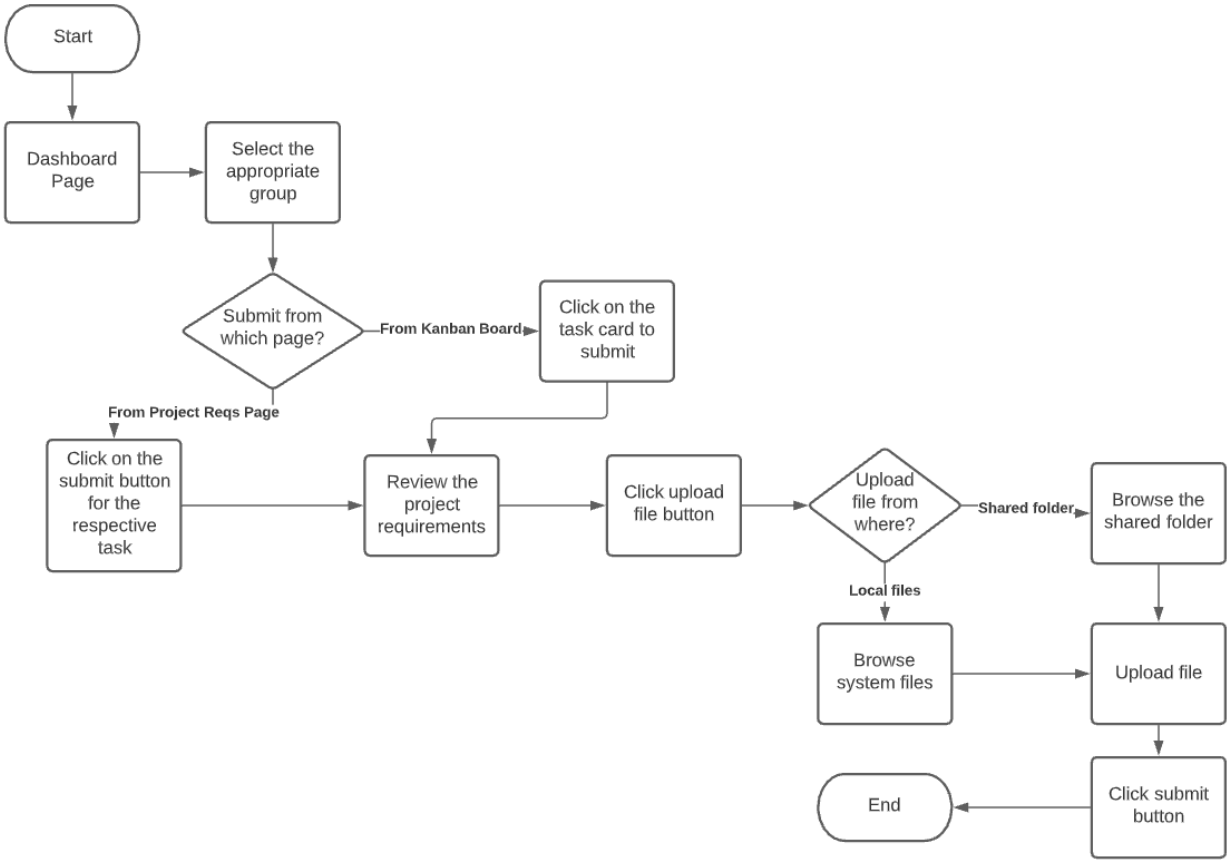
This is the final information architecture, after the revisions we made during the process of creating our prototype. Our information architecture diagram represents the hierarchy of pages within the application. Going down the tree represents navigating further into the application, with the yellow blocks representing the actual functions and affordances.



Interaction Architecture

A student wants to submit a file that they already uploaded to the teams' shared folder for an assignment.

Interaction Architecture: Submitting a file for the project.

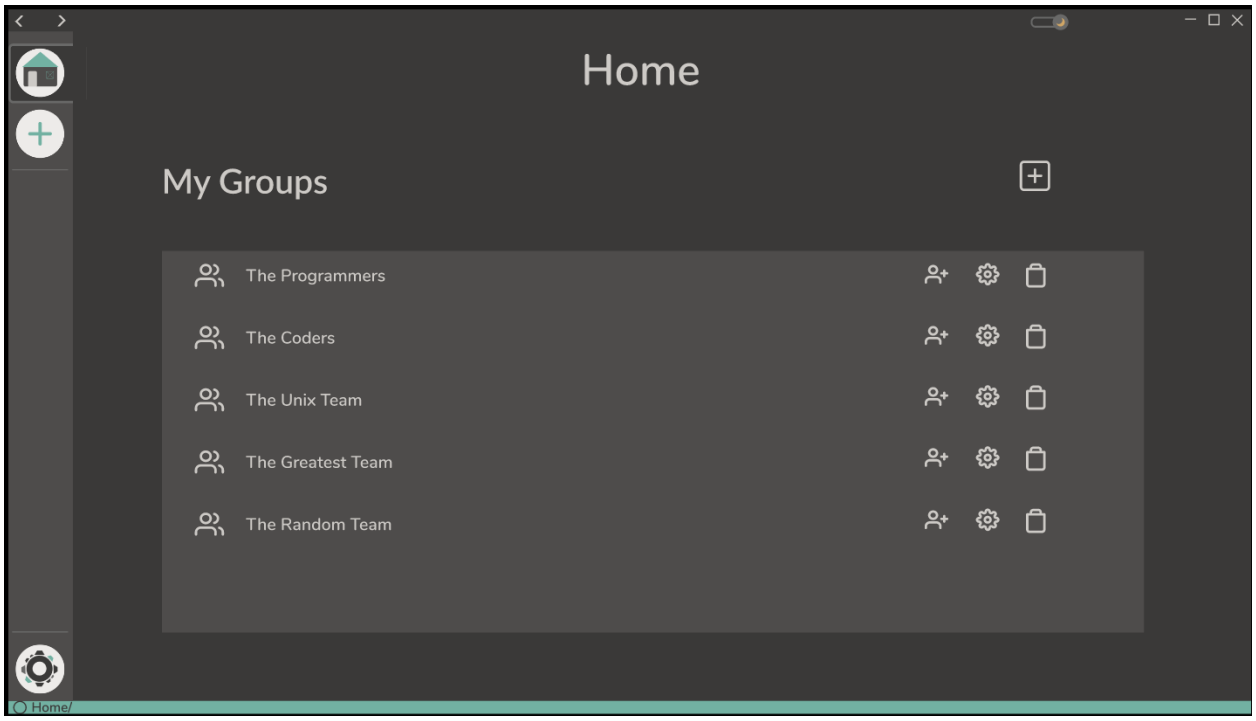


Prototype

Our prototype was designed using “Figma”, an application for designing and prototyping. Clicking the link below will redirect you to our prototype, where you can interact with the current version of our prototype.

Link to Prototype:
https://www.figma.com/proto/11Vz8KkjLnnWuoAtxtpQNk/DarkTheme_CSS480-(Copy)?node-id=208%3A675&scaling=contain.

Home Screen

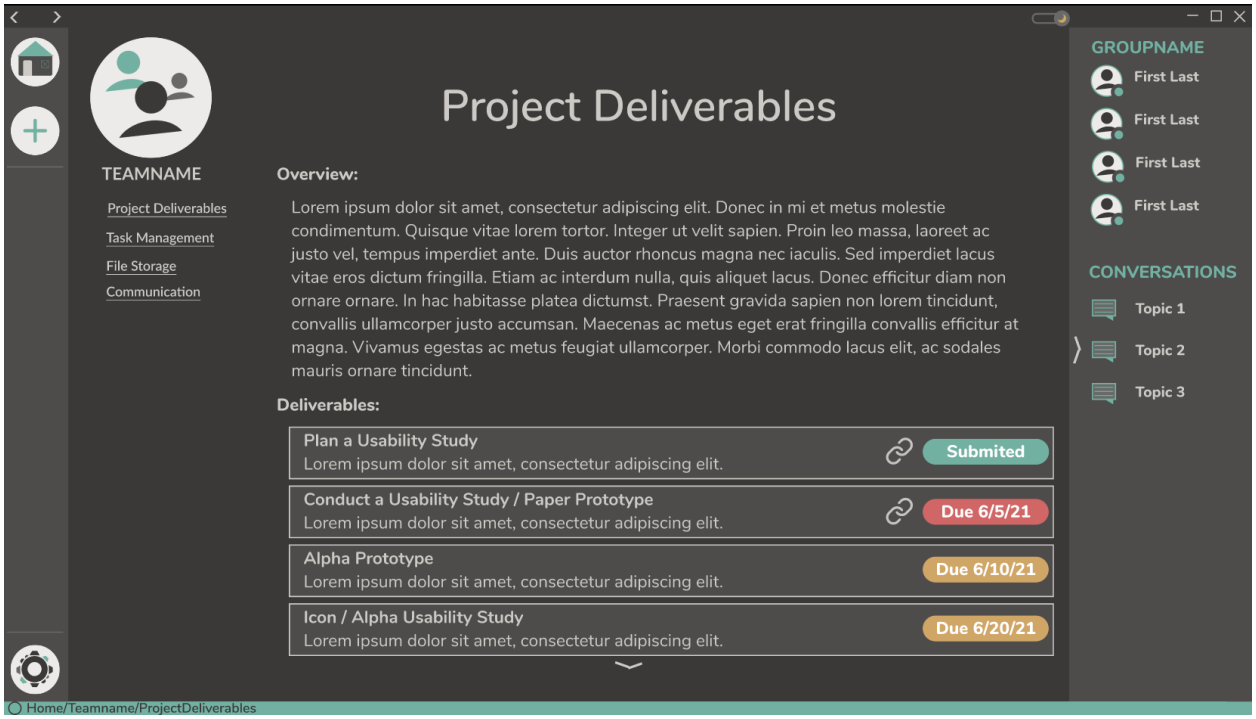


The Home page is where users start out in the application, and can always be quickly accessed by clicking the “Home” button on the left-hand navigation bar. The page includes a list of links to all the groups that the user is a member of, and also allows the user to create and change settings for a group.

Key features of the home page include:

- A list of all the groups that the user is a member of, that navigates the user to a group’s page when clicked on.
- The ability to create and name a new group.
- The ability to add new members to a group, change settings for the group, or to remove the group.

Project Deliverables

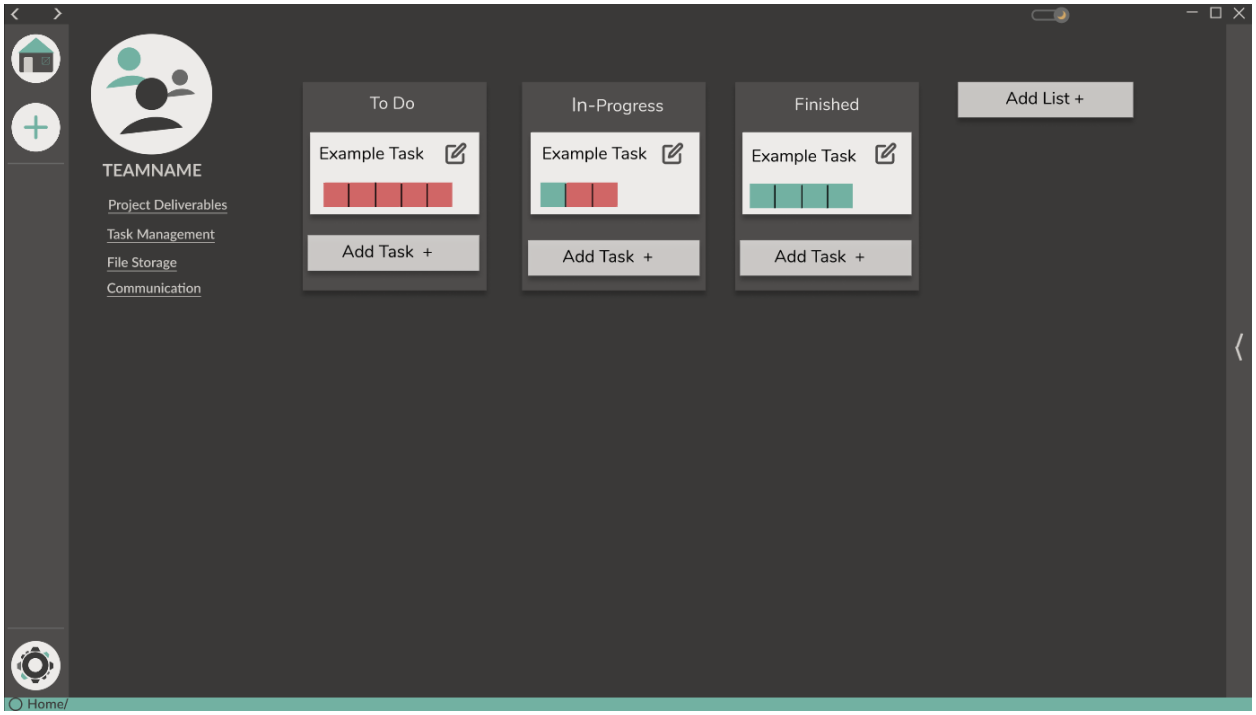


The Project Deliverables page is where users can go to read about the project their team is working on, look through the deliverables for the project, read the requirements for each deliverable, and submit their work. In a student team, this would allow students to interface with their assignments like on canvas. On a workplace team, this would allow employees to plan the project they are working on and submit deliverables to their upper management for deployment.

Key features on this page include the following:

- Deliverables are objects that display their due date, priority (through color labeling), and whether or not work has been linked to that deliverable.
- Each deliverable can be added as a task or as a milestone to the task management board.
- Requirements listed for a deliverable can be “checked off” to show automated progress on the deliverable in the task management tab.
- When a deliverable is clicked, an overlay pops up allowing users to preview the linked work (if any) alongside the requirements and “check off” requirements that are completed.
- A user can easily navigate from this page to many other useful tasks relating to their team including task management, file storage, communication channels, or hopping directly into a chat using the sidebar.

Task Management

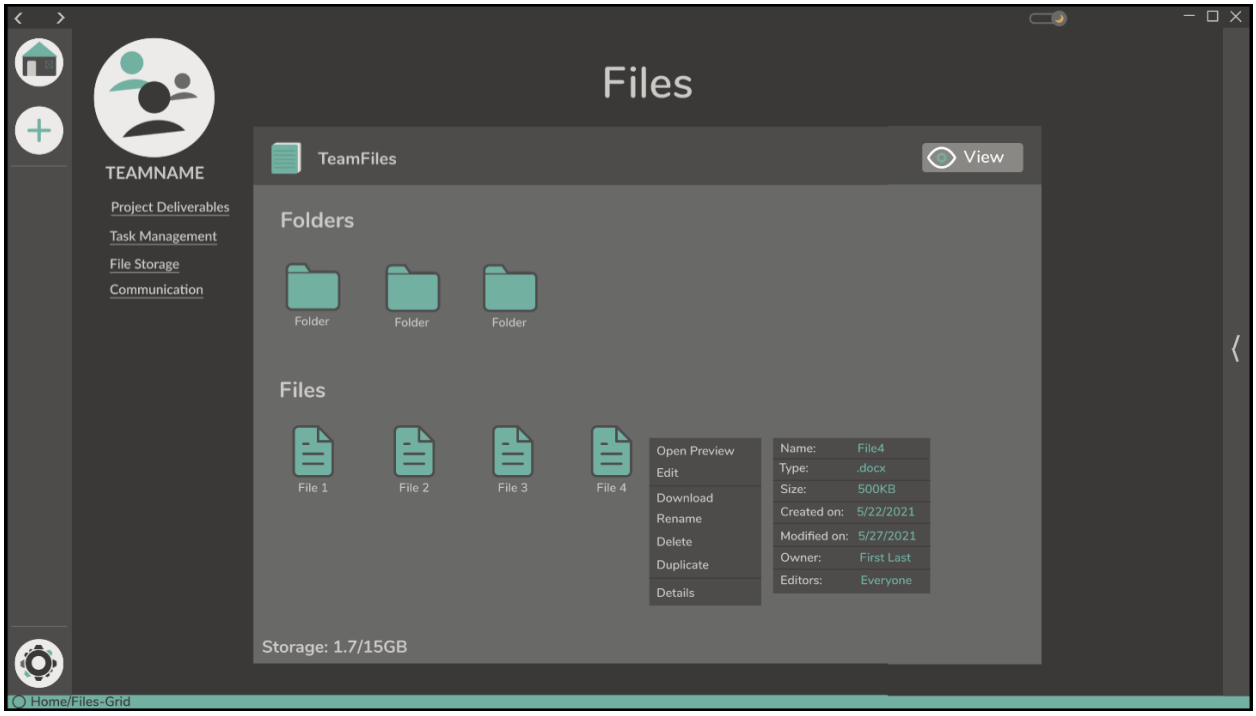


The task management page will redirect users to their respective groups' kanban board. This provides a way to view progress on specific tasks, as well as delegating new/different tasks to members of the group. All group members will be able to view and edit or update kanban board tasks and progress.

Key features of the task management page include:

- Basic kanban board features
    - Customized workflows
    - Task delegation
    - Task sorting via lists
- Task partitioning for easy viewing. Essentially each box on the task progress bar represents a project milestone that has been input by the group. As each milestone gets checked off, the respective box on the progress bar will turn green for completion.
- Allows users to add tasks, edit tasks, and add a new list.

File Storage

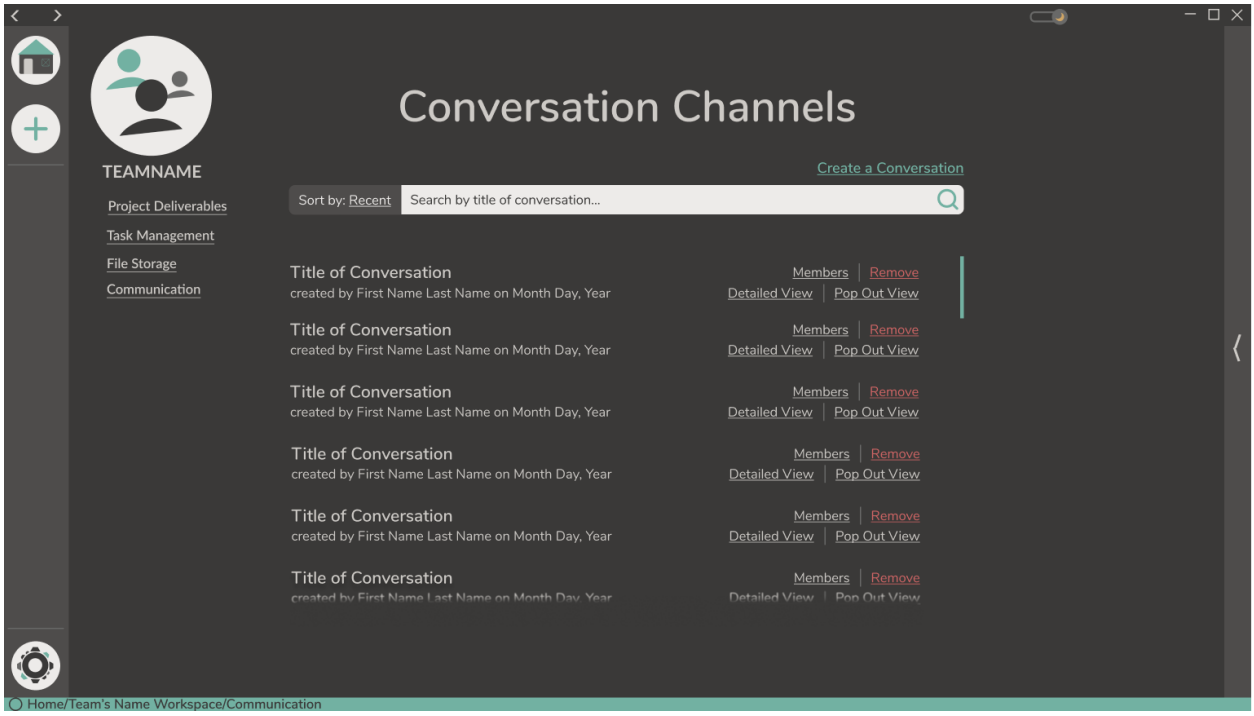


The files page allows students to upload files to a shared storage space accessible to every team member. This allows students to share files that are important to completing the project. Additionally, our files system allows the files to be edited collaboratively in real time, allowing all team members to work on the same document at the same time.

A few important design decisions make regarding this part of the application are

- Place the folders at the top in one alphabetized group, and put the files below them in a separate alphabetized group. This makes navigating into the folders much easier, since the user does not have to spend time searching a list for a folder.
- Allow the user to right click a file or folder to view a list of options. One of these options is the “Details” option, which when clicked lists all relevant information for the file such as the name of the file, type of the file, size of the file, and permissions.
- Allow the user to view a preview of the file before editing or downloading it so that they can be sure that they have selected the correct file.

Communication



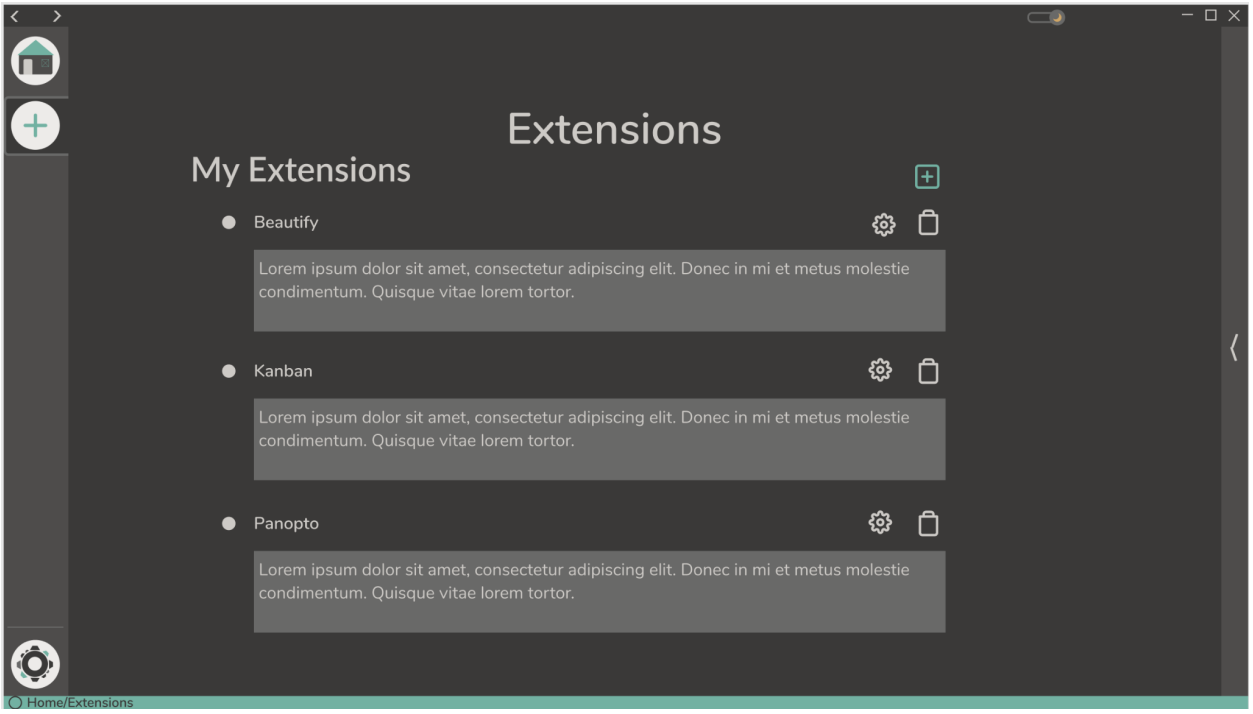
The Communication feature allows the user to create conversation channels with their team members. The purpose of this feature is to allow users to communicate effectively and efficiently with their team members. On the “Conversation Channels” screen, the user can do the following:

- Create a conversation
- Sort a conversation by recent, unread, newest, or oldest conversations
- Search a conversation by title
- View members of a conversation channel
- View a conversation channel through either a pop-out view or a detailed view

The pop-out view allows the user to view the conversation thread and post replies. The detailed view allows the user to do the following:

- View the conversation thread
- Post replies
- Edit their posted responses
- Delete their posted responses
- View unread messages in the thread
- View members

Extensions



The extensions feature allows the user to customize Divvy with a variety of helpful extensions. We wanted to give our users the ability to install extensions that will help them complete their tasks more efficiently.

In this page, the user can

- Add a new extension
    - When adding an extension, the user can search it by its name, and then install it
- Change the extensions settings
- Delete an extension

Usability Testing

Our methodology for usability testing was to first have the participant complete an icon quiz to evaluate how a user would interpret our icons before even laying eyes on our application. Next, we had them start on the home page of our application and navigate to the right part of the page to complete a specific task. Each of our team members was responsible for testing a different task to make sure we tested the whole application.

Findings

The most important thing we learned from the usability tests was that it can be very easy to assume something is clear as the designer, only to find that it is not clear to the user. As the

designers, we know the context of the things we created, so it is not possible to see it from a new perspective again. Our interpretation of our application is framed by the series of decisions made to take us to this point in the design. However, the user does not have any of this context. When they first use our application, they do so with a completely new perspective. We learned that the user often has different interpretations of our application than we intended.

The first example of this was in regards to our icons. What we found during our icon quiz was that our icons did not always mean the same thing to the user that we wanted. The user confused the project description icon with the file storage icon. Also, the user did not think that our project management icon was representative of that section.

The next time we encountered this was in regards to our home page. During one of the usability tests, the participant started on the home page where he was presented with a list of courses that when clicked on would take him to the group for that course. The icon for this page was our “Groups” icon, listed third on the left navigation panel. After completing the task, the participant was instructed to return to the home page. However, this caused some confusion, as he was not sure which icon to click to navigate there. Once he did arrive back at the home page, he did not recognize it as the home page, so he was not sure that he had been successful. This caused us to realize that our home page was not designed well. Furthermore, we recognized that having the icon for our home page be the Groups icon and appear third on the list was very confusing, and hurt the navigability of our application. We recognized just how important it is to have a strong home page that is supported by a good navigation bar.

Other lessons we learned from usability testing was that our interactable elements needed to be more obvious. Specifically, the contrast between interactable objects and static non-interactable design elements needed to be more clear.

Impacts

The impacts of our findings were to revise certain elements of our design to make things clearer. The first major change we had to make was to completely redo the homepage to make it a better start page to our application. We recognized that we needed to make it more clear within the page that this was our home page, so a user could identify what it was when they navigated to this page.

The next major change we made was to simplify the navigation bar that appears on the left side of every page of our application. We reduced the number of icons on this navigation bar from five things, to just three things. Furthermore, we created a new icon for the home page, and used that to navigate the user back to the home page.

Next, we made all of our buttons lighter in color to emphasize that they are interactable. And finally, we made significant changes to the project description page, including renaming it to project deliverables. We made the interactable elements on this page more obvious and implemented an assignments and requirements tab to this page to let the user view more things.

Reflections

We all feel very satisfied with the final product of our design efforts. However, given more time to spend on this, there are some things we would have done differently.

Our prototype was really fleshed out and we spent a majority of the time on refining it. If we had more time we would have liked to make a simpler version of the prototype first before finalizing the details. With the time constraints of quarterly based classes, we felt pressured that the first design we created, also had to be the final design. We think it would have been beneficial to the overall design process if we just focused on the major design components of our application before we got into the finer details. We also would have liked to test out a few different design ideas rather than just one. Utilizing A/B prototyping and testing would likely have been our main design process. Having multiple different designs also lets us get more informative feedback from user testing and we could take the best components of each design to decide on a final design. Finally, if we had a lot more time, coding it up would have been a fun project to do. With the effort we put into the design of the application, it would have been great to see what functions of the design we could actually implement in a reasonable timeframe.

While working on this project, our group learned a lot about the design process. We found that we shouldn't assume anything about how the user will perceive our application because they are not the ones who designed it. We may have understood what certain icons were or why certain components were placed where they were, but it may not always be that clear to the user because they won't be as comfortable as us with the application. Another learning point was how much time we ended up spending on the visual design. We had to put a solid amount of effort into the visuals and designing a good UI to increase the usability of our application. We also spent a large amount of time on prototyping and user testing to refine our visual design and potentially uncover any flaws that we may have missed. Finally, we learned how important communication was to make sure we were all on the same page regarding our design. We were able to incorporate the best of each member's ideas and didn't experience much indecisiveness with the only "issue" being the difference between our applications dashboard and the my groups page (this was easily solved with a quick group discussion).

Overall, we are quite happy with how our design ended up given time constraints and online/remote group and class work. We were able to learn a lot about the design process, as well as designing something as a team. Making sure we all contributed something to design was important to us and we felt that we were able to succeed in that regard.