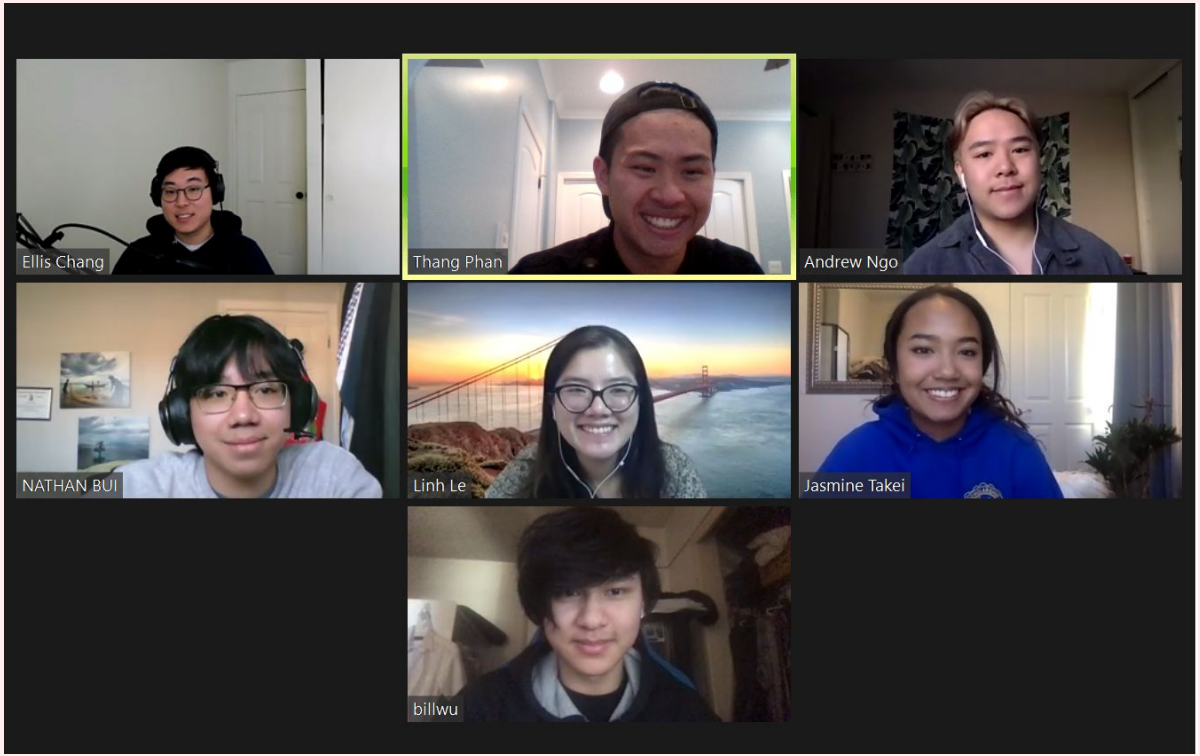
Boolean Barbz

Pitch Document

Statement of Purpose

Boolean Barbz's purpose is to complete the CSE 110 group project which is creating a timer that has the correct properties of a pomodoro timer

Our team


Ellis Chang
Thang Phan
Andrew Ngo
NATHAN BUI
Linh Le
Jasmine Takei
billwu

Problem

Most pomodoro timers out on the web do not employ the true pomodoro techniques. In addition, some either have too many distracting visuals or are too complex to use. Below are 3 prominent features that shouldn't be on a pomo timer that we have found in numerous pomo timers on the web .

01 | The ability to pause your work or break session

02 | Unnecessary settings and tabs

03 | Distracting visuals

Solution

Our solution is to create a minimalistic timer that employs pomodoro techniques that has an easy to use, non-distracting, and skeuomorphic interface. Our framework for our solution will focus on addressing the issues of effective time use, focus, and reducing anxiety

User Personas and Use Cases

Martha

- Female, 26, Office Job Professional
- Uses computer a lot
- Very structured

Martha wants to use a pomo timer is order to time manager her professional work as well as have structure breaks.

John

- Male, 32, Works from home
- Writer
- Bad time management

John wants to use a pomo timer is order to not burn out while writing. He tends to just keep writing if not structuring his breaks.

Andrew

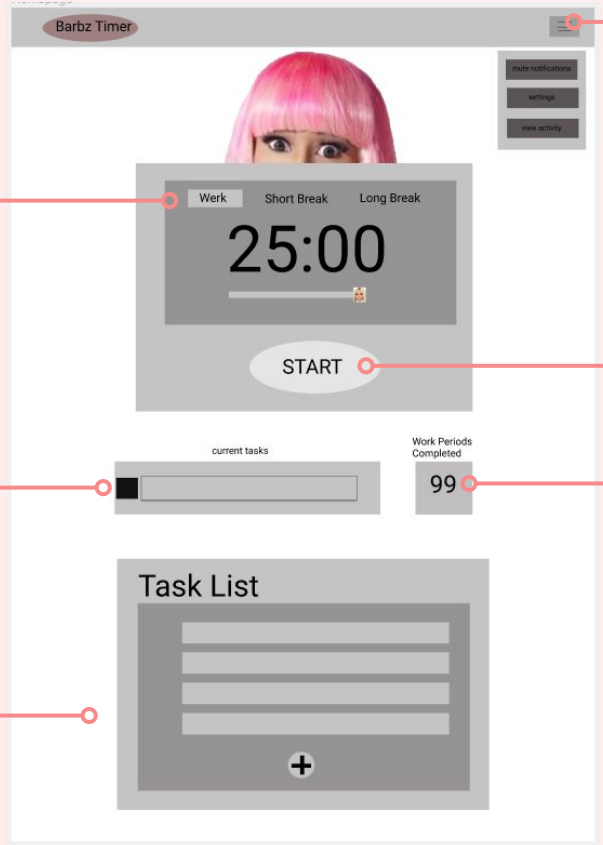
- Male, 20, UCSD Undergraduate
- Computer Science major
- Gets distracted e

Andrew wants to use a pomo timer in order to stay on track when doing school year. His major requires tons of hours a week so when he's distracted he falls behinds a lot

Basic Features And Nice-To-Haves

Features	Start Button	Stop Button	Display Remaining Time	Checklist	Ability to change times of the various categories	Notifications	Time tracking for individual tasks	User Analytics
First Iteration	✅	✅	✅	✅				
Second Iteration	✅	✅	✅	✅	✅	✅		
Third Iteration	✅	✅	✅	✅	✅	✅	✅	✅

Visualize The Solution (first iteration)



Timer categories

Werk, short break, or either long break will be highlighted depending on what category the timer is on

Current Task

Shows user their current task and has a checkbox, when a user is done with their task they check it off and their next task appears

Task List

Shows users all the tasks they have, have an add button where users can add new tasks

Hamburger menu

Hamburger menu that contains user settings, mute notifications option, and activity log

Stop/Stop button

Start starts the timer and when users click it the stop button shows, vice versa. When users clicks stop, the timer resets and the work periods reset.

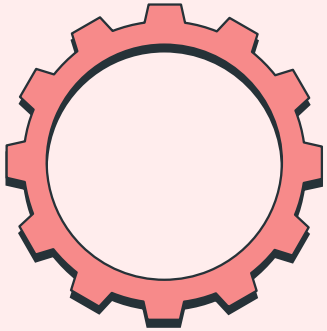
Work Periods

This counter tracks all work periods completed by the user, resets when user clicks stop

Constraints

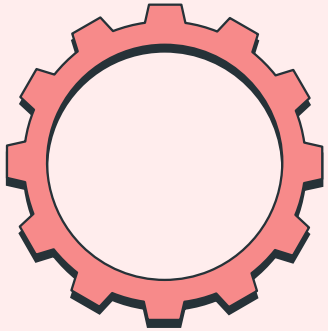
Timeline

We have until finals week to complete and have a fully functional timer



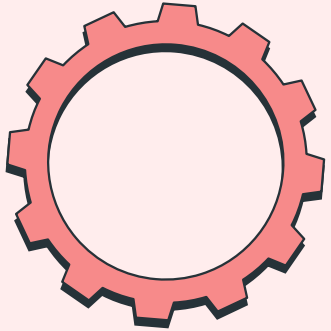
Technology

We will only use vanilla javascript and must prioritize front end aspects



Features

We will not overload our timer with features/cluttered UI



Risks

Since we are all relatively new to software development and the basic functionality of our timer is relatively simple, a lot of potential risks relate to process and organization such as:

- Over complicating the design and functionality of our timer
- Not enough planning before development
- No collaboration because of remote environment
- No QA standard
- The risk of rushing of project due to the time constraint
- Focusing on the wrong things (i.e backend instead of frontend)

Rabbit Holes

Although our application is relatively simple, we will still be looking at for the following rabbit holes. We plan to patch our rabbit holes via a quality assurance process

Functionality

We need to make sure we take into account all the possible user flows when scoping out and implementing the functionality of our timer

Visuals

We will have to take into account all the buttons and settings users need in the front end in order to have a good user experience

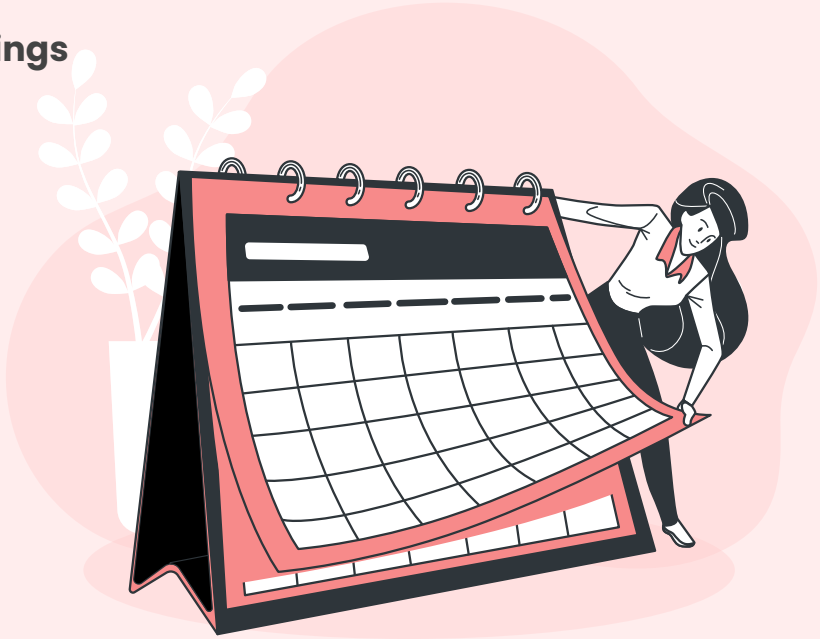
Organization Structure

Sub-teams and Meetings

We split our group into two groups: Design/Planning and Development/QA. Each team meets separately once a week and the whole team meets twice a week to go over updates.

Task Tracking

We plan to track team tasks on notion via a kanban board and Github for development tasks



Agile

We plan to operate in 1 week sprints and in lieu of daily stand ups, everyone will write daily summary notes on our notion

Tools

For design our team will be using mainly miro and figma while using github and vscode for development

Roadmap

Week 6

- Finish High-fi wireframes
- Start 1st iteration development

Week 7

- 1st iteration development

Week 8

- Finish up 1st iteration development
- Q/A for 1st iteration
- Design modifications for 2nd iteration

Week 9

- Start 2nd iteration development
- Design modifications for 3rd iteration

Week 10

- Finish up 2nd iteration development
- Start 3rd iteration development
- 2nd iteration Q/A