

# **Interactive Media Design Prototyping**

**Jan Dočekal**

I took a previously “developed” idea of a **Moodboard generator** which I came up with in the class Ideas Generation, and developed only in the means of a simple html wireframe.

The main function was supposed to be pulling images and then allowing the user to select the images he likes and then export it to a PDF/JPEG/PNG.

## 4# Moodboard Creator

API'S : google image - search

- imgaqa smart cropping
- pinterest

bio : automagically create moodboards based on color, tag, theme, artists(?)

\* PDF  
(instant PDF)

## Mood Board Creator

#333333 #Chain, #Pe y more options



save

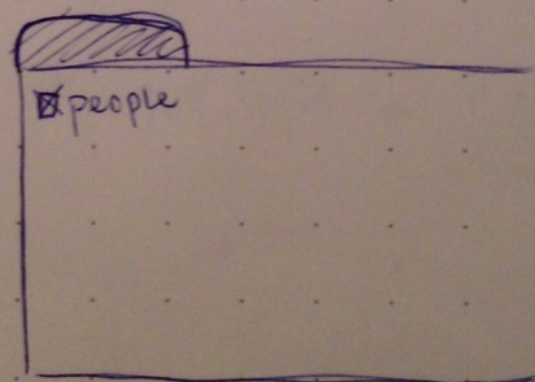
PDF file



replace

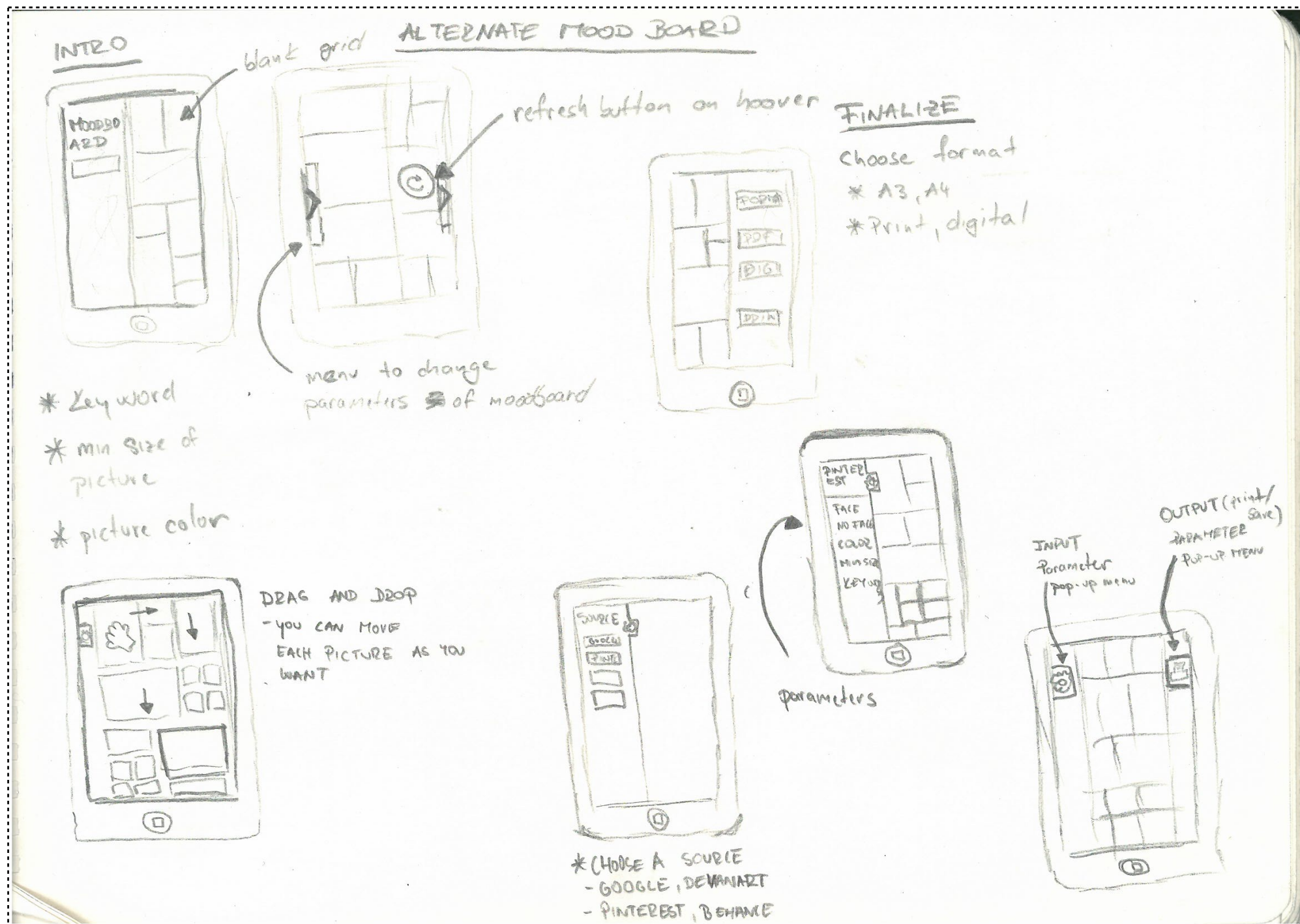
generate different image

additional API'S : Face-recognition.  
nudity recognition  
Flickr  
deviant art  
stumbleupon  
pinterest



**A moodboard generator sketched concept.**





**Scaling possibilities sketch (ipad app)**



#tag, @designer, keyword

DeviantART

Flickr

Pinterest

Google Images

Dribbble

Color

People



Small Medium High HD

Save me!

\*We ensure that nothing nasty  
(pomography, etc.) gets into  
your moodboard by using Nudity  
detection API.

The initial prototype of the full working website.

**Coding / How it works?**

**The “application” uses Flickr API to fetch it’s images.**

flickr.com/services/api/

**it does so by using the `getJSON()`; function, in jQuery**

```
$.getJSON(FinalURL, function(photos) {  
    var photo = photos.photos.photo;  
    console.log(photo);
```

**and the `each()`; function to create the images, and construct the URLs**

```
$.each(photo, function(i, item) {  
  
    $('#content').append('');  
    $('#' + item.id).load(function() {
```

**On every finished each() function I use the ISOTOPE plugin to handle the arrangement of images in the space.**

```
$('#content').isotope( 'reloadItems' ).isotope( { sortBy: 'original-order' } );
```

**And for storing the selected images I use an array coupled with the .push() function and \_.pull() from lodash.js**

```
$( "body" ).delegate( ".pic", "click", function() {  
    if ( $("#finish").is(":visible"))} else {  
        $(this).toggleClass("selected");  
        if ( $(this).is( ".selected" ) ) {  
            keepers.push($(this).attr('src'));  
        } else {  
            _.pull(keepers, $(this).attr('src'));  
        }  
        if ( $(".pic").is( ".selected" ) ) {  
            $("#info").fadeIn("fast");  
        } else {  
            $("#info").fadeOut("fast");  
        }  
    }  
});
```



I have several of those `getJSON` and each functions running in the code, in total of 271 code lines, to maintain infinite loop, additional loops, to maintain the saving of the array and transferring it to a “new page”.

To finally create the file out of the images I use a simple method. Print, which I struggled with the most, and I would have come up with some external saving solution if I had more time.

I use the *CSS Flexbox* to maintain the same position of elements for the print document *+ some other properties*.

```
.pic[style]{  
    flex: 1;  
}
```

```
#container[style]{  
    display: flex !important;  
    flex-direction: row !important;  
}
```

**This project is also publicly available at**

**<https://github.com/violetzie/Moodgen>**