

SW Engineering CSC648-848-05 Spring 2024

Project Title: ToDo-Today

Team: 02

Names of Group Members:

Name	Email	Role
Alex Nikols	anikols@mail.sfsu.edu	Team-Lead
Jason Tran		Frontend-Lead
Sulav Jung Hamal		Backend-Lead
Jadon Hoang		Github Master
Randale Reyes		Database Admin
Christian Gopez		Docs-editor

Milestone 2

04/04/2024

History Table

Date	Revision
4/4/2024	Milestone 2 Revision 1
3/25/2024	Final Milestone 1 Revision 2
3/1/2024	Submitted Milestone 1 Revision 1

I. Brainstorming Ideas

1. Dungeon Finder:

An application similar to a dating app but for people, Dungeons and Dragons players. Using this app people who are looking for a group to play with would be able to connect with others. Users would be able to fill out a profile consisting of things such as the type of player they are, their character information, and also what they are looking for in a group.

Pros:

- An easy way for players to find a group
-

Cons:

- ?
- ?

2. ToDoToday (WIP):

A ToDo list that is able to support the ability to mark down a certain degree of percentage done for a task. The idea of not being able to mark down a task as done is demoralizing to the person and doesn't help in productivity. With this feature, we are able to make sure that we can keep track of and visualize the progress that we are making in the tasks that we mark down. There will also be a journal feature that can help the users to reflect and further digest how they are doing for a task.

Pros:

- Providing a motivating way for marking tasks off
- Journaling feature to help reflect on tasks

Cons:

- Many other competitors
- ?

3. RentABook:

Textbook rental service similar to that of Airbnb. We all have been in a position where either we do not have the funds to get a new textbook that is required for a class that we are taking or we bought a textbook and no longer have a use for it rendering it useless. This app fixes both of these problems because on this app you would be able to rent textbooks that you would need

without having to pay the full price of that textbook. On the other hand, you would also be able to put the textbook that you have up for rent so that you can make money off of a book that you might not have a use for anymore.

Pros:

- Can help provide texts and education for cheaper prices
- Can be another income source for those with old textbooks

Cons:

- Legality?
 - Would be hard to monitor and make sure the textbook owners do not get scammed
-

II. Executive Summary

ToDo TODay is a To-Do List and daily task tracker app, with integrated calendar and journaling functions. What makes the ToDo TODay app stand out are a couple of unique twists on the classic To Do List apps currently available on the market. The first of which is the ability to allow for partial completion of tasks and to mark them by percentage. This task system is also embedded into a nesting structure allowing for smaller tasks to be nested inside of larger tasks so you can understand your overall progress on projects or daily goals cleanly and intuitively. The other unique addition to this sort of To Do List app is a dynamic journaling function. This dynamic journal will use an AI GPT system to dynamically create new journal prompts for the user daily based on what is the highest priority on their current to-do list.

ToDo TODay is specifically designed to be an easy-to-use way of organizing anything from an individual's small tasks in their daily life to massive projects that may take months or years to complete. The reason why my team and I decided to create ToDo TODay was to look into every other To Do List app and how they work. Every single application that we found was binary either the task was complete or it wasn't. This kind of binary choice can be very demoralizing for large projects and does not give you an overall understanding of your progress. We created this app to take the best parts of large project trackers that are used in the industry and bring them to a more public audience. As well as including tasks and tools to help users stay on track with their projects and make sure that they have enough time to get everything done.

ToDo TODay's flexibility and structure help reinforce the habit of maintaining and updating their to-do list while also clearing off tasks from it. This positive reinforcement loop will not only help users be on top of all the regularly scheduled things they may need to do in their regular lives but also allow them to find time for themselves. The other thing ToDo TODay will do is encourage the

user to journal their experience and where they might be struggling with their tasks it will help them engage more fully in thinking about what they might be struggling with and how they might improve over time. Say somebody has practice guitar as a regular task they do weekly, then TODO TODay will bring that up in the journal and have the user talk about how they feel they are progressing in their guitar practice. When the user interacts and journal their experience it will help them crystallize and coalesce their feelings on how the practice has been going and how they have been doing which will help them better understand what they can do to keep improving and getting better at playing the guitar. This is just one example of the type of positive reinforcement and benefit you get out of the journaling and reevaluation process that TODO TODay is built around.

III. Main Use Case

1. Student managing their schedule

- **Actor:** Student, TODOtoday Application

- **Assumptions:**

- The student has access to the internet
- The student has access to a device (desktop or laptop)

- **Use Case:**

This student is very busy, they are taking a heavy workload this semester and must have a way to organize and complete these tasks in a timely manner. Stressed out and anxious, the student decides to download an app, ToDoToday, which will be able to help organize and keep track of their classwork. In one of these courses, the student has been assigned a large project with multiple milestones and checkpoints. With the help of the ToDoToday app, the student assigns the whole project as the main task and milestones as a subtask to the main project. In addition to this, the student also adds the checkpoints as subtasks to the milestone subtasks. As the student completes the checkpoints, the progress bar for the milestone and the whole project increases. On top of this, they also use the calendar to keep track of what time their various assignments may be due and what's coming up next. As a student works through his tasks and completes them they get marked as complete on the calendar. The student also got into the habit of using the journaling to reflect on how they did their assignments and where they can improve. With this in mind, the student feels much more at ease, feeling much less anxious about the project. Overall, the ability to observe the progression in the progress made on the project in its entirety induces a feeling of ease and relaxation. The newfound ability to journal these reflections also enforced what the student had learned from projects and assignments.

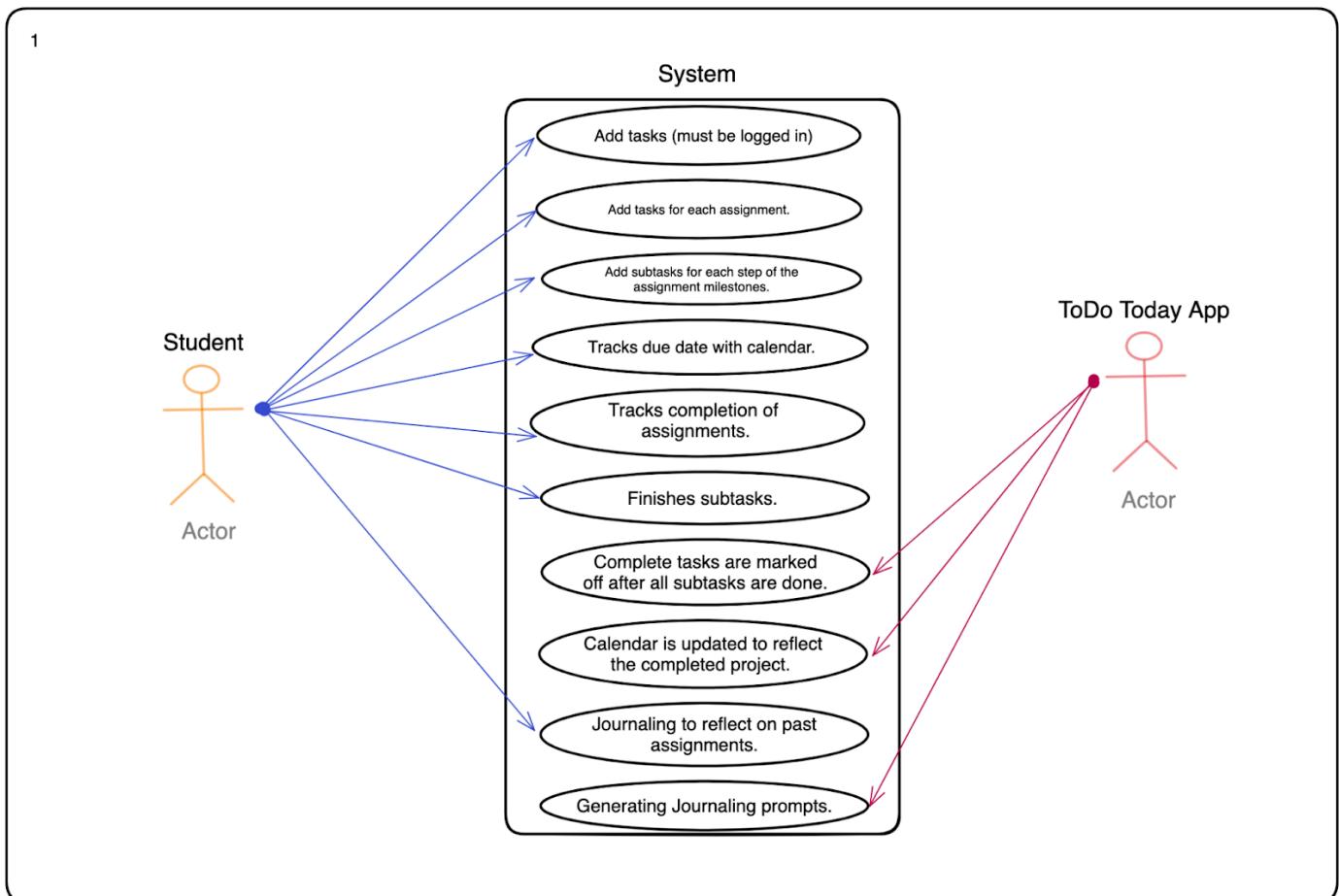
- **Benefits:**

- Organization: The information and tasks are easily available for the student to see and manipulate

- Progress Tracking: The student can have a visual representation of tasks completed for the day and could have an idea of how much more is left to do.

• Potential Requirements:

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.



2. A mother trying to manage both their kid's differing schedule and varying tasks

- **Actor:** Parents, TODOtoday Application

• Assumptions:

- Parents have access to the internet
- Parents have access to a device (desktop or laptop)

- **Use Case:**

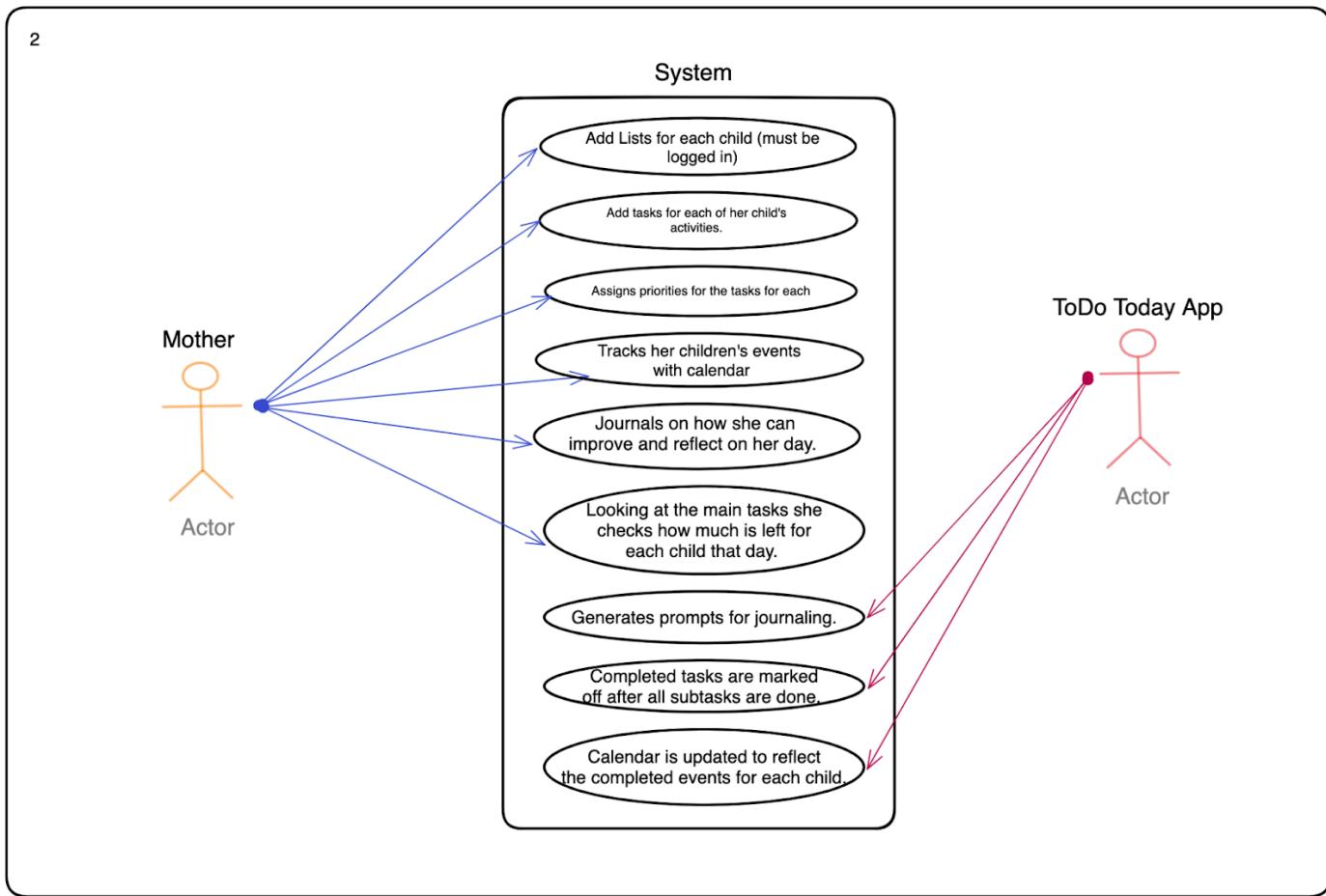
A mother is having a hard time trying to keep track of her children's schedules throughout the school year due to them being in different extracurriculars. One of the kids is more athletic and is involved in more sports and physical activities. The other child is more into music and computers. The two children have varying schedules and the mother needs a way to schedule and organize them. This app will help keep the two schedules separate for easier viewing. Within these sub-lists, she will be able to list the different things she has lined up for the specific child. As she gets things done the progress percentage will reflect and show how many of the activities or tasks were completed for that child's schedule. There will also be a feature that the mother can use to mark certain tasks to be more important so that she would know which task she would have to focus on more at that time. Lastly, the journal feature will be able to give her time to reflect on the tasks done, what she could have done better, and overall reflect on her day.

- **Benefits:**

- Organization: The information and tasks are easily available for the mother to see and manipulate
- Progress Tracking: The mother can have a visual representation of tasks completed for the day and could have an idea of how much more is left to do.
- Motivation: Rather than giving up, the journal will help with the ability to contemplate what she has done so far if things are a little overwhelming she be able to contemplate what to for next time.

- **Potential Requirements:**

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.
- Priority Setting: A feature to set the priority status of a task. This would allow people to set more important tasks to have a higher priority vs the lesser priority ones



3. Patients utilizing the app to keep track and log their daily life

- **Actor:** Patients, Life Coach

- **Assumptions:**

- Patients have access to the internet
- Patients have access to a device (desktop or laptop)

- **Use Case:**

A life coach is helping patients day by day to help control and improve their patients lives. The life coach recommends the patient an app that would help them schedule their life schedules and tasks while also keeping track of their progress. The patient will be able to use the to-do list app to make lists of tasks needing their completion. They would also be able to set up subtasks to break down larger tasks to make them more manageable and less overwhelming. They can use the calendar to not only keep track of scheduled appointments but also to manage their time throughout the week. They also would then be able to use the function of the journal app to keep tabs on how they are doing. The journal app will be a great feature for them as they would also be able to read and use the talking points they talked about in the journal section to their life coach and they would be able to debrief on the patient's week or day easier. With the

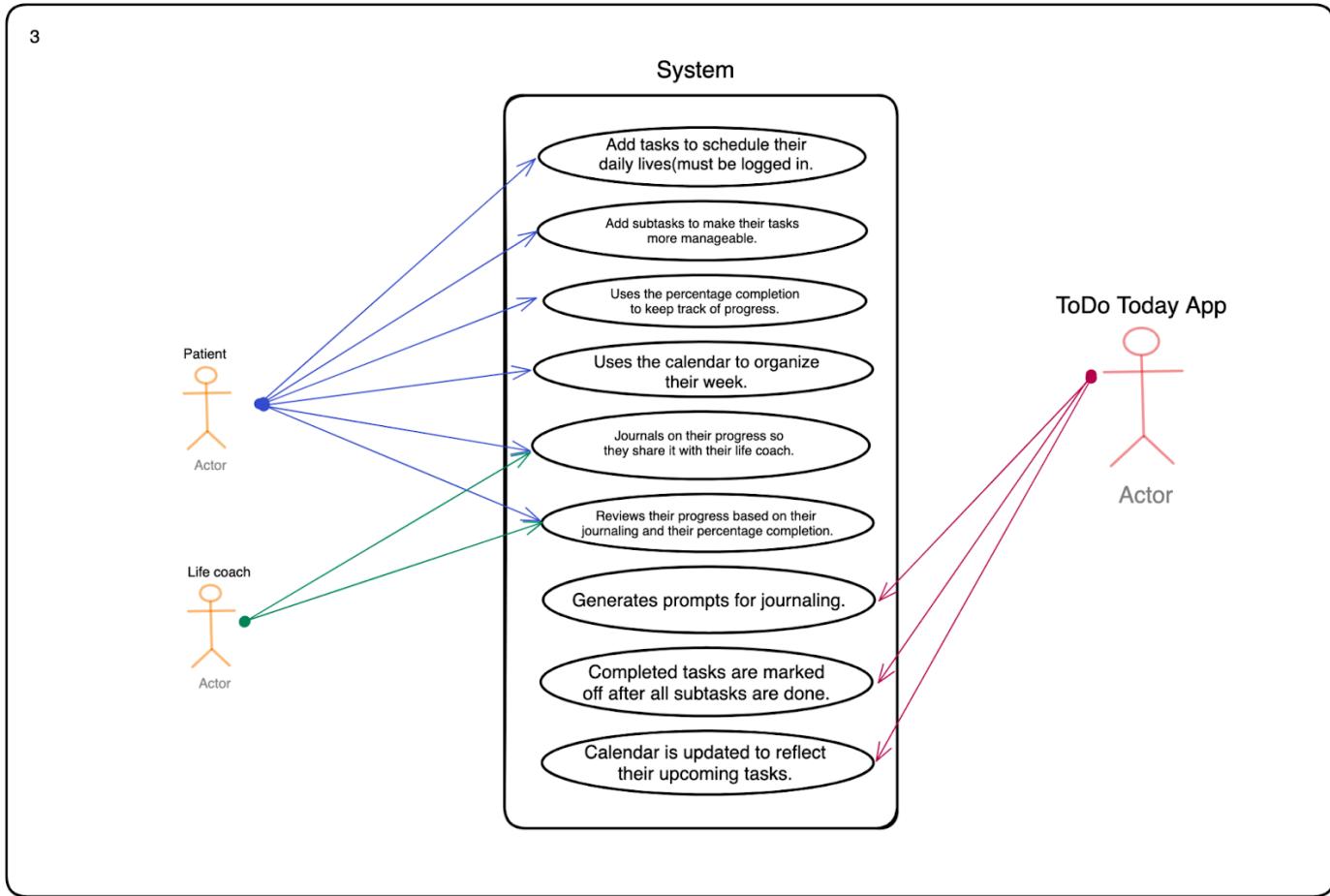
addition of the progress bar the completion of tasks is no longer demoralizing but rather makes it motivating to get the percentage of the bar to 100.

- **Benefits:**

- Self:Reflection: Through the journal app the patient will be able to keep track of not only themselves but also the tasks they need to do through the app
- Progress Tracking: The patient can have a visual representation of tasks completed for the day and could have an idea of how much more is left to do.
- Motivation: The visualization of the progress bar will be motivating to the patient to be able to get their tasks done.

- **Potential Requirements:**

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.
- Priority Setting: A feature to set the priority status of a task. This would allow people to set more important tasks to have a higher priority vs the lesser priority ones
- Journal feature: This feature will help the user debrief and think about the tasks that they are doing. Essentially having a recap and potentially being able to think about what worked for them and what didn't. Could also have a part where they keep track of their results or progress on tasks.



4. Project Manager and Members using the app to organize tasks for the project

- **Actor:** Project Manager

- **Assumptions:**

- The Project Manager has access to the internet
- The Project Manager has access to a device (desktop or laptop)

- **Use Case:**

The project manager needs to lead multiple teams to develop a new software project. These teams need to work in harmony if they want to succeed in building this project. By using the ToDoToday app the teams working under the project manager would be able to keep track of their work and progress while working through the project. The teams will be able to mark the tasks while also keeping track of the dates so that they make sure they can make their due dates for certain tasks. This is also easily viewable in a calendar format so that they can visualize the task due dates better. This allows the product manager to manage each team in a much more efficient and organized manner. As each team completes its subtasks, the tasks check off what has been completed, gradually increasing the progress bar for the main project.

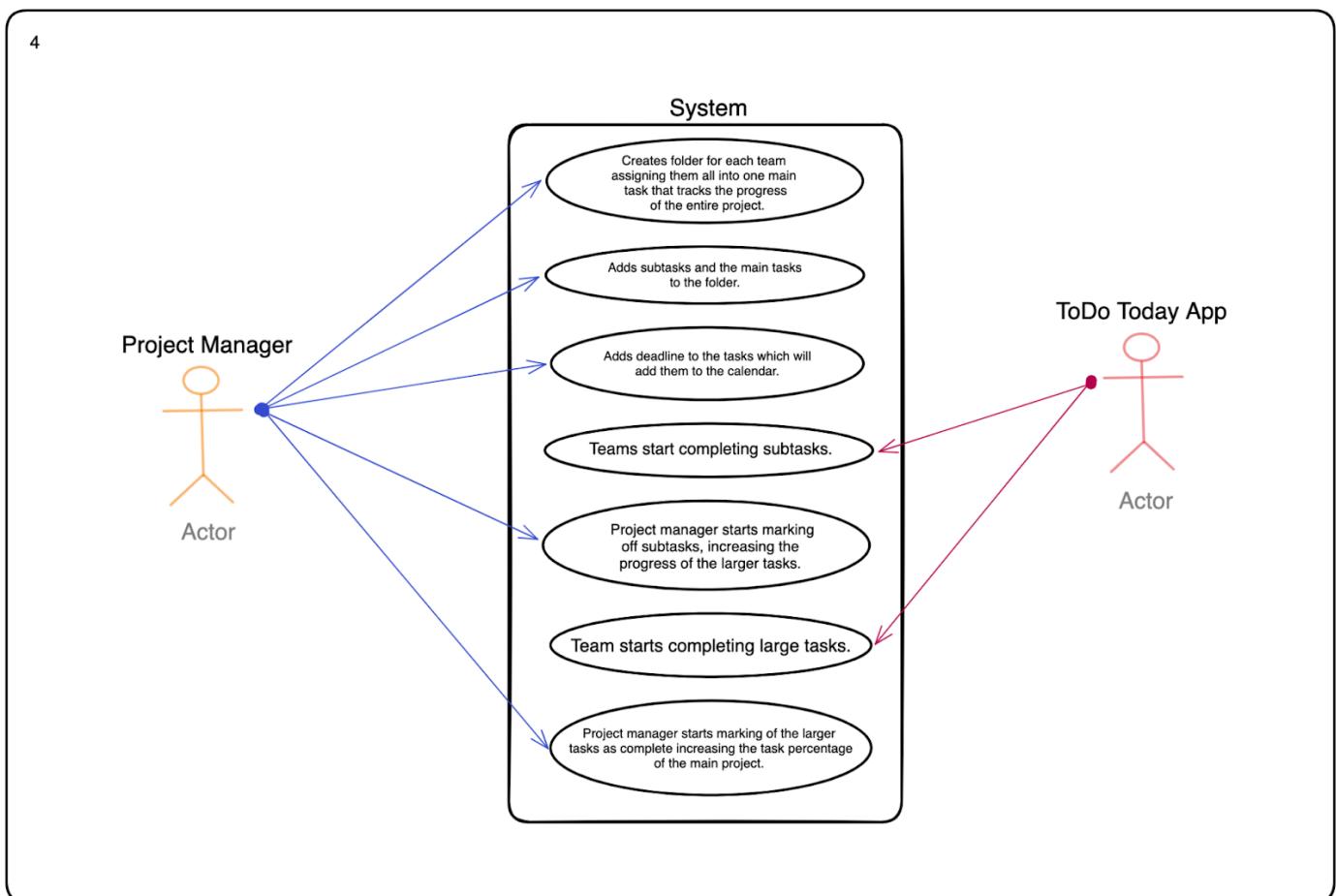
Overall, the organization allows the project manager to keep up with deadlines and pinpoint which tasks teams need to focus on to work in unison.

- **Benefits:**

- Organization: The Employees or project members can make different lists for their tasks which could also have subtasks to break down the tasks even more.
- Task Percentage: The Task Percentage will be helpful in that there is a visualization of how much the task has been done.
- Priority Setting: Being able to choose which things are more urgent than others will be of help to make sure that the things that need to be done are done before moving on in their project.

- **Potential Requirements:**

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.
- Priority Setting: A feature to set the priority status of a task. This would allow people to set more important tasks to have a higher priority vs the lesser priority ones



5. Teachers using the app to plan their lessons for the different subjects that they are teaching

- **Actor:** Teacher

- **Assumptions:**

- The Teacher has access to the internet
- The Teacher has access to a device (desktop or laptop)

- **Use Case:**

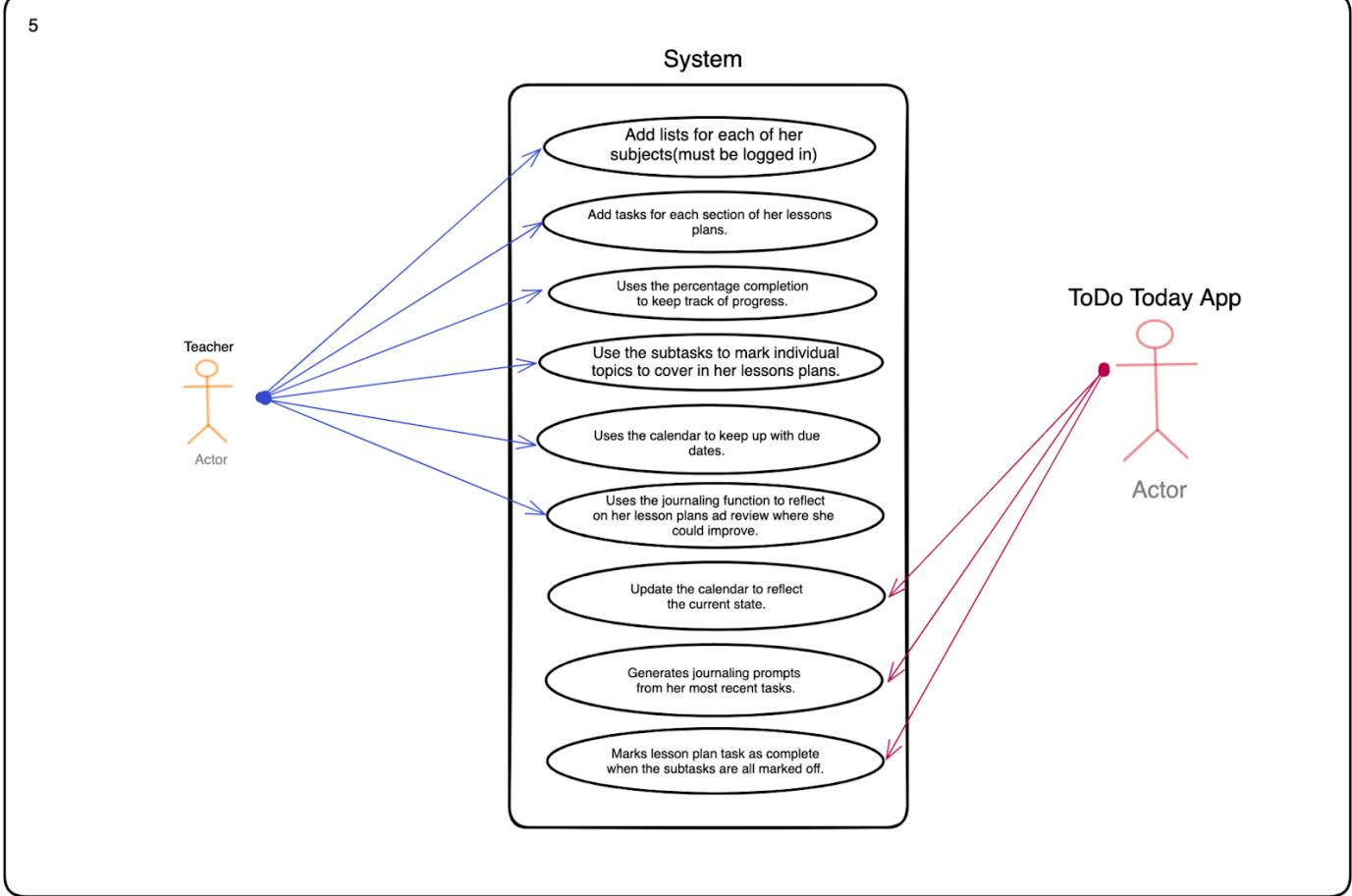
The teacher is planning on teaching multiple sections of courses varying in different subjects. With the app, they would be able to list out specific tasks and lists for that specific class. Keeping the different to-do lists organized and sectioned for readability. They would be able to make different sub-lists pertaining to subjects such as math, science, and history. They would be able to have the lesson plan and then another list under that for checking off the material and steps on that lesson plan. Then depending on how the different subjects went the teacher would be able to log the lesson and talk about the things that she liked or things that she could improve on using the journal function of the app. If there is an important thing that is going on in a specific class such as one of them having a quiz or a test on a certain day, they would be able to mark that quiz or test as a high priority so that it is not missed. Using the calendar the teacher will be able to keep track of when quizzes, exams, and other class milestones are approaching to make sure that she is keeping up with the schedule that she has laid out for the class.

- **Benefits:**

- Organization: The Teacher can have separate lists for their needs. Through this, they have multiple to-do lists that they can easily access and find
- Task Percentage: The Task Percentage will be helpful in that there is a visualization of how much the task has been done. They would also be able to see how much of the lesson plan they did by looking at the percentage of that task's bar.
- Priority Setting: Being able to choose which things are more urgent than others will be of help to make sure that important things are not forgotten.

- **Potential Requirements:**

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.
- Priority Setting: A feature to set the priority status of a task. This would allow people to set more important tasks to have a higher priority vs the lesser priority ones
- Easy UI: The user interface should be easy to use and easy to find the different lists without looking too cluttered.



6. A Traveler planning their trip and preparations

- **Actor:** Traveler

- **Assumptions:**

- The traveler has access to the internet
- The traveler has access to a device (desktop or laptop)

- **Use Case:**

Sarah is planning a vacation to Europe. She uses the app to create a packing list with items such as clothes, toiletries, and travel documents. As she packs each item, she checks it off in the app. She could see the completion rate of each place and uniquely identify which place she still needed to visit. As for the rip itself, she can make a subtask that can take care of activities and things that she wants to do on the trip. She can also assign dates to these tasks and will be able to view them on the calendar for ease of view. This gives her a way to have an overview of the trip and what she can expect on certain days. As she checks off the activities it also gives her an indication of how much is left in the trip that she has planned. This can help her in many ways one being that if the progress bar is almost full but she has more time

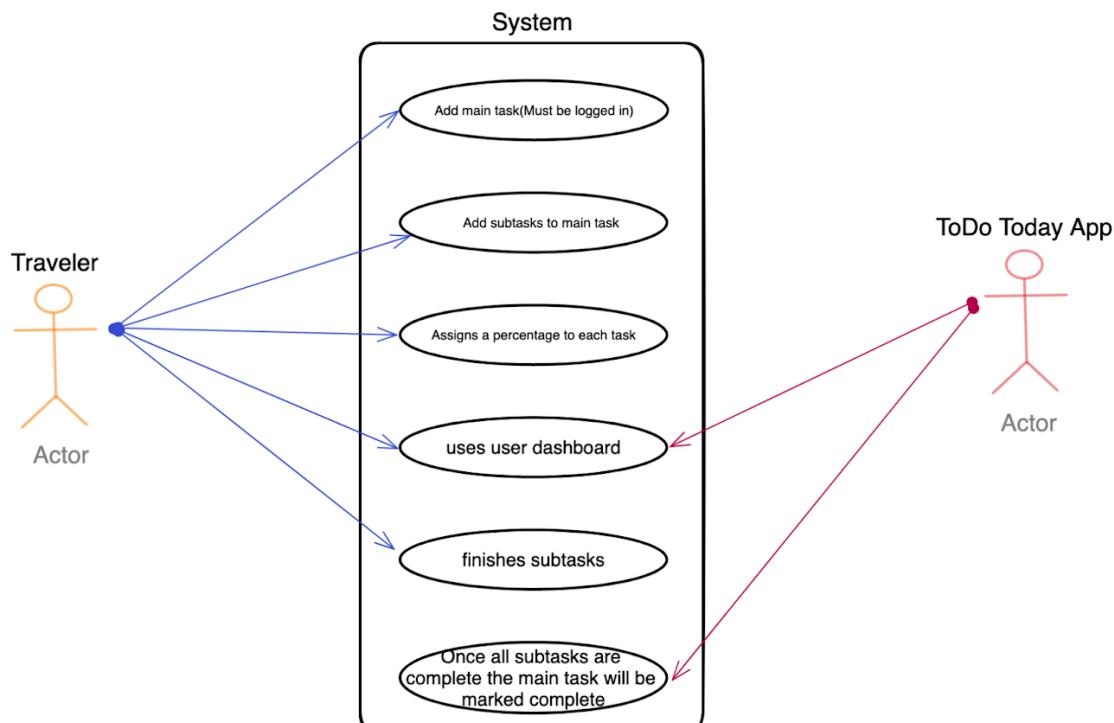
wherever she is staying this tells her that there is more room for other activities and she can plan accordingly.

- **Benefits:**

- Organization: It is easy for Sarah to make a list to track things such as things to pack. She can make lists of any of her needs during the trip through the app
- Tracking: The lists that she makes can display the progress that she has made so that she can visually see how much else she has to pack, or how many more things she has to do in a certain city.
- Planning: The app will make her planning easier and more accessible through the ability to create, change, and manage the different lists available through the app.

- **Potential Requirements:**

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.
- Priority Setting: A feature to set the priority status of a task. This would allow people to set more important tasks to have a higher priority vs the lesser priority ones
- Easy UI: The user interface should be easy to use and easy to find the different lists without looking too cluttered.
- Manageability: The user should be able to manage and change things such as priority status and other general things like the name of the task or where the task is.



7. Hobbyist working on multiple small side projects/models.

- **Actor:** Hobbyist

- **Assumptions:**

- The hobbyist has an Internet connection
- The hobbyist has access to a device (desktop or laptop)

- **Use Case:**

This hobbyist is attempting to build a bookshelf in their workshop. They heard about a to-do list app from one of their friends for managing projects. He decides to check out a to-do list web app, (ToDo ToDay). He then makes a new to-do list using the app to organize his bookshelf-building project. He then set up the main task for that to-do list first being researching the type of bookshelf he wants to build. Then he creates his first task, research. Next, he creates subtasks for researching styles of bookshelves, researching materials, and researching the hardware he is going to use. He quickly decides to use the hardware he has on hand and gets that first subtask done then he moves on to research materials. He finds a deal on walnut wood and decides that that will be the material he'll use for the project. Moving over to one of the other tasks, the purchase of materials, he goes down to purchase, the wood for the body

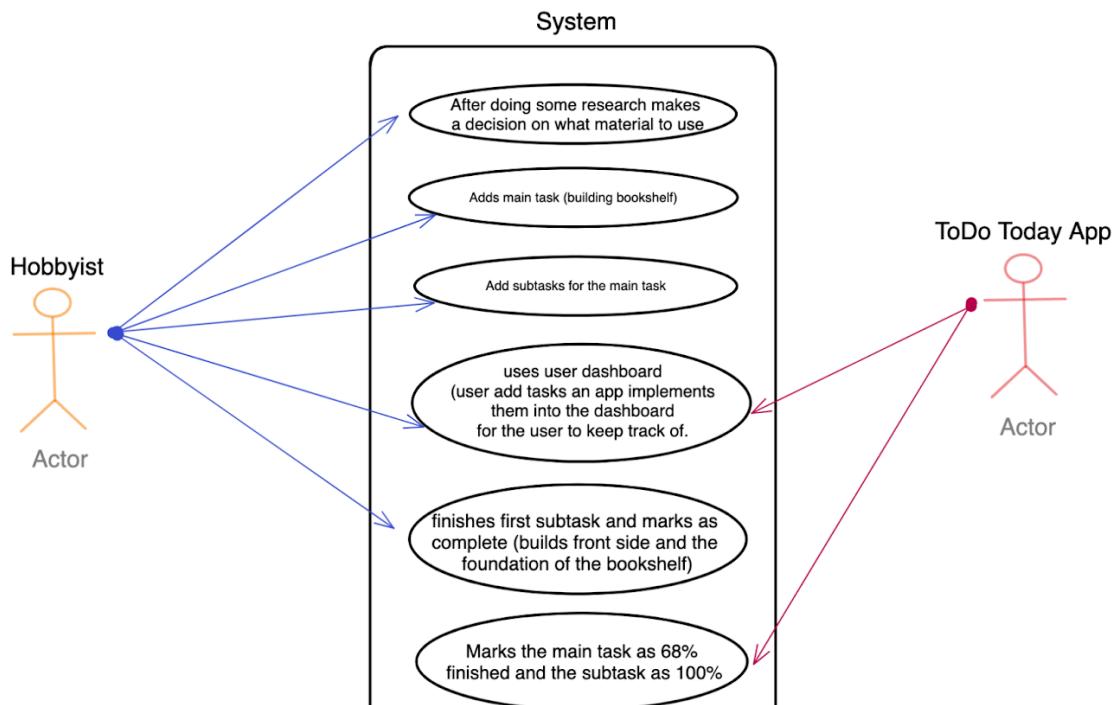
section and marks it off. He continues down the to-do list marking off tasks, building and setting everything up until he has everything ready for the bookshelf. He starts building the outside frame first and gets that done the first day. He marks that off his list and sees that he only has the shelves and the backside of the shelf to do next so he's already 68% of the way through his project. Feeling more confident he continues to build the body of the bookshelf and now he is closer to 85% completion, all that's left is finishing that up and then doing the final stages of staining and putting in hardware can check off everything from the list and it shows his completion at 100%. Once he is finished he then uses the Journal to reflect on what he struggled with and how he can improve for his next project.

- Benefits:

- Organization: Easily break down your projects into smaller, manageable tasks.
- Progress Tracking: The percentile completion allows you to gauge your overall progress at a glance.
- Prioritization: Quickly identify and focus on tasks that need attention, ensuring efficient use of your time.
- Flexibility: As you juggle multiple projects, adapt and prioritize tasks based on changing circumstances.
- Motivation: Seeing progress percentages can be motivating and help you stay on track with your hobby projects

- **Potential requirements:**

- Nested Task Management: The app should support the creation of hierarchical task structures, allowing users to break down larger projects into smaller, more manageable subtasks.
- Task Percentage Completion: The app needs to have a feature that allows users to assign and visualize completion percentages for each task, providing a quick overview of the overall progress.
- Prioritization: There should be functionality for users to prioritize tasks within a project, helping them focus on what needs immediate attention.
- Flexibility and Adaptability: The app should allow users to easily modify and rearrange tasks as project requirements change or evolve over time.
- Multi-Project Support: Since the hobbyist is working on multiple small projects simultaneously, the app should enable users to organize and switch between different projects efficiently.
- User-Friendly Interface: The app should provide an intuitive and user-friendly interface to enhance the overall user experience, making it easy for the hobbyist to manage tasks without a steep learning curve.
- Motivational Features: Including visual elements or notifications that celebrate task completion milestones could contribute to a more motivating user experience.



8. A zoo keeper keeps an eye on and maintains different species of animals in the zoo using the app to keep track of each enclosure.

- **Actor:** Zoo-Keeper

- **Assumptions:**

- The Zoo-Keeper has access to the internet
- The Zoo-Keeper has access to a device (desktop or laptop)

- **Use Case:**

Zookeeper was overwhelmed when trying to care for and keep track of the dietary and other needs of all of the different animals that were under his care. He found an app called To Do Today, that allowed him to set tasks for each animal that he had to take care of, and under that task, it set subtasks for their dietary requirements and any other task that he might need to do for each individual animal. This app was very convenient for him because it allowed him to not only keep track of feeding schedules using their calendar but also the percentage of completion of taking care of each animal during his shift. He uses the journaling function to keep track of the animal's preferences for what they eat and other quirks. His Journal becomes a record of

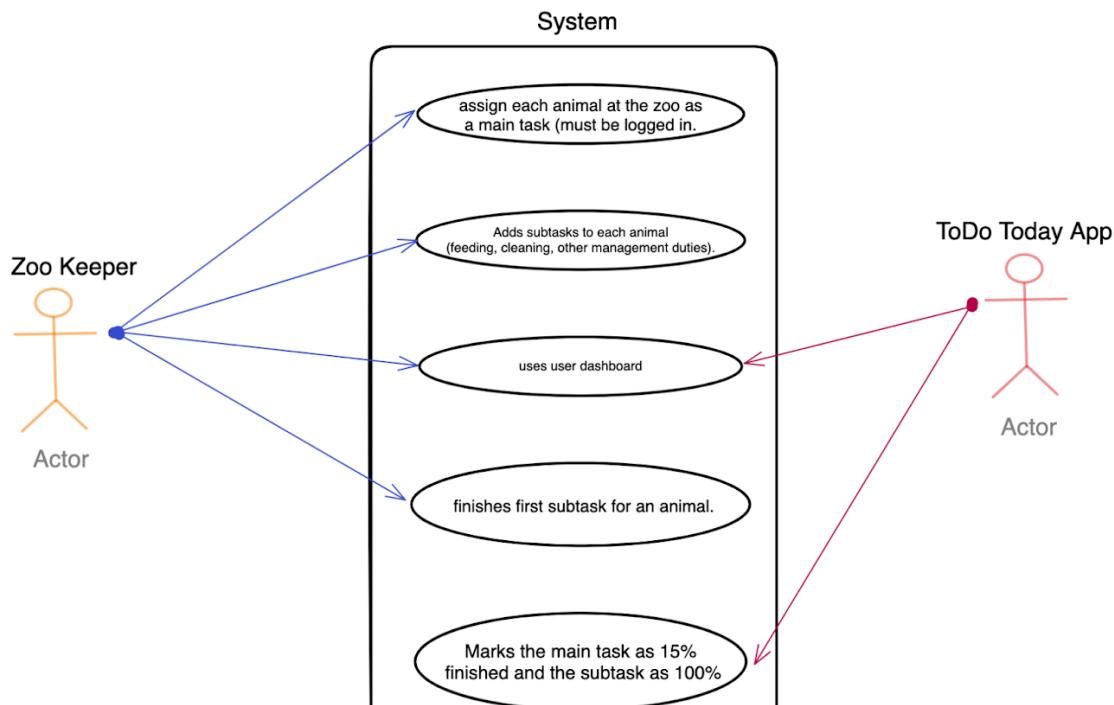
how to handle each animal. With the app, he's found his job significantly more enjoyable and a lot less stressful as he does not have to worry about getting confused and either doing tasks twice or missing tasks.

- **Benefits:**

- Organization: Easily break down your projects into smaller, manageable tasks.
- Progress Tracking: The percentile completion allows you to gauge your overall progress at a glance.
- Prioritization: Quickly identify and focus on tasks that need attention, ensuring efficient use of your time.
- Flexibility: As you juggle multiple projects, adapt and prioritize tasks based on changing circumstances.
- Journal Log: The journal can be used to keep track of each enclosure or habitat, take notes, or just write general reflections on the tasks regarding it in the journal.

- **Potential Requirements:**

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.
- Priority Setting: A feature to set the priority status of a task. This would allow people to set more important tasks to have a higher priority vs the lesser priority ones
- Easy UI: The user interface should be easy to use and easy to find the different lists without looking too cluttered.
- Manageability: The user should be able to manage and change things such as priority status and other general things like the name of the task or where the task is.



9. A new dog owner wants to keep track of things to do for their dog.

- **Actors:** Dog owner, ToDoToday

- **Assumptions:**

- The dog owner has access to the internet
- The dog owner has access to a device (desktop or laptop)

- **Use Case:**

A happy dog owner just got their first puppy and wants to provide the best care for it. However, they feel a little overwhelmed with the new responsibilities of owning a pet. They decided to download ToDo Today to keep track of these new tasks. The dog owner starts listing each task needed to get done for their dog and starts prioritizing each task. The dog owner wants to prioritize training their dog as they think they should train a dog as soon as possible. The dog owner creates the main task "Teach the puppy to sit" and creates subtasks such as: buying training treats, guiding the puppy to sit down, and rewarding the puppy with treats when sitting. The dog owner utilizes giving weights to each subtask and wants to start the task. They decide to go to the pet store to buy treats. Once they finished buying the training treats, they marked it

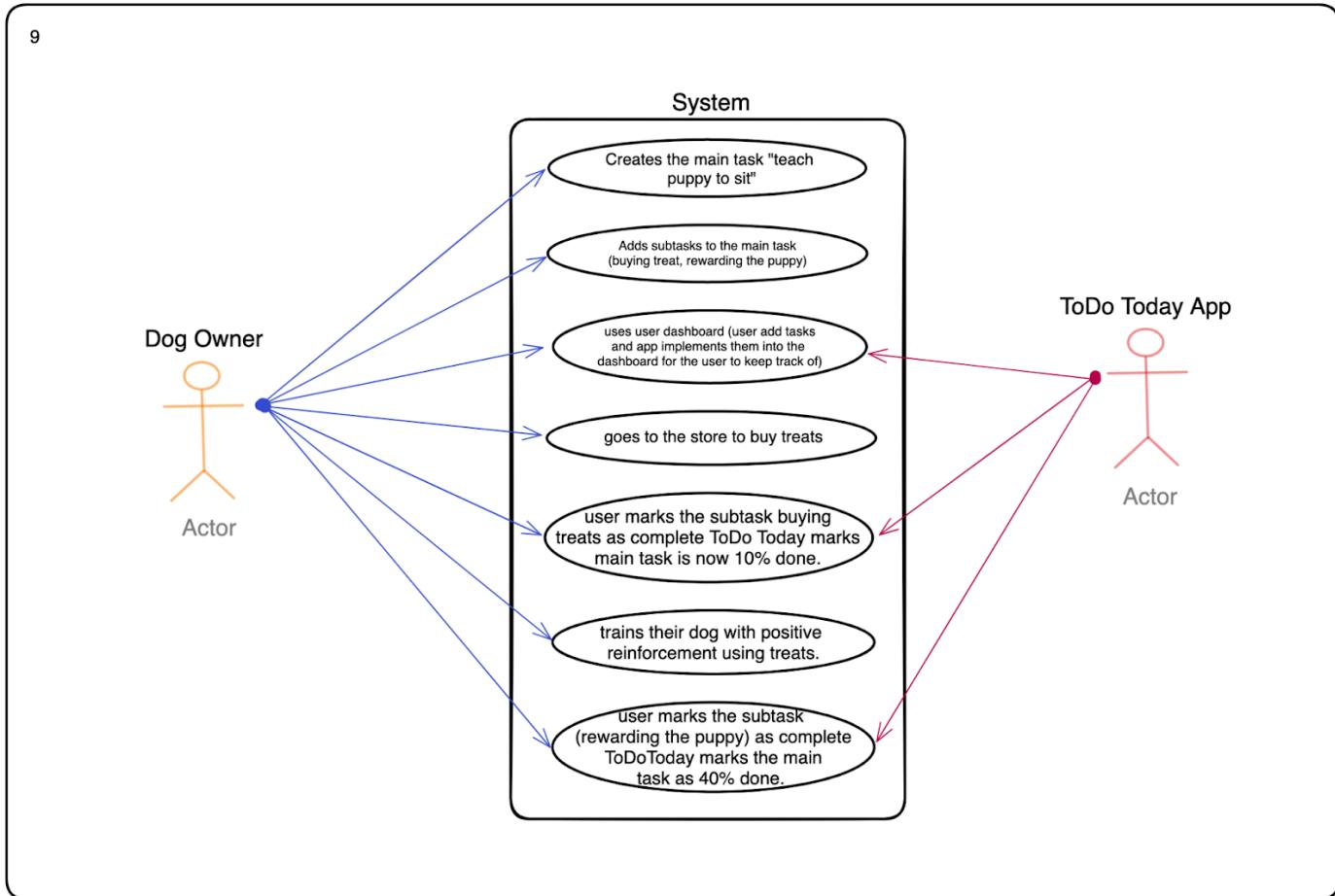
as complete and saw that they were now 10% done with the main task. When the dog owner arrives home, they start training their puppy right away. They train their dog with positive reinforcement using treats every time the dog sits down, marking the subtask complete and now at 40% total completion. For the next couple of hours, the dog owner would be trying to train the dog to sit on command while still using positive reinforcement. The dog owner finally sees progress and has successfully trained their dog to sit on command. Feeling joyful, they mark the last subtask as complete, showing 100% completion. After using the app, the dog owner was pleased with the experience of a progress bar for their task.

- **Benefits:**

- ToDoToday gave the dog owner a pleasant todo list experience with the progress bar
- Able to assign weights to each subtask
- The weights and priorities can determine which tasks are more important for the dog at that time
- Subtasks for bigger tasks such as training for a trick. The dog owner would be able to break that process down into different tasks
- Progress is shown through the bar visualization showing the dog's progress in learning tricks

- **Potential Requirements:**

- Nested Scheduling: The ability to have different lists that can be easily differentiable from one another. Maybe being able to name certain lists
- Task Percentage: Rather than fully checking off a task, provide a feature where they can check a certain amount of percentage and also be able to display the progress bar to the user.
- Priority Setting: A feature to set the priority status of a task. This would allow people to set more important tasks to have a higher priority vs the lesser priority ones
- Easy UI: The user interface should be easy to use and easy to find the different lists without looking too cluttered.
- Manageability: The user should be able to manage and change things such as priority status and other general things like the name of the task or where the task is.



IV. Main Data Items and Entities

Here are some of the main data items and entities.

1. Task:

- Meaning:** Since it is a to-do app, it should have at least a task. It represents an activity or item that needs to be completed.
- Usage:** Users can create, view, edit, and delete tasks within the application.
- Data Structure:** Contains fields such as task ID, description, due date, priority, status, associated user ID, and many more.

2. User:

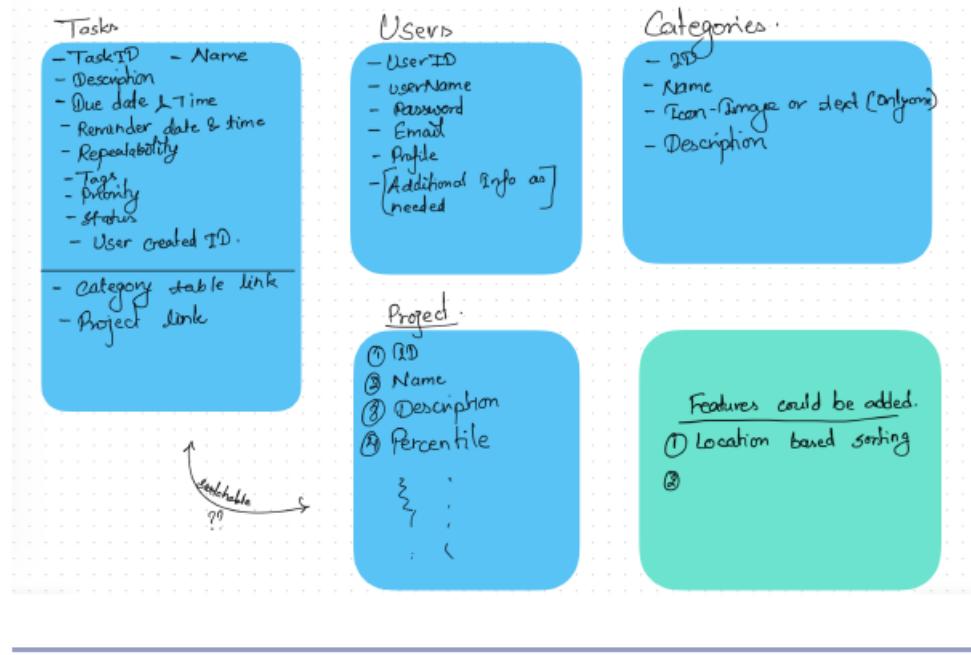
- Meaning:** Users are anyone who will be using the application.
- Usage:** Users can register, log in, and manage their tasks.
- Data Structure:** Includes fields like username, email, password, and possibly additional profile information.

3. Project:

- **Meaning:** Groups tasks together under a common theme or goal. This could be something that is interchangeable with tasks. If the task itself has a child task, it should be a project.
- **Usage:** Helps organize tasks and allows users to focus on specific objectives.
- **Data Structure:** Consists of a name, description, and possibly a list of associated tasks.

4. Category:

- **Meaning:** Classifies tasks into different types or categories. The user is the one categorizing the tasks.
- **Usage:** Aids in organizing tasks based on their nature or purpose.
- **Data Structure:** Contains a name and possibly a description.



Few More:

There are not-so-important items that in themselves aren't data items but do consist of categorical data.

1. Priority:

- **Meaning:** Indicates the level of importance or urgency assigned to a task.
- **Usage:** Helps users prioritize their tasks and focus on what needs to be done first.
- **Data Structure:** Typically includes levels such as high, medium, and low, each associated with a corresponding value or weight.

2. Status:

- **Meaning:** Reflects the current state or progress of a task.
- **Usage:** Allows users to track the status of their tasks, whether they are pending, in progress, or completed.
- **Data Structure:** Includes options like pending, in progress, completed, or archived.

3. Percentile:

- **Meaning:** Shows the completion percentage of tasks or projects.
- **Usage:** Provides users with a visual representation of their progress and helps them stay motivated.
- **Data Structure:** Represents the percentage of completed tasks relative to the total number of tasks in a project or category.

4. Deadline:

- **Meaning:** Specifies the date and time by which a task should be completed.
 - **Usage:** Helps users stay organized and meet their deadlines effectively.
 - **Data Structure:** Contains fields for the due date and possibly the time of day.
-

V. Initial List of Functional Requirements

Functional Requirements:

“Users shall be able to”

User

1. A user shall be able to register for a new account using a username.
2. A user shall be able to register for a new account using an email.
3. A user shall be able to register for a new account using a password.
4. A user shall accept the terms and conditions before creating an account.
5. A user shall be able to log into their account using their email/username and password.
6. A user shall be able to nest subtasks inside of other subtasks.
7. A user shall be able to access the “app” only when they have an internet connection.
8. A user shall be able to mark a task done while being able to specify the percentage at which that task has been completed.
9. A user shall be required to give a valid email upon creating an account.

10. A user shall be able to edit tasks
11. A user shall receive a notification for upcoming deadlines for tasks.
12. A user shall be able to add a new task, regardless if a previous task is incomplete.
13. A user shall be able to delete tasks
14. A user shall be able to name and rename their new and existing tasks.
15. A user shall be able to add new subtasks even if the parent task is complete.
16. A user shall be able to review their previous journal entries.
17. A user shall be able to undelete a task
18. A user shall be able to edit their profile information
19. A user shall be able to update their email address
20. A user shall be able to change their password

Task

21. A task shall be able to be deleted and added
22. A task shall be able to hold dates and times
23. A task shall be able to hold certain priority flags depending on the importance
24. A task shall be able to nest subtasks inside of other subtasks.
25. A task shall display the percentage completed
26. A task shall display the overall percentage completion of its subtasks
27. A task shall be raised up when given a high priority
28. A task shall be pushed down when given a low priority
29. A task shall warn the user if they reach the limit for subtasks

Journal

30. A Journal shall generate a prompt for the user from their highest-priority tasks
31. A Journal shall give the user a new entry and prompt each day.
32. A Journal shall ask the user how they can improve.
33. A Journal shall remind them to reflect on their work.
34. A Journal shall be deleted when the user chooses to delete it
35. A Journal shall be able to be undeleted within 12 hours of deletion
36. A Journal shall prompt a user to delete old journals if there is not enough space

Calendar

37. A calendar shall show the user's upcoming deadlines
38. A calendar shall notify the user of upcoming deadlines
39. A calendar shall keep track of passed deadlines as well as upcoming deadlines
40. A calendar shall be able to differentiate between todo lists

41. A calendar shall have a week view
42. A calendar shall have a month view
43. A calendar shall be able to transition between those two views.

Account

44. An account shall be created by one user
45. An account shall only be created by a user able to properly consent to our terms and conditions without the assistance of a parent or guardian
46. An account shall be created with a username.
47. An account shall be created with an email.
48. An account shall be created with a password.
49. An account shall be deleted upon the decision of the user
50. An account shall be accessible across devices (Desktop/Laptop)
51. An account's tasks shall be synchronized across devices (Desktop/Laptop)

List

52. A To Do list shall be able to delete tasks when the user chooses to delete it
 53. A To Do list shall upon deletion delete all tasks and subtasks beneath it.
 54. A To Do list shall be able to be differentiated from other To Do lists
 55. A To Do list shall be shareable
 56. A To Do list shall be importable
-

VI. List of Non-Functional Requirements

Non-functional requirements:

Usability:

- The user interface should be intuitive and easy to navigate, with feedback loop as needed.
- Prompt generation is run by a GPT2 API.

Compatibility:

- The Application should be compatible Firefox version 123
- If optimized for a desktop webpage environment, the consistent behavior and UI clues should be maintained.
- Optimization should be done with the use of native API's as much as possible.

Performance:

- The app should be able to run on a virtual system with 1 VCPU 1 GiB RAM efficiently without significant performance degradation.

Scalability:

- The system should be scalable to accommodate an increasing number of users and tasks without compromising performance.

Security:

- Critical user information should be encrypted.
- The app should be protected against security risks such as SQL injection.

Code maintainability:

- The codebase should be well-structured, modular, and thoroughly documented.

Privacy:

- User data should be handled securely and if possible, encrypted.
-

VII. Competitive Analysis

- **Competitive Analysis Table #1**

Feature/Company	Minimalist	Toodledo	Microsoft To Do	TeuxDeux	AnyDo	TickTick
Strengths	no login, great onboarding features and small text reminders on how to operate the app. The features are all intuitive and gesture based built specifically for the iPhone. One of the 1st results when you search for To Do List app on the App Store.	easy onboarding and good user experience once most features are learned, gives users enhanced to-do list experience with the amount of features it has. It is also multi-platform where users can use with iOS, Android, and the web.	Love the UI looks very clean and easy to use. There are quite a lot of customization options such as being able to change themes, sorting events or adding your very own custom themes.	Simple in nature making it easy for the user to get used to. Cloud based application allows for use on and syncing with multiple devices. Not OS specific so it is also available on the web.	Easy to add new tasks. Two task viewing modes. Importable calendar. Famili mode/plan allows you to sync up tasks with family members/others. A news/support tab.	Easy to use in terms of adding tasks and checking them off. There is a way to view your schedule through the calendar feature. Can add different lists and types of styles. The app allows for customization, letting the user choose color themes fonts, and app icons.
Weaknesses	no social media presence. No way to transfer A to do list between devices. A separate premium app that means you have to download an entirely new application instead of just upgrading the one that you currently have downloaded.	at first, user interface can seem a little confusing since there are so many features included in this app, a lot to learn to fully understand most features on the app	Limits flexibility of the app due to the user needing a microsoft account. This is especially a big issue considering that not all work related emails will be sent through one email.	Cannot set specific time for any task. Cannot write a description and summary for task. Cannot set reminder and notifications for task. Unable to make sub tasks and does not support partial completion.	Layout can be confusing, a lot of buttons and no guidance or tutorial to explain.	Does not have a subtask feature. Partial progression isn't implemented. Some of the features are behind a paywall. Email notifications can be annoying.
Pricing	\$0/month, free	\$0/month, free	\$0/month, free	This application is free on mobile and offers pricing option of \$4/month or \$36/year option for the web version.	Various Pricing Plans including, 1 month for \$9.99, billed monthly, 12 months for \$4.99, billed annually, 6 months for \$7.99, billed every 6 months	\$0/month, free but offers other paid features
Social Media	minima list has no social media presence. And neither does the individual who created it. With a bit of extra research I found that he makes and sells simple apps and video games but has no real social media presence.	Toodledo has "Help" section that includes: FAQ, a Forum page, and a Blog page. Not present on major social media platforms. Last Instagram post was on Aug 22, 2023. Semi-active on Twitter giving occasional updates and advertising their platform.	Microsoft to do has blog posts on Microsoft's official website for information for issues and for FAQ. These forum pages are helpful and are regularly updated and archived. They're able to do this because Microsoft already has a robust blogging and forum system already built that Microsoft to do just uses.	They have blog posts where they post on how to use the application and other interesting stories. They have Instagram and X (Twitter) accounts but not that active on them.	Semi-active on Instagram and Twitter (X) for their main page (@Anydo), also has a support account (@Anydo_Support) that posts product updates and general questions.	They have multiple social medias in the settings tab including Twitter, Facebook, Instagram and more.
Onboarding Experience	Onboarding is immediate and steps you right into the tutorial of how to operate the app. As the app is designed to be as minimalist as possible there are a few gestures that might not come immediately to mind so the tutorial walks you through everything and makes sure you understand how it works.	easy onboarding experience, account is simple to make: all you need is a name, email, password, and verify you are not a robot	The onboarding experience was immediate as long as you have a Microsoft account. It was about 3 clicks and then I was in and sitting at the home page. There was a single pop up for explaining what Microsoft to do was. But apart from that there was no other real onboarding of any kind.	seamless onboarding experience with very little friction. All you need is an email and password to create an account. They also offer 30 day free on web without any card information. That alone reduces all doubt for user who is thinking of trying it.	Just had to make an account using an email. After it asks how you plan to use it and then prompts to create a task to teach you how to use the basic features. After this got lost trying to find out about other features.	Prompted for me to create an account but you are able to use without. Creating tasks is intuitive due to the layout and the bottom taskbar makes it easier for maneuvering between list view, calendar view, and the settings.

• Competitive Analysis Table #2

Feature	Minimalist	Toodledo	Microsoft To Do	TeuxDeux	AnyDo	TickTick	Our Product
Priority Setting	-	+	+	-	+	+	+
Calendar Viewing	-	-	+	+	+	+	+
Widgets	-	-	-	-	+	++	-
Progress Bar	-	-	-	-	-	-	++
Journal Feature	-	-	-	-	-	-	++
Multiple Lists	+	+	+	+	+	+	+
Different Styles	+	+	+	-	+	+	-
SubTasking	+	-	-	-	-	-	++

1. Minimalist:

In minimalist using the app can be a little frustrating at times since it is designed to be as minimal as possible. The design has certain tasks and operations that are not bound to a specific key press or normal action, instead, they are bound to swipes taps, and shakes of your device. The visual layout is relatively clean clutter-free and pleasant to look at but sacrifices its usability in its pursuit of the absolute minimalist To Do List app. It allows you to create multiple folders and multiple to-do lists but this is only unlocked after you've created the 1st To Do List. The undo and delete functions are tied to the gyroscope of your phone you shake to delete. Which I feel is a little dangerous just in case you accidentally drop your phone or it misreads the vibrations of a particularly rough ride as you saying you want to delete A to-do list. One thing that the app does well though is that it allows you to have one main List task and then a couple of sub-tasks underneath it but it will only let you do one layer deep it will not let you do multiple sub-layers of subtasks nesting into each other.

2. Toodledo:

In Toodledo, adding a task is relatively simple as well as searching for a task, these features are at the top of the page and hard to miss. Creating side notes for a task is something Toodledo gives its users to add more context and clarification to a task. Toodledo also displays its task attributes on the main page such as task name, the folder the task is in, task start date, task due date, how frequently the task should be repeated, and task priority ranging from -1 to 3.

3. Microsoft To Do:

Microsoft to do software is solid and simple and a good baseline for what A to do list software should be. It has a simple and intuitive interface but little to no onboarding steps you literally open it up and it says welcome to Microsoft to do and that's about it there's no tutorial there's no walkthrough for how to access and how to set up your to do lists you have to figure it out yourself. One of the key things that makes Microsoft to do stands out from other To Do List applications is its integration into the Microsoft ecosystem. Since it is a first party product it has direct connections to outlook to other office suite products and allows for much more direct connections between them. As long as you have an Office 365 account you can use Microsoft to do for free. As it is sort of an add-on it's hard to define its price as an individual package.

4. TeuxDeux:

TeuxDeux is a straightforward to-do app that helps you stay organized, whether you're on your computer or on the go with your phone. Its simplicity makes it easy to use, and since it's cloud-based, you can access your tasks from anywhere and keep them synced across all your devices. So, my experience with this application was very smooth but I kept missing some of the major feature that I was used to with other applications. For instance, you can't set specific times for

tasks or add detailed descriptions. Reminders, sub-tasks, and the option for partial completion are also missing. While the mobile version is free, the web version offers additional features for \$4 per month or \$36 per year. That felt backward or opposite to me as that didn't make any sense. Despite not being very active on social media, TeuxDeux offers a hassle-free sign-up process and a generous 30-day free trial for the web version, making it easy for new users to give it a try and see if it suits their needs.

5. AnyDo:

The onboarding process was extremely quick. After giving it an email, your name, and making a password, it first asks if you're planning to use it just for yourself, or with a team. I chose the option called "Just me" and it prompted me to add my first tasks. At most, it's three tasks, but one task is enough to proceed. It briefly shows if I already want to make a subscription to upgrade to Premium by showing me the different prices, but I skipped it. After that, that was it, I already made multiple tasks, scheduled them on different times and days, and imported my calendar. However, I spent more time being lost and finding things out than it took for me to get started. It lets me import my calendar that's integrated with my phone, so now it shows me what's already on my calendar, but doesn't notify me of anything from it. It also has the option to use AI to "Suggest a task for me", which I'm not sure exactly what that would be like. It's a Premium function, so I haven't been able to test it. As of now, I haven't paid for anything, and I'm not sure I will in the future.

6. TickTick:

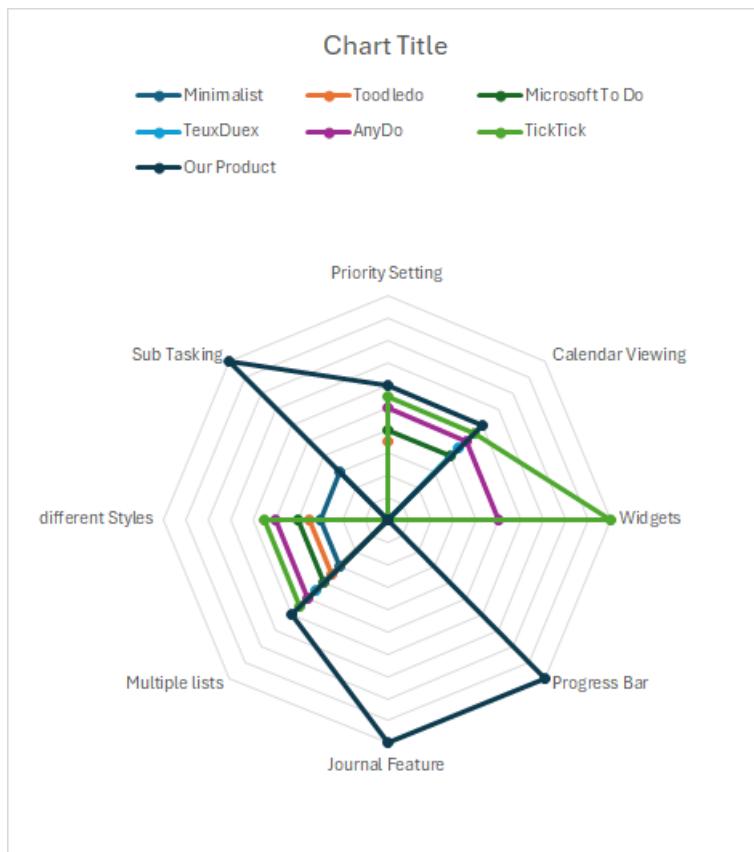
TickTick allows the user to make different lists for their differing needs. Making the tasks for the list is very simple and intuitive. You can set different priorities for the tasks depending on how urgent the task is. There is also a feature where you can view the tasks in a calendar view which makes it easy to view to the user. Other features include things such as a habit tracker and even a tool for using the Pomodoro technique to boost focus. The app also allows for many different customization features in the app regarding how the app looks visually through different themes.

Planned Advantages:

From our competitive analysis our application stands out in three key ways the first of which being the progress bar which allows for a more granular understanding of where you're at in a current project and how close you are to being finished. The next is our AI powered journaling feature which uses a GPT 2 open source API to generate journaling prompts to keep you on task and keep you thinking about what your next steps should be for your project. Finally there was only one other application that allowed you to create subtasks, that was minima list, and that application would only let you create a single layer of subtasks. Our apps implementation of subtasks will be more flexible as it going to use a recursive structure allowing for a sub task to have a subtask of its own and so on and so forth down to the granularity that you may need.

Overall our app stands out on those three key metrics allowing us to Stand out in an already crowded market.

Additional Graphs:



IX. Checklist

Task	Status
Team found a time slot to meet outside of the class	Done
GitHub master chosen	Done
Team decided and agreed together on using the listed SW tools and deployment server	Done
Team ready and able to use the chosen back and front-end frameworks and those who need to learn are working on learning and practicing	Done
Team lead ensured that all team members read the final M1 and agree/understand it before submission	Done
Github is organized as discussed in class (e.g. master branch, development branch, a folder for milestone documents, etc.)	Done

X. High-level Architecture

1. **Software stack:**
 2. **Server Host:** Amazon AWS EC2 1VCPU 1 GiB RAM
 3. **Operating System:** Ubuntu Server 22.04 LTS
 4. **Database:** MySQL Community Server 8.0.33
 5. **Web Server:** Node.js 20.11.1 and Express.js 4.18.2
 6. **Server-Side Language:** JavaScript
 7. **Additional Technologies:**
 - **Web Framework:** React
 - **IDE:** VSCode, Webstorm
 - **APIs:** Huggingface GPT2 API
 - **Packet manager:** NPM
 - **Password encryption:** bcrypt 5.1.1
-

XI. List of team contributions (Milestone 1):

Name	Code	Documentation
Alex Nikols	2	7
Jason Tran	2	5
Sulav Jung Hamal	10	9
Jadon Hoang	2	6
Randale Reyes	2	5
Christian Gopez	2	7

XII. Data Definitions

- **ToDoToday:** The application name.
- **Dashboard:**
 - The main page of the application where the tasks are displayed.
- **Calendar:**

- The calendar allows for tasks to be displayed on the dates they are assigned.
- **Journal:**
 - The journal is where the user would be able to write down their journal entries and view past entries.
- **Prompt:**
 - A prompt will be provided to the user for the journal entry.
- **Journal Entries:**
 - Journal entries would hold the input the user enters for the journal.
- **ToDo List:**
 - The ToDo List is what will be holding all the tasks that the user creates.
- **Todoltem:**
 - the Todoltem class represents individual tasks with various attributes such as priority, status, and due date.
- **Main Task:**
 - A Main Task would be a task that can hold subtasks or also be independent itself.
- **Sub Task:**
 - A Sub Task would be tasked under the main task usually splitting the main task into separate parts.
- **Priorities:**
 - Priorities are used to differentiate the importance of the different tasks
 - High: Very Important
 - Medium: Important
 - Low: Not as Important as now
- **Progression Percentage:**
 - This would be used to name the progression bar that we plan to implement. The progression percentage will show the progress that the user has on certain tasks visually through a progression bar.
- **Due Date**
- **Log In**
- **Log Out**
- **Register**
- **Username**
- **Email**
- **Password**
- **Account**
- **Settings**

XIII. Prioritized Functional Requirements

Priority 1:

User:

1. A user shall be able to register for a new account using a username.
2. A user shall be able to register for a new account using a valid email.
3. A user shall be able to register for a new account using a password.
4. A user shall be able to accept the terms and conditions.
5. A user shall be able to log into their account using their email.
6. A user shall be able to log into their account using their username.
7. A user shall be able to log into their account using their password.
8. A user shall be able to change their password.

Task:

9. A task shall be able to be deleted.
10. A task shall be able to be added.
11. A task shall display the percentage completed.
12. A task shall display the overall percentage completion of its subtasks.
13. A task shall be able to nest subtasks inside of other subtasks.
14. A task shall display the progress of a task using a progress bar.
15. A task shall update depending on the subtask's progress.

Journal:

16. A Journal shall generate a prompt for the user based on their higher-priority tasks.
17. A Journal shall have an input section for user reflections.
18. A journal shall be accessible by an account.
19. A Journal shall contain a prompt generated from the user's top 5 priority tasks.

Calendar:

20. A calendar shall have a monthly view.
21. A calendar shall present the title of the task on the task's due date.
22. A calendar shall have the due date of the task on the day the task is due.

Account:

23. An account shall be created by one user.
24. An account shall only be created by a user able to properly consent to our terms and conditions without the assistance of a parent or guardian.
25. An account shall be created with a username.
26. An account shall be created with an email.
27. An account shall be created with a password.
28. An account shall be created with a confirmed password.
29. An account shall be able to access the “app” only when they have an internet connection.
30. An account shall be able to mark tasks as done.
31. An account shall be able to delete tasks.
32. An account shall be able to name new tasks.
33. An account shall be able to rename existing tasks.
34. An account shall be able to add a new task, regardless if a previous task is incomplete.
35. An account shall have multiple journal entries.
36. An account shall display the account username.

ToDo List:

37. A ToDo list shall be searchable by the user.
38. A ToDo list shall delete all subtasks beneath it upon deletion of a task.
39. A ToDo list shall upon deletion of a to-do list delete all tasks inside that list.

Settings:

40. A setting shall allow the user to change their username.
41. A setting shall allow the user to change their email.
42. A setting shall allow the user to change their password.

Dashboard:

43. A dashboard shall allow the user to see an overview of their tasks.
 44. A dashboard shall allow the user to see an overview of their upcoming tasks.
-

Priority 2:

User:

1. A user shall be able to edit their profile information.
2. A user shall be able to update their email address.

Task:

3. A task shall be able to hold dates and times.
4. A task shall be able to hold certain priority flags depending on the importance as high, medium, or low.
5. A task shall be raised up when given a high priority.
6. A task's deadline shall be able to be shown in the calendar.

Journal:

7. A Journal shall ask the user how they can improve.
8. A Journal shall remind them to reflect on their work.
9. A Journal shall be deleted when the user chooses to delete it.
10. A journal shall have a collection of previous journal entries from the user.
11. A journal shall have a prompt of less than 150 words.
12. A journal shall have an open-ended prompt.
13. A journal shall have a date created.

Calendar:

14. A calendar shall show the user's upcoming deadlines.
15. A calendar shall keep track of passed deadlines as well as upcoming deadlines.
16. A calendar shall present the time the task is due on the day of the task's due date.
17. A calendar shall present the priority of the task on the day of the task's due date.
18. A calendar shall have color-coded tasks.

Account:

19. An account shall be deleted upon the decision of the user.
20. An account shall be able to review their previous journal entries.
21. An account shall be able to specify completion via percentage.
22. An account shall be able to sort through all their tasks by priority.
23. An account shall be able to search through all their subtasks and main tasks.
24. An account shall have a profile icon.
25. An account shall allow the user to edit profile information.
26. An account shall display the account icon.
27. An account shall have a delete profile option.

ToDo List:

28. A ToDo list shall be able to delete a task when the user chooses to delete it.
29. A ToDo list shall be able to be renamed by the user.

30. A ToDo list shall be named by the user upon creation.
31. A ToDo list shall have a name to differentiate it from other To Do lists.

Dashboard:

32. A dashboard shall allow the user to see an overview of their overdue tasks.
 33. A dashboard shall allow the user to mark completed quickly and easily.
-

Priority 3:

Task:

1. A task shall be pushed down when given a low priority.
2. A task shall warn the user if they reach the limit for subtasks.
3. A task shall have a category listed.
4. A task shall be able to be color-coded.

Journal:

5. A Journal shall be searchable by date.
6. A Journal shall be able to be undeleted within 12 hours of deletion.
7. A Journal shall prompt a user to delete old journals if there is not enough space.
8. A Journal shall mark days that the user has not written in.

Calendar:

9. A calendar shall be able to differentiate between todo lists.
10. A calendar shall have a weekly view.
11. A calendar shall be able to transition between different views.
12. A calendar shall have a QR code that will allow the user to view a mobile view of their calendar.
13. A calendar shall have an add task button allowing the user to add tasks from the calendar view.
14. A calendar shall show the main or sub-task completion percentage on their due date.
15. A calendar shall be able to mark tasks as complete.

Account:

16. An account shall be able to undelete a task.
17. An account shall be able to add new subtasks even if the parent task is complete.

18. An account shall be able to sort through all their tasks by percentage.
19. An account shall be able to sort through all their tasks by date.
20. An account shall be able to sort through all their tasks by subtask.
21. An account shall be able to sort through all their tasks by main task.
22. An account shall be able to sort through all their tasks by amount of subtasks.
23. An account shall be able to add an icon next to their task.
24. An account shall be able to color code their tasks.
25. An account shall display how many in-progress tasks the user has.
26. An account shall display how many high, medium, and low tasks the user has completed.

ToDo List:

27. A ToDo list shall be shareable.
28. A ToDo list shall be importable.
29. A ToDo list shall have a QR code that will allow the user to view a mobile view of their ToDo list.
30. A ToDo list shall be exportable as a static PDF.
31. A ToDo list shall be printable.

Settings:

32. A setting shall allow the user to change the app from light mode to dark mode.
33. A setting page shall allow the user to set the default dashboard page on login.
34. A setting page shall allow the user to set the default view mode for all task views.

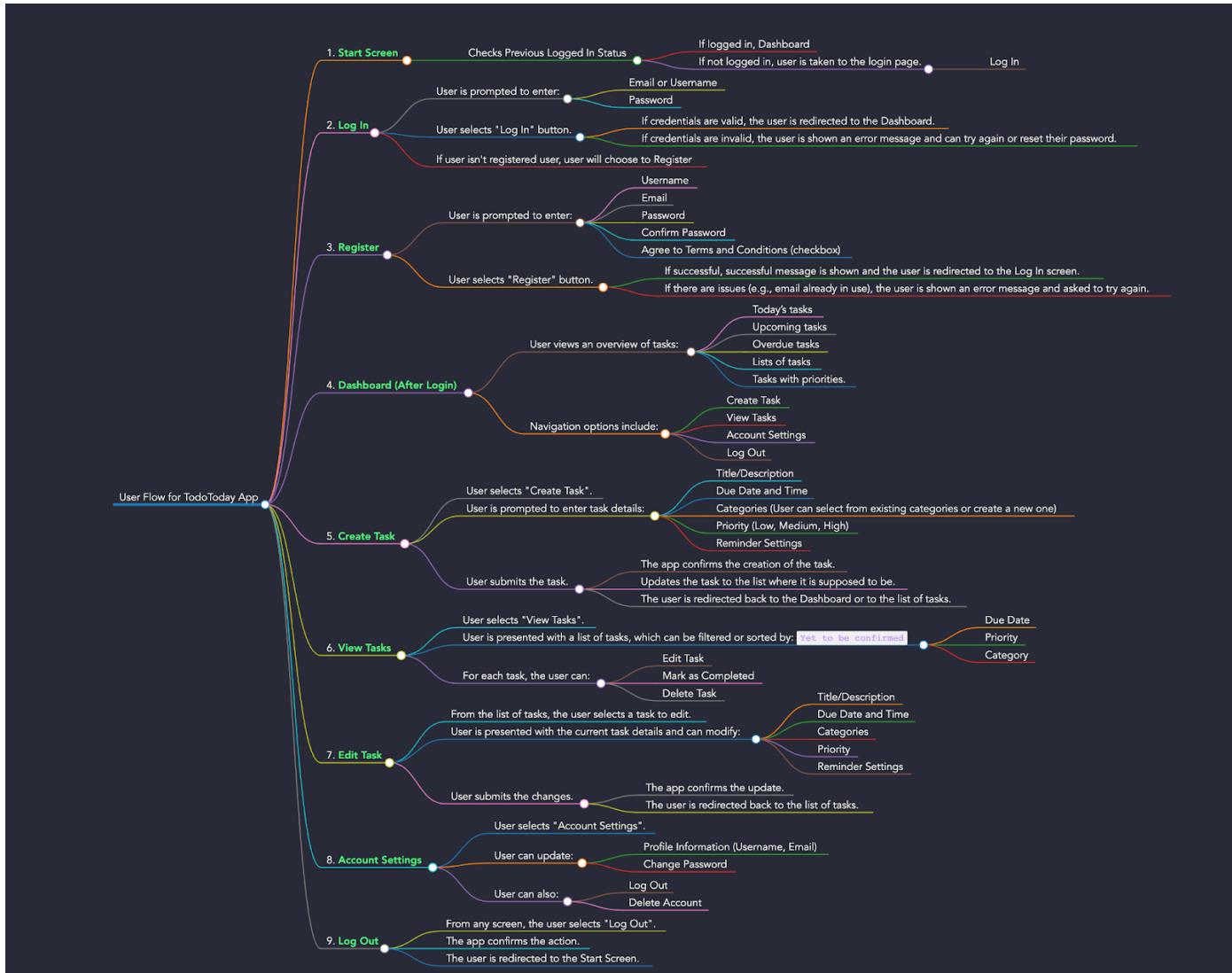
Dashboard:

35. A dashboard shall allow the user to see an overview of their completed tasks.
36. A dashboard shall allow the user to sort tasks based on due date.
37. A dashboard shall allow the user to sort tasks based on priority.
38. A dashboard shall allow the user to view stubs for their tasks on a mini calendar.
39. A dashboard mini calendar view shall allow the user to click and mark it complete from within the view.
40. A dashboard mini calendar view shall allow the user to move the task at different times and dates.
41. A dashboard mini calendar view shall allow the user to see the priority of the task.
42. A dashboard shall allow the user to toggle between different categories of tasks.
43. A dashboard shall reflect the newly added task in the appropriate category.

XIV. UI Mockups and Storyboards (high level only)

UI Mockups:

User Flow Diagram



Register Account Page:

ToDo
ToDay

Register Account

Username
Enter username

Email
Enter email

Password
Enter Password

Confirm Password
Confirm Password

I agree to the Terms of Service and
Privacy Policy.

Register or **Login**

Login Page:

ToDo
ToDay

Login

Email / Username
Enter email or username

Password
Password

Login

or

Register Account

Account Page:

Account

Settings

Username
@username Change

Email
email@email.com Change

Password Change

Logout

Change Username X

New Username...

Save

Change Email X

New Email...

Save

Change Password X

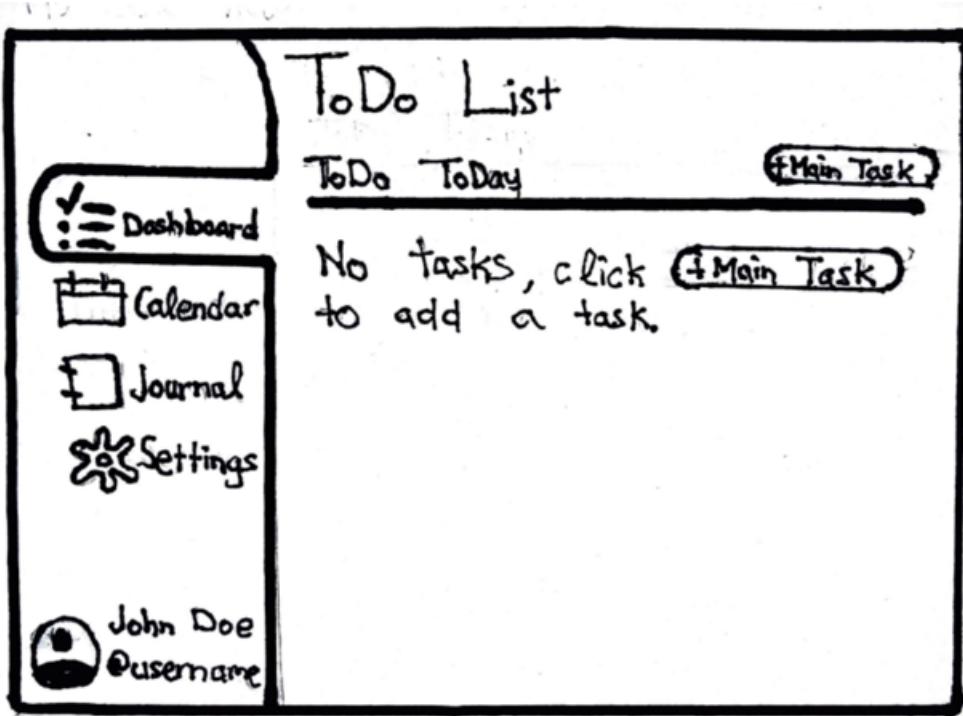
Current Password...

New Password...

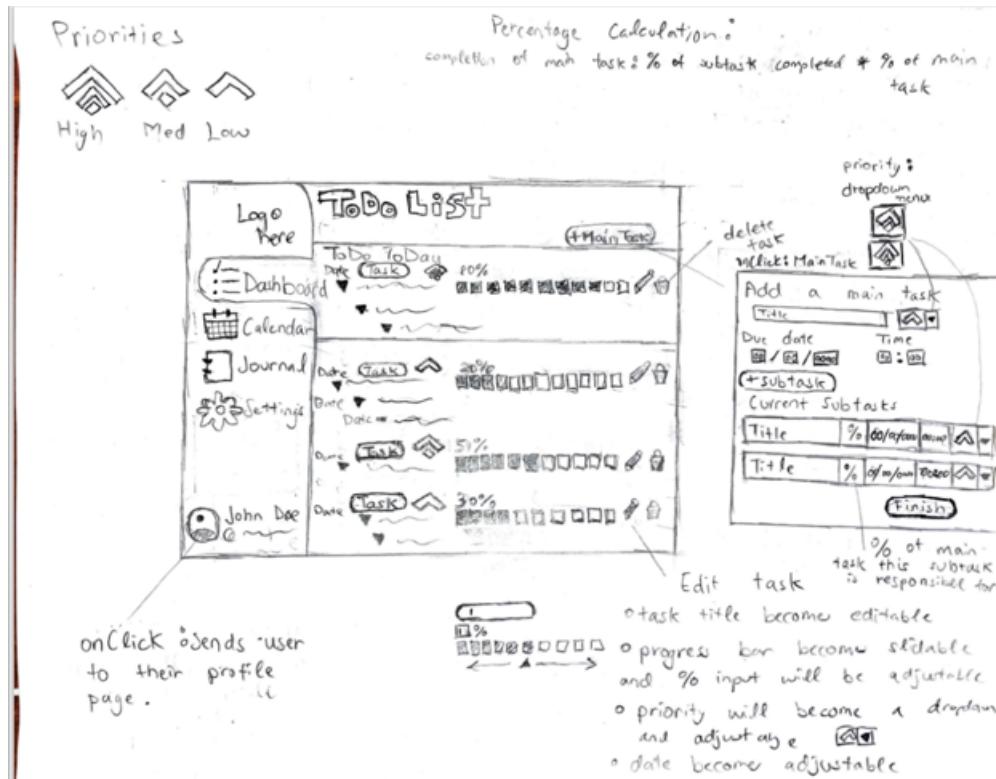
Confirm New Password...

Save

Empty Dashboard Mockup:



Dashboard Mockup:



OnClick + Main Task Mockup:

Add a main task

Title	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Due date	D/□/□	
Time	□:□□	

+ Subtask

Current Subtasks

Title	+ / /	:		
Title	+ / /	:		

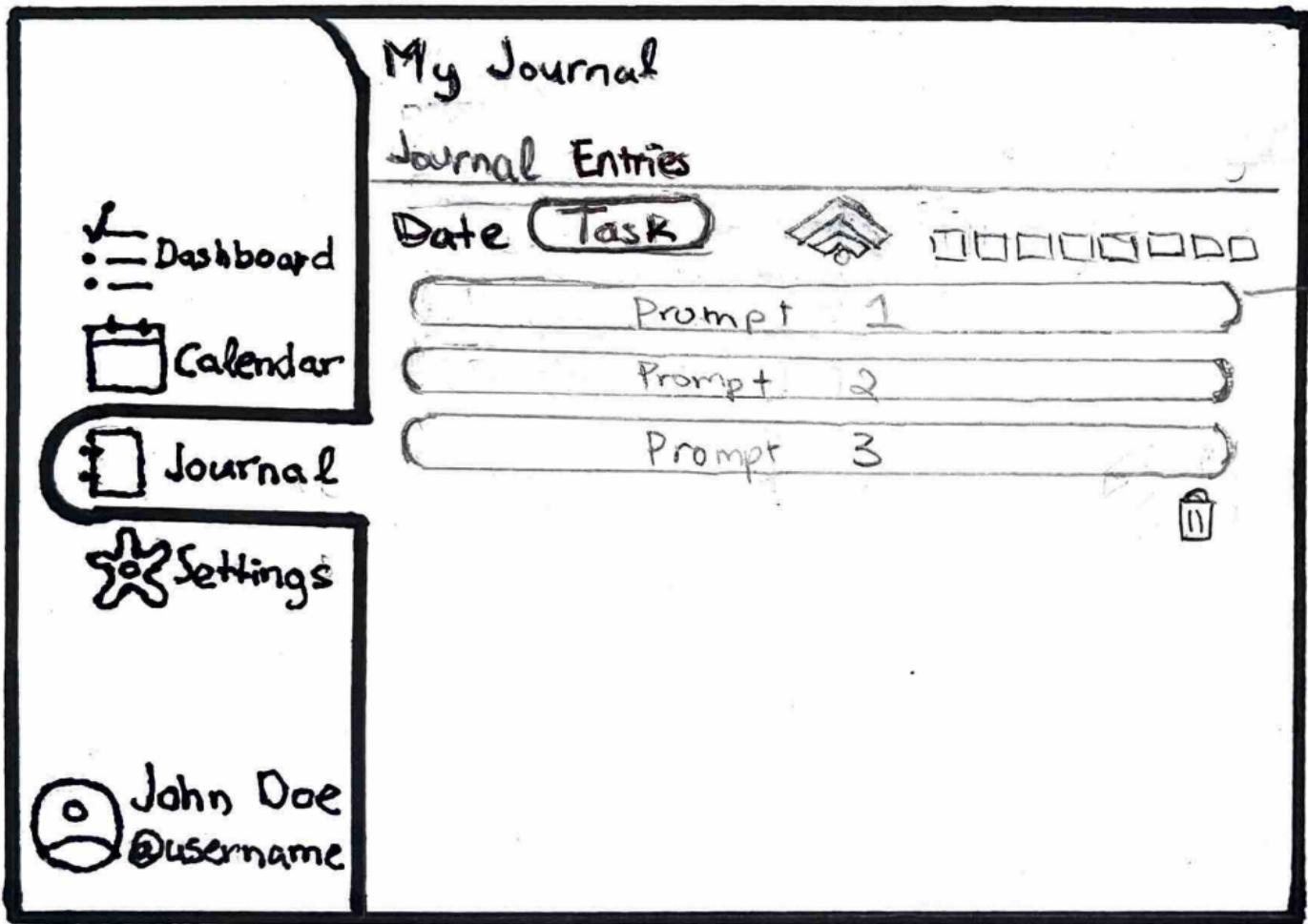
Finish

Calendar View Mockup:

A hand-drawn mockup of a mobile application interface. On the left is a sidebar with a logo at the top, followed by five menu items: Dashboard, Calendar (which is selected and highlighted with a red oval), Journal, Settings, and a user profile section for John Doe (@username). The main area is titled "Month" and contains a 4x7 grid calendar for a month. The days of the week are labeled: Mon, Tue, Wed, Thu, Fri, Sat, Sun. The dates are filled in sequentially from 1 to 31. A horizontal arrow points right above the 7th column, and a vertical arrow points down above the 4th row.

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Journal View Mockup:



On Click & Reflections

Reflections

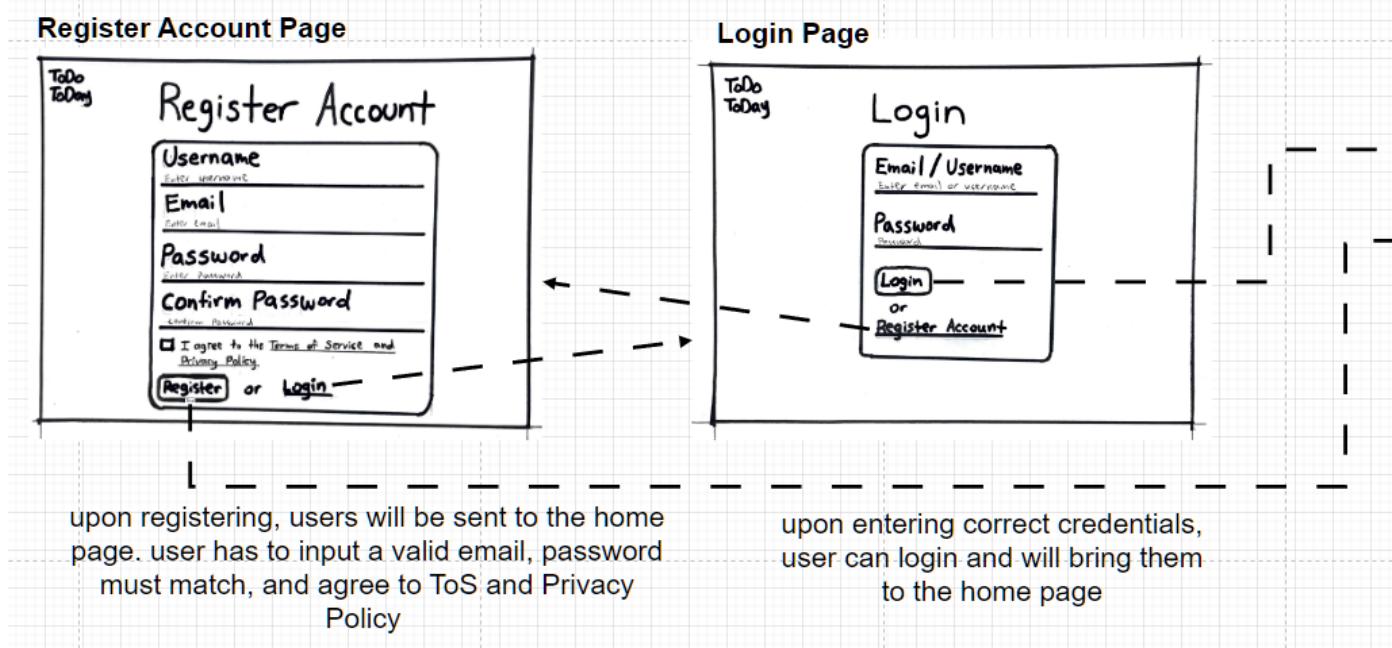
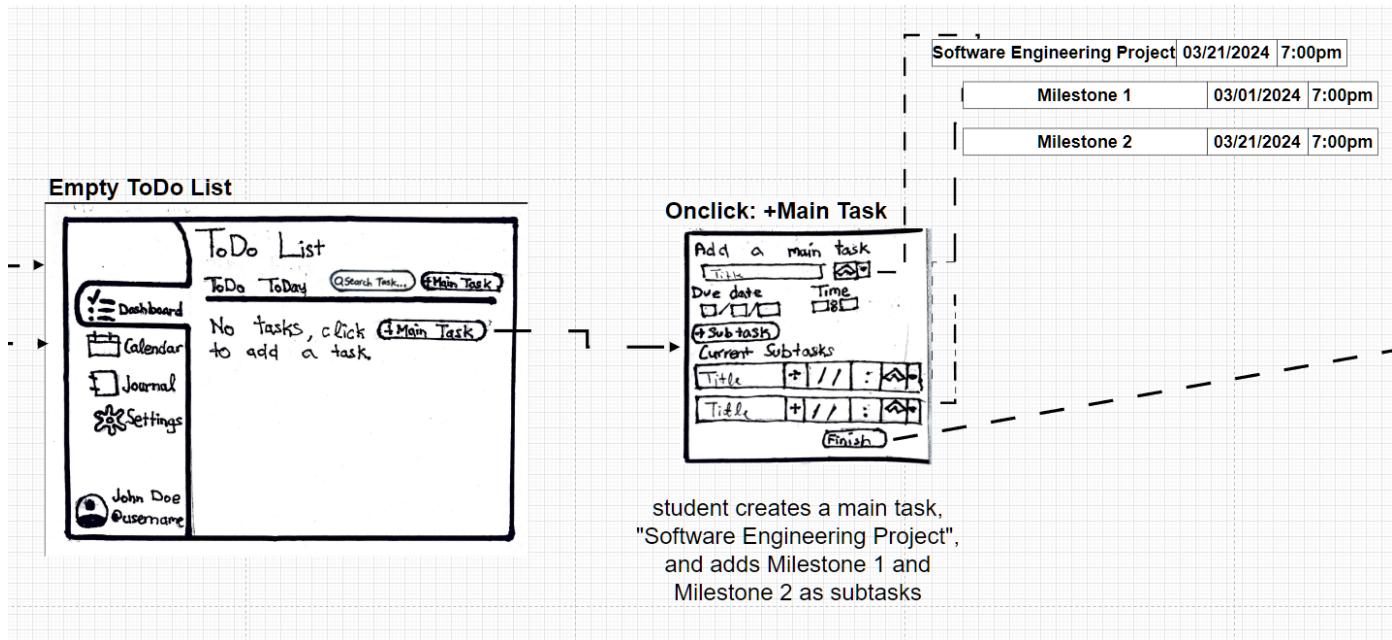
Type your reflections here...

Save

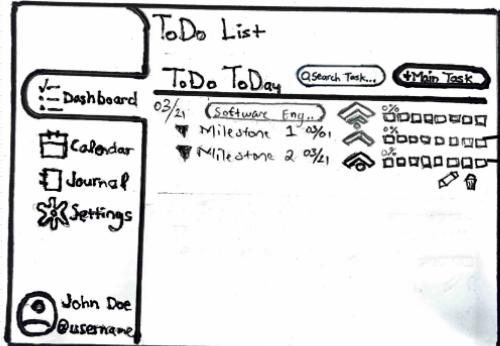
Use Case Storyboards:

The storyboards progress from left to right while moving downwards.

Use Case 1:

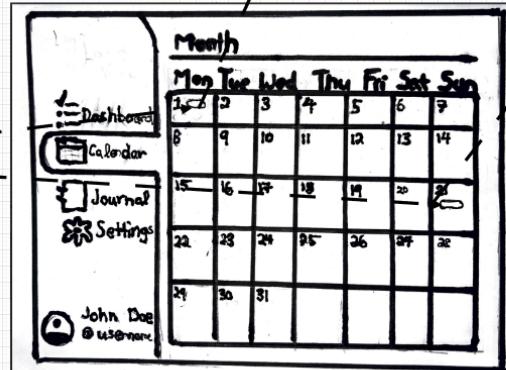


Added main task and subtask to Dashboard



student clicks finish and the main task along with its subtasks now appears in the dashboard showing the due dates, priorities, and progress bar of each task

Calendar View

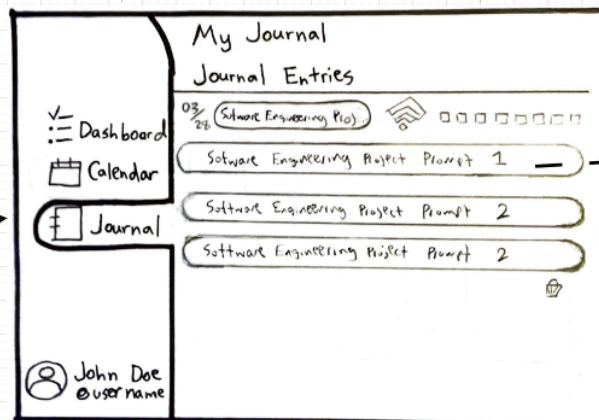


student checks the calendar and each subtask appears on its due date

Software Engineering Project Prompt 1

Type your reflections here...

Save



the student can use the journal to make reflections and talk about what he worked on and how he can improve

Milestone 1

Milestone 2

Use Case 2:

Register Account Page

Register Account

Username
Email
Password
Confirm Password
 I agree to the Terms of Service and Privacy Policy.
Register or Login

Login Page

Login

Email / Username
Password
Login or Register Account

upon registering, users will be sent to the home page. user has to input a valid email, password must match, and agree to ToS and Privacy Policy

upon entering correct credentials, user can login and will bring them to the home page

Empty ToDo List

Dashboard
Calendar
Journal
Settings
John Doe
Username

ToDo List
ToDo Today (Search Task...) +Main Task
No tasks, click +Main Task to add a task.

a mother wants to keep track of her two son's schedules so she creates a main task called "Alex's Schedule" and adds his schedule as subtasks

On Click: +Main Task

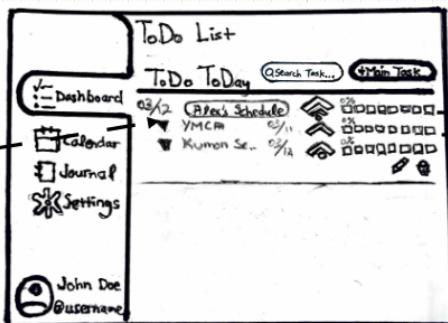
Add a main task
Title
Due date
Time
+Sub tasks
Current Subtasks
Title
Title

Alex's Schedule 03/12/2024 7:00pm

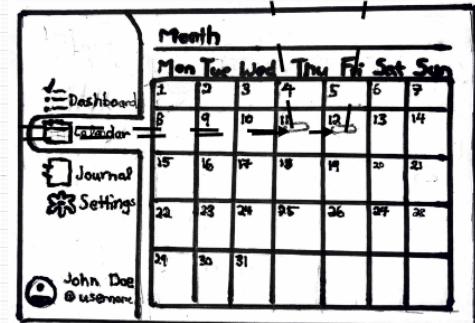
YMCA 03/11/2024 3:00pm

Kumon Session 03/12/2024 3:00pm

Added main task and subtask to Dashboard



Added subtasks to Month view



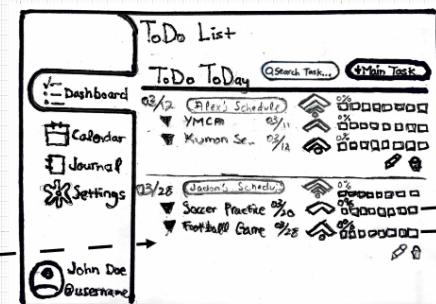
"Alex's Schedule" is now in the home page along with its subtasks and can be viewed in the calendar

On click: +Main Task

the mother now creates a second main task for her other son titling it "Jadon's Schedule" and creates subtasks for this schedule

Jadon's Schedule 03/28/2024 7:00pm

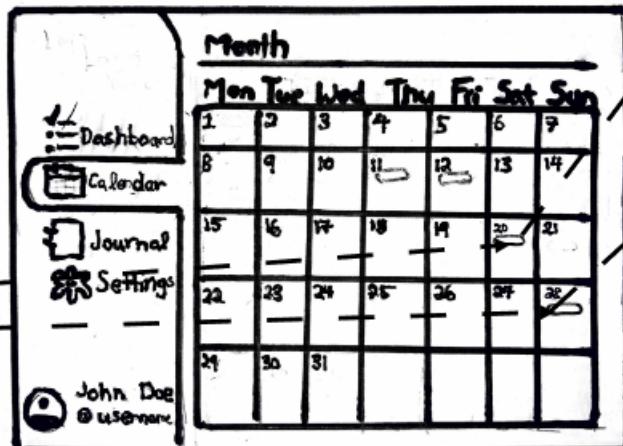
- Soccer Practice 03/20/2024 3:00pm
- Football Game 03/28/2024 9:00pm



both tasks now appear in the dashboard with their corresponding due dates, progress bars, and priorities

Soccer Practice

Football Game

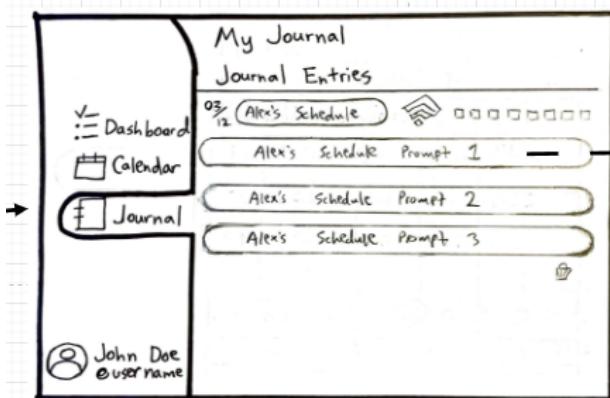


the calendar shows each task
on its respective due date,
making it easier for the mother
to keep track of her son's
schedules

Alex's Schedule - Prompt 1

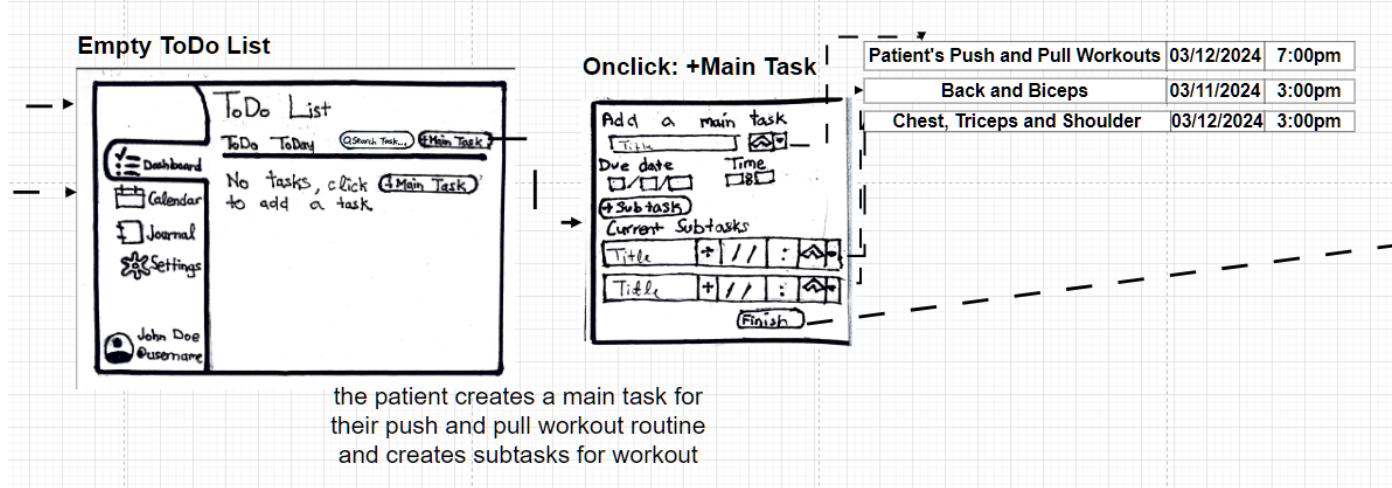
Type your reflections here...

Save



the mother uses the journal to
reflect on how she manages her
son's schedule

Use Case 3:



Register Account Page

Register Account

Username
Email
Password
Confirm Password
I agree to the Terms of Service and Privacy Policy.
Register or Login

upon registering, users will be sent to the home page. user has to input a valid email, password must match, and agree to ToS and Privacy Policy

Login Page

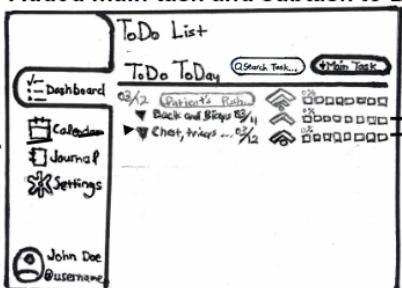
Login

Email / Username
Password
Login or **Register Account**

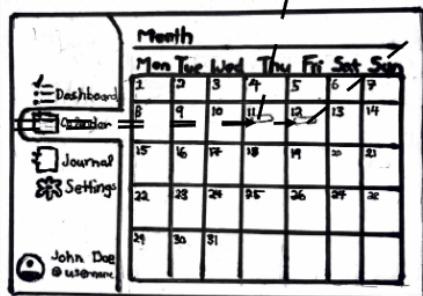
upon entering correct credentials, user can login and will bring them to the home page

Back and Biceps

Added main task and subtask to Dashboard

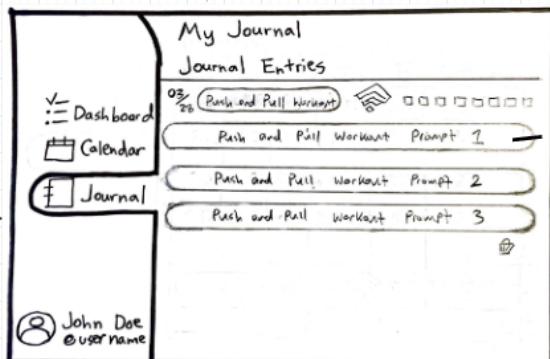


the workout routine is displayed on the dashboard



the patient utilizes the calendar to have a better visualization of their workouts throughout the week

Chest, Triceps and Shoulder



the patient uses the journal feature to talk about what they like and didn't like about their workout

Push and Pull Workout Prompt 1

Type your reflections here...

Save

Use Case 4:

Register Account Page

Login Page

upon registering, users will be sent to the home page. user has to input a valid email, password must match, and agree to ToS and Privacy Policy

upon entering correct credentials, user can login and will bring them to the home page

Empty ToDo List

On click: +Main Task

UI/UX Design 03/12/2024 7:00pm

Team 1 03/12/2024 7:00pm

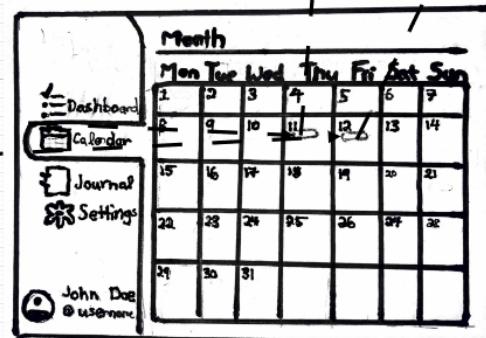
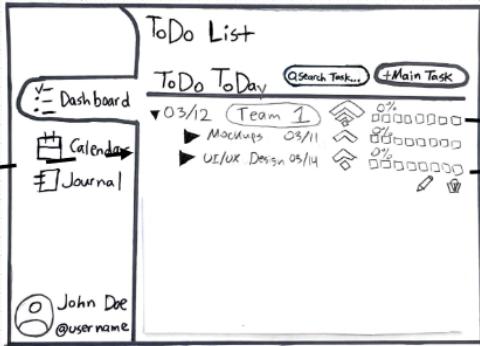
Mock ups 03/11/2024 3:00pm

a project manager has multiple teams to manage and creates Team 1 as a main task while the subtasks are Mockups and UI/UX Design

Mockups

UI/UX Design

Added main task and subtask to Dashboard



the due dates of the Mockups and
UI/UX Design can now be seen in
the calendar view

On click: +Main Task

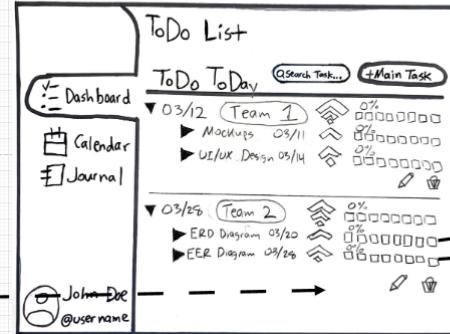
A modal dialog box titled 'Add a main task'. It has fields for 'Title' (with a placeholder 'ERD Diagram'), 'Due date' (set to '03/20/2024'), 'Time' (set to '3:00pm'), and 'Current Subtasks'. Below these are two rows of input fields for 'Title' and 'Time' with a 'Finish' button at the bottom.

Team 2 03/28/2024 7:00pm

ERD Diagram 03/20/2024 3:00pm

EER Diagram 03/28/2024

9:00pm



the project manager adds Team 2
with ERD Diagram and EER
Diagram as the subtasks

ERD Diagram

EER Diagram

Month

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

John Doe @username

ToDo List

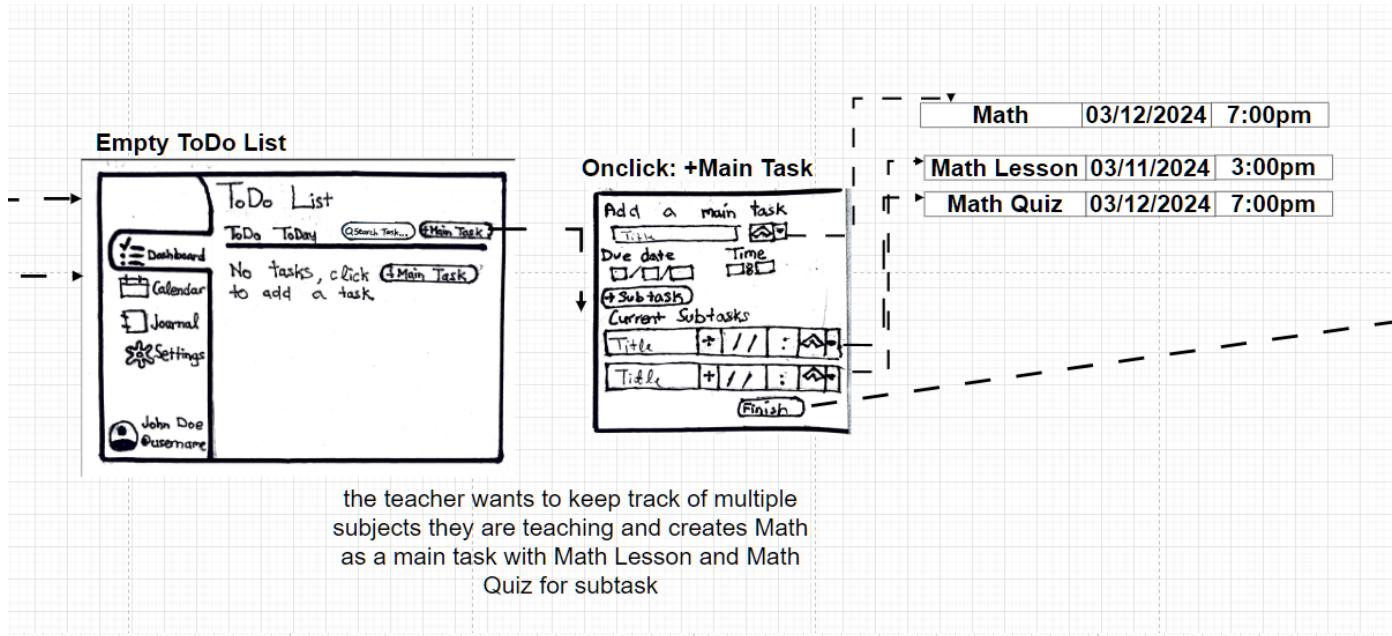
To Do	To Day	Search Task...	Main Task
▼ 03/12 Team 1			
► Mockups	03/11	33%	50%
► UI/UX Design	03/14	25%	75%
▼ 03/20 Team 2			
► ERD Diagram	03/20	50%	50%
► EER Diagram	03/20	75%	25%

John Doe @username

all the subtasks for Team 1 and Team 2 are now seen in the same calendar view

as each team progresses through their subtasks, the project manager marks them partially complete with the progress bar, increasing the progress bar for each team

Use Case 5:



Register Account Page

Register Account

Username
Email
Password
Confirm Password
 I agree to the Terms of Service and Privacy Policy
Register or login

upon registering, users will be sent to the home page. user has to input a valid email, password must match, and agree to ToS and Privacy Policy

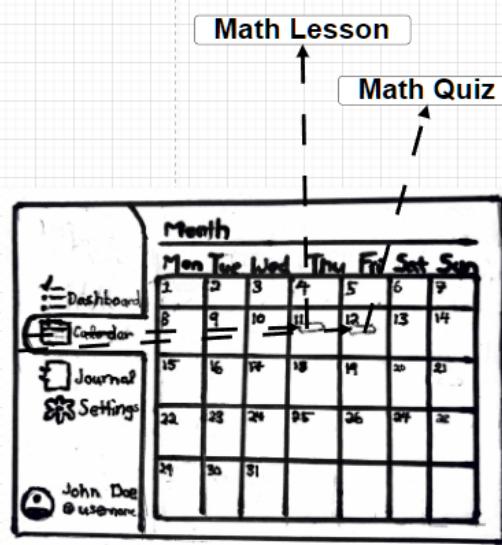
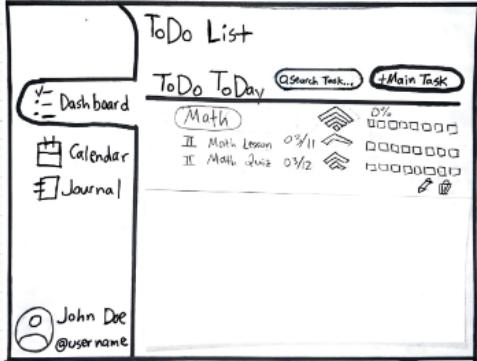
Login Page

Login

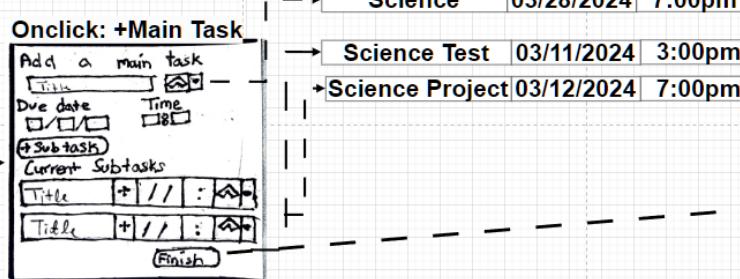
Email / Username
Password
Login or **Register Account**

upon entering correct credentials, user can login and will bring them to the home page

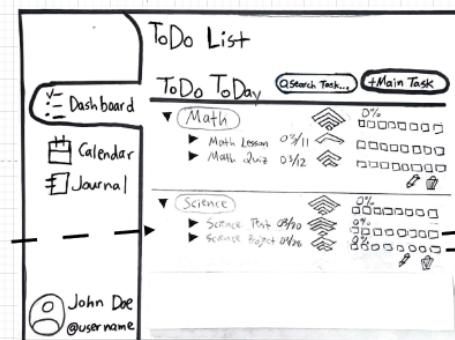
Added main task and subtask to Dashboard



Added main task and subtask to Dashboard

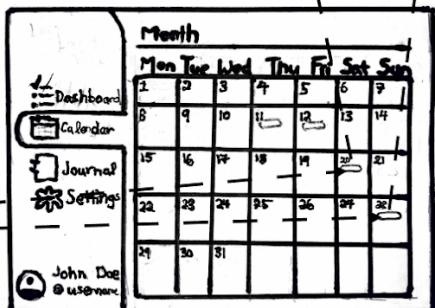


teacher creates main task Science
and adds Science Test and
Science Project

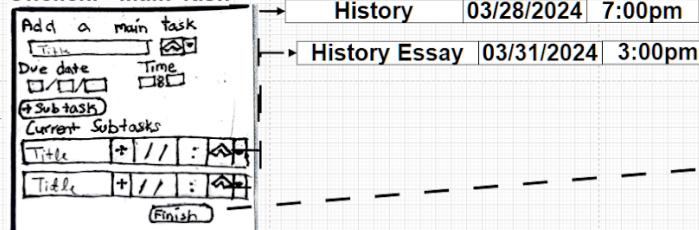


Science Test

Science Project



OnClick: +Main Task



lastly, the teacher creates a History
task to keep track of when the History
Essay is due for his students

Added main task and subtask to Dashboard

all the tasks and subtasks can be seen in the dashboard with each priority next to each task as well as in the calendar for better tracking and organization for the teacher

My Journal

the teacher uses the journal feature to talk about things she liked about her lessons or things that she could improve on

Use Case 6:

Register Account Page

Login Page

upon registering, users will be sent to the home page. user has to input a valid email, password must match, and agree to ToS and Privacy Policy

upon entering correct credentials, user can login and will bring them to the home page

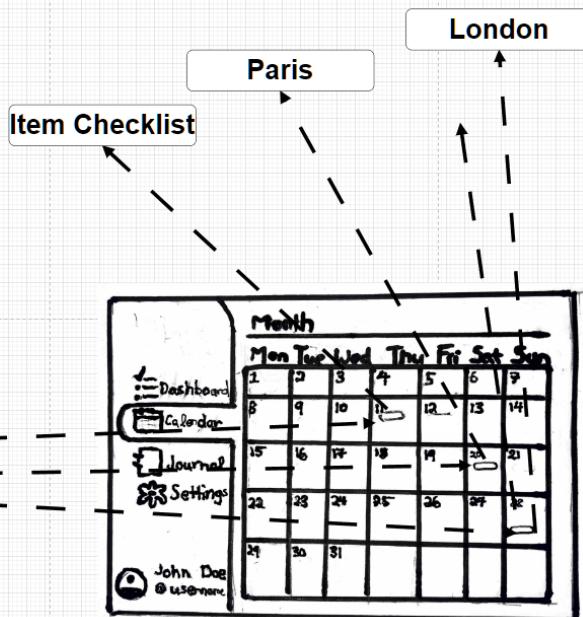
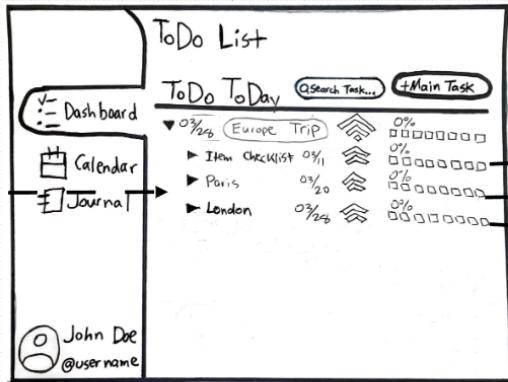
Empty ToDo List

Sarah is planning a Europe trip. She wants to be more organized and wants a better way to keep track of her trip. She decides to use ToDoToday

On click: +Main Task

Sarah creates Europe Trip as the main task. She Item Checklist, Paris, and London as the subtasks

Added main task and subtask to Dashboard



On click: +Subtask Item Checklist

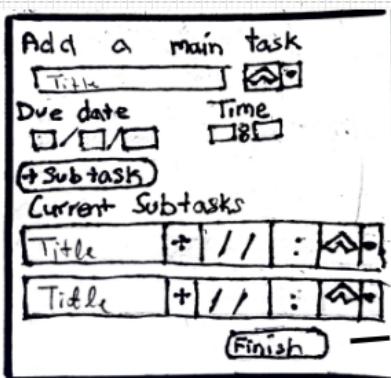
The screenshot shows a modal dialog titled 'Add a main task'. It has fields for 'Title' and 'Due date' (with a date picker), and a 'Time' field. Below these is a section for '+Sub-task' with a heading 'Current Subtasks'. It shows two rows of subtask fields, each with 'Title' and a checkbox. A 'Finish' button is at the bottom.

Clothes 03/11/2024

- Toothbrush 03/11/2024
- Shoes 03/11/2024
- Vlog Camera 03/11/2024
- Passport 03/11/2024
- Phone 03/11/2024

Sarah wants to make sure she has everything she needs for her trip.
She adds items to the Item Checklist as subtasks

Onclick: +Subtask Paris

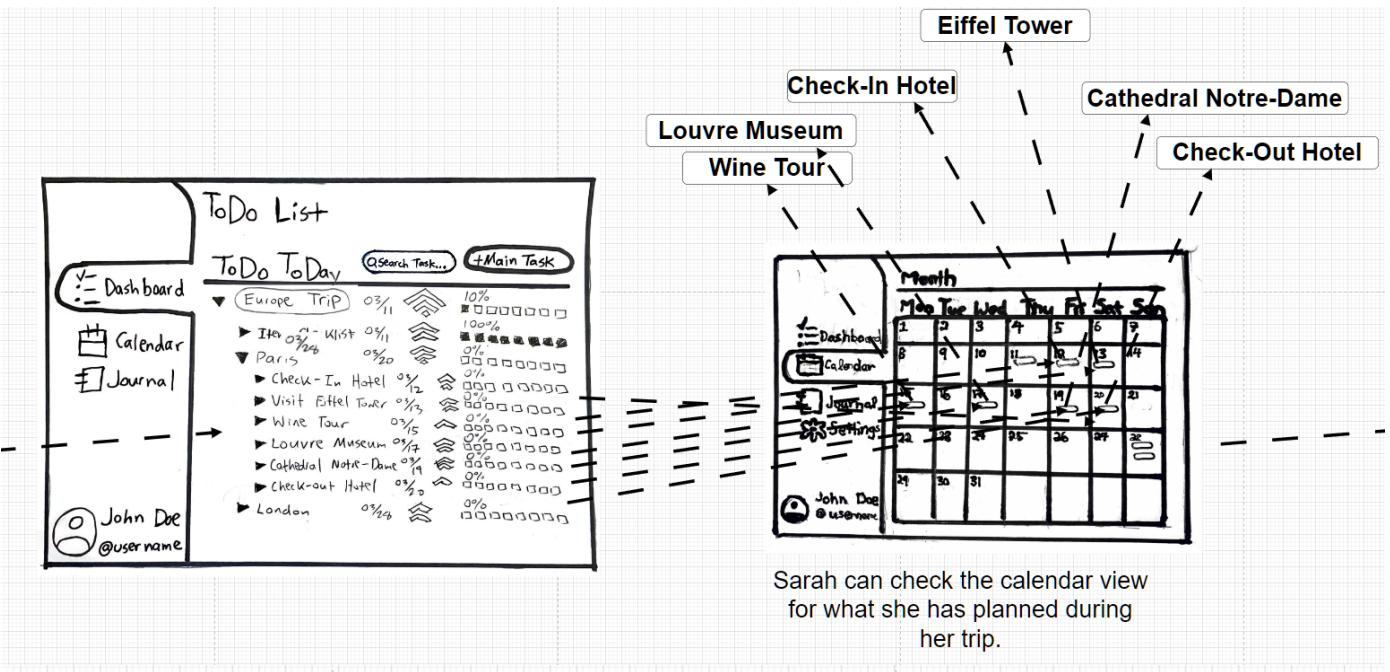


Check-In Hotel	03/12
Eiffel Tower	03/12
Wine Tour	03/15
Louvre Mueseum	03/17
Cathedral Notre-Dame	03/19
Check-Out Hotel	03/20

In the Paris subtask, she adds her main itinerary for her Paris trip and has a better idea of what she is going to do in Paris

Item	Progress	Completion %
Clothes	0/11	0%
Toothbrush	0/11	0%
Shoes	0/11	0%
Vlog Camera	0/11	0%
Passport	0/11	0%
Phone	0/11	0%
Paris	0/20	0%
London	0/26	0%

After adding everything, she checks them off as she finishes packing them

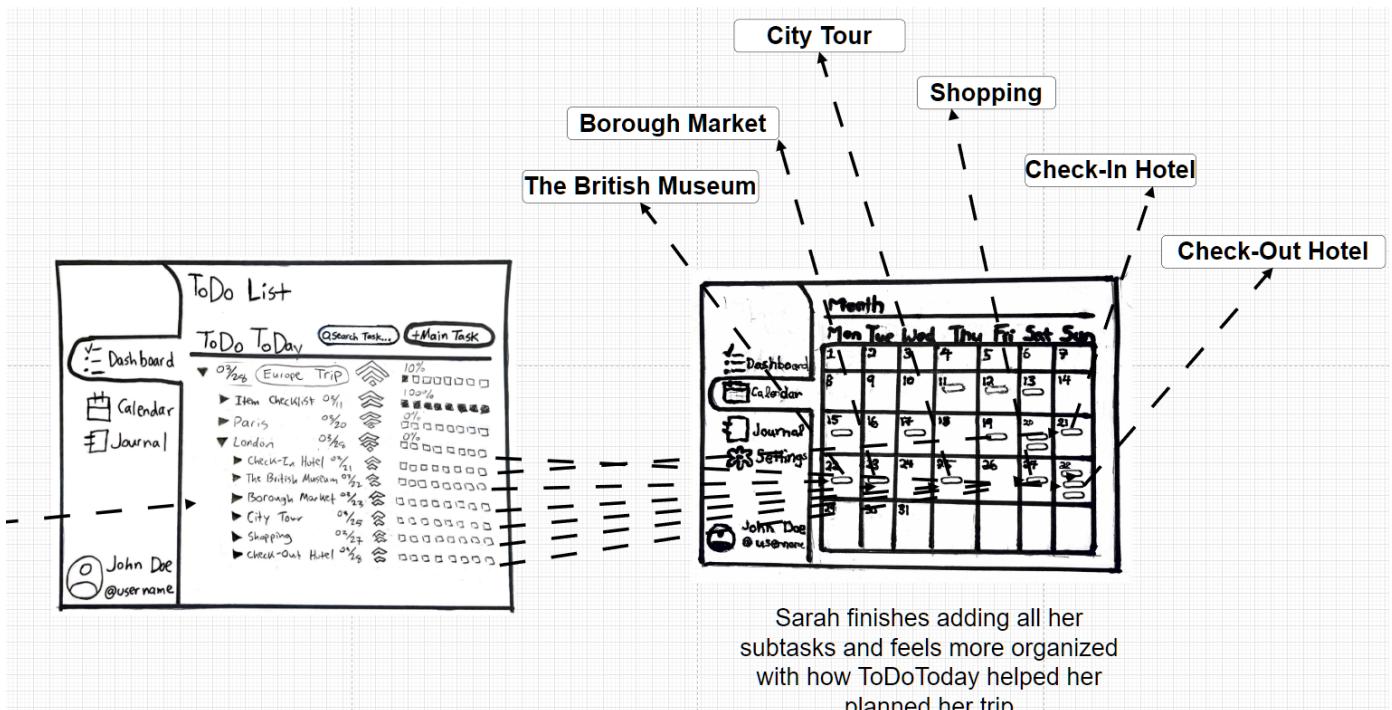


On click: +Subtask London

A modal dialog titled "Add a main task" is shown. It has fields for "Title" (with a placeholder "London") and "Due date" (set to "03/28"). Below these are sections for "Time" (set to "8:00") and "+Subtasks". Under "+Subtasks", there is a table titled "Current Subtasks" with two rows. Each row has columns for "Title", a plus sign, a checkmark, a colon, and a trash bin icon. The first row is empty, and the second row has a placeholder "Title". At the bottom is a "Finish" button.

- | | |
|--------------------|-------|
| Check-In Hotel | 03/21 |
| The British Museum | 03/22 |
| Borough Market | 03/23 |
| City Tour | 03/25 |
| Shopping | 03/27 |
| Check-Out Hotel | 03/28 |

In the London subtask, she adds her main itinerary for her London trip and has a better idea of what she is going to do in London



Sarah finishes adding all her subtasks and feels more organized with how ToDoToday helped her planned her trip

Use Case 7:

Register Account Page

Todo
ToDay

Register Account

Username
Enter Username

Email
Enter Email

Password
Enter Password

Confirm Password
Confirm Password

I agree to the [Terms of Service](#) and [Privacy Policy](#).

Register or Login

Login Page

Todo
ToDay

Login

Email / Username
Enter email or username

Password
Enter password

Login or [Register Account](#)

upon registering, users will be sent to the home page. user has to input a valid email, password must match, and agree to ToS and Privacy Policy

upon entering correct credentials, user can login and will bring them to the home page

Empty ToDo List

ToDo Today +Main Task

No tasks, click [+Main Task](#) to add a task.

Dashboard
Calendar
Journal
Settings

John Doe
Username

A hobbyist is planning a project to build a bookshelf. He heard about ToDoToday from a friend and decides to use it for his bookshelf project.

On click: +Main Task

Add a main task

Title Due date Time

+Subtask

Current Subtasks

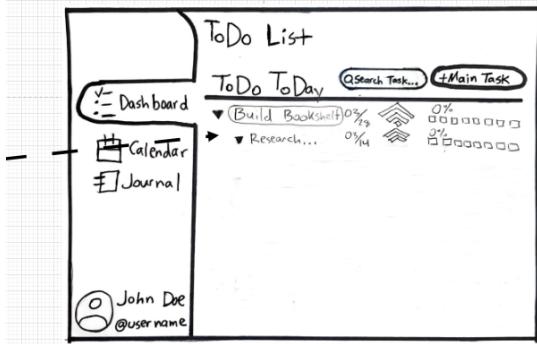
Title	Due Date	Time

Finish

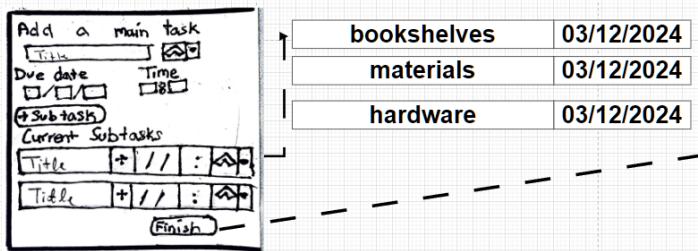
Build Bookshelf 03/28/24
Research... 03/11/24

The hobbyist creates the main task of building the bookshelf and creates a subtask for research

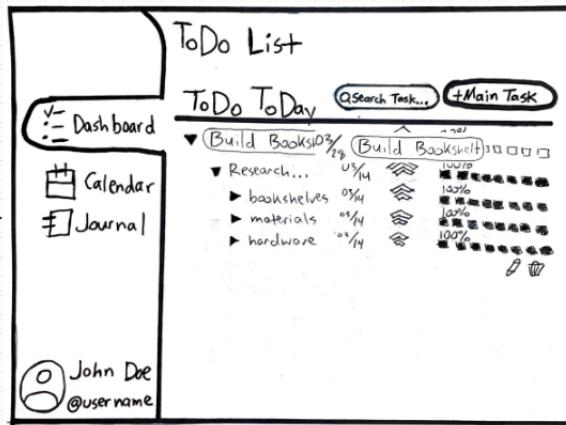
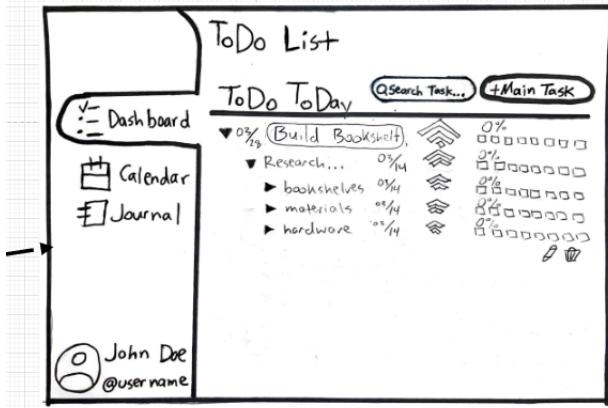
Added main task and subtask to Dashboard



On click: +Subtask Research...

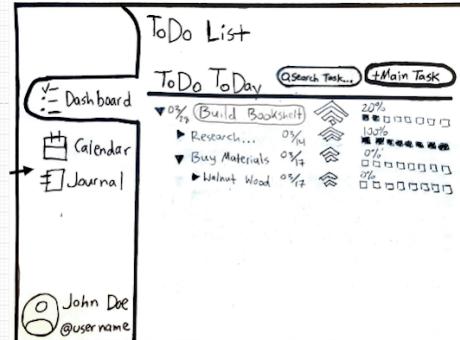
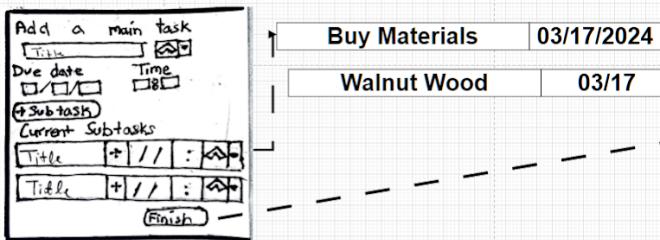


He adds subtasks: bookshelves, materials, and hardware as his research subtasks



While the hobbyist is researching for his bookshelf, he starts marking them off in ToDoToday

On click: +Subtask Build Bookshelf



After researching, the hobbyist knows what kind of wood to use. He creates a subtask to buy materials. Under the buy materials subtask, he adds walnut wood

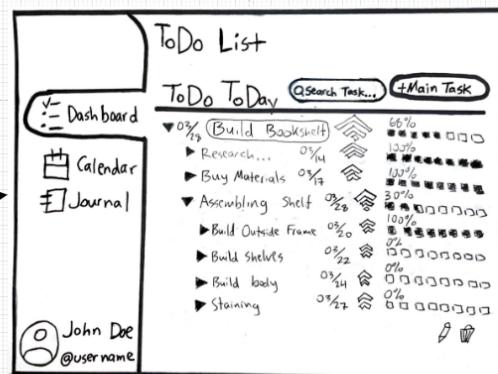
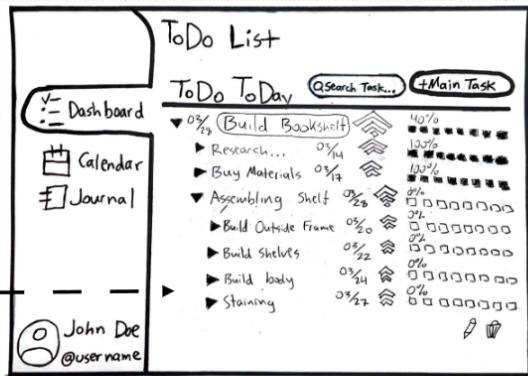


As he finished buying the walnut wood, he checks it off, completing the buy materials task

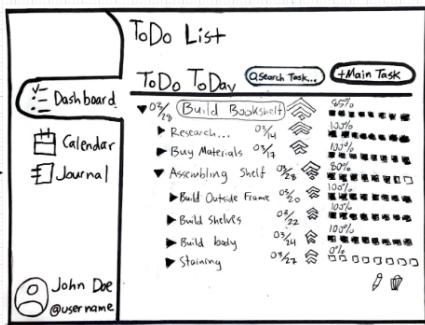
On click: +Subtask Build Bookshelf

Assembling Shelf	03/12/2024
Build Outside Frame	03/12/2024
Build Shelves	03/12/2024
Build Body	03/12/2024
Staining	03/12/2024

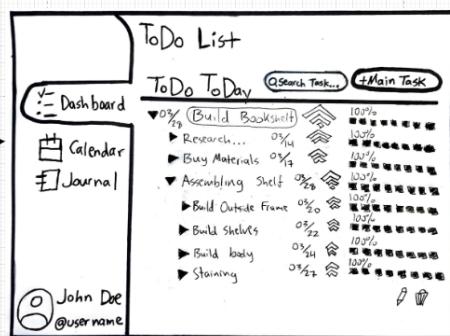
With all the materials, the hobbyist creates the final subtask of assembling the shelf. The subtasks he creates for this are: Build Outside Frame, Build Shelves, Build Body, and Staining



The hobbyist finishes building the outside frame, marking the whole subtask complete. The assembling shelf subtask is now 30% complete and the whole main task is now 68% complete.



The hobbyist goes through his subtasks as he is now done with building the shelves and the body of the bookshelf. Assembling Shelf is now 80% complete and the main task is 85% complete



Now, the hobbyist finishes his bookshelf staining. He marks it complete, making assembling the shelf 100% complete as well as the whole task.

The diagram illustrates a user interface flow. On the left, a vertical sidebar contains icons for Dashboard, Calendar, Journal, and a user profile (John Doe). An arrow points from the Journal icon to a main content area. This area is titled "My Journal" and "Journal Entries". It shows a date "03/28" next to a button labeled "Build Bookshelf". Below this are three entries: "Build Bookshelf Prompt 1", "Build Bookshelf Prompt 2", and "Build Bookshelf Prompt 3". A dashed line leads from the third prompt to a separate window titled "Build Bookshelf Prompt 1", which contains a text input field with the placeholder "Type your reflections here..." and a "Save" button.

My Journal

Journal Entries

03/28 Build Bookshelf

Build Bookshelf Prompt 1

Build Bookshelf Prompt 2

Build Bookshelf Prompt 3

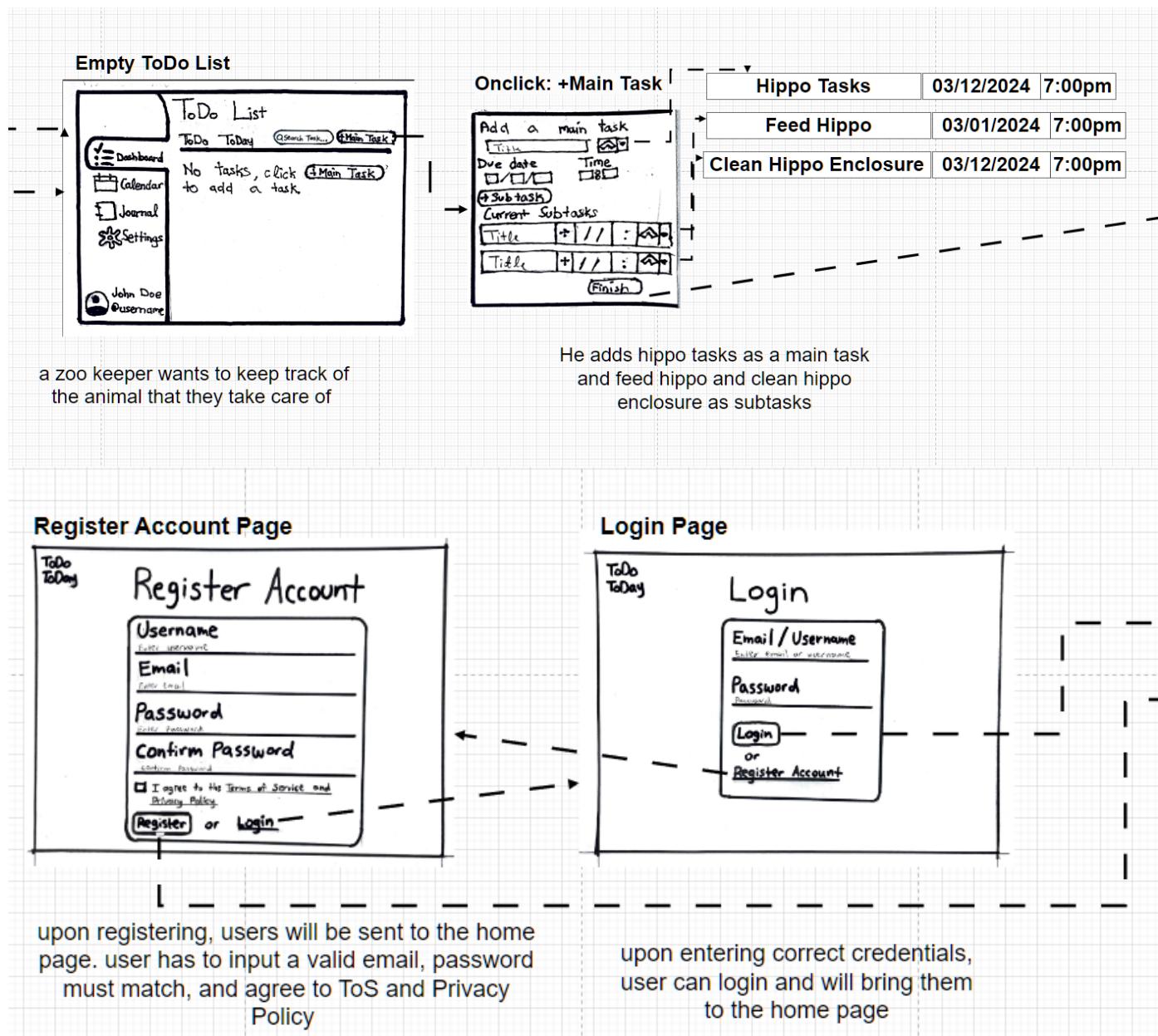
John Doe
username

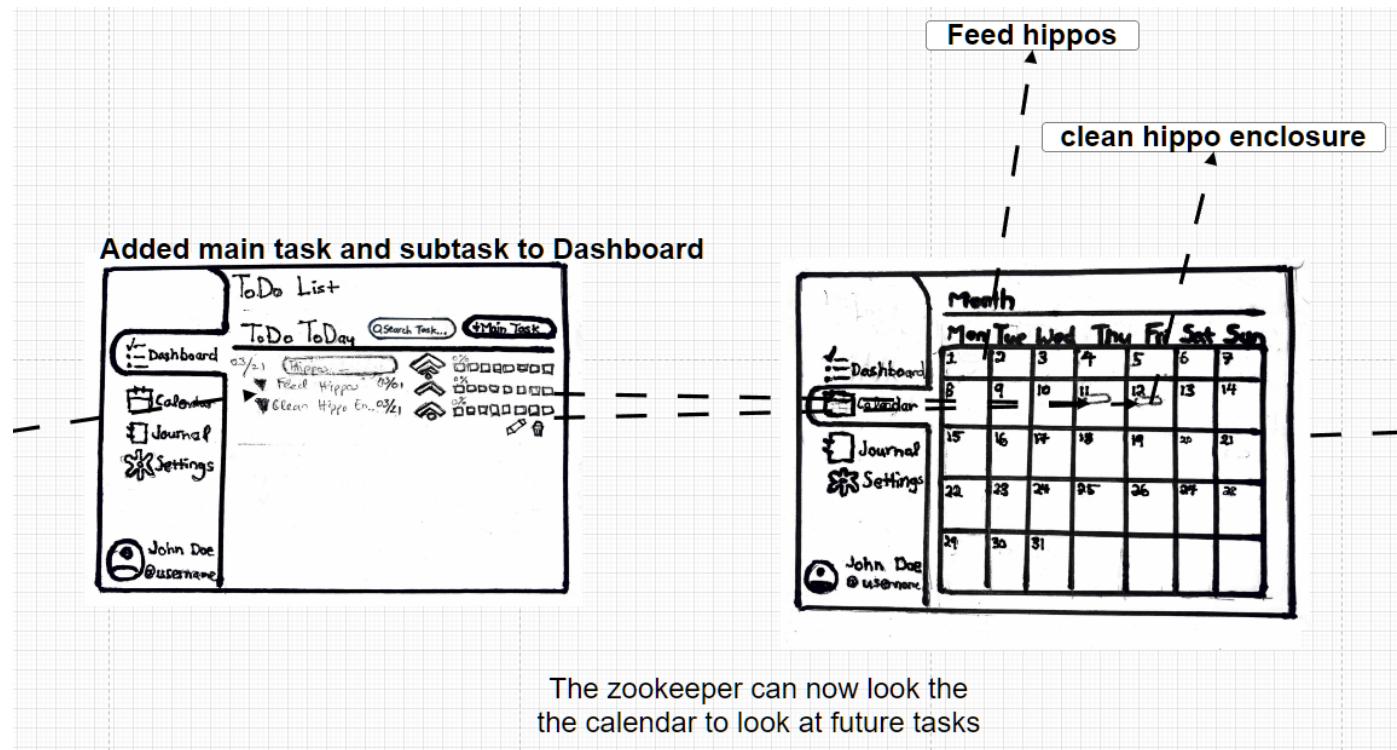
Type your reflections here...

Save

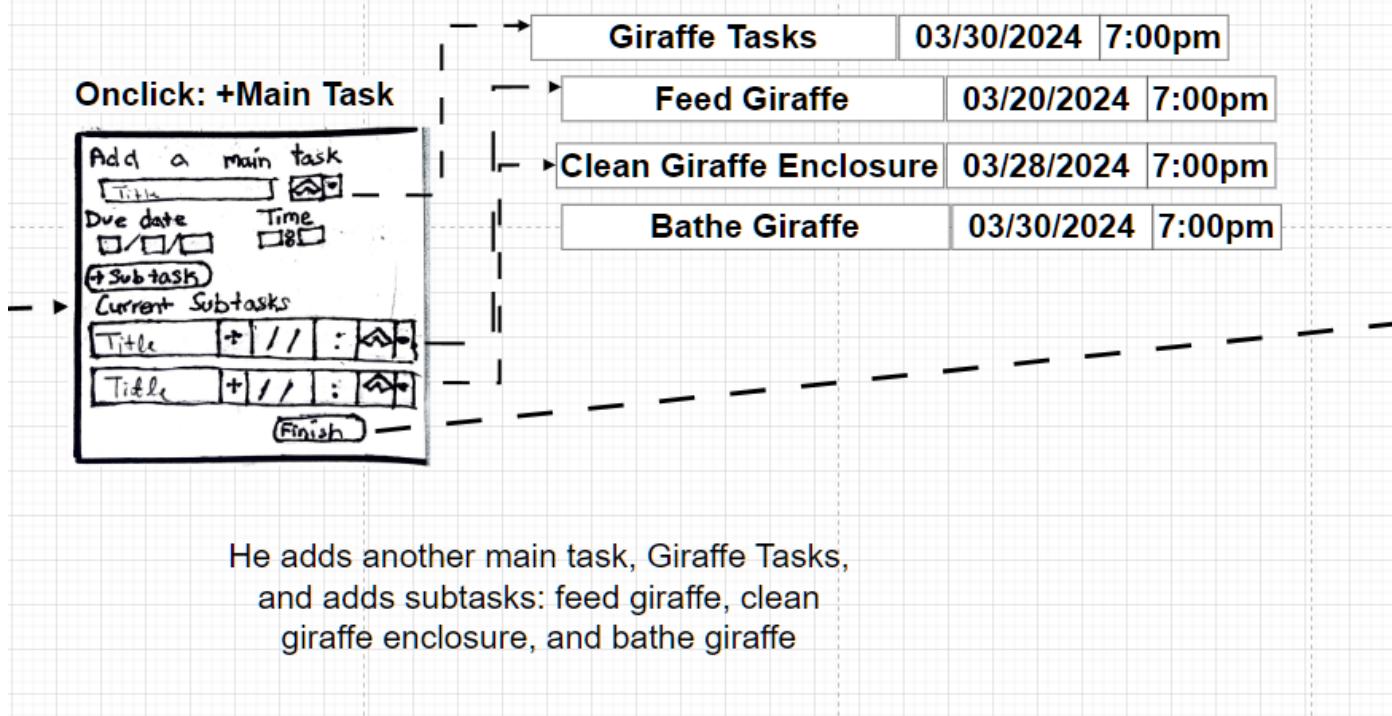
The hobbyist has finally finished his bookshelf project and decides to use the journal to reflect on what he struggled with and how he can improve for future projects

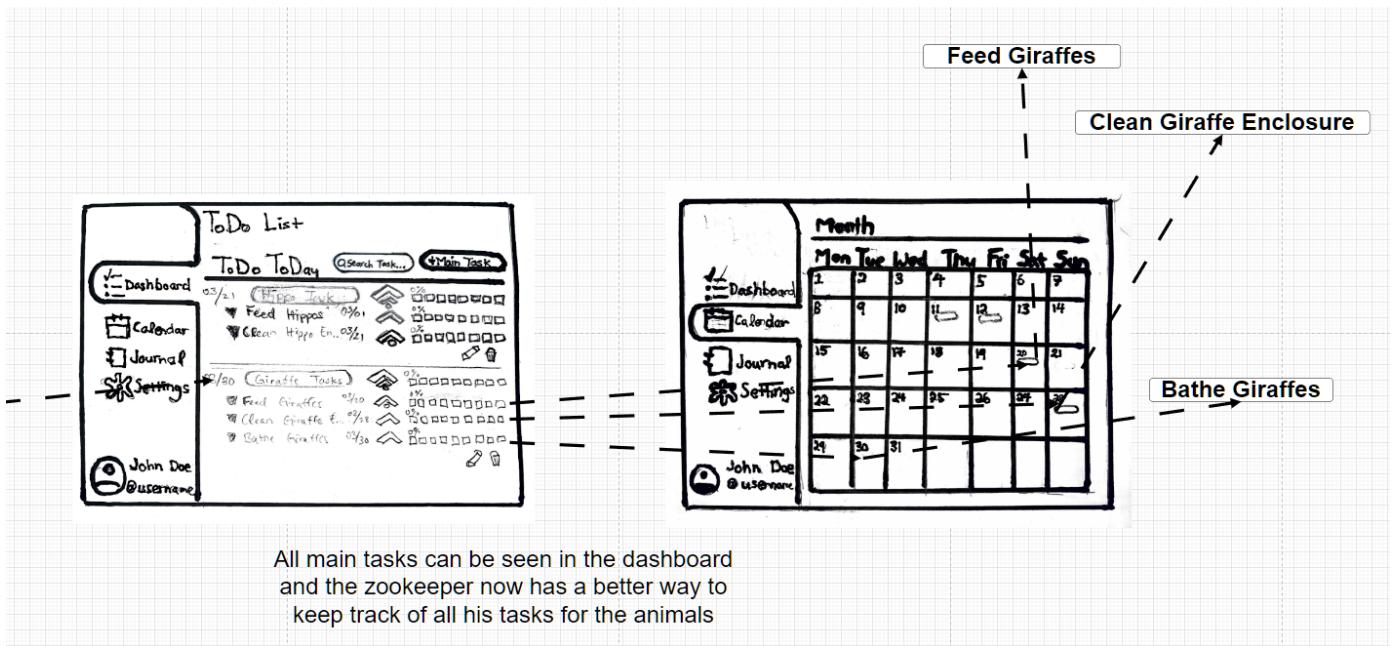
Use Case 8:





The zookeeper can now look the calendar to look at future tasks





Use Case 9:

Register Account Page

Login Page

upon registering, users will be sent to the home page. user has to input a valid email, password must match, and agree to ToS and Privacy Policy

upon entering correct credentials, user can login and will bring them to the home page

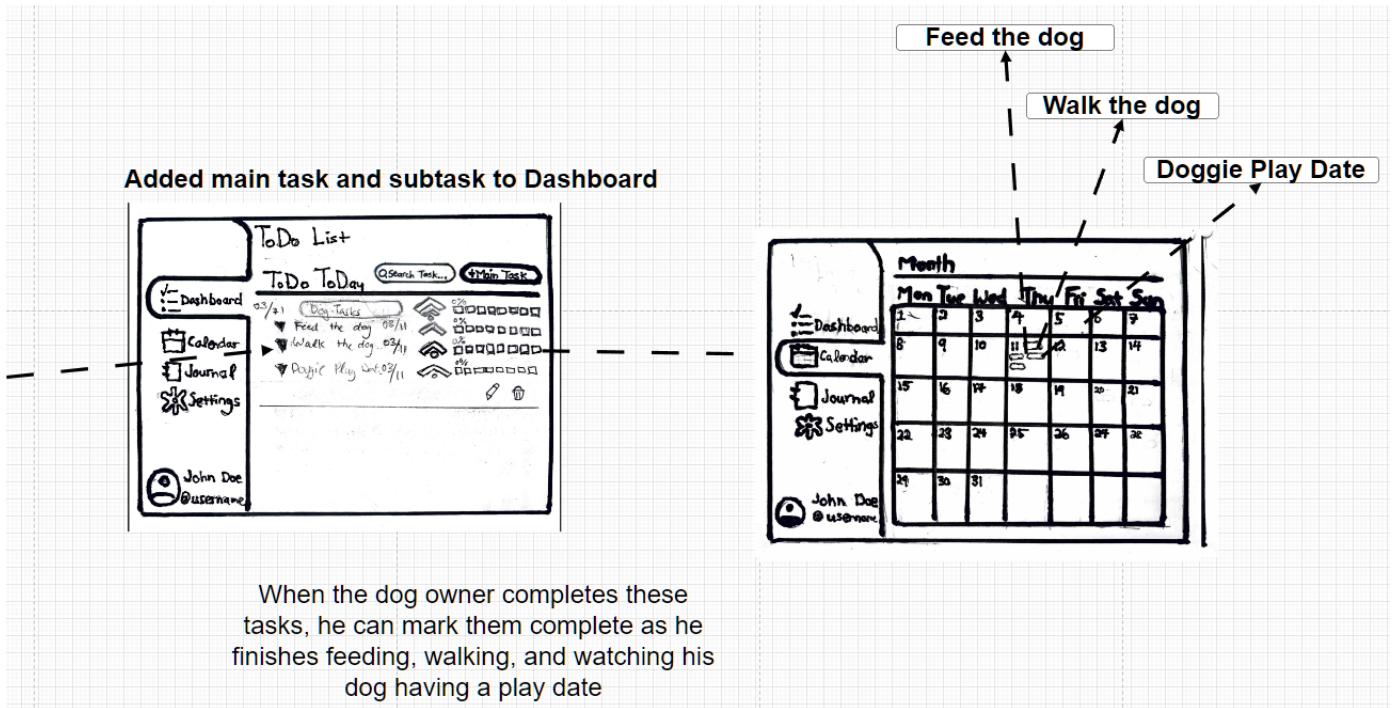
Empty ToDo List

A dog owner wants to keep track of the things they want to do for their dog

Onclick: +Main Task

Dog Tasks	04/11/2024	7:00pm
Feed the dog	04/11/2024	3:00pm
Walk the dog	04/11/2024	3:00pm
Doggie Play Date	04/11/2024	3:00pm

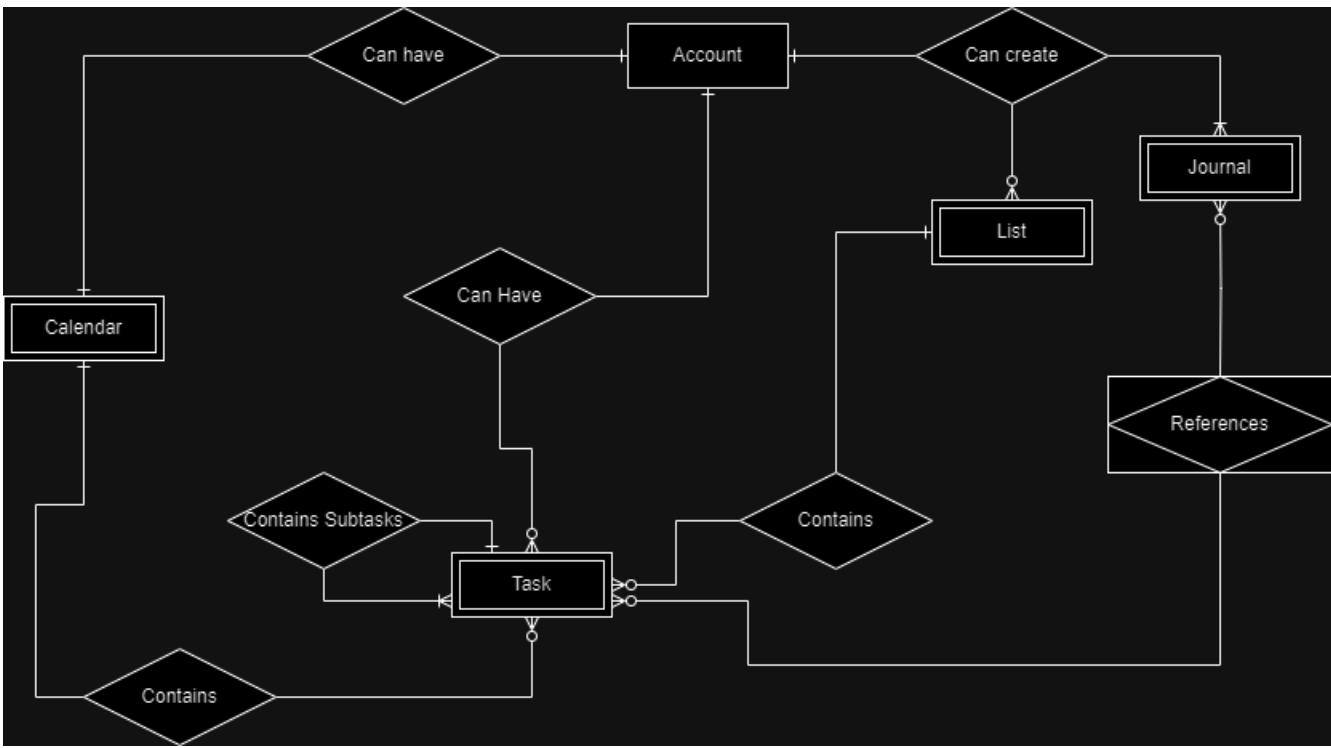
He creates a main task titled Dog Tasks and creates subtasks to feed the dog, walk the dog, and have a dog play date



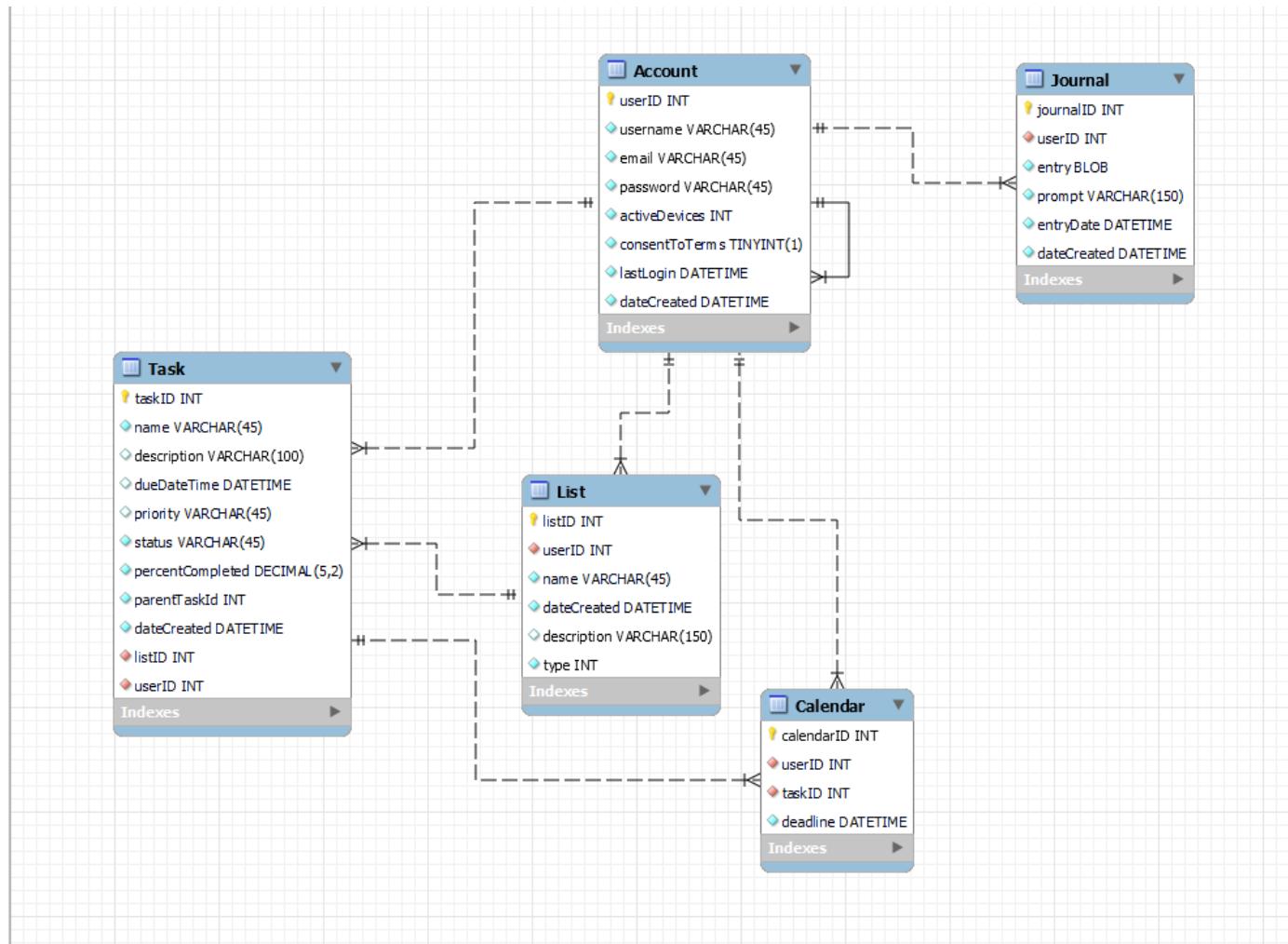
XV. High-level database architecture and organization

MySQL is a great tool to use due to its ability to efficiently handle the relational structure needed for organizing entities, such as tasks, users, lists, and other entities. It's also easy to use, as well as having the ability to convert ERDs to EERs easily and can export the actual SQL database

ERD:



ER:



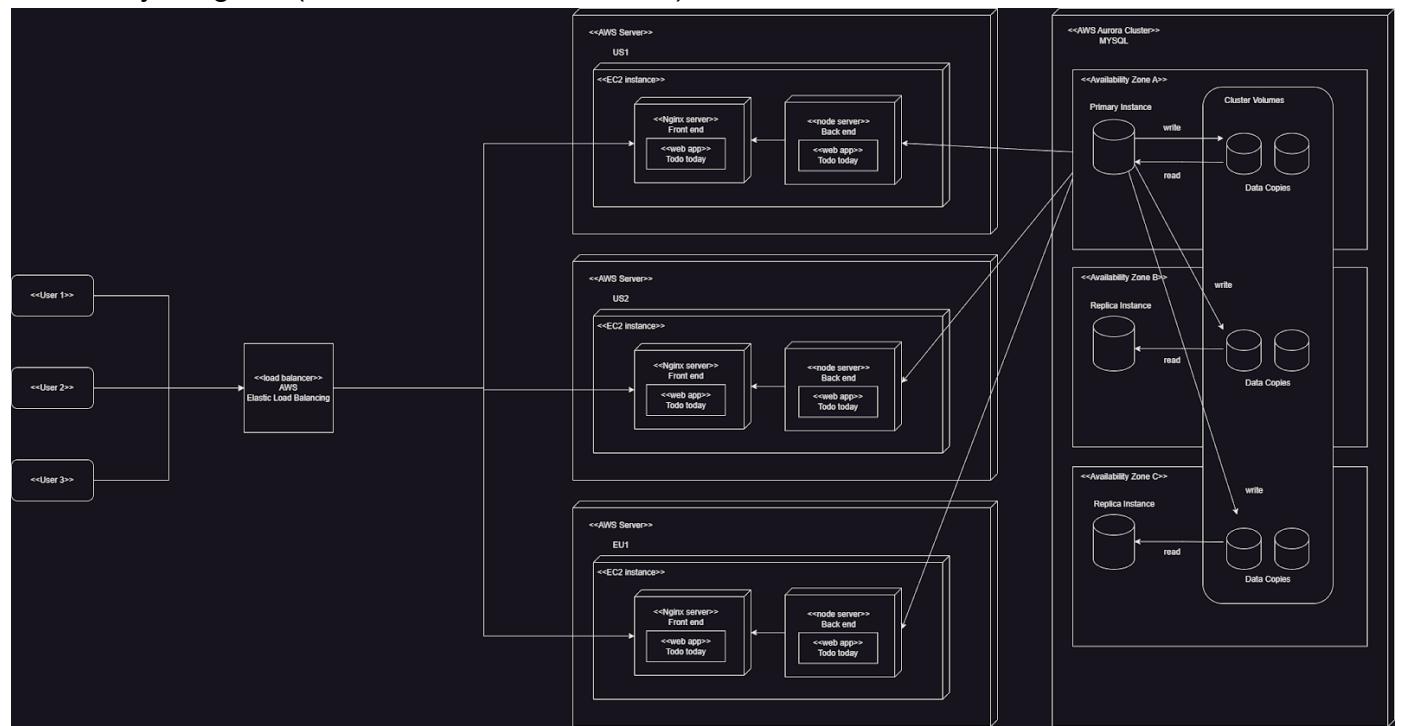
XVI. High-Level APIs and Main Algorithms

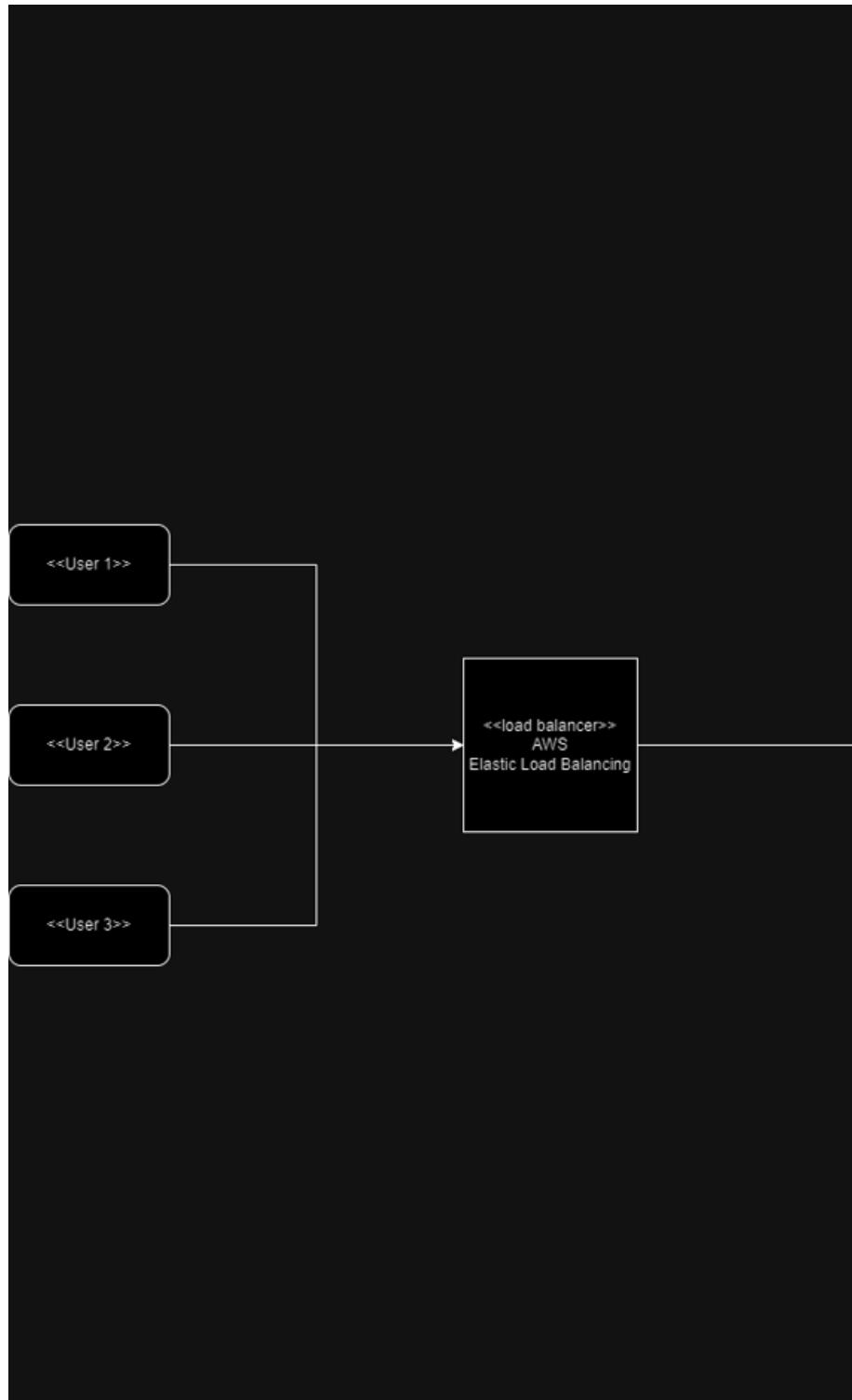
5. High-Level APIs and Main Algorithms

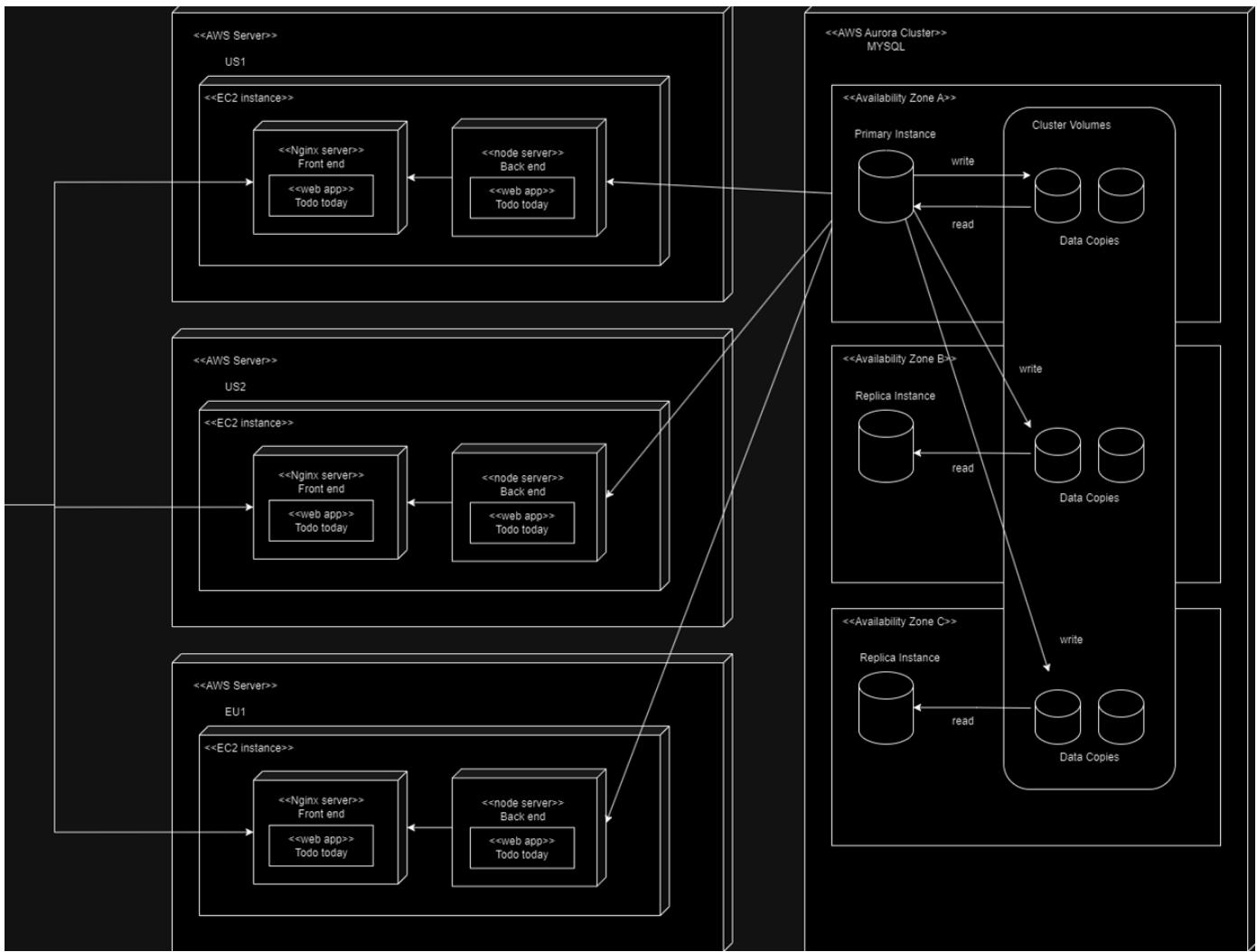
- Priority sorting algorithm
- Sorts subtasks based on priority but only for layer. Each set of subtasks is sorted independent of subtasks under a different task.
- Tree search algorithm
- Searches the tasks and subtasks in the tree structure of our todo lists
- GPT2 API(HuggingFace)
- Calls to the GPT2 API to generate text for our prompts
- List Search
- Basic search of both the name and description

XVII. System Design

Scalability Diagram (Zoomed in sections below):







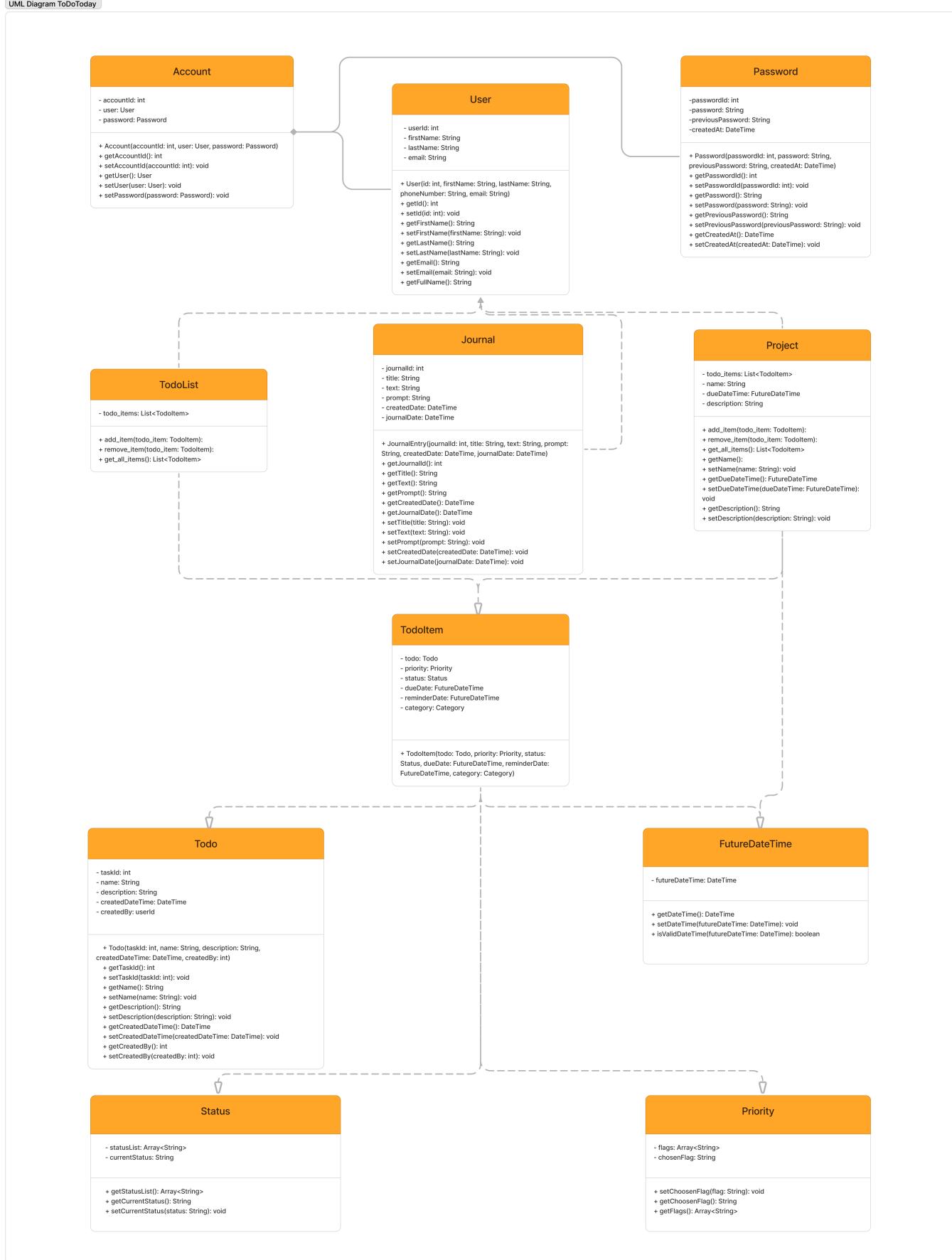
Scalability Diagram Summary:

Starting at the user end on the left side of the diagram the first microservice we're using is Amazon Web Services elastic load balancing to manage incoming traffic for the instances of our system. Beyond the load balancing we have 3 instances of our application one on the East Coast, one on the West Coast, and one in Europe, the reasoning being that no critical failure could have happened in all three locations simultaneously, which makes the entire system much more fault tolerant allowing for an entire critical failure in one of those regions without interruption to service. The next microservice we would use would be Amazon Web Services Aurora data cluster system specifically with the MySQL compatibility. This allows us to keep our same database structure but replicate it over multiple instances with a clustered volume backup. In the version that is in the diagram, we have one primary instance with two replica instances ready and immediately waiting to take over if there is a critical failure, along with 6 supplemental copies of the data. This gives us an incredible level of fault tolerance and error correction as if something goes wrong or something gets overwritten Aurora is designed to be robust and have safeties in place that will help us recover in the case of accidental or intentional data corruption or deletion. Another advantage of using Amazon Web Services systems for our upgrade path is that it keeps everything in one ecosystem. This unification of the ecosystem means that all of our systems are in one place

which is convenient for maintenance and upkeep and also is a potential security vulnerability depending on the security for our login for admin. But that is the only major drawback is that since we are using only one system if Amazon Web Services goes down so does our entire application we would have to recover what we can and then try to transfer to the Azure cloud or some other system but we would lose significant time as everything is integrated into amazons web ecosystem. The main advantage to keeping it all within the AWS framework is that this basic structure is scalable so if we need more space and if or if we need to increase our throughput we can add new instances and increase the redundancy and distribution for our data cluster very easily with just a couple clicks. If we were using a disparate set of components from various companies it would be much harder for us to scale up and down depending on usage. If the application becomes extremely popular we might need to migrate out of Amazon Web Services into a more proprietary system and that will take a lot of research and development time but for the current scale and the current scope that we are looking at this should be viable even as a phase two for development of our product.

UML Class diagram:

UML Diagram ToDoToday



Summary

System design patterns are categorized into three types: Creational, structural, and Behavioral patterns. We explored different design patterns, including the Factory Method and Singleton patterns.

While suitable for creating objects of related types without specifying their concrete classes, the Factory Method pattern did not provide the level of flexibility and customization we needed for configuring `Account` and `TodoItem` instances. Additionally, the Singleton pattern, which ensures a class has only one instance and provides a global point of access to it, did not apply to our scenario. The reason is, that it would limit our ability to create multiple `TodoItem` instances with different configurations.

Other patterns within Structural Patterns and Behavioral Patterns seemed the next step for the scalability. So, after evaluating these patterns and their suitability for our application's architecture, we determined that the Builder pattern offered the best fit. Its ability to encapsulate the construction process and provide a flexible way for creating complex objects made it an ideal choice for constructing `TodoItem` instances with varying configurations. By adopting the Builder pattern, we ensure that our application remains scalable, maintainable, and adaptable to future changes and requirements.

How are we using this pattern in our System?

Here is how it is being used in the above UML Diagram:

1. Building `Account` with `Password` and `User` Class:

In our system architecture, the `Account` entity comprises user credentials and personal information, which are represented by the `Password` and `User` classes, respectively. To streamline the creation of `Account` instances with varying configurations and to make each class responsible for small tasks, we employed the Builder pattern.

The `Account` class serves as the primary builder for `Account` objects. It uses the construction process by utilizing methods to set the properties of the `Password` and `User` components directly within the `Account` class.

By utilizing the Builder pattern in this context, we ensure that the creation of `Account` objects remains flexible and maintainable. The encapsulation of construction logic within the `Account` class and dividing the responsibilities across multiple other classes simplifies the instantiation process and promotes code readability.

2. Building `TodoItem` with Different Configurations:

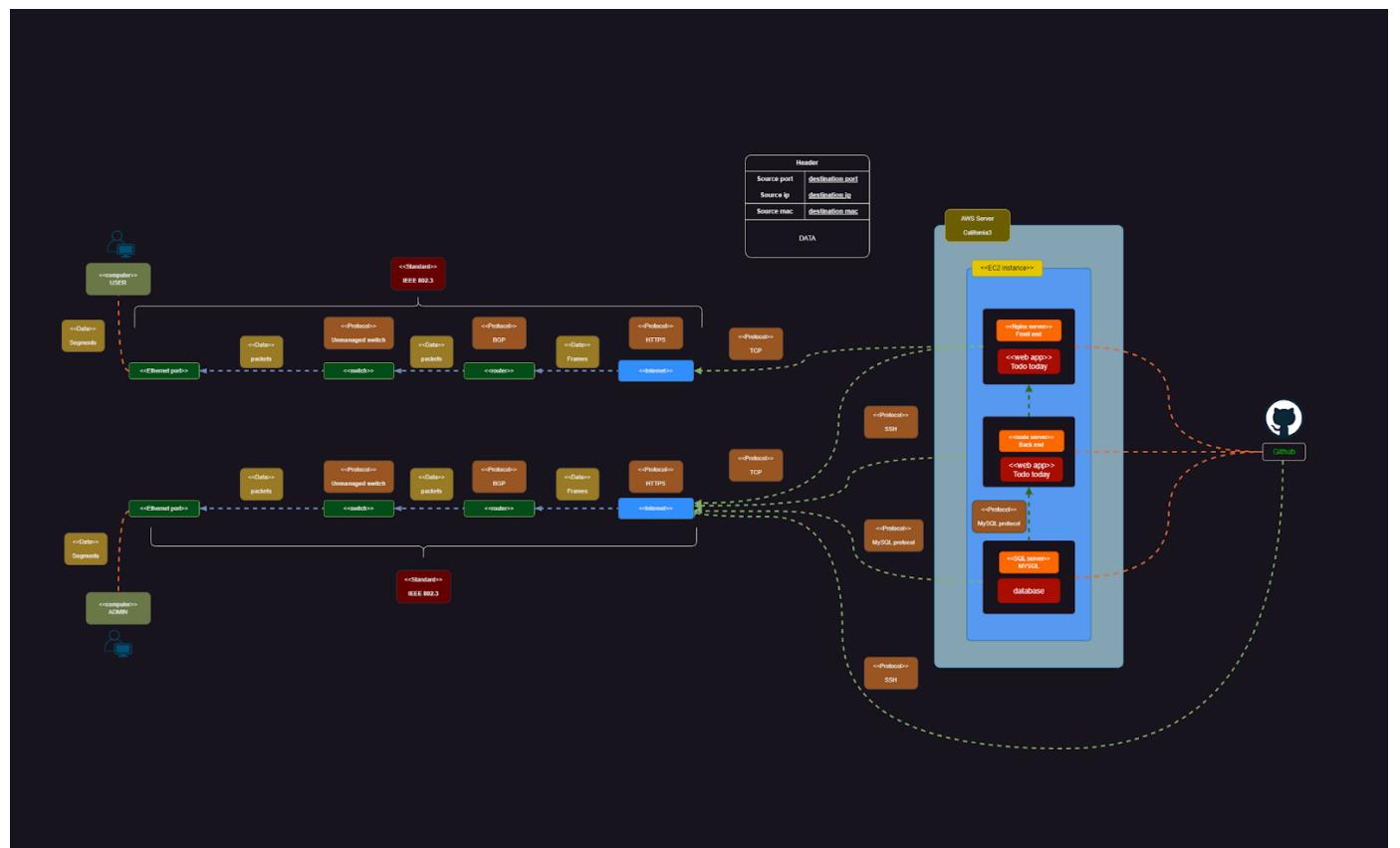
Within our task management module, the `TodoItem` class represents individual tasks with various attributes such as priority, status, and due date. To facilitate the creation of `TodoItem` instances with customizable configurations, we adopt the Builder pattern.

The `TodoItem` class itself acts as the builder for `TodoItem` objects. It provides methods for setting the attributes of the `TodoItem` with `Todo`, `Priority`, `Status`, and `Due Date`, allowing clients to specify the desired configuration through method calls.

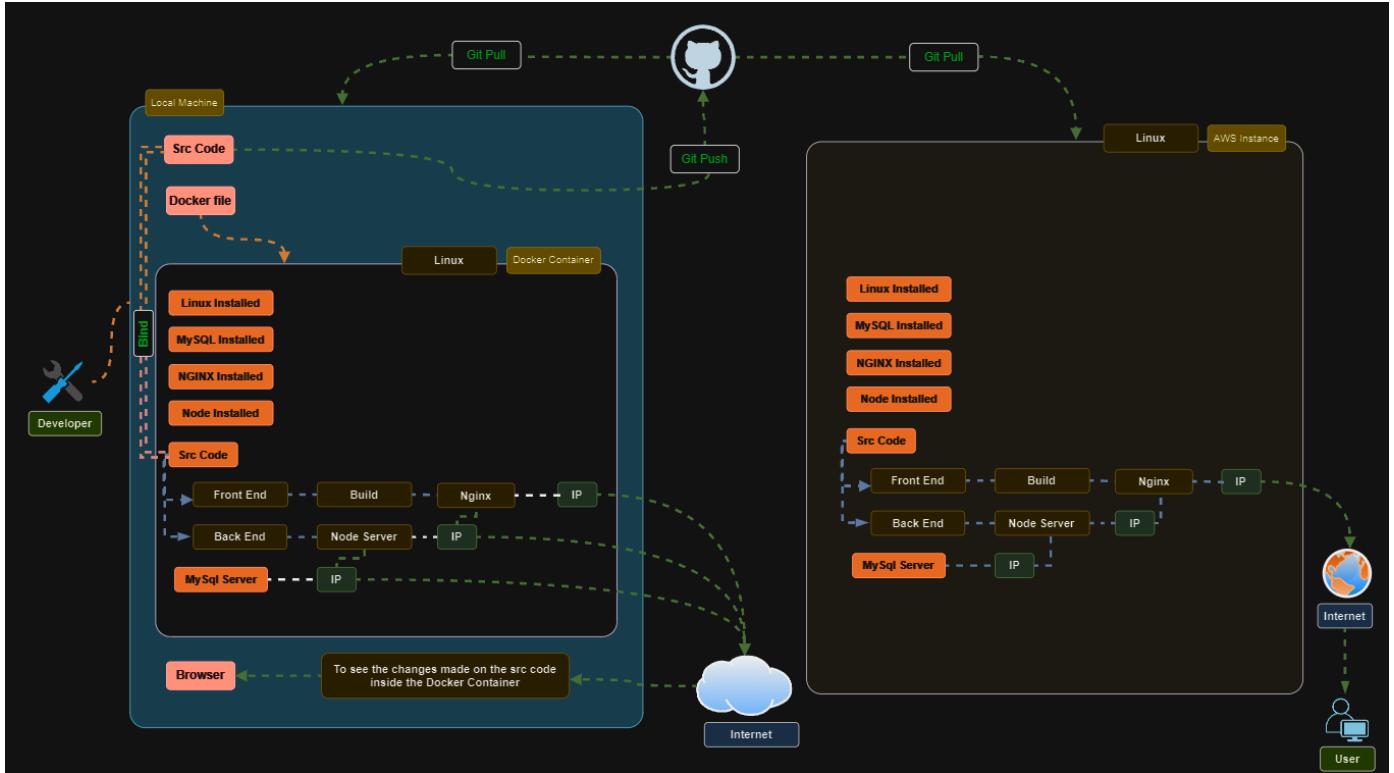
By leveraging the Builder pattern for `TodoItem` creation, we ensure that the construction process remains clear and adaptable. The `TodoItem` class encapsulates the logic for building `TodoItem` objects. It enables clients to construct instances with ease while maintaining code's flexibility and maintainability.

XVIII. High-Level Application Network and Deployment Design

Network Diagram:



Deployment Design:



XIX. Identify actual key risks for your project at this time

Identify only actual and specific risks in your current work such as (list those that apply:

- skills risks (do you have the right skills),
- schedule risks (can you make it given what you committed and the resources),
- technical risks (any technical unknowns to solve),
- teamwork risks (any issues related to teamwork);
- legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright)

Known Risks:

1. Application Network Diagram Knowledge (skill)
2. Initial lack of AWS knowledge (technical / skill)
3. Chat GPT (technical unknown)
4. Data Leakages (security risk)

5. Lack of communication
 6. Lack of knowledge regarding legal terms of condition content
 7. Difficulty in splitting up tasks
-

XX. Project management (Team Lead)

During milestone one as the team lead I used an excel spreadsheet to manage our time and completion of tasks. It was a very manual system that I constantly was having to update as I was the only one with access to that data. Moving over to notion that allowed me to share the tasks between all of the teammates made it much easier for us to understand exactly where we were at and who had finished what and what still needed to be done. It also allowed us to define subtasks and build out a solid structure for the step by step sub steps that we need to do to complete the larger pieces that are needed for completion of our milestone 2. It also allowed us to divide and conquer in a much more effective way than me just sort of handing out assignments directly I was able to hand out high level assignments saying OK you're in charge of this task you're in charge of that task and then you tell us what we need to do to help you. That sort of subdivision of Labor has significantly improved both the quality and speed of our work.

XXI. Detailed list of contributions (Team Lead)

	Alex Nikols	Jason Tran	Sulav Jung Hamal	Jadon Hoang	Randale Reyes	Christian Gopez
overall Code	1	4	5	2	1	1
overall Documentation	5	4	4	4	3	5
Data Definitions	2	1	4	1	2	4
Prioritized Functional Requirements	3	3	3	3	3	3
UI Mockups and Storyboards (high level only)	2	5	2	5	1	1
High level database architecture and organization	3	1	3	1	5	1

	Alex Nikols	Jason Tran	Sulav Jung Hamal	Jadon Hoang	Randale Reyes	Christian Gopez
High Level APIs and Main Algorithms	4	1	4	1	1	4
System Design	4	1	5	1	1	2
High Level Application Network and Deployment Design	5	1	4	1	1	2
Identify actual key risks for your project at this time	3	3	3	3	3	3
Project management	5	2	3	2	1	2
Vertical SW Prototype	2	4	5	2	2	2