

petivity

productivity's best friend

The Team:



Sidra I.



Crystal Z.



Ivan S.



Jonathan M.

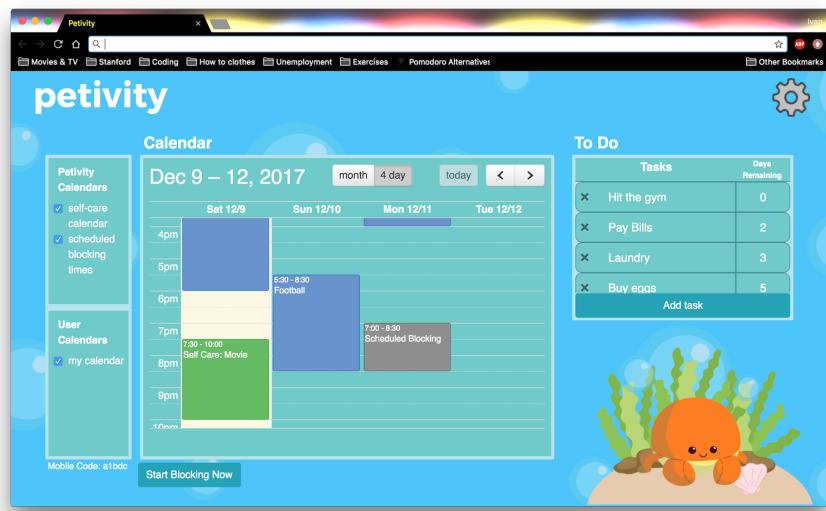
Problem/Solution Overview

People often have difficulty completing tasks because of distractions, poor time management, and/or unhealthy work-life balance. Petivity aims to increase productivity by effectively eliminating distractions and keeping people on track in a fun and engaging way.

With Petivity, users can easily access and manage their to-dos and Google Calendar-integrated calendar from a centralized platform using our Google chrome extension, with completion of tasks animating the virtual pet. The calendar also allows the user to manage the blocking feature and schedule self-care items. Additionally, the user can manage their blocked sites and enable/disable blocking through this platform, while also syncing with the mobile app.



Mobile Component



Web Component

Tasks & Final Interface Scenarios

***Note:** We re-evaluate our tasks in complexity after Medium-Fi prototyping and therefore Task 2 (Self-Care Scheduling) is listed after Task 3 (Blocking) in complexity to adjust for the increased complexity from adding a mobile component.

Web:

Task 1 (Simple): Complete a task from to-do list and add new task to it.

Task 2 (Complex): Add a website to block to your list of blocked sites and enable/disable blocking..

Task 3 (Medium): Schedule one of Petivity's self-care activity suggestions to your calendar.

Mobile:

Also addresses Task 2: Start and end blocking as well as interact with pet.

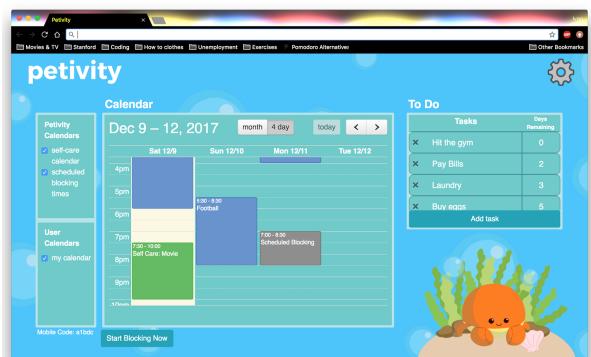
Why we chose task 1 (Simple): Showed the main functionality of to-do list and highlighted interaction with pet through our web-interface

Why we chose task 2 (Complex): Showed the main functionality of the blocking component through web and mobile, showing added functionality of the google chrome extension plug-in being present every time one opens a new tab and the ease of managing blocking through the plug-in.

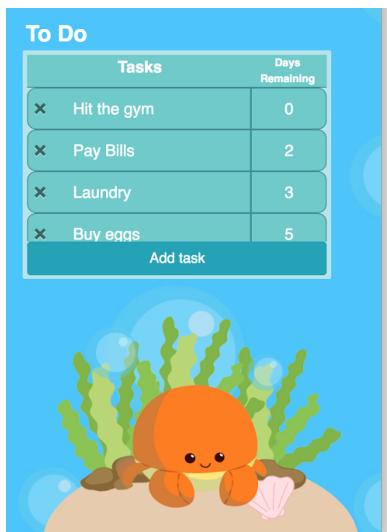
Why we chose task 3 (Medium): See how users interact with the functionality of self-care

Storyboards for 3 tasks

- [Task 1] Complete a task from to-do list and add a new task (low complexity)



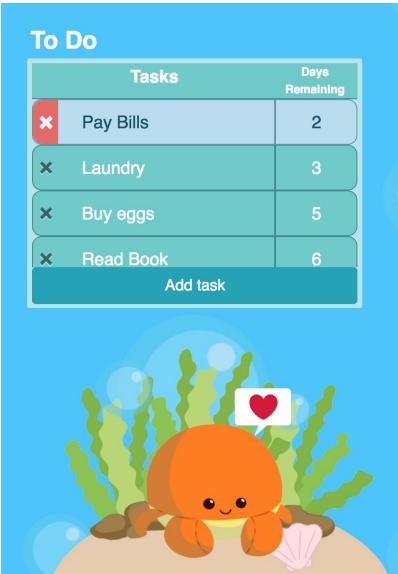
Open new tab in browser and application's home screen is displayed.



Clicking a “X” in to-do list to complete a task.



Completing a task causes food to fall from list to feed the pet.



The pet is happy afterwards as shown by the heart bubble

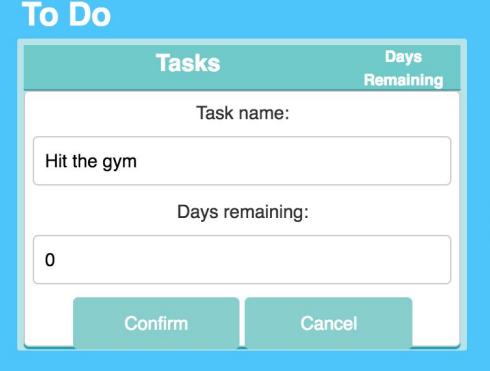


To add a task to to-do list, user clicks on 'Add Task' button

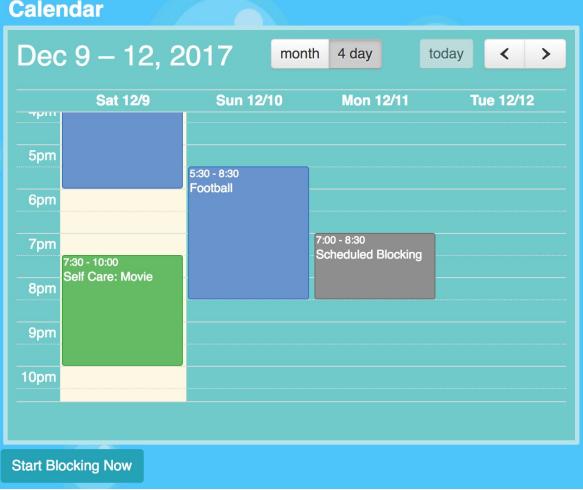
This screenshot shows a modal window titled 'To Do' for adding a new task. It has two input fields: 'Task name:' and 'Days remaining:', both represented by empty text input boxes. At the bottom are two buttons: 'Add' and 'Cancel'.

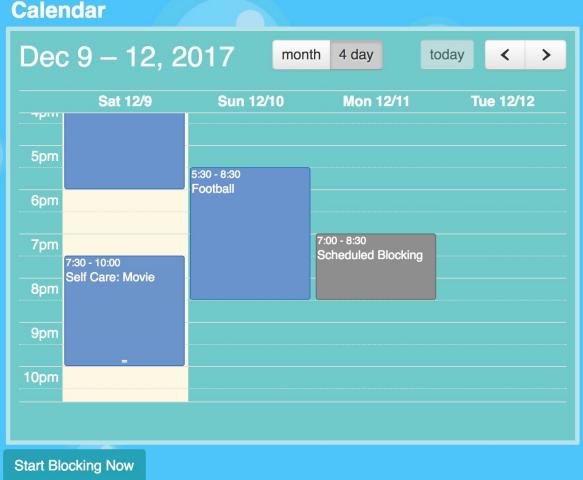
Tasks	Days Remaining
Task name:	
Days remaining:	

A popup box enables users to input new task information, including name and days remaining

	User inputs data
	Confirming new input, task is entered in to-do list, organized by due date.

2. [Task 3] Schedule a self care activity (Medium complexity)

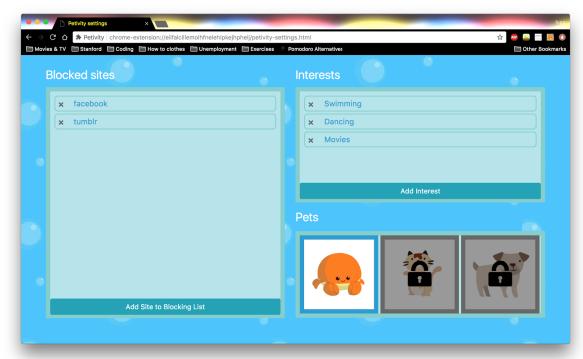
	Self-care suggestions automatically appear as green entries in calendar for user to press. Suggestions are based on user specified hobbies/activities.
---	--



The calendar shows a weekly view from Dec 9 to Dec 12, 2017. On Saturday, there is a blue block from 5pm to 6pm labeled "Self Care: Movie". On Sunday, there is a yellow block from 6pm to 7pm labeled "Football". On Monday, there is a blue block from 5:30pm to 6:30pm labeled "Football" and a grey block from 7pm to 8pm labeled "Scheduled Blocking". A "Start Blocking Now" button is at the bottom.

Clicking the self care activity automatically adds the event to your calendar

3. [Task 2] Block distractions (*High Complexity*)



The Petivity settings screen shows a "Blocked sites" section with "facebook" and "tumblr" listed, and an "Add Site to Blocking List" button. It also shows an "Interests" section with "Swimming", "Dancing", and "Movies" listed, and an "Add Interest" button. Below these are sections for "Pets" with icons of an orange cat, a brown dog, and a grey dog.

From home screen, user can navigate to settings to specify websites to block (among other options).



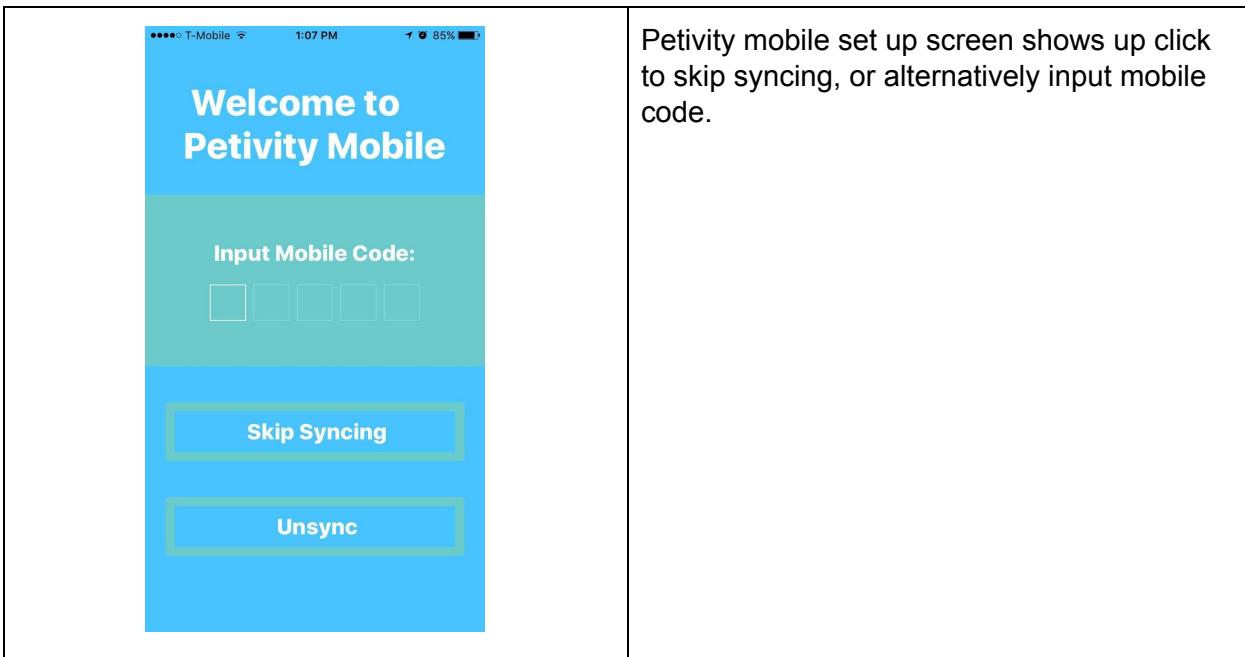
The Petivity home screen shows a calendar for Dec 9 - 12, 2017. The "Self Care: Movie" event from Saturday is now highlighted in green, indicating it has been blocked. The "Football" events remain blue. A "To Do" list on the right includes tasks like "Hit the gym", "Pay Bills", "Laundry", and "Buy eggs". A "Start Blocking Now" button is at the bottom.

Returning to home screen and pressing the block button activates blocking mode.



Block screen

Task 3 Mobile- Blocking (Complex)



Petivity mobile set up screen shows up click to skip syncing, or alternatively input mobile code.



Tap pet to go to blocking page



Click start to initiate blocking



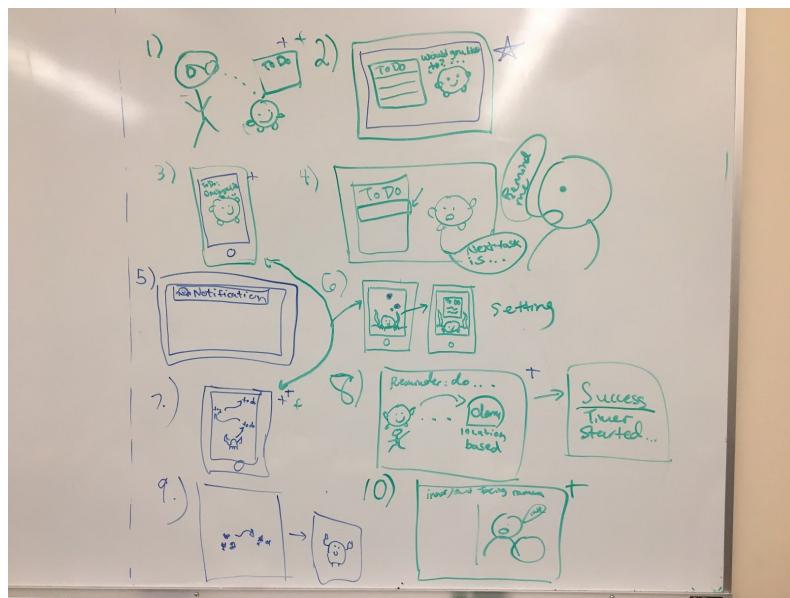
Timer runs until you click stop or navigate away from app



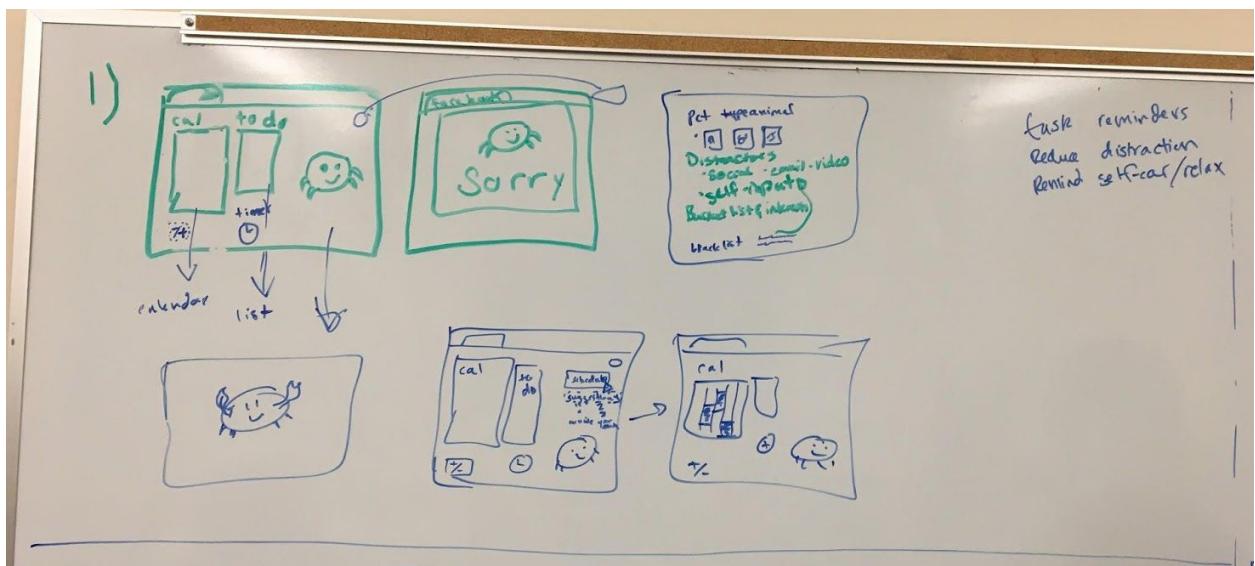
Pop up shows up congratulating you on blocking for whatever time you were not using your phone

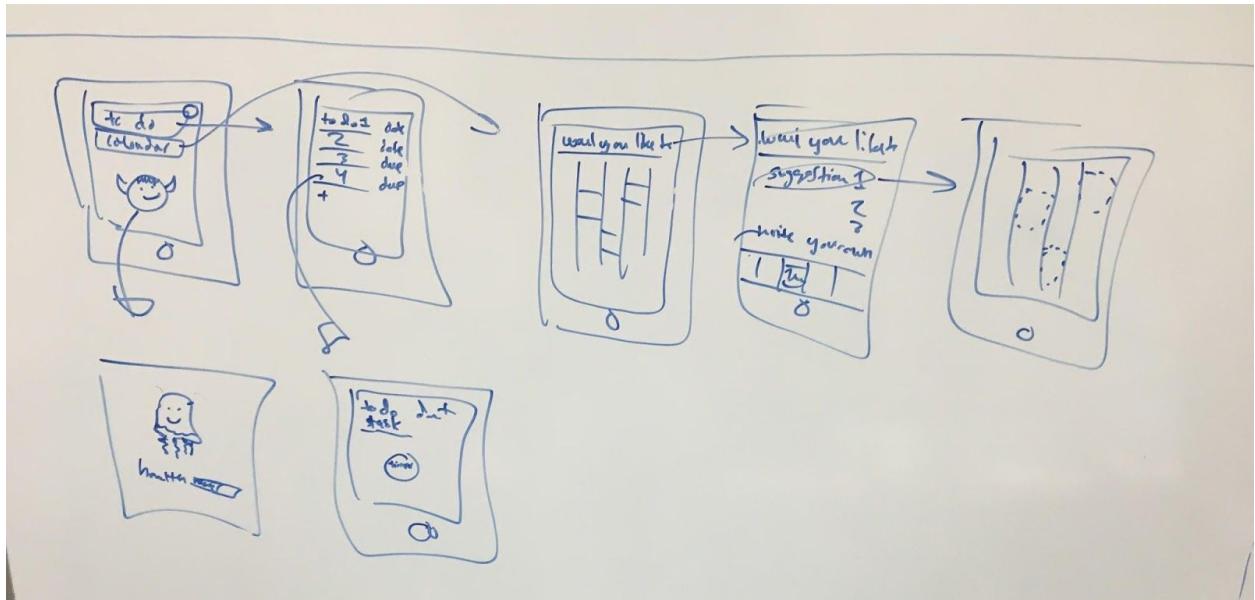
Design Evolution

Initial Sketches



Brainstorming stage of 10 possible designs





Initial Web and Mobile Sketches

Design 1: Web Application

Pros	Cons
<ul style="list-style-type: none"> Larger Screen Greater effectiveness in reminding user of upcoming tasks/deadlines as it displays with your new tab in browser (likely in the midst of user working) Can completely block distracting websites Requires less effort to open, as the app automatically opens with new tab in browser 	<ul style="list-style-type: none"> Isn't mobile, so it may only be effective on your computer Can't block mobile phone related distractions Can't receive reminders if not using computers

Design 2: Mobile Application

Pros

- Could use app to discourage use of other distracting apps
- Follows you everywhere, can receive reminders even when you're not actively using phone
- Be able to manage users' mobile usage (Perhaps similar to Forest app)

Cons

- Reminders of tasks/deadlines aren't as effective since you have to keep navigating to app to view them
- Would be difficult to fit a calendar, to do list, etc on one screen, would be inconvenient to have them on separate screens

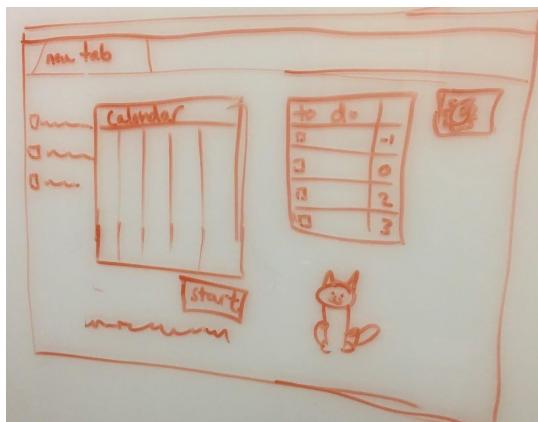
Pros/Cons for design

Initial Sketch Decisions:

Our two design storyboards: one for a web-based application, another for mobile. At this stage, from weighing pros/cons, we ultimately selected web-based because a lot of users stated that they mostly work on their computers and would like an effective way to stay focused while on the computer. Furthermore, a web application can provide a more convenient interface to access functions, since there is more screen space than a phone.

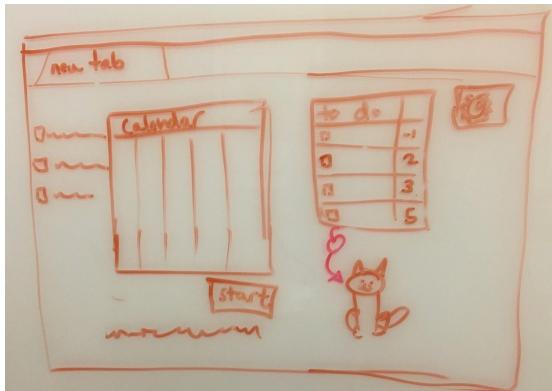
Initial Sketch Storyboards for 3 tasks

Task 1. Be reminded/aware of oneself's tasks and schedule

	<p>Open new tab in browser and application's home screen is displayed.</p>
---	--



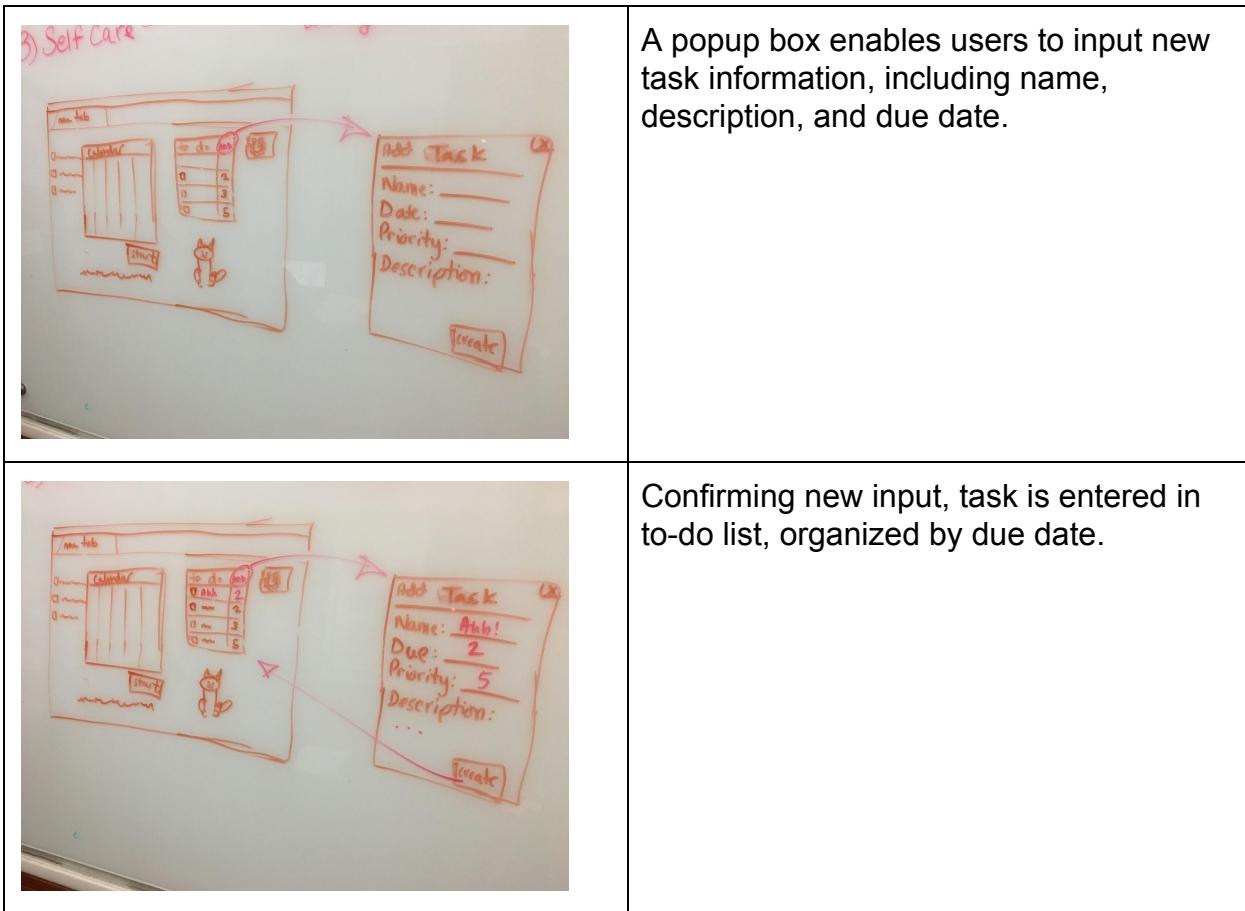
Clicking a checkbox in to-do list to complete a task.



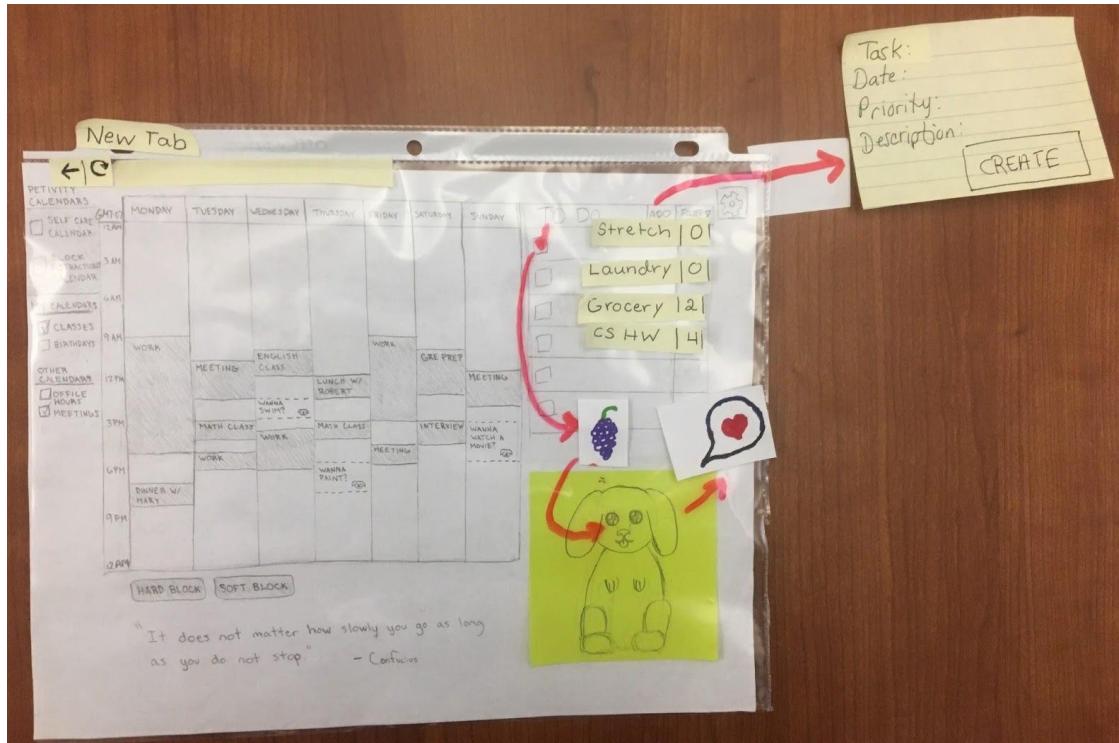
Completing a task causes food to fall from list to feed the pet.



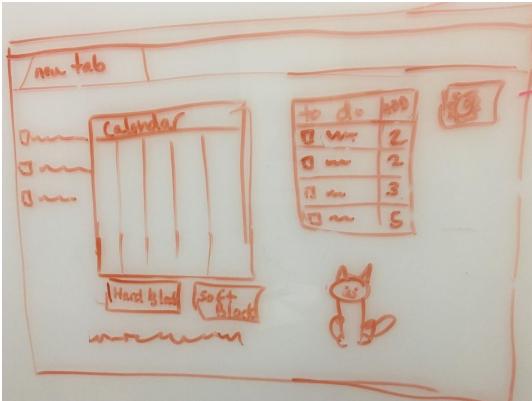
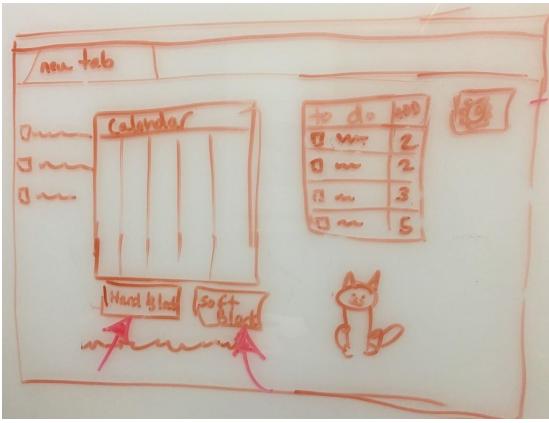
Adding a task to to-do list, user clicks on 'ADD' button or empty task slot



Task 1 Lo-Fi prototype storyboard:

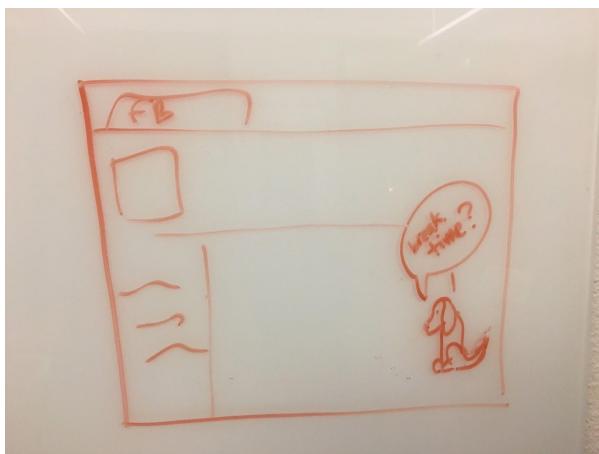


Task 2. Block distractions

	From home screen, user can navigate to settings to specify websites for blacklist (among other options).
	Return to home screen after.
	Pressing the block button activates blocking mode (either hard or soft).

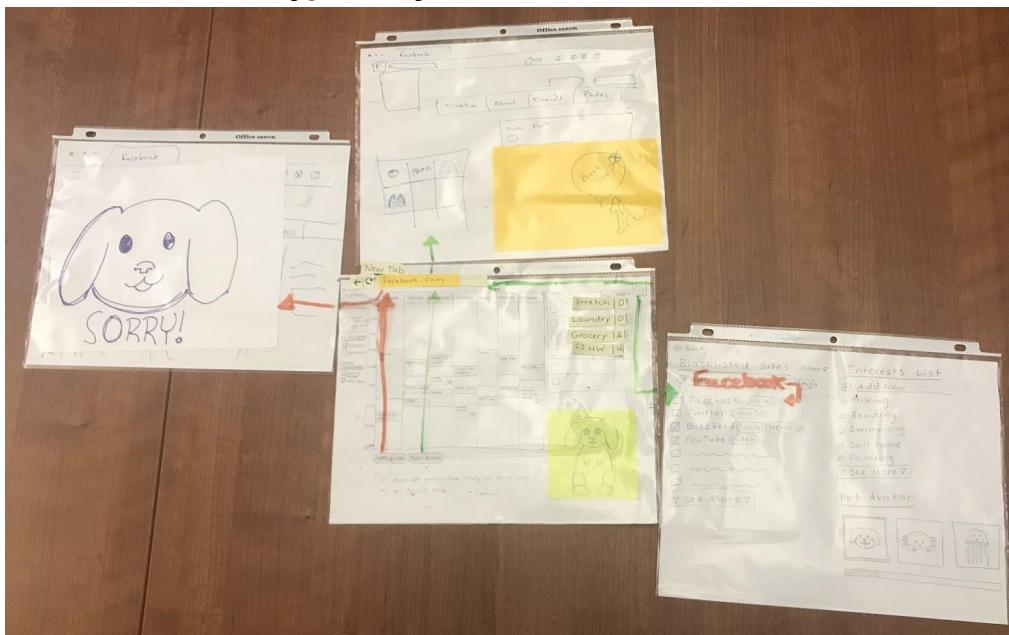


Hard block screen

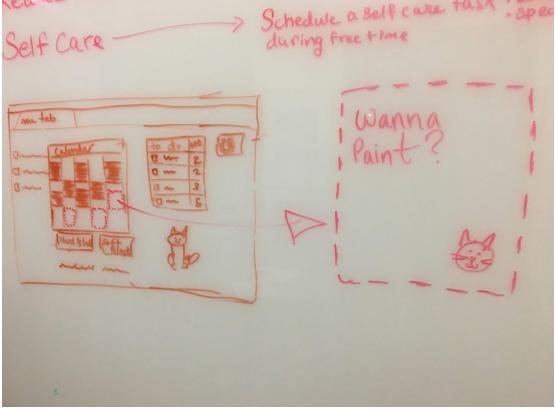
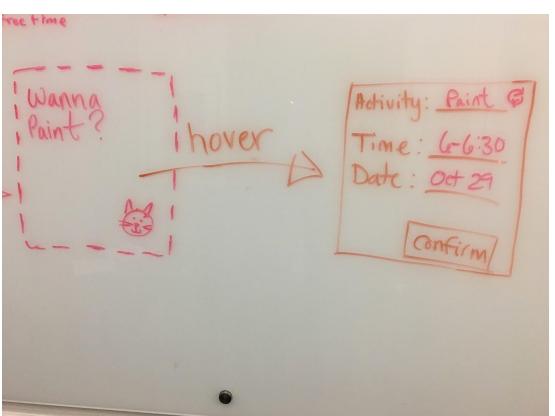
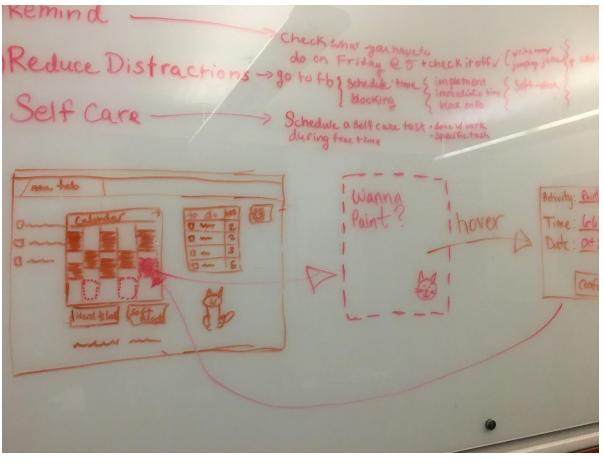


Soft block screen

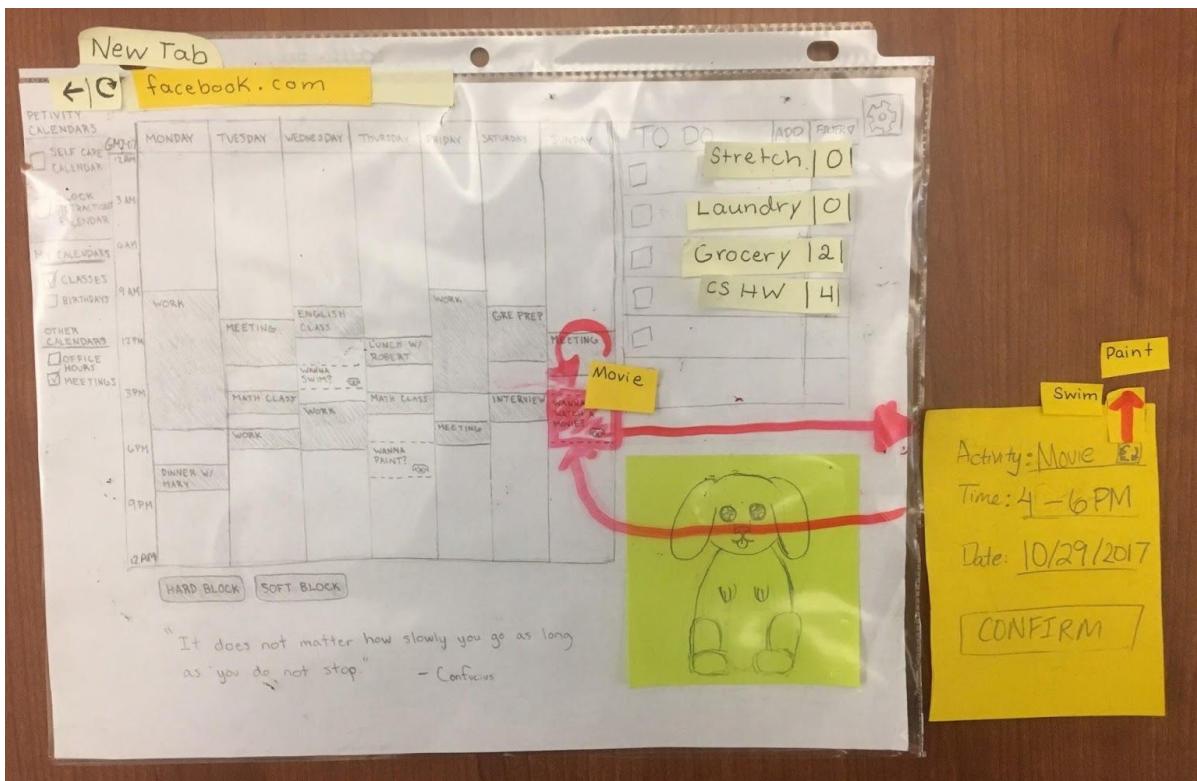
Task 2 Lo-Fi Prototype storyboard:



Task 3. Remind oneself to pursue personal activities

	<p>Self-care suggestions automatically appear as dotted entries in calendar for user to press. Suggestions are based on user specified hobbies/activities.</p>
	<p>Hovering over calendar suggestion, a popup box allows user to edit activity, time, and date. Refresh button changes activity to different one.</p>
	<p>After confirmation, activity is entered into calendar.</p>

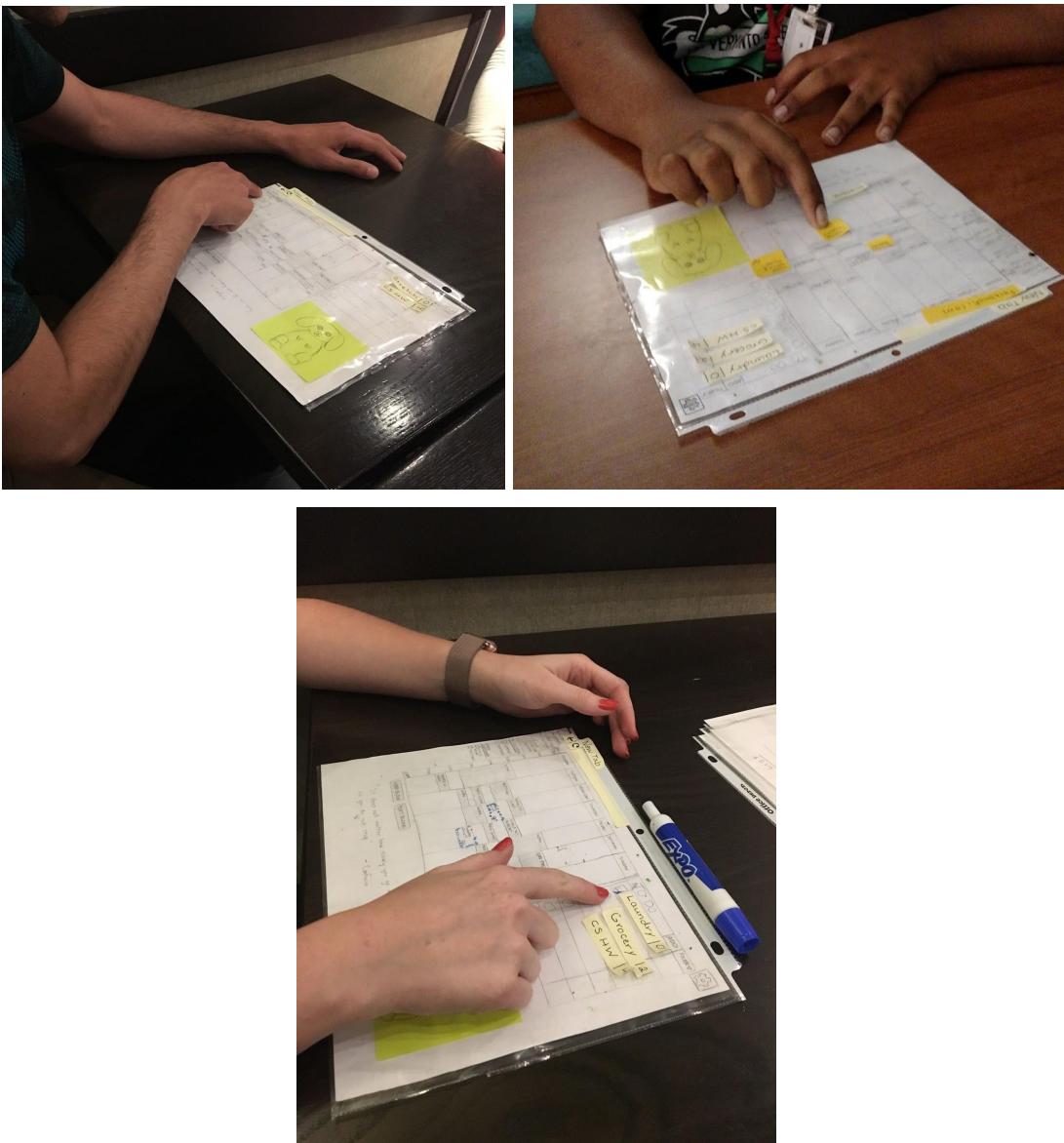
Task 3 Lo-Fi Prototype storyboard:



Complete Lo-Fi Prototype:



Lo-Fi Usability testing:



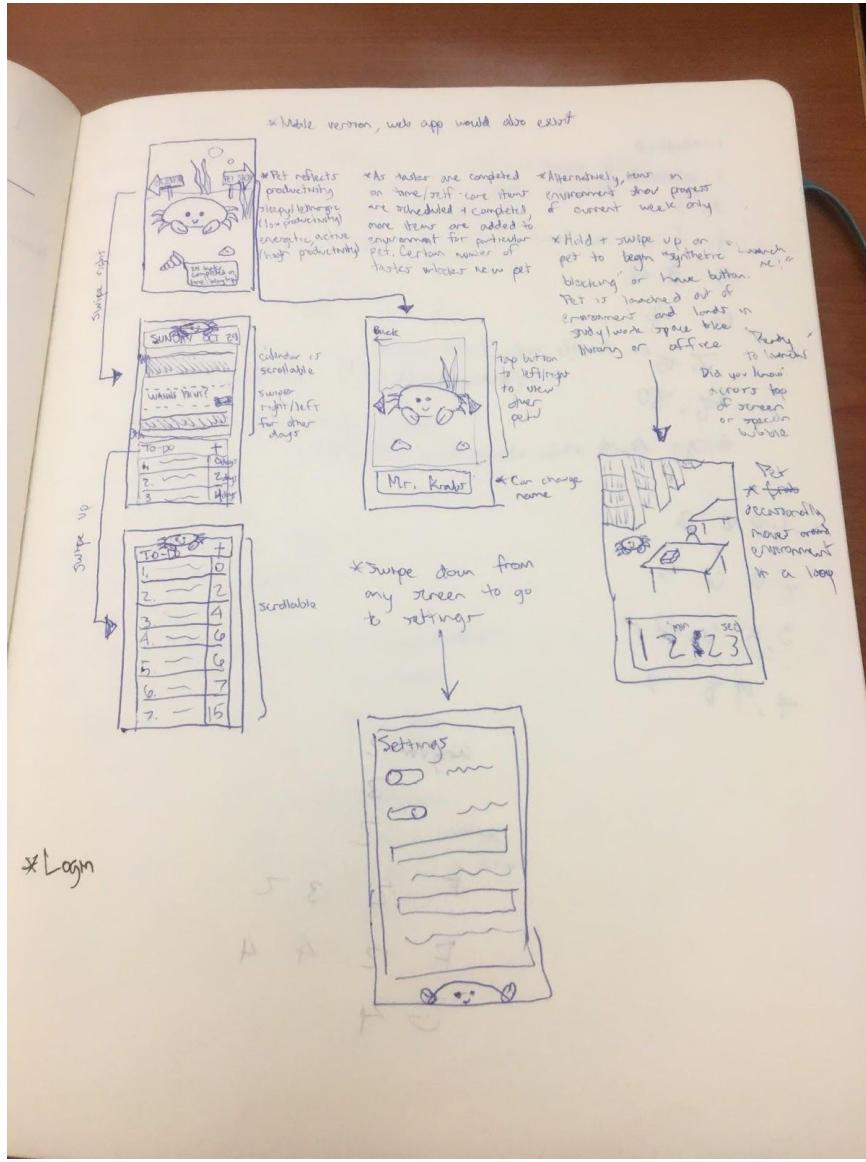
Lo-Fi Prototype Evaluation and Feedback:

Regarding task 1, we learned that it was relatively intuitive for users to complete and add to-dos to their to-do list. Users suggested that the setting task priority was superfluous. As a fix, we decided to remove task priorities and organize tasks by number of days left until deadline. Regarding task 2, we learned that we should make the settings button more noticeable or include a more convenient way for quickly adding distracting websites on the main screen. Users were confused by the “hard block” and “soft block” terms. Thus, we needed to change the word choice and integrate block buttons into a single button. Regarding task 3, we realized users had trouble distinguishing suggested self-care items from scheduled items in calendar. So we noted

to use color, shape, and/or animation to further distinguish them in the Hi-Fi prototype. Overall, all our participants appreciated the product and commented that they would use it, especially enjoying the pet. All users visibly showed enthusiasm for the pet reward system for Task 1, and one even commented that it made the application “unique.”

Medium-Fi Prototype:

Major Design Change:



Detailed Mobile storyboard sketch:

Added mobile component to augment our application to be a hybrid web-mobile application based on feedback from course instructor and TA.

Web Medium-Fi Prototype:

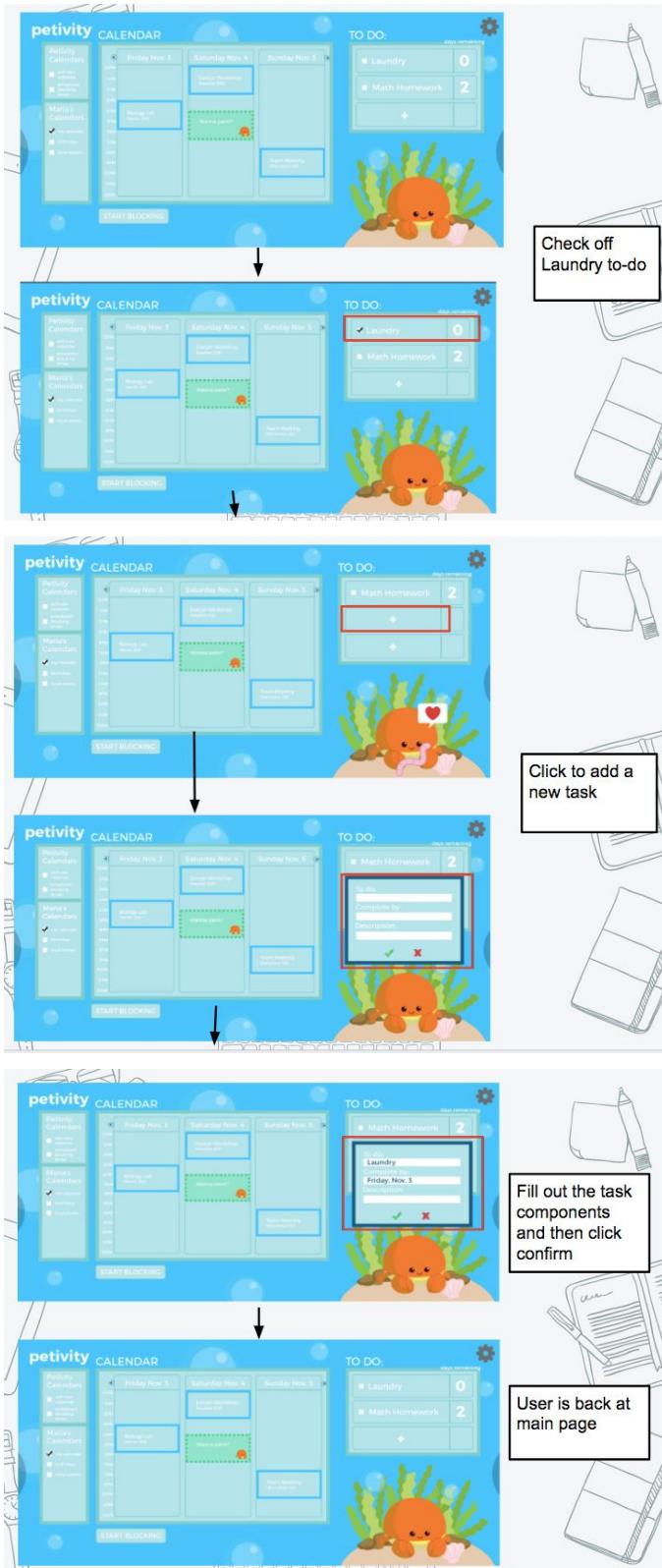


Mobile Medium-Fi Prototype:

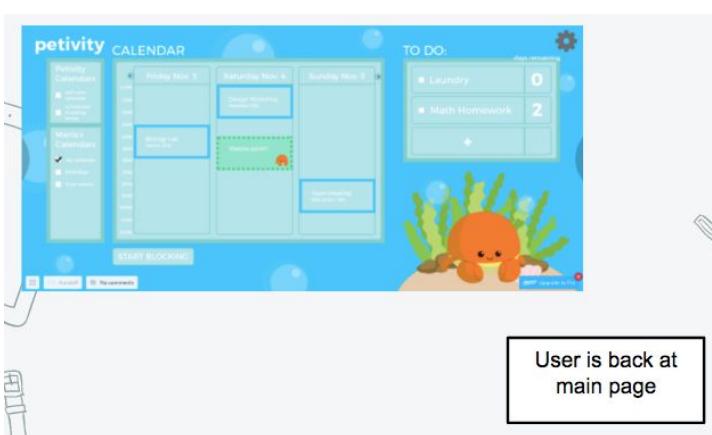
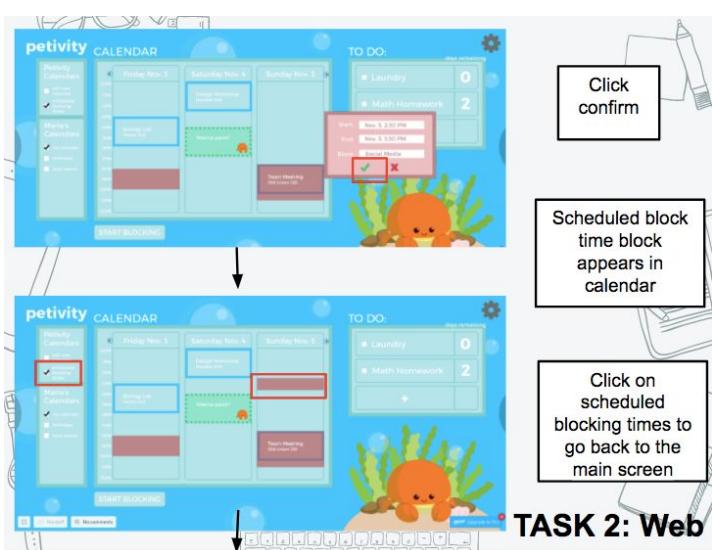
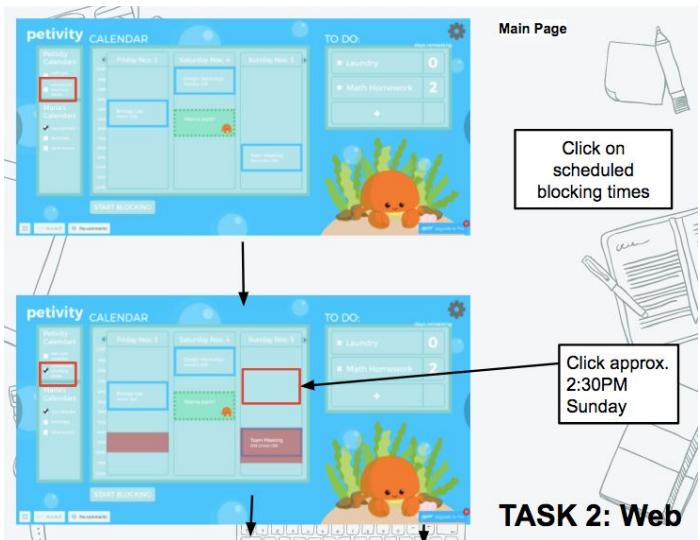


Medium-Fi Prototype Task Flow:

Task 1-Web: Complete and add tasks



Task 2 - Web: Block Distractions



BLOCKED SITES

- ✓ Facebook
- ✓ YouTube
- ✓ Buzzfeed
- ✓ Reddit
- ✓ Twitter
- ✓ Instagram
- ✓ Pinterest
- ✓ Vimeo

INTERESTS

- Painting
- Swimming
- Movies

PET AVATAR

Blocking of Buzzfeed should be enabled now

Select to add a new website to block

BLOCKED SITES

- ✓ Facebook
- ✓ YouTube
- ✓ Buzzfeed
- ✓ Reddit
- ✓ Twitter
- ✓ Instagram
- ✓ Pinterest
- ✓ Vimeo

INTERESTS

- Painting
- Swimming
- Movies

PET AVATAR

www.tumblr.com

Type in new website (tumblr) to add it to the blocked sites list

BLOCKED SITES

- ✓ Facebook
- ✓ YouTube
- ✓ Buzzfeed
- ✓ Reddit
- ✓ Twitter
- ✓ Instagram
- ✓ Pinterest
- ✓ Vimeo
- ✓ Tumblr

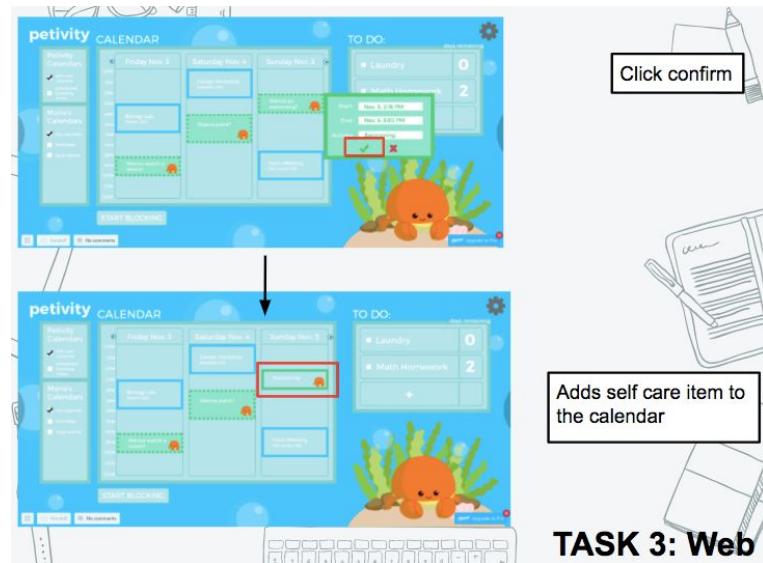
INTERESTS

- Painting
- Swimming
- Movies

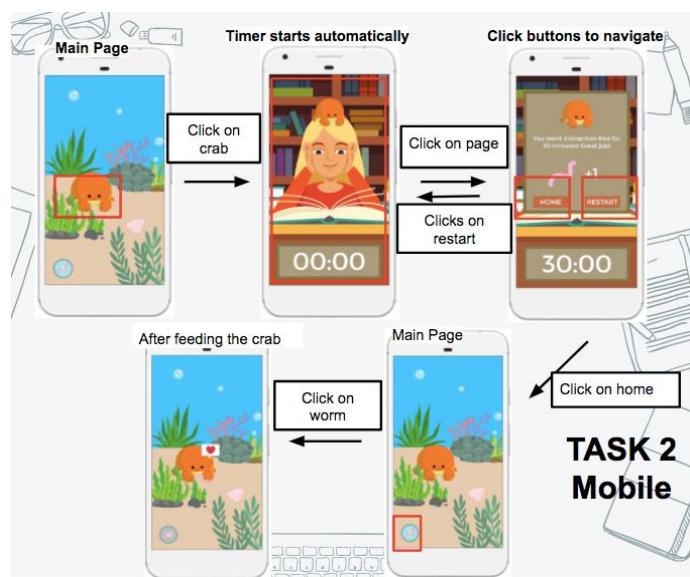
PET AVATAR

TASK 2: Web

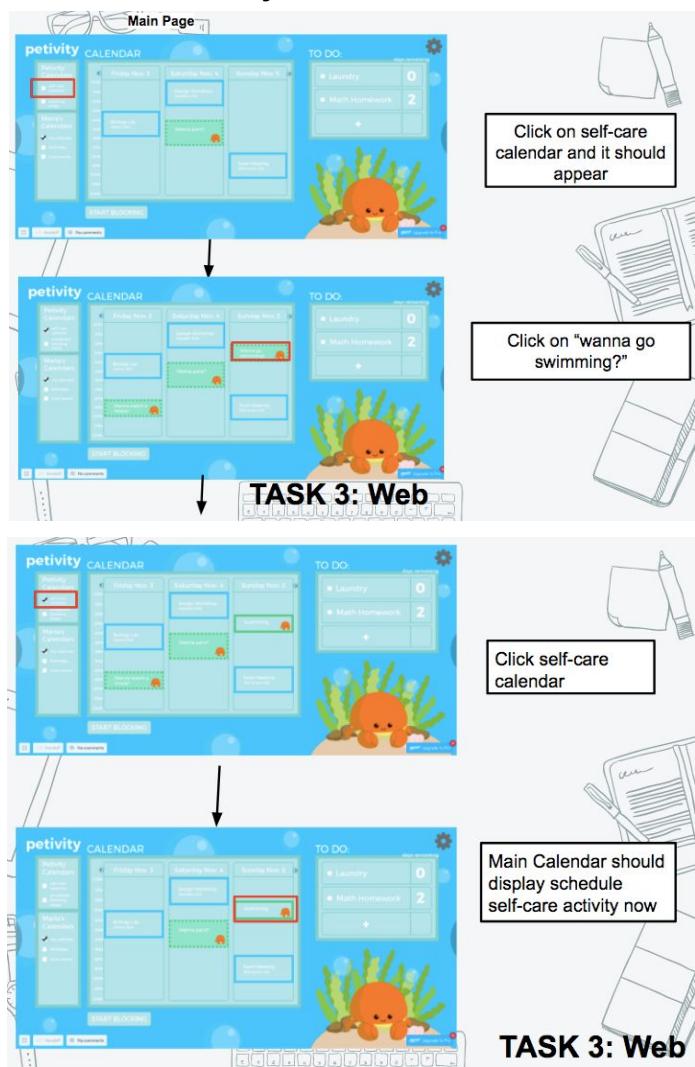
TASK 2: Web



Task 2-Mobile:



Task 3-Web: Schedule self-care activity



At this stage we realized that task 2 is more of the complex task than task 3 due to the addition of the mobile component and added functionality/steps. Description and discussion of final changes due to HE contributing to Medium-Fi prototype is explained in section below.

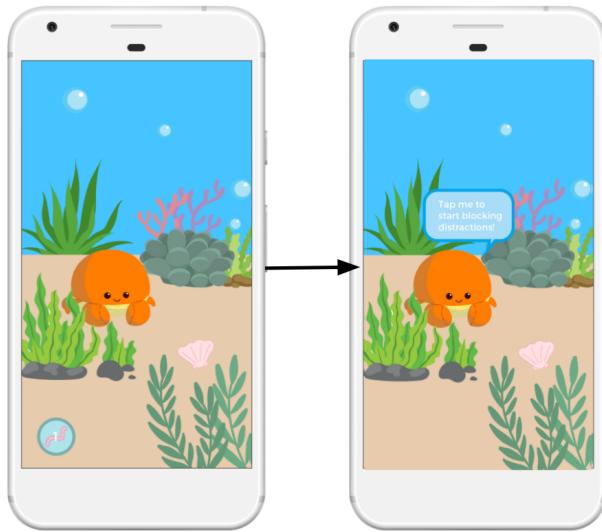
Major Usability Problems Addressed

H10 Help and Documentation: Severity 4

Problem: No user instructions on mobile home page after sign in

In our medium-fi prototype the user would sign in and be brought to the main screen of the mobile app, where the pet would be in its own environment and there was a food item in the bottom left corner of the screen, but nothing else. A new user would see this screen and would not know what to do next.

Solution: We added a speech bubble to the pet that says “Tap me to start blocking distractions!” so that a new user would know to tap on the pet to navigate to the mobile blocking screen.

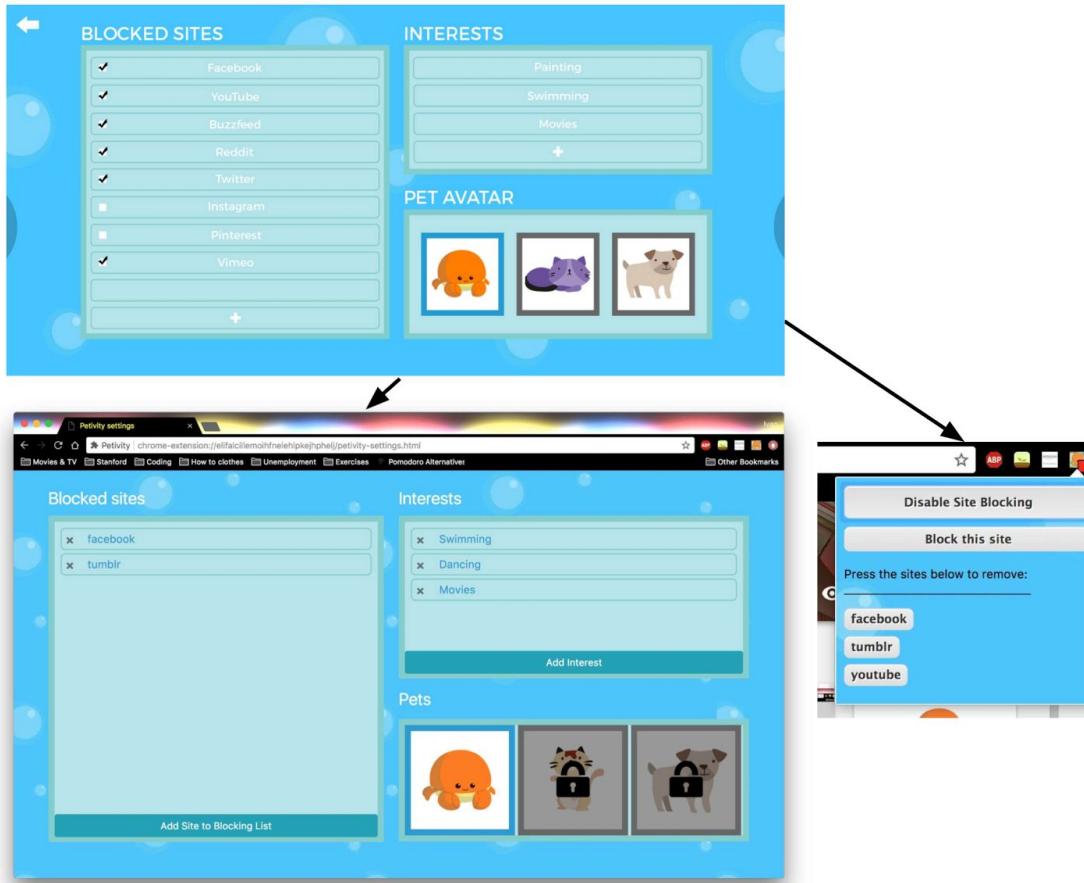


H5 Error Prevention: Severity 3

Problem: Website's Blocked Sites section does not check if URL is valid/handles error

Our medium-fi prototype had no way of handling errors/checking validity of websites when users entered them (though for this prototype we created a task flow that only allowed for valid websites to be entered) and a malformed input could potentially cause the blocking feature to not work.

Solution: Although we would like the ability to check whether www.randomwebsite.com is an actual website and produce an error message if it's not, it was beyond our technical capabilities. As a workaround, we made it so that you can use common names to block websites instead of URLs, so the user does not have to worry about malformed URL inputs (facebook instead of <https://www.facebook.com/>) causing blocking distractions to not work. In addition, we added a button to the chrome extension which automatically copies the current website's url to the list of blocked websites to avoid invalid user input.

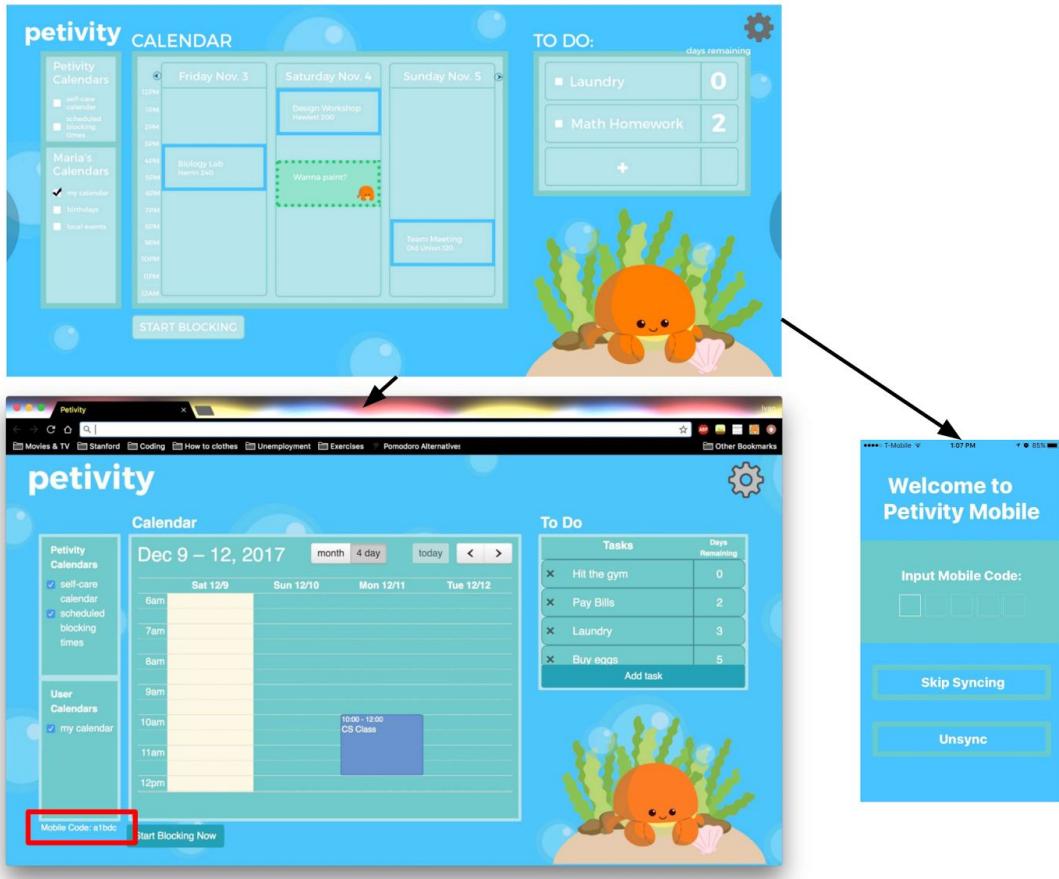


H4 Consistency and Standards: Severity 3

Problem: No sign-in flow on web, while there is one on mobile

Our mobile app required that the user logged in to use it while the chrome extension did not require any login information in our medium-fi prototype. This may have confused users, especially since the mobile app and chrome extension are supposed to be synced and users may wonder how they are synced if there's no login on the chrome extension.

Solution: We removed the login pages from the mobile app so that both the mobile app and chrome extension would not require a user to log in. To sync the mobile app and chrome extension, users instead are to enter a unique mobile code on the mobile app, given to them on the chrome extension.

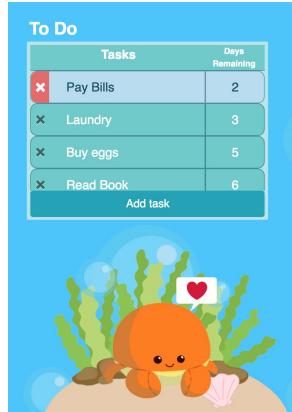


H1 Visibility of System Status: Severity 3

Problem: Pet does not have a status state, no indication of progress

In our medium-fi prototype users expected that there would be some way of checking up on a pet's status/state that would represent the progress the user has made in completing tasks. Although we planned to implement this, it was difficult to show in a medium-fi prototype where such changes are triggered after significant periods of time or inactivity (pet goes from happy to sad) or are triggered by the completion of numerous tasks/multiple days of staying on track (pet grows/new pets unlocked).

Solution: In our high-fi prototype, we currently have completion of tasks triggering animation of a treat and heart pop-up. For our identified task flows, showing the pet status was deemed not completely necessary and not very easy to implement under the time constraints. Nevertheless, in the future, we would like to implement additional functionality addressing this aspect. One idea would be to have the pet change emotion over time based on task completions or other positive user activities as part of a positive feedback system.

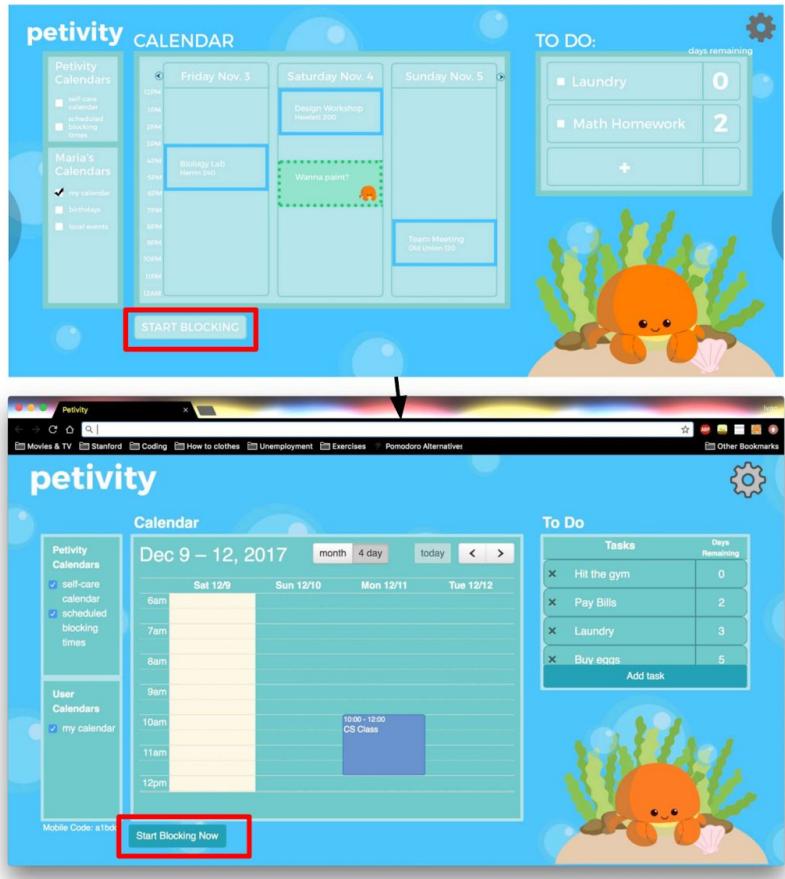


H2 Match between System & World: Severity 3

Problem: Confusion over what “Start Blocking” button does

Users were confused with the “Start Blocking” button in the medium-fi prototype due to the various mechanisms in place for blocking distractions, such as the “scheduled blocking times” calendar in addition to this button. It was hard for them to understand how this button fit into the distraction blocking task flow. The “Start Blocking” button is intended for users to activate the distraction blocking feature immediately.

Solution: To address this problem, we added more clarification by changing the wording of the button text to “Start Blocking Now” in our high-fi prototype. That way users would know that the button is only for activating the distraction blocking feature now, having no relationship with the “scheduled blocking times” calendar.



Other changes:

- We altered the color scheme to improve readability/visibility of text
- Added a clear “Add Task” button
- We changed the design of the calendar since we had difficulties implementing our own calendar from scratch. We decided to use FullCalendar and alter its design to fit our aesthetic instead
- The chrome extension also included buttons/displayed information related to our blocking feature in the browser’s toolbar for easy access, which was not present at all in the medium-fi prototype

Prototype Implementation

■ Mobile Tools:

- Expo XDE
- ReactNative

■ Web Tools:

- JQuery
- Bootstrap

- Google Chrome Extension Developer Tools
- Google Calendar
- FullCalendar

■ How the Tools Helped:

To build the mobile app prototype, we used Expo XDE and ReactNative, which allowed us to build an iOS mobile app as well as test it. To build the chrome extension we used jQuery, Bootstrap, and Google Chrome Extension Developer Tools, Google Calendar, and FullCalendar. Google Chrome Extension Developer Tools allowed us to build a chrome extension and test the chrome extension as it would appear for a user. Bootstrap was used mainly for its grid system which simplified the process of creating a layout/structure of our chrome extension. jQuery was used for easier DOM manipulation and animations. To build the calendar for our chrome extension we used FullCalendar and imported events from a Google Calendar. For version control we used GitHub, which allowed us to develop the mobile app/chrome extension collaboratively and gave us a repository to store our files.

■ How the Tools Did Not Help:

The tools were not helpful in that some, like ReactNative, were a bit difficult to set up before we could actually use them. It was also difficult for us to find/learn the parts of the tools that would allow us to accomplish what we wanted for our mobile app/chrome extension, such as using jQuery for our animations. We also found that utilizing various combinations of tools produced conflicts in trying to accomplish what we wanted, which happened when we attempted to use Bootstrap and AngularJS together or the chrome extension with the Google Calendar API. This often halted our development for hours at a time as we searched for and tested workarounds/decided on an alternative set of tools to use.

■ Wizard of Oz:

- Pet selection: We gave the illusion that the user only had one pet unlocked and there were other pets available for use only unlocked once they acquire certain number of points.
- Self-Care Suggestion Artificial Intelligence: We gave the illusion that the self-care activities would auto-populate your calendar based on available free space, and how often you did the activity, when you did it last.
- Self-Care Scheduling: We gave the illusion that clicking one of the self-care suggestions
- Syncing: We gave the illusion that the mobile app and web chrome extension synced together so you implement blocking at the same time.
- Blocking Calendar Scheduling: We gave the illusion that if you schedule a blocking event on your calendar, the Google Chrome Extension will automatically block distracting websites during that time.

■ Hard-Coded Data

- Calendars: We hard-coded the events by using Google Calendar for our backend, and creating all the events and timings of events for demonstrative purposes for our self-care calendar, regular calendar, and blocking calendar.
- Customization: We hardcoded the pet selection, the backgrounds and the food option.

■ Missing and Desired Future Implementations:

We weren't able to implement a point or earning system to keep track of how productive a user is. Thus we weren't able to implement a pet that actually grows or develops as you become more productive, nor a system for unlocking new pets (although we used Wizard of Oz for it).

We also had to use some other Wizard of Oz techniques/hard coded data and ideally in the future we'd want to have these features actually implemented. We would have self-care suggestions completely developed. Actual artificial intelligence would be implemented so that suggestions populate the calendar based on empty blocks of time, and be based on how often the user engages in particular activities and the last time they engaged in it, etc.

In addition, while our mobile application featured the blocking aspect of the chrome extension, we would wish to incorporate more features in the mobile app to bridge the gap between functionality between the two platforms and bring more portability. We also used Wizard of Oz for syncing and wish to actually create syncing for the mobile application.

Ideally, we'd like to offer customization for our prototype (although we did offer some through Wizard of Oz to for pet selection). We would provide multiple environments and pets, and a way to name the pet. We'd also like to offer more interactions and animations with the pet based on how productive you are and what activities you do, as well as a variety of food items to feed the pet.

Summary

Developing Petivity and seeing its evolution through the various stages has been a very rewarding process for all of us on the team. We were able to tackle challenges and explore a novel problem area, applying various techniques we learned in studio and lecture. Under time constraints, we were able to learn a lot of new technologies and tools very quickly and apply them effectively to produce prototypes. We are glad that through iteration and refinement, we were able to create a product that people want/enjoy and find value in.