IVÁN
YEPES ARROYO

Frontend Developer

Bringing Ideas into products

Recent Projects Summary

Pixelcrush Watermarker

January 2018 - March 2018

One of our clients asked us if we could develop an app for its Shopify store to create watermarks on demand and automate their creation each time a product is created or added images to it.

As we had an own product which is in charge of manipulating images in real-time (pixelcrush.io), we were very familiar with all the processes involved so the client knew in advance we were a good option for this job.

As the image manipulation in real-time was already built into pixelcrush.io (adding parameters to a URL using your account's CDN endpoint), this was an almost-pure frontend app, but at the end, I had to deal with some backend stuff I enjoyed a lot while developing it.

Once the user installs the app in his Shopify store he can play with a watermark editor which allows using its shop logo, any other logo or a text watermark, and set its size, placement, and opacity.

These were saved as 'Templates' which were used later to automate the processes.

The user could load a product and manually apply one of the saved templates, or go to the automation section to define a template and leave the app to watermark all actual or new images.

Any watermarked image were previously backed and available to be reverted to its original image at any time, even revert all images from all products in a single click.

As doing backups and reverting them is an expensive process, I had to create a worker on top of Redis to manage the jobs who upload/download the images so all these expensive operations are asynchronous.

Apart from the workers and the middleware API that manages the communication between the app and Shopify, I created the full app from scratch, using 'Shopify Polaris' component library as its almost a mandatory requirement to build apps on their platform to have the same look and feel.

Tech Stack

Frontend

ReactJS
Redux
Webpack
Shopify Polaris
Bull Queue Manager

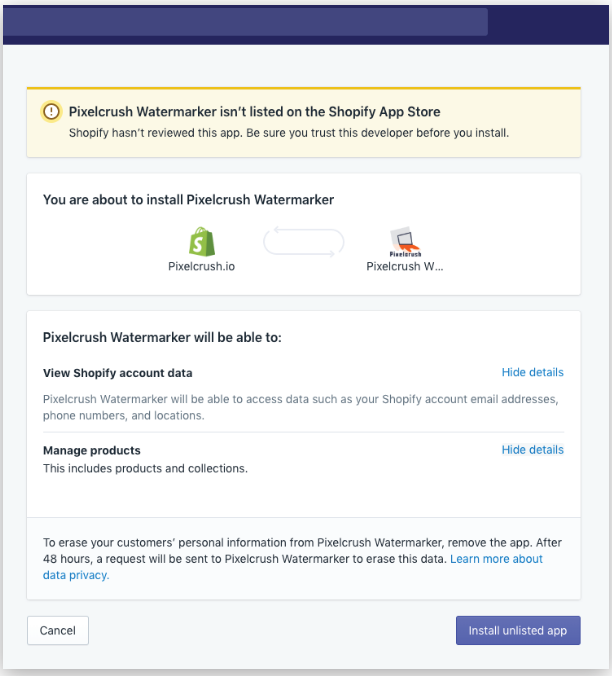
Backend

NodeJS (Express)
Shopify API
Redis
MySQL
JSON Web tokens

Pixelcrush Watermarker

Screenshots

Installing App on Shopify Store



Pixelcrush Watermarker isn't listed on the Shopify App Store
Shopify hasn't reviewed this app. Be sure you trust this developer before you install.

You are about to install Pixelcrush Watermarker

Pixelcrush Watermarker will be able to:

View Shopify account data — Hide details

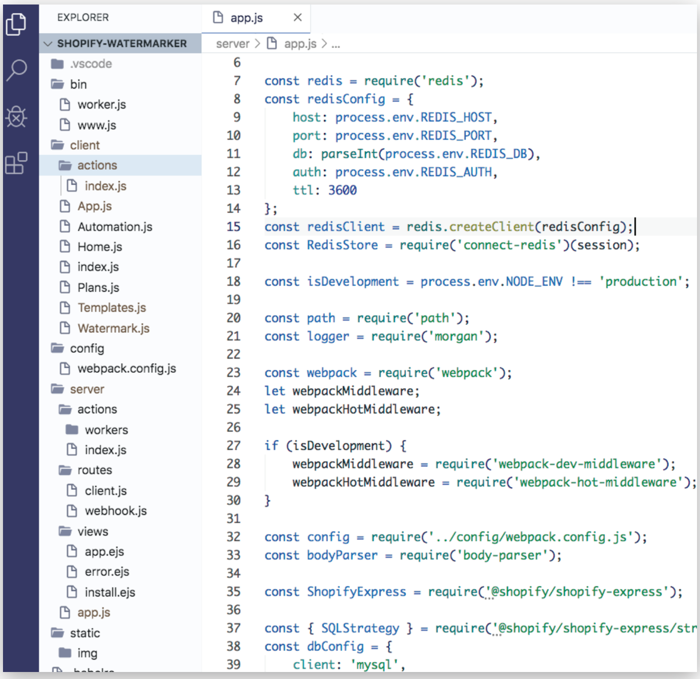
Pixelcrush Watermarker will be able to access data such as your Shopify account email addresses, phone numbers, and locations.

Manage products — Hide details

This includes products and collections.

To erase your customers' personal information from Pixelcrush Watermarker, remove the app. After 48 hours, a request will be sent to Pixelcrush Watermarker to erase this data. Learn more about data privacy.

Project Structure / Code Sample

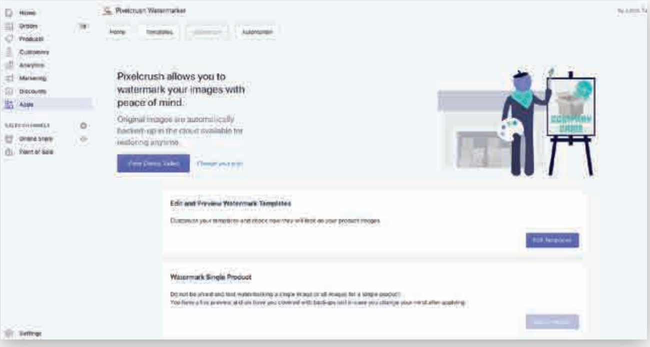


```
EXPLORER
SHOPIFY-WATERMARKER
  .vscode
  bin
    worker.js
    www.js
  client
    actions
      index.js
    App.js
    Automation.js
    Home.js
    index.js
    Plans.js
    Templates.js
    Watermark.js
  config
    webpack.config.js
  server
    actions
      workers
      index.js
    routes
      client.js
      webhook.js
    views
      app.ejs
      error.ejs
      install.ejs
    app.js
  static
    img
```

app.js — server > app.js > ...

```
const redis = require('redis');
const redisConfig = {
    host: process.env.REDIS_HOST,
    port: process.env.REDIS_PORT,
    db: parseInt(process.env.REDIS_DB),
    auth: process.env.REDIS_AUTH,
    ttl: 3600
};
const redisClient = redis.createClient(redisConfig);
const RedisStore = require('connect-redis')(session);

const isDevelopment = process.env.NODE_ENV !== 'production';

const path = require('path');
const logger = require('morgan');

const webpack = require('webpack');
let webpackMiddleware;
let webpackHotMiddleware;

if (isDevelopment) {
    webpackMiddleware = require('webpack-dev-middleware');
    webpackHotMiddleware = require('webpack-hot-middleware');
}

const config = require('../config/webpack.config.js');
const bodyParser = require('body-parser');

const ShopifyExpress = require('@shopify/shopify-express');

const { SQLStrategy } = require('@shopify/shopify-express/str
const dbConfig = {
    client: 'mysql',
```

App Homepage



Pixelcrush allows you to watermark your images with peace of mind.

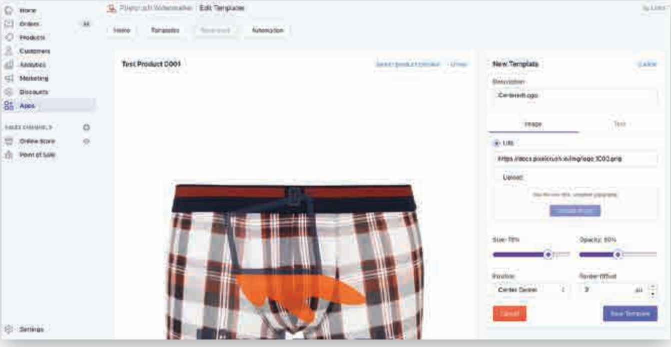
Original images are automatically backed-up in the cloud available for restoring anytime.

Edit and Preview Watermark Templates

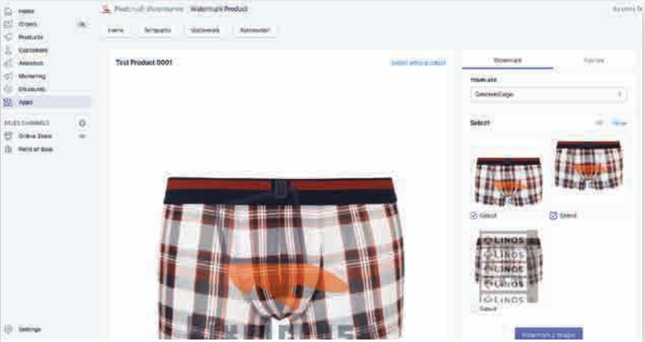
Customize your templates and check how they will look on your product images

Watermark Single Product

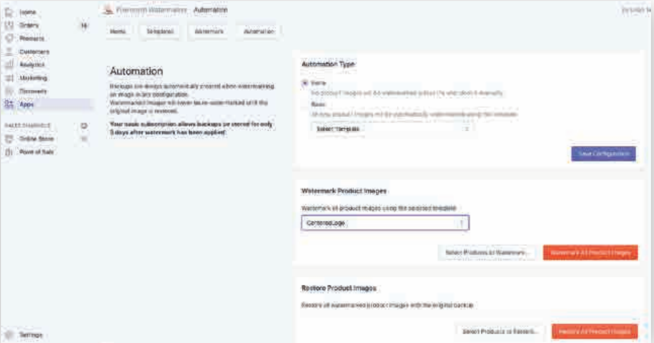
Templates



Manual Watermarking



Automation



Automation

Your basic subscription allows backups be stored for only 5 days after watermark has been applied!

Automation Type

Select Template

Watermark Product Images

Watermark all product images using the selected template

CenteredLogo

Restore Product Images

Restore all watermarked product images with the original backup

Letiquet

October 2018 – August 2019

An entrepreneur reached us to build an Instagram-like app with extra functionality to be able to manage a personal closet, which will be filled into the app scanning barcodes / QR codes from physical shop tickets or clothes' inner labels.

We quickly started to define the requirements of the app and I created an initial mockup in Adobe XD, then after agreeing with the client the final needed functionalities for the MVP, I started creating the React- Native app using 'Expo' app bootstrapping, as I wouldn't need any native module to get into the final stage of the project, and it makes the development experience more convenient and the client can get the signs of progress of the app almost in real-time because there's no need to ship .apks or upload the app to TestFlight or Play Store Beta program.

I built the entire app to React Native without the need of using any native modules and help on creating some of the functionalities within the API (NodeJs).

There aren’t quite many options of styling apps/components in React Native as there are in ReactJS, and I didn’t felt the ones available were good enough for bringing the app a consistent and fast styling method, so I created a custom 'framework' with a bootstrap-like syntax to quickly compose components instead defining styles individually on the own component file. This dramatically improved development time and design consistency across the app and OSes (iOS / Android).

Tech Stack

Frontend

React Native
Expo
Metro Bundler
Custom Styling Framework
IcoMoon
JSON Web-Tokens
QR/Barcode Generators
Adobe XD (Interactive Mockup)

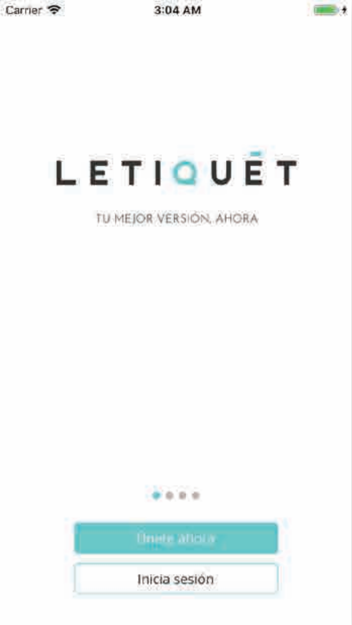
Backend

NodeJS API
Google Cloud Storage

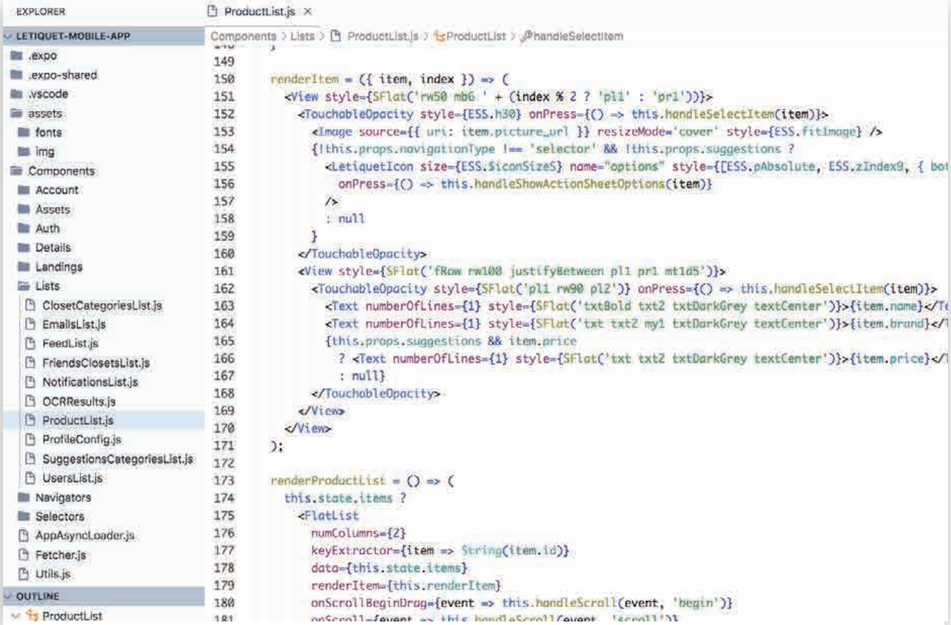
Letiquet

Screenshots

Login / Register



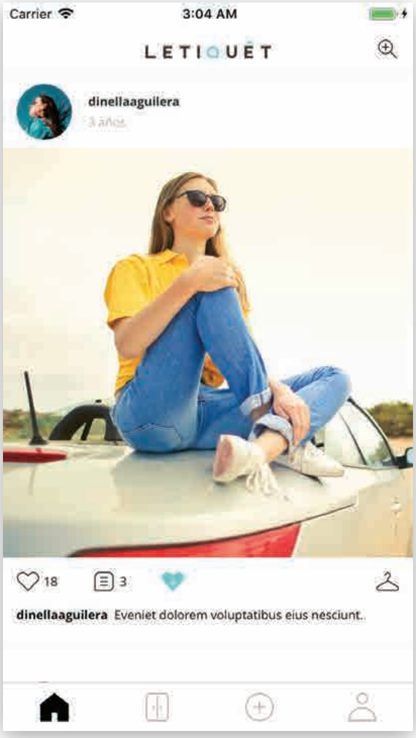
Project Structure / Code Sample


EXPLORER
ProductList.js
Components > Lists > ProductList.js > ProductList > handleSelectItem

```
LETIQUET-MOBILE-APP
  .expo
  .expo-shared
  .vscode
  assets
    fonts
    img
  Components
    Account
    Assets
    Auth
    Details
    Landings
    Lists
      ClosetCategoriesList.js
      EmailsList.js
      FeedList.js
      FriendsClosetsList.js
      NotificationsList.js
      OCRResults.js
      ProductList.js
      ProfileConfig.js
      SuggestionsCategoriesList.js
      UsersList.js
    Navigators
    Selectors
    AppAsyncLoader.js
    Fetcher.js
    Utils.js
OUTLINE
  ProductList
```

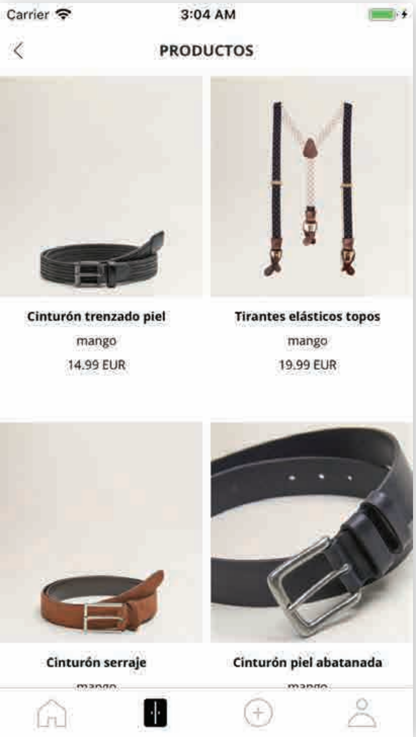
```
  }

  renderItem = ({ item, index }) => (
    <View style={SFlat('rw50 mb6 ' + (index % 2 ? 'pl1' : 'pr1'))}>
      <TouchableOpacity style={ESS.h30} onPress={() => this.handleSelectItem(item)}>
        <Image source={{ uri: item.picture_url }} resizeMode='cover' style={ESS.fitImage} />
        {!this.props.navigationType !== 'selector' && !this.props.suggestions ?
          <LetiquetIcon size={ESS.$iconSizeS} name="options" style={[ESS.pAbsolute, ESS.zIndex9, { bot
            onPress={() => this.handleShowActionSheetOptions(item)}
          />
          : null
        }
      </TouchableOpacity>
      <View style={SFlat('fRow rw100 justifyBetween pl1 pr1 mt1d5')}>
        <TouchableOpacity style={SFlat('pl1 rw90 pl2')} onPress={() => this.handleSelectItem(item)}>
          <Text numberOfLines={1} style={SFlat('txtBold txt2 txtDarkGrey textCenter')}>{item.name}</Te
          <Text numberOfLines={1} style={SFlat('txt txt2 my1 txtDarkGrey textCenter')}>{item.brand}</T
          {this.props.suggestions && item.price
            ? <Text numberOfLines={1} style={SFlat('txt txt2 txtDarkGrey textCenter')}>{item.price}</T
            : null}
        </TouchableOpacity>
      </View>
    </View>
  );

  renderProductList = () => (
    this.state.items ?
      <FlatList
        numColumns={2}
        keyExtractor={item => String(item.id)}
        data={this.state.items}
        renderItem={this.renderItem}
        onScrollBeginDrag={event => this.handleScroll(event, 'begin')}
        onScroll={event => this.handleScroll(event, 'scroll')}
```

Feed



dinellaaguilera Eveniet dolorem voluptatibus eius nesciunt.

Closet Products



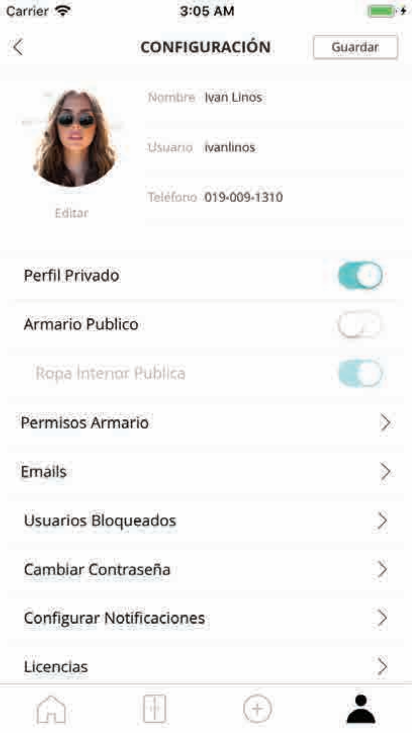
Cinturón trenzado piel
mango
14.99 EUR

Tirantes elásticos topos
mango
19.99 EUR

Cinturón serraje
mango

Cinturón piel abatanada
mango

Account



Nombre Ivan Linos
Usuario ivanlinos
Teléfono 019-009-1310

Perfil Privado ●
Armario Publico ○
Ropa Interior Publica ●
Permisos Armario
Emails
Usuarios Bloqueados
Cambiar Contraseña
Configurar Notificaciones
Licencias

AirSensei

May 2019 – August 2019

This client sells and manages IOT devices that control air pollution and some other metrics, so they had to have a place where they can control where those devices are installed, which are their metrics in real-time and their historic data.
After listening to their requirements, I created the initial mockup and helped the backend team to create the data model.

They created the API in python and I started to create the app, I had a couple of remote juniors developers I assigned some tasks while I was creating the core of the app.
As I had to manage a lot of forms and tables, I decided to create a custom 'framework' to have an easy way to both show the data and manage data submit/editing form and tables the easiest and most reliant way.

Extending JSON-schema specification and after defining the fields which their validation rules for a given entity, we used Table component to show the data of this JSON-schema, same for a submit/edit form, which both of them are attached to a global store and API which automatically retrieves/post the data to the relevant endpoints (also defined in the custom JSON-schema) so at the end, form handling is minimal and everything follows a chain event, this is the key of this project apart of the OpenStreetMap integration to allow showing markers and popup-slider for showing locations/devices data.

Tech Stack

Frontend

ReactJS
React Context API
Webpack
Material UI + Reactstrap
JSON-Schema
Open Street Map / Leaflet
i18n (Translation system)
Adobe XD (Interactive Mockup)

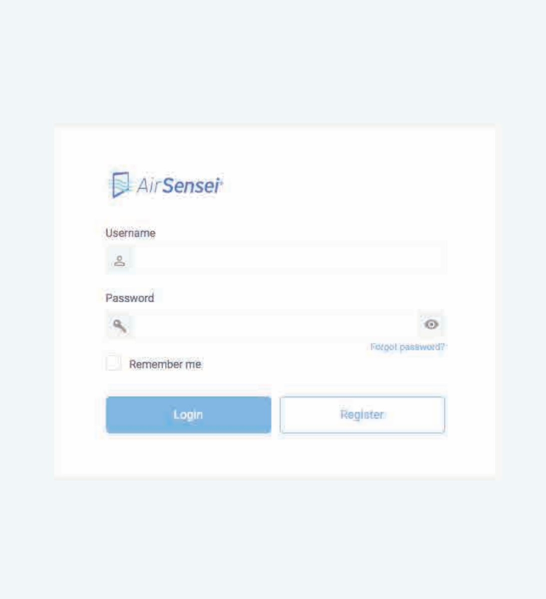
Backend

Python API (Developed by Backend Team)
Google Cloud Storage

AirSensei

Screenshots

Login / Register



AirSensei

Username

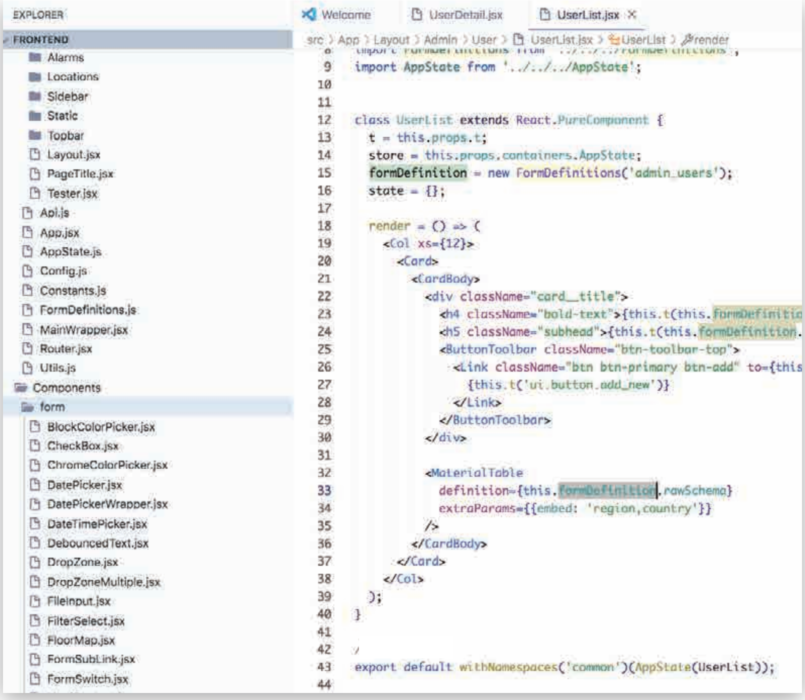
Password

Forgot password?

□ Remember me

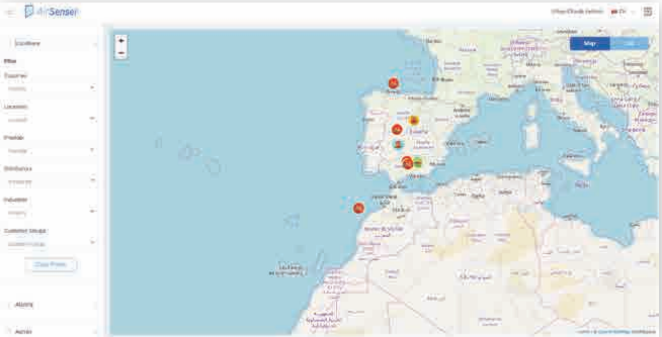
Login | Register

Project Structure / Code Sample

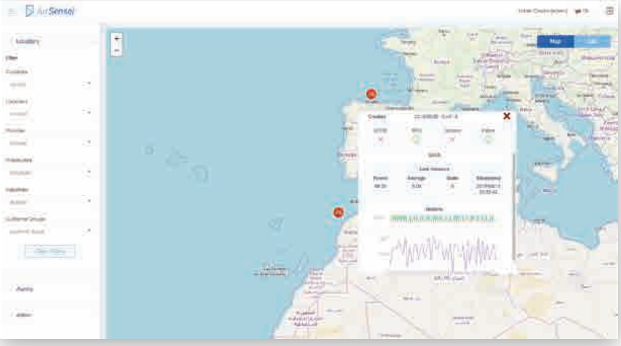


```
import FormDefinitions from '../../../FormDefinitions';
import AppState from '../../../AppState';


class UserList extends React.PureComponent {
  t = this.props.t;
  store = this.props.containers.AppState;
  formDefinition = new FormDefinitions('admin_users');
  state = {};

  render = () => (
    <Col xs={12}>
      <Card>
        <CardBody>
          <div className="card__title">
            <h4 className="bold-text">{this.t(this.formDefinitio
            <h5 className="subhead">{this.t(this.formDefinition.
            <ButtonToolbar className="btn-toolbar-top">
              <Link className="btn btn-primary btn-add" to={this
                {this.t('ui.button.add_new')}
              </Link>
            </ButtonToolbar>
          </div>

          <MaterialTable
            definition={this.formDefinition.rawSchema}
            extraParams={{embed: 'region,country'}}
          />
        </CardBody>
      </Card>
    </Col>
  );
}

/
export default withNamespaces('common')(AppState(UserList));
```

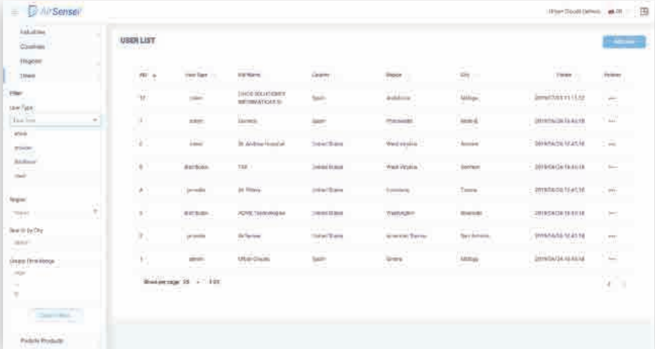
Devices on Map



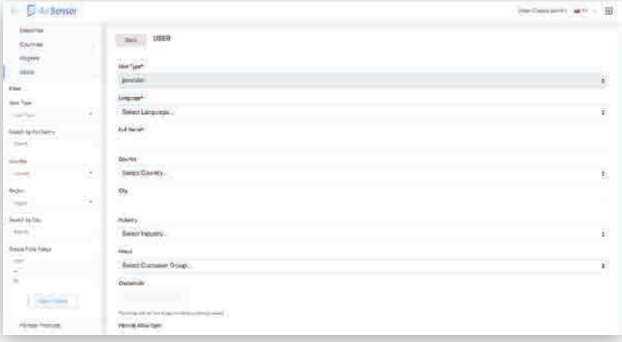
Device Real-Time Data



Listing



Form



Semana Santa AR

March 2019 – June 2019

Partnering with Telefonica (biggest telecom company in Spain), they wanted to create an Augmented Reality app to create a proof of concept integrating AR the technology in their next-gen commercial products.

The app shows you information and locations where the user should go to capture with the phone the original location a video was recorded. Then they can see the throughscreen and the rear camera an augmented version of the real-live view with the enhanced alpha-video of relevant events that happened at that exact point weeks ago and were recorded.

I was in charge of creating the initial mockups and build the React Native app from scratch. This app is a pure React Native app as I needed to integrate on it the 'ARView' native module a colleague created using Java and Objective-C then I only needed to import it to create the actual augmented view experience in the final location detail screen of the APP navigation flow.

Tech Stack

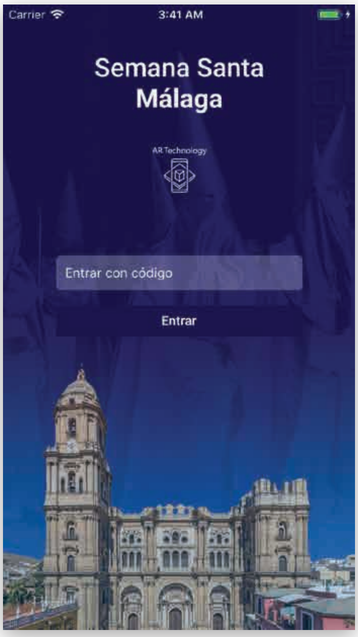
Frontend

React Native
Metro Bundler
SQLite
IcoMoon
Google Maps
Apple Maps
Jest

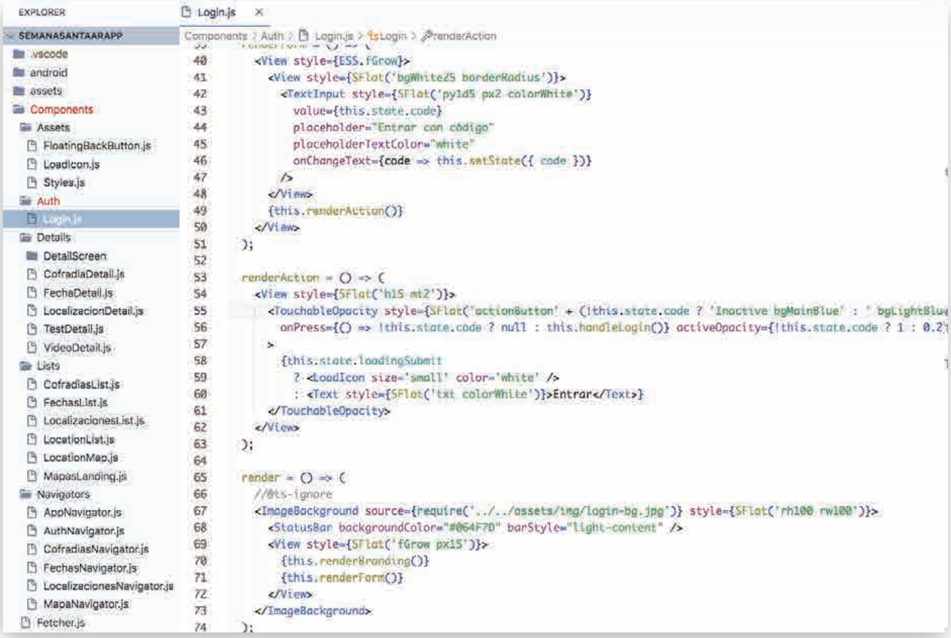
Semana Santa AR

Screenshots

Login

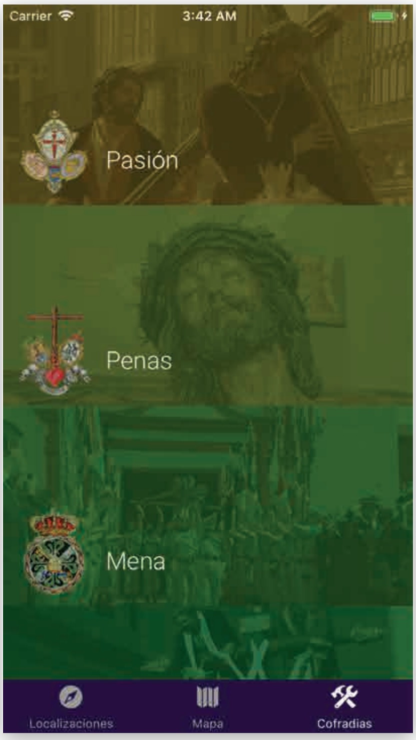


Project Structure / Code Sample

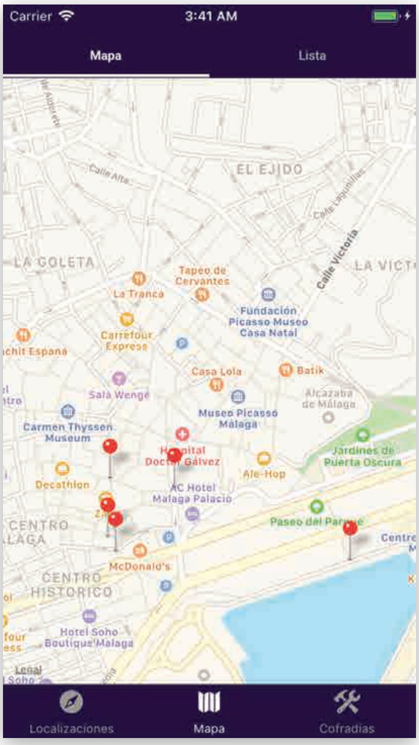


```
    <View style={ESS.fGrow}>
      <View style={SFlat('bgWhite25 borderRadius')}>
        <TextInput style={SFlat('py1d5 px2 colorWhite')}
          value={this.state.code}
          placeholder="Entrar con código"
          placeholderTextColor="white"
          onChangeText={code => this.setState({ code })}
        />
      </View>
      {this.renderAction()}
    </View>
  );

  renderAction = () => (
    <View style={SFlat('h15 mt2')}>
      <TouchableOpacity style={SFlat('actionButton' + (!this.state.code ? 'Inactive bgMainBlue' : ' bgLightBlue
        onPress={() => !this.state.code ? null : this.handleLogin()} activeOpacity={!this.state.code ? 1 : 0.2}
      >
        {this.state.loadingSubmit
          ? <LoadIcon size='small' color='white' />
          : <Text style={SFlat('txt colorWhite')}>Entrar</Text>}
      </TouchableOpacity>
    </View>
  );

  render = () => (
    //@ts-ignore
    <ImageBackground source={require('../../assets/img/login-bg.jpg')} style={SFlat('rh100 rw100')}>
      <StatusBar backgroundColor="#064F7D" barStyle="light-content" />
      <View style={SFlat('fGrow px15')}>
        {this.renderBranding()}
        {this.renderForm()}
      </View>
    </ImageBackground>
  );
```

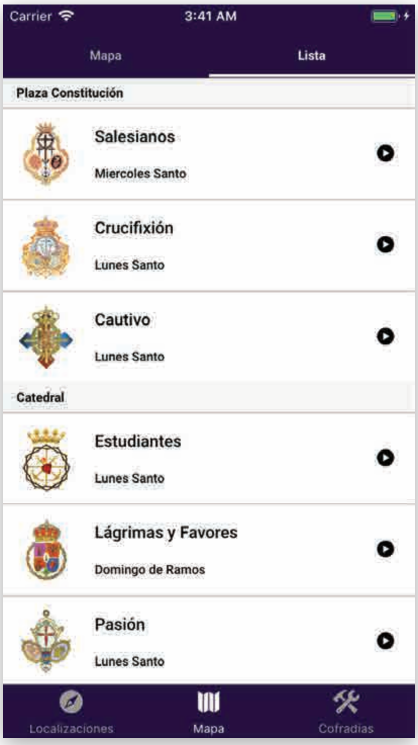
Directory



Map



Location List


Carrier 3:41 AM

Mapa | Lista

Plaza Constitución

- Salesianos — Miercoles Santo
- Crucifixión — Lunes Santo
- Cautivo — Lunes Santo

Catedral

- Estudiantes — Lunes Santo
- Lágrimas y Favores — Domingo de Ramos
- Pasión — Lunes Santo

Localizaciones | Mapa | Cofradias

WeDoing

September 2019 – December 2019

I left Linos in October 2019 as I felt I needed to take a step further in my career and create a product I had been thinking about for a couple of years.

So I created WeDoing, a digital platform to solve a modern problem: connect like-minded people with common interests who don’t have anyone for going or doing that thing with.
For instance: going to a concert of hip-hop music that you like, but your friends don’t.

I also used this project as an opportunity to consolidate some technologies I had used but wanted really digging into them and apply all of them into a single project fully managed by myself, from concept to app store launch, including all things backend, testing, deploying and of course, the app itself.

I have been lucky enough to find an early investor who bought me the entire platform in an early stage (but fully developed) last December, hence I am ready for a new challenge (and if you are reading this, is because I'm applying for a position of my interest).
Because in the end, what I enjoy the most is building things, (I'm a developer at heart) and WeDoing was already done in a development point of view and it’s in a marketing/growth stage now.

Tech Stack

Frontend

React
React Native
MobX
Bing Maps
Enzime

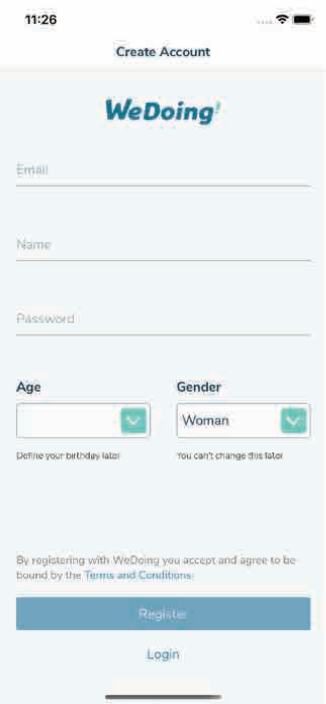
Backend

NodeJS
Parse Platform
Serverless
MongoDB
API Scrapping

WeDoing

Screenshots

Register



11:26

Create Account

WeDoing

Email

Name

Password

Age

Gender

Woman

Define your birthday later

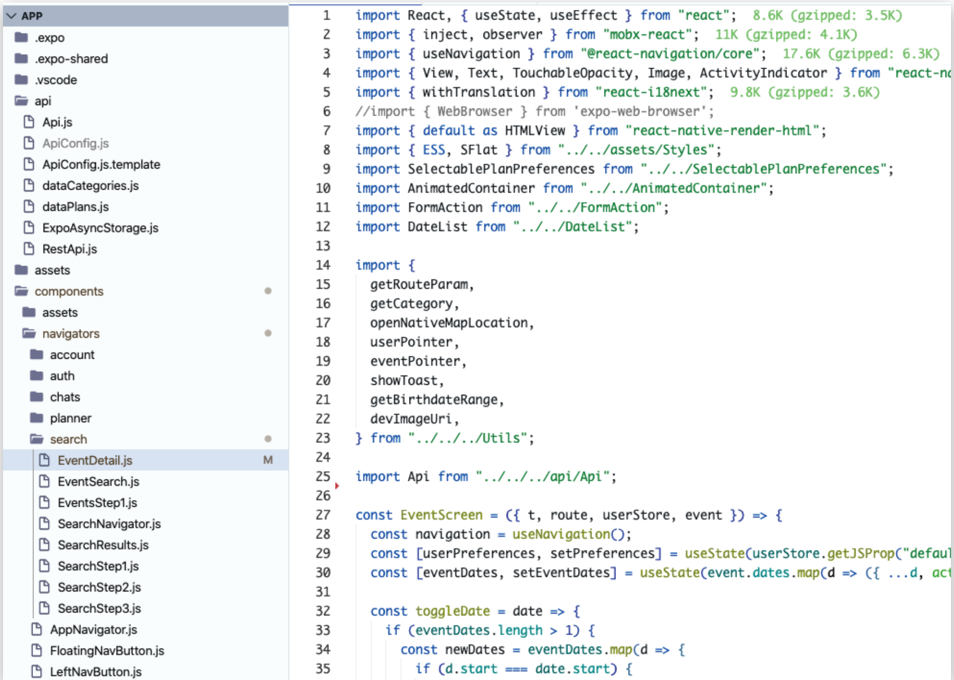
You can't change this later

By registering with WeDoing you accept and agree to be bound by the Terms and Conditions.

Register

Login

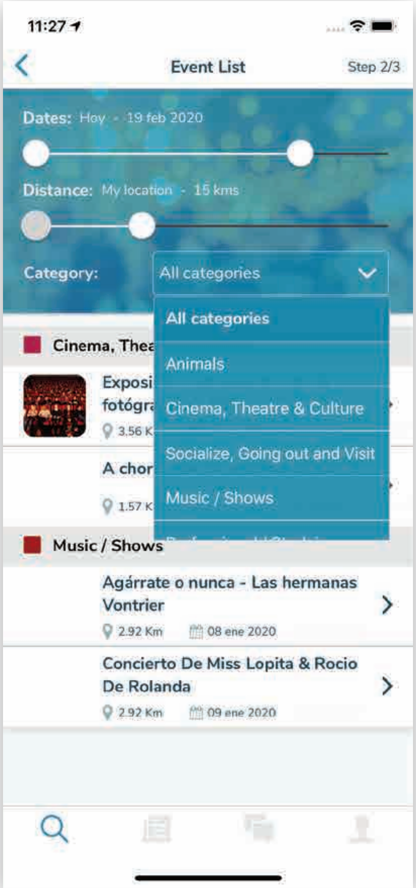
Project Structure / Code Sample



```
APP
  .expo
  .expo-shared
  .vscode
  api
    Api.js
    ApiConfig.js
    ApiConfig.js.template
    dataCategories.js
    dataPlans.js
    ExpoAsyncStorage.js
    RestApi.js
  assets
  components
    assets
    navigators
      account
      auth
      chats
      planner
      search
        EventDetail.js        M
        EventSearch.js
        EventsStep1.js
        SearchNavigator.js
        SearchResults.js
        SearchStep1.js
        SearchStep2.js
        SearchStep3.js
      AppNavigator.js
      FloatingNavButton.js
      LeftNavButton.js
```

```
import React, { useState, useEffect } from "react";  8.6K (gzipped: 3.5K)
import { inject, observer } from "mobx-react";  11K (gzipped: 4.1K)
import { useNavigation } from "@react-navigation/core";  17.6K (gzipped: 6.3K)
import { View, Text, TouchableOpacity, Image, ActivityIndicator } from "react-na
import { withTranslation } from "react-i18next";  9.8K (gzipped: 3.6K)
//import { WebBrowser } from 'expo-web-browser';
import { default as HTMLView } from "react-native-render-html";
import { ESS, SFlat } from "../../assets/Styles";
import SelectablePlanPreferences from "../../SelectablePlanPreferences";
import AnimatedContainer from "../../AnimatedContainer";
import FormAction from "../../FormAction";
import DateList from "../../DateList";

import {
  getRouteParam,
  getCategory,
  openNativeMapLocation,
  userPointer,
  eventPointer,
  showToast,
  getBirthdateRange,
  devImageUri,
} from "../../../Utils";

import Api from "../../../api/Api";

const EventScreen = ({ t, route, userStore, event }) => {
  const navigation = useNavigation();
  const [userPreferences, setPreferences] = useState(userStore.getJSProp("defaul
  const [eventDates, setEventDates] = useState(event.dates.map(d => ({ ...d, act

  const toggleDate = date => {
    if (eventDates.length > 1) {
      const newDates = eventDates.map(d => {
        if (d.start === date.start) {
```

Search Event



11:27

Event List

Step 2/3

Dates: Hoy - 19 feb 2020

Distance: My location - 15 kms

Category: All categories

- All categories
- Animals
- Cinema, Theatre & Culture
- Socialize, Going out and Visit
- Music / Shows

Cinema, Thea

Exposi fotógra

3.56 K

A chor

1.57 K

Music / Shows

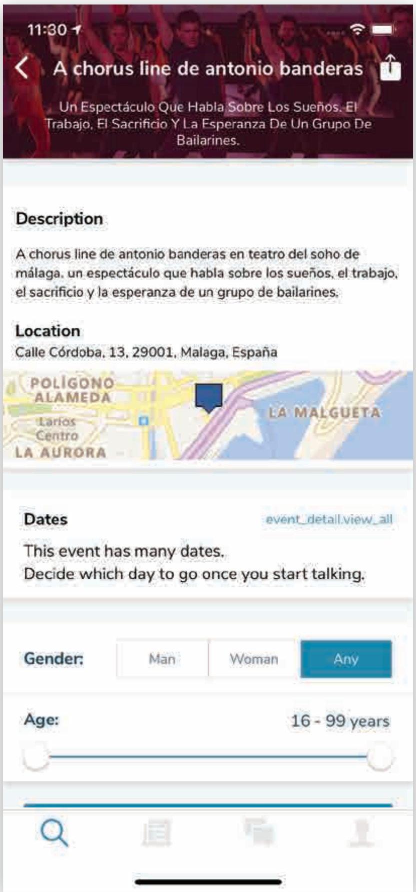
Agárrate o nunca - Las hermanas Vontrier

2.92 Km 08 ene 2020

Concierto De Miss Lopita & Rocio De Rolanda

2.92 Km 09 ene 2020

Event Detail



11:30

A chorus line de antonio banderas

Un Espectáculo Que Habla Sobre Los Sueños, El Trabajo, El Sacrificio Y La Esperanza De Un Grupo De Bailarines.

Description

A chorus line de antonio banderas en teatro del soho de málaga. un espectáculo que habla sobre los sueños, el trabajo, el sacrificio y la esperanza de un grupo de bailarines.

Location

Calle Córdoba, 13, 29001, Malaga, España

Dates

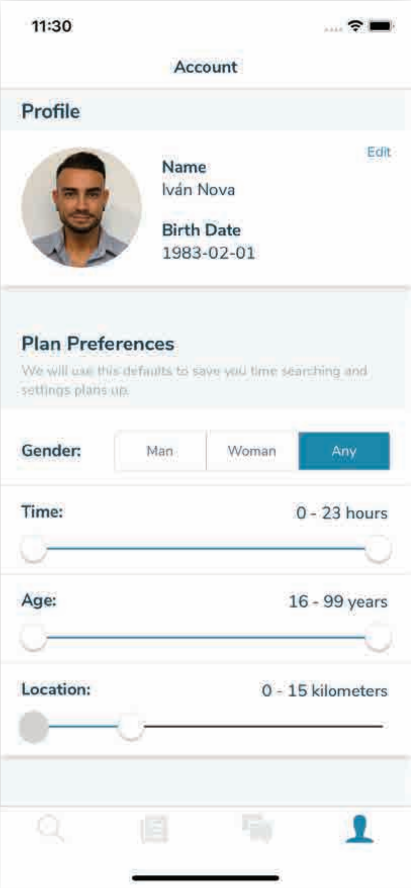
event_detail.view_all

This event has many dates.
Decide which day to go once you start talking.

Gender: Man | Woman | [✓ Any]

Age: 16 - 99 years

Account



11:30

Account

Profile

Edit

Name
Iván Nova

Birth Date
1983-02-01

Plan Preferences

We will use this defaults to save you time searching and settings plans up.

Gender: Man | Woman | [✓ Any]

Time: 0 - 23 hours

Age: 16 - 99 years

Location: 0 - 15 kilometers