Intro

Welcome to the nixa.io React challenge. Your assignment is to create a simple React application based on the Tractive UI Kit. The prototype is designed for the mobile screen and we'll be only concerned with testing on mobile browsers. We're assuming the website should work and appear correctly when emulating iPhone 6/7/8 in Google Chrome at 375x667px resolution.
The **total points** you can get in this challenge is the sum of points from user stories and non-functional requirements. This adds up to **100 points**.
The goal of this challenge is to check you UI coding skills, your ability to use React and its patterns as well as get a grasp on your approach to problem solving.

Specification

Prototype & design spec

Here's a video of the basic user flow described in the user stories below:
https://drive.google.com/open?id=1J56l05JuPllkOtXJZX_FY5sEwfxfsm71
You can see a clickable prototype here:
https://xd.adobe.com/view/751c01b1-8c05-4af0-695f-d34d90280861-032d/
Here are the design specs with font-sizes & assets (including SVG and image files) :
https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/
You can download the source Adobe XD file here:
https://drive.google.com/open?id=11FyEaP90ymup4l7TZPRa8jH6p9rlUGJi

Fonts

Use the following stylesheet to get access to the Europa and Rift Soft fonts used in the project:

```
<link rel="stylesheet" href="https://use.typekit.net/dke8rty.css">
```

Assets

For your convenience the icons and images can be downloaded from this design specification screen:
https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/screen/fac802bf-a440-4c32-a901-133e511a55cd/Assets/

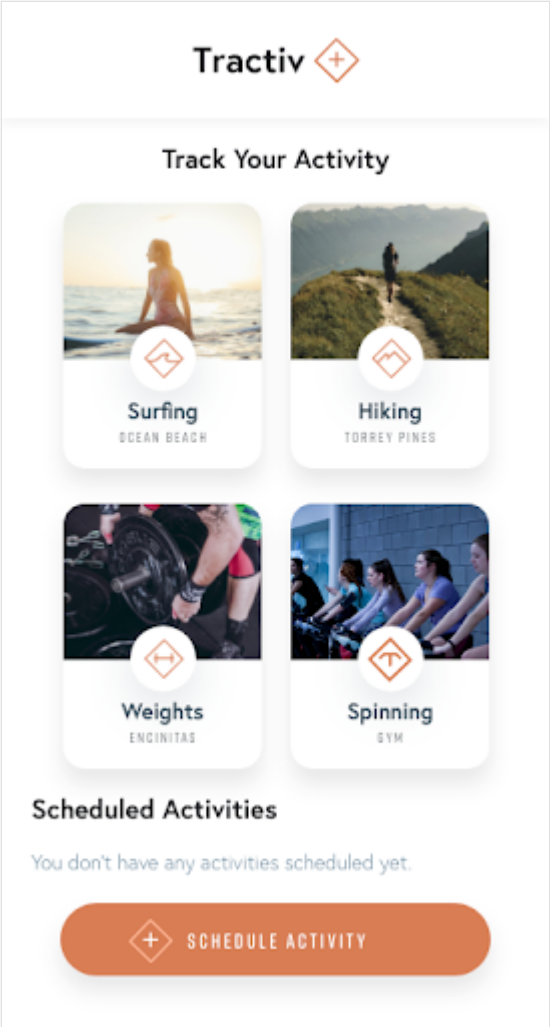
User Stories

The Tractiv app allows a user to track and schedule their activities. Below are specific user stories that need to be implemented. You can also find the information about the points awarded for completing each of the user stories.

US01: Home Screen - first run [10 points]

As the user when I open the app for the first time I see a grid of activities that they are able to track. There are 4 types of activities:

1. Surfing
2. Hiking
3. Weights
4. Spinning



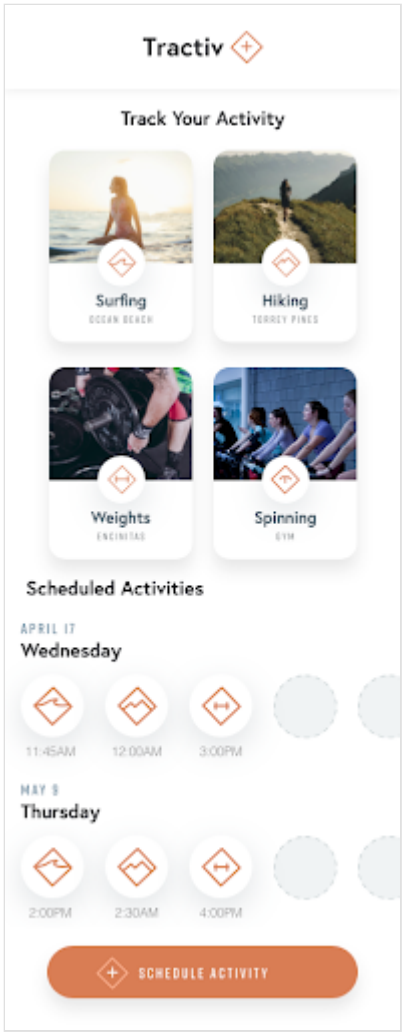
When there are no activities scheduled I should see a header and information that there are not activities scheduled.

Design Specs

https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/screen/57c69037-f4bc-4eef-9a18-ab89b79ebc99/Start/

US02: Home Screen - scheduled activities [10 points]

As a user when I have scheduled some activities to do in the future I should see them listed on the home screen. The activities should be displayed grouped by date with day of the week and date formatted as **September 1**.

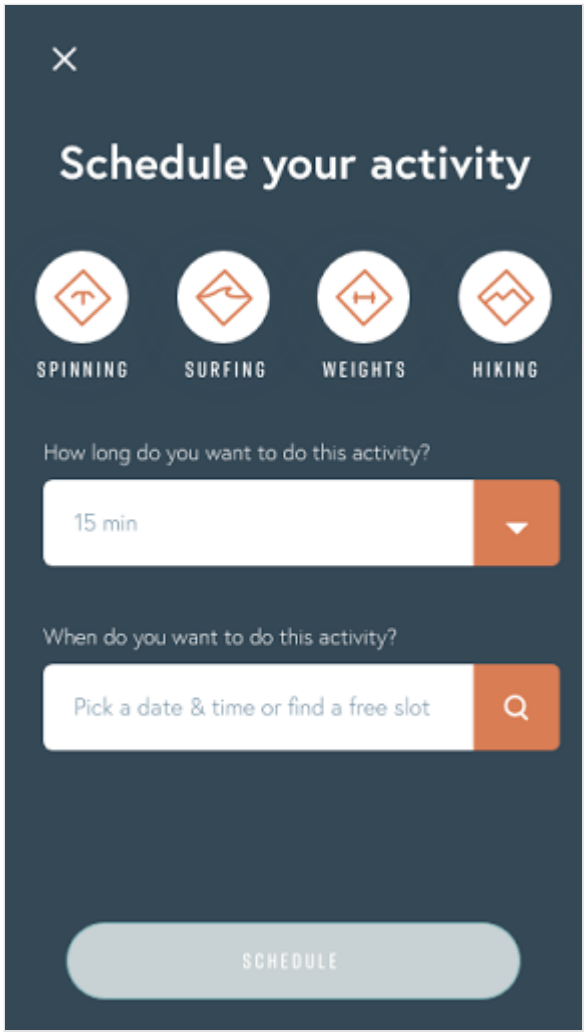


Design Specs:

https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/screen/300f78b4-2b9d-4c06-ae78-ad376bf9de05/Activity-Schedule/

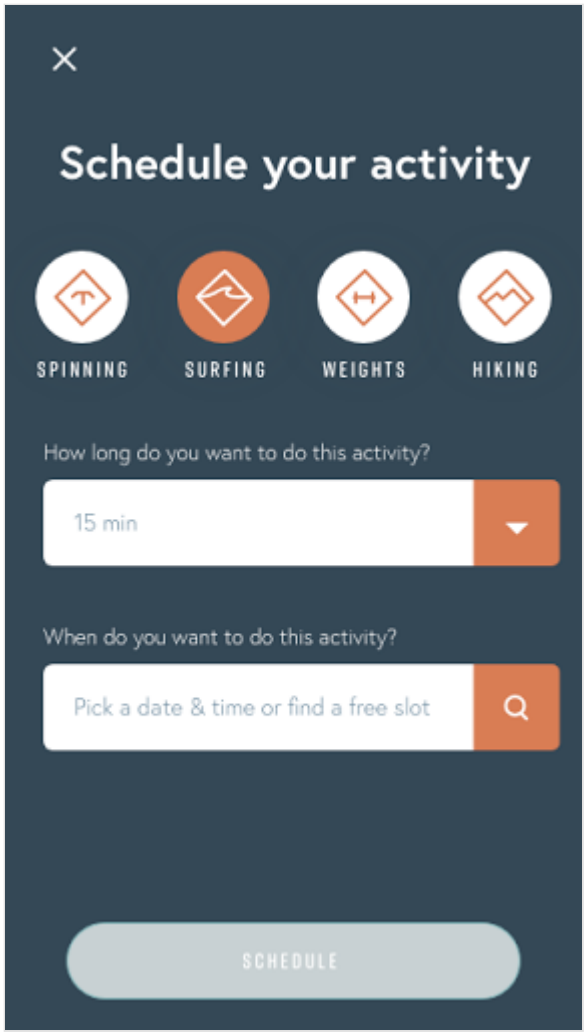
US03: Scheduling Activity [20 points]

As a user when I am on the Home screenand I click the **SCHEDULE ACTIVITY** button a full-screen modal window shows up where I can schedule the details of an activity I want to do in the future. In the initial stage the **Schedule** button is disabled which is indicated by a lighter background color of the button.

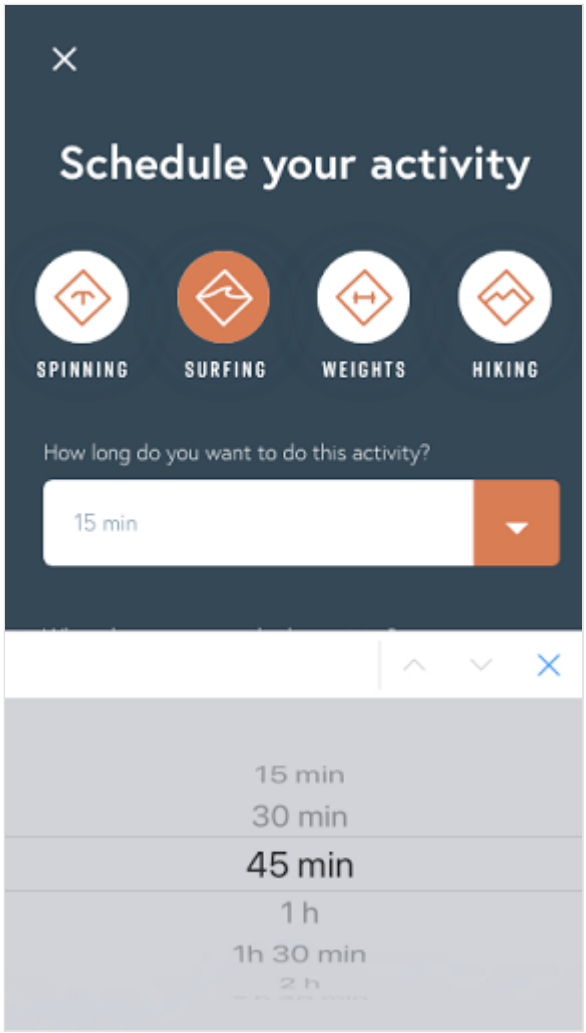


On this screen I can select one of the four activities by tapping on the icons they represent, which are labelled:

- Spinning
- Surfing
- Weights
- Hiking
  Tapping an icon highlights which of the activites has been selected by changing the background color and icon color. See:

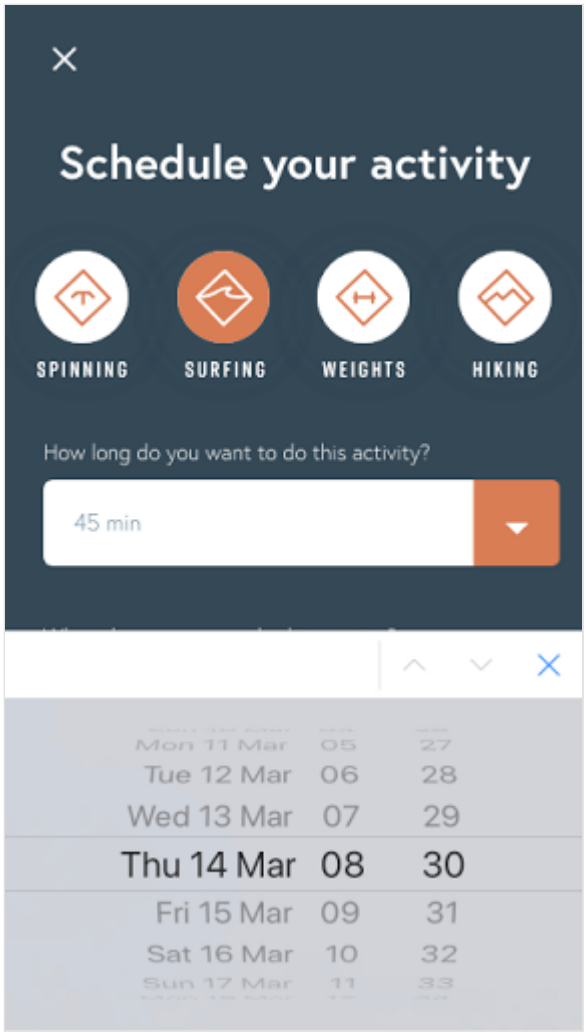


Then I need to pick for how long I want to do this activity. I can tap either on the white field or the dropdown indicator on the right. By default **15 min** is the selected duration. Opening the dropdown should use the native dropdown of a platform. For example on iOS it will show like this:



After I picked the duration I can pick when I want to do this activity. For this I have two options:

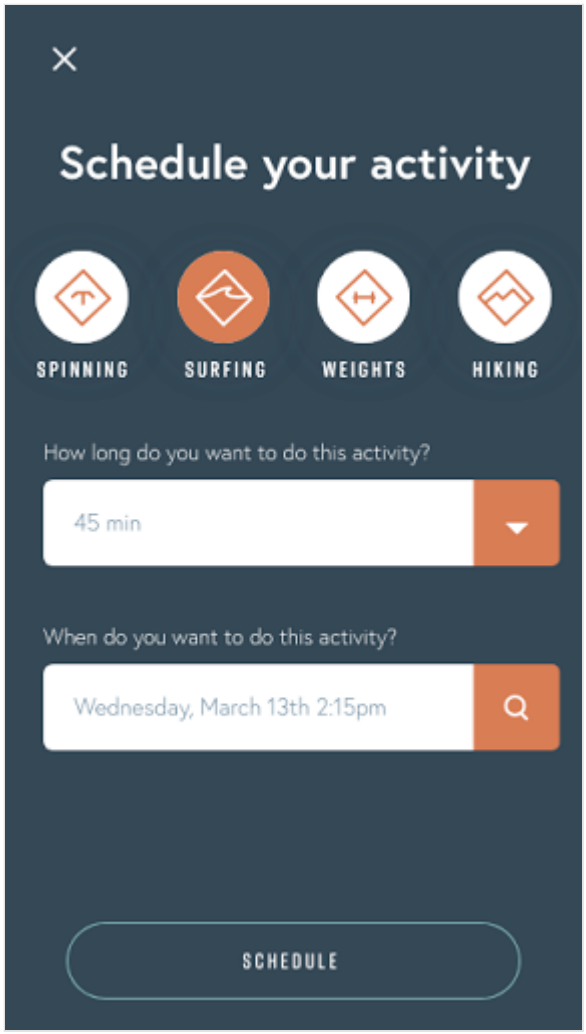
1. If I tap on **Pick a date & time or find a free slot** labelled field I will see a default date/time picker for the specific platform, where I can specify the right date. For iOS it will look like this:



When I select the right date it will be set on the field in the following format: **Wednesday, March 13th 2:15pm**.

2. If I tap the search icon, the app is supposed to propose free slots as described in the specific use case **US04: Scheduling Activity – finding free slots**.

When all fields have been selected then the **SCHEDULE** button changes state to active. See:



When I click this button the scheduled activity should be stored on the server side and I will be navigated to the home screen showing my scheduled activities as described in US03: Home Screen - scheduled activities**US03: Home Screen - scheduled activities**.

Design Specs:

Schedule Activity modal: https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/screen/d9eae2e2-a53f-462e-80ca-6c43e1cac949/Schedule-Activity/
Highlighted icon: https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/screen/b3543a17-50de-473a-99dd-5ae2eb8c01c3/Schedule-Activity-1/
Active Schedule button: https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/screen/c7cdfe00-10a9-4654-9b46-e46392f25e77/Schedule-Activity-Duration-Filled-1/

US04: Scheduling Activity – finding free slots [30 points]

As a user when I clicked the search icon on **Schedule Activity** modal view, the app should generate a list of available slots when I can perform the activity at the selected duration considering my schedule.
The app should propose time slots for the next seven days starting the day after today. We do not want activities scheduled during night time which is defined as 22:00 – 8:00 (or 10pm – 8am). The algorithm should pick the first available time and proposed time slots should not overlap. No overlap means, that if there is an activity with 30 min duration and we have availability in the schedule between 8:00 and 8:45 we only want to show 8:00 as the first time available, but not 8:15, cause it would overlap with the 8:00 – 8:30 proposed slot. In other words the next proposed slot must not start before the end of the previous proposed slot.
The time slots should be presented to me in the following format: **Wednesday, March 13th 2:15pm**

Example

Consider the following example schedule:


```
{
  "Monday": [
    {
      "Start": "8:30",
      "End": "11:00"
    },
    {
      "Start": "12:00",
      "End": "15:00"
    },
    {
      "Start": "16:10",
      "End": "18:00"
    },
    {
      "Start": "20:00",
      "End": "22:00"
    }
  ],
  "Tuesday": [
    {
      "Start": "8:00",
      "End": "11:00"
    },
    {
      "Start": "12:30",
```

```
      "End": "17:30"
    },
    {
      "Start": "19:00",
      "End": "21:00"
    }
  ],
  "Wednesday": [
    {
      "Start": "10:00",
      "End": "13:00"
    },
    {
      "Start": "15:00",
      "End": "20:00"
    }
  ],
  "Thursday": [
    {
      "Start": "9:00",
      "End": "15:00"
    },
    {
      "Start": "17:00",
      "End": "20:00"
    },
    {
      "Start": "21:10",
      "End": "22:00"
    }
  ],
  "Friday": [
    {
      "Start": "8:00",
      "End": "11:00"
    },
    {
      "Start": "12:30",
      "End": "17:30"
    }
  ]
}
```

> This schedule should be hardcoded in the app in JSON format, so that we can alter the schedule and make sure the algorithm works properly when testing the solution.

Assuming the day is Tuesday, then we're looking into possible slots from Wednesday this week till Tuesday next week.
If the activity duration is 1.5h (90 min), then the following time slots for the activity should be proposed:

- Wednesday 8:00
- Wednesday 13:00
- Wednesday 20:00
- Thursday 15:00
- Friday 11:00
- Friday 17:30
- Friday 19:00
- Friday 20:30
- Monday 18:00 (next week)
- Tuesday 11:00 (next week)
- Tuesday 17:30 (next week)

Design Spec:

https://xd.adobe.com/spec/c6e728d7-1ab8-45be-6d57-e40c22a028d7-b0d6/screen/c7cdfe00-10a9-4654-9b46-e46392f25e77/Schedule-Activity-Duration-Filled-1/

Non-functional requirements

UI Library/Framework

You should use React as the UI library, other than that you are free to use any state management you feel comfortable with and that will display your skills. Most companies using the platform use Redux in their stack.

API [15 points]

To implement the API used for storage and retrieval of activity history and scheduled activites you can use any kind of serverless service or server mock. Here are some options:

1. JSON Server
2. Google Cloud Firestore
3. Back4App
4. Backendless
5. Hoodie
6. etc.
   The score for this requirement will not by affected by choice of a particular service/mock, but by proper use of patterns for interacting with the back-end, separation of concerns, state management, etc.

Testing [10 points]

If you create a single test case for one of your components you will be awarded additional 10 points.

PWA [5 points]

If you implement the page as a Progressive Web App you will be awarded additional 5 points.

README

Your solution must come with a README.md file, that explains how to run and test the solution – including all prerequisities that need to be installed.

Submitting Results

Please submit your result using one of:

1. A private github repository
2. A private bitbucket repository

3. E-mail to challenge@nixa.io
   The users you should share repository access to are provided below:

Github user names

1. `maksymilian-majer`
2. `kielon`

Bitbucket user names

1. `maksymilianmajer`
2. `MaciejAdamczyk`

Final notes

Usage disclaimer

The solution you share with nixa.io team isn't going to be used commercially. It's only purpose is to demonstrate and validate your skills. If you pass the full assessment process, then your code and our evaluation of your solution will be displayed as part of your interactive profile on the nixa.io platform.

Pilot feedback

Please bear in mind, that we're in the pilot phase. If you worked in an agile team you know it's important to release fast and often, so please bear with any imperfections of this task and report your questions, concerns and feedback to challange@nixa.io. After all 1.0 Is the Loneliest Number 😉