

Nap Cycle

Product definition statement: Nap Cycle verifies the user is asleep before setting a timer for the ideal napping period, guaranteeing the user a refreshing nap at just the press of a button.

Reason: We want the app to be as simple as possible, We want the app to be synonymous with getting a cup of coffee, It should be as stress free as possible. And simply click a button or two, fall asleep and we take care of the rest.

User: Is anyone that feels under rested, or tired in the middle of the day. That's a wide spectrum of people so we are going to be mostly focusing on college students, but could also cater to stay at home moms, people working multiple jobs, and busy self employed/ entrepreneurs with flexible schedules. All these groups have some but minimal disposable income. Age, Gender, Ethnicity, Native language, Religion, and Political affiliation vary widely among suspected user base, so it's important to build the UI to be user friendly to all demographics. For example building the UI to not rely so much on english text and more on intuitive inputs.

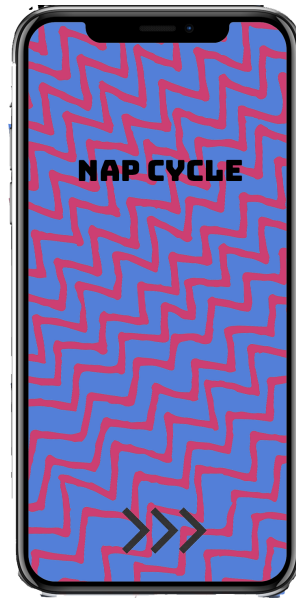
Market analysis: Based on research from Statista a median of 8% of the US population uses a sleep tracking app regularly that's about 24 million users, 24% have at least tried a sleeping app and at least 68% of people can imagine using one. That's 72 million and 204 million people respectively. The range of our user base is anywhere between 24 and 204 million users. When looking at similar apps, getting a piece of that market share ends up being anywhere between 10k and 400k downloads depending on level of success. Resulting in anywhere between 5k and 700k total revenue.

As for competition, there are no major apps presently on the market that fulfill our app's exact role. The closest things that exist are sleep trackers, customizable alarms, and rem-cycle-based wake-up alarms that go off at an optimal time. None of these are marketed or optimized towards nappers, and none use our key innovation of starting a timer once the user falls asleep. Even so, the apps are quite popular; some with hundreds of thousands of downloads. As elaborated on above, there is a clear market for our app.

Budget: Based on the market a relatively unpopular sleep related app had 10k downloads and the most successful app had 400k downloads. Selling at a dollar per app would bring in about \$10-400k dollars. After Apple takes their cut we are left with 7-280k dollars. After taxes we are left with 7 - 182 thousand dollars. Assuming 7k is too low to be taxed and 280 will be taxed at 35%. We are unsure how to estimate advertising so we will put 30 percent into advertising. Being left with 4.9k - 127.4k left over as revenue for payroll split two ways is 2.45k - 63.5k left over for salary. Obviously that's a wide range—we're covering our bases and calculated a realistic scenario versus an ideal scenario.

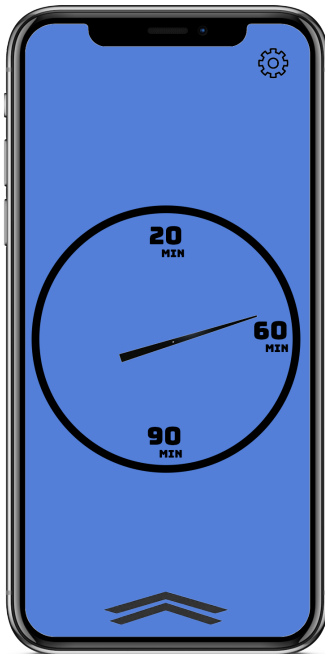
User scenario:

Brian isn't getting enough sleep. He has a mid-day slump around 4:30 everyday between work and school. He hears about an app from a friend that helps people nap the perfect amount of time. He downloads and opens the app. He is met with a hypnotic welcome screen with moving zig zags and the title reading "nap cycle".



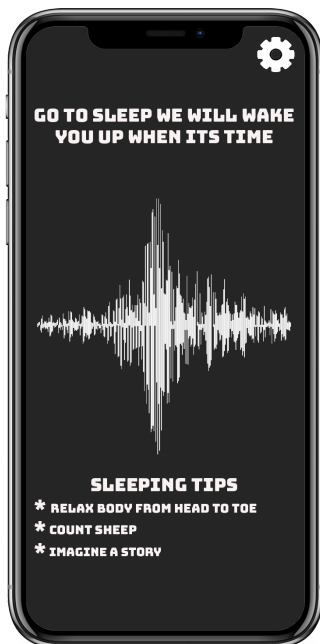
He gestures in the form of a tap or a swipe to get to the next view. He is met with a brief loading screen displaying instructions on the three steps required to start the nap. The instructions are in single words, accompanied by symbols designed to communicate how to use the app.

Confused as to which one to select, Brian sees a bouncing arrow at the bottom of the page. He swipes up on it and is met with a description of the 3 durations of nap and the benefits they provide.



He decides against the 60 min nap because grogginess is a side effect, and does feel he has time for a 90 minute nap. So he swipes down and turns the dial to “20min”.

Next he is met with a view titled “safe guard” with a text that reads “what time do you absolutely have to be up by”. Underneath the text there is an iphone or 80s alarm clock style time picker. He scrolls through the hour and minute to select 5:23. Then toggles the Am/Pm to Pm.



He swipes to the next view titled “go to sleep”. Under the title the caption reads “we’ll wake you up when it’s time”. Under that a visualizer shows the noise in the room to signal the brian that the app is listening and working. Below are tips on how to fall asleep. Brain lays in bed for a minute restlessly but eventually falls asleep.

He is awoken by his phone vibrating and sounding an alarm, 20 minutes after he fell asleep and 50 minutes after setting the alarm. This is three minutes before his worst case scenario time. He is greeted with a welcome back screen, telling him the current time and how much sleep he actually got.



He feels refreshed. He clicks on the setting button that is in the top right corner of every view. A menu slides over from the left. On it includes the options “connect to your device”, “view sleep data”, and “set custom alarm”. He clicks “view sleep data” but these features are not available yet.

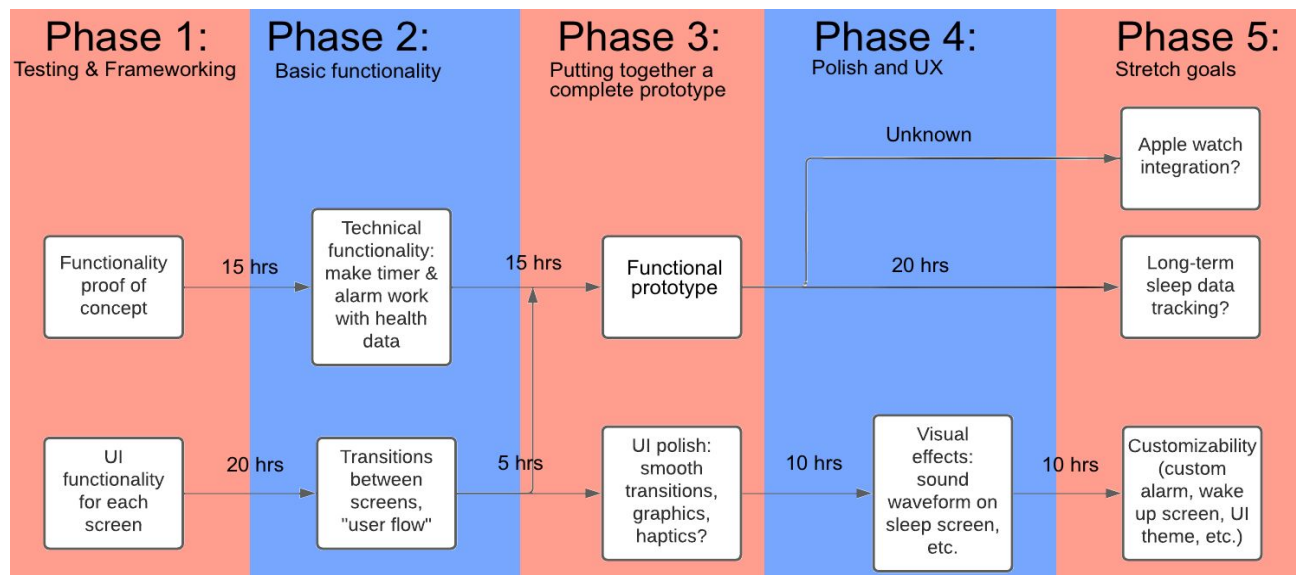
App data requirements: Our app requires pulling data from Apples Health Kit. Apple Health kit uses the M-Motion Chip inside the last few models of Iphone to determine whether a user is asleep or awake based on noise type in the room. This data is populating within the security of HK (health kit). With user permission our app can periodically pull this data to evaluate whether the user is asleep yet.

We get our data from the HK database located on the phone drive using a query with the following API: <https://developer.apple.com/documentation/healthkit/hksamplequery>

Milestone list and tasks:

Functionality proof of concept	<ul style="list-style-type: none">- Create global constant powerNap = 20, memoryNap = 60, remNap = 90- Create global variables safeGuardTime, napInterval, isSleep- Create successful query to health kit- Parse data to get previous moments sleep data- Create sleep loop that only breaks when sleep data is "asleep"
Technical functionality: make timer & alarm work with health data	<ul style="list-style-type: none">- Create timer loop that checks to see if the interval of time has passed- Create breaks from loop only when interval complete or phone is less 15% power- Create alarm function that uses speaker (to be triggered at the break of timer loop)
Functional prototype:	<ul style="list-style-type: none">- Combine all screens so a user can go through the motions of how the app would be used and watch it work
UI functionality for each screen	<ul style="list-style-type: none">- Create background for start screen- Create swipe to next screen from start- Create background for load/instruction screen- Create swipe to next screen from load/instruction screen- Build spinnable dial for selection screen- Link dial position to napInterval- Create submit button (for setting napInterval and screen transition)- Create date Picker for Save Guard screen- Create submit button for Save Guard screen- Like date from date picker to safeGuardTime on submit- Add cancel alarm button to wake up screen
Transitions between screens, "user flow"	<ul style="list-style-type: none">- Make sure the app flows in an intuitive way- Visual screen transitions- Bouncing "swipe up" indicator for info panel
UI polish: smooth transitions, graphics, haptics?	<ul style="list-style-type: none">- See above, plus:- Animated backgrounds and buttons- Final pass on UI graphics- Haptic feedback on button presses and timer dial
Visual effects: sound waveform on sleep screen, etc.	<ul style="list-style-type: none">- Gain access to microphone- Create animation linked to some variable received from microphone

Pert Chart:



Our PERT chart has two parallel workflows with relatively low dependency. The top one revolves around the backend programming and functionality of the core systems, while the bottom line deals with the UX, presentation, and polish. The one main dependency is that a functional prototype requires both the backend systems and the UI functionality / "user flow" to work.

We've (very) roughly estimated how long we think each step might take to complete, giving particular time to work through the bugs and kinks of UI design. We also divided the project into phases, to better visualize progress towards the end goal.

Future work:

Apple Watch integration: at some point we want to integrate with apple watch. Using the heart rate sensor in apple watch to get sleep data for more accurate nap percussion. We could also use the smart alarm feature in the app watch to gently wake the user.

Customizable alarms: currently we are using set time intervals but at some point we want to add a screen where users can set a custom nap interval and add it to the dial

Long term nap tracking: Very long term we would like to be able to collect napping data to use to predict a user's napping habits. Knowing things like how long on average the user takes to fall asleep, how deep they sleep, or what condition helps them fall asleep faster, could help inform the user on how to optimise their nap.

