Epicture

Introduction

Epicture is a school project where the goal is **to use and implement online photo sharing API platforms**.
We developed a mobile application using the **Imgur API** to display, manage and share images.
To build this app we work in a group composed of **Marlon l'Huillier** as the project lead and myself, **Laurent Coloma**.

note: Every Figure in this document can be found in the 'src/Documentation/Figures' folder.

Installation

To properly run this apps you will need:

- Node
- Gradle
- React-Native

Organization

First and foremost, before typing any lines of code, we had a meeting to setup the project guidelines.
During it, we looked at the **Android Mobile App** application and tested it to see how it was working and where to go.
The following Figures, will show you the baseline design of the app.

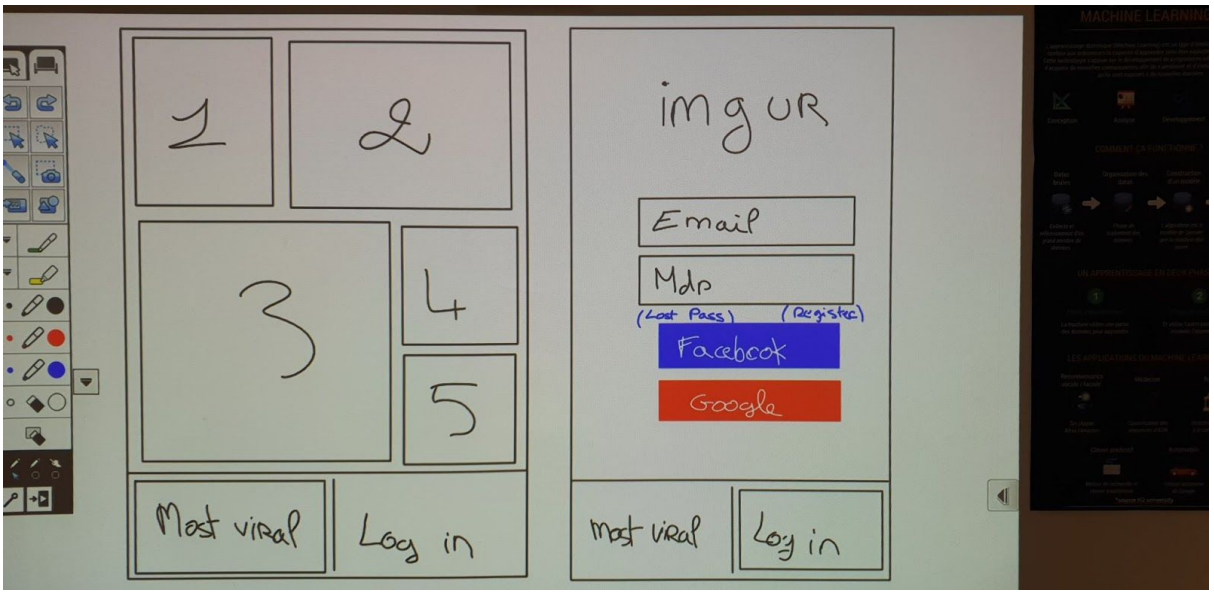

Figure 1
Un-Authed Home Page: Most Viral and Login View

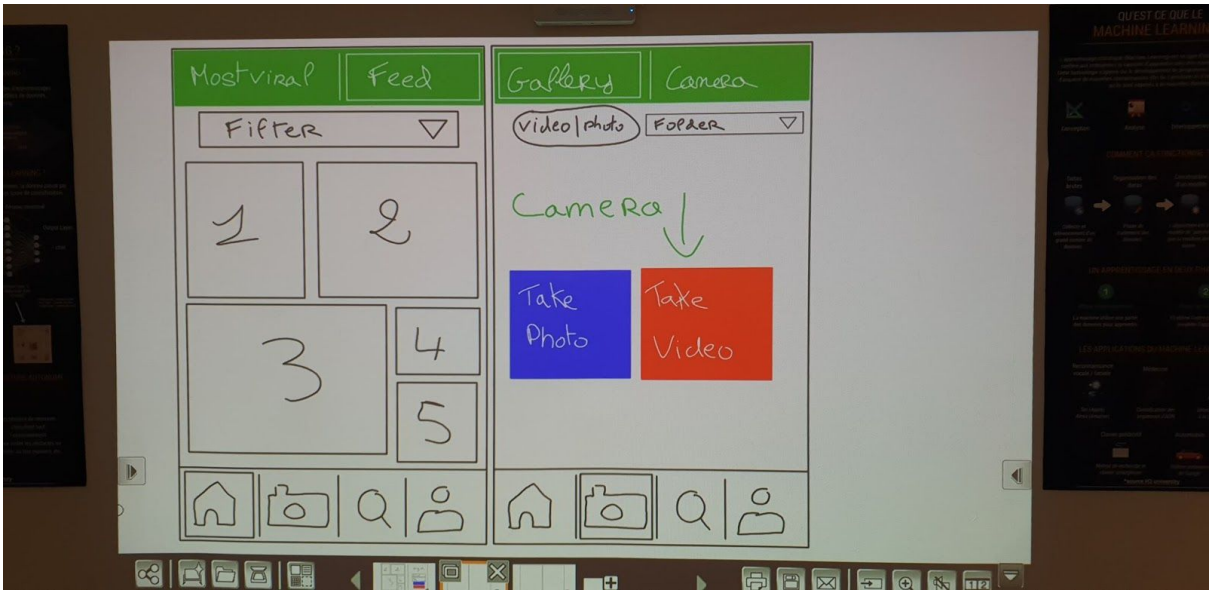

Figure 2
Authed Home Page and Upload View

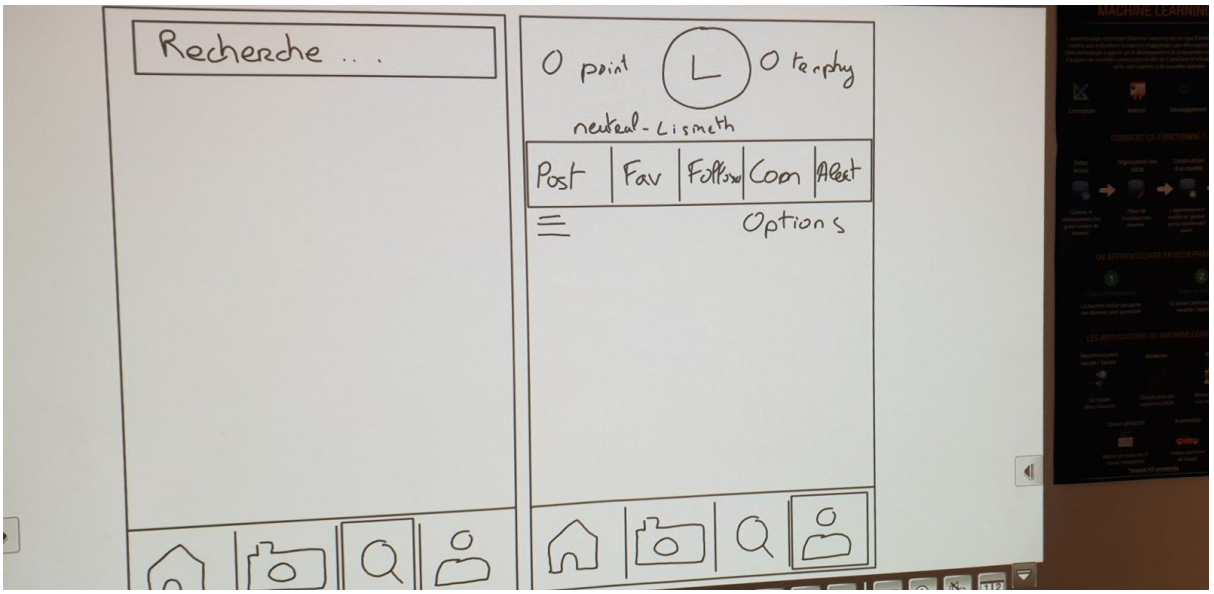

Figure 3
Search and Profile View

To follow our guidelines we followed a Trello referencing all the tasks needed to realise the app.

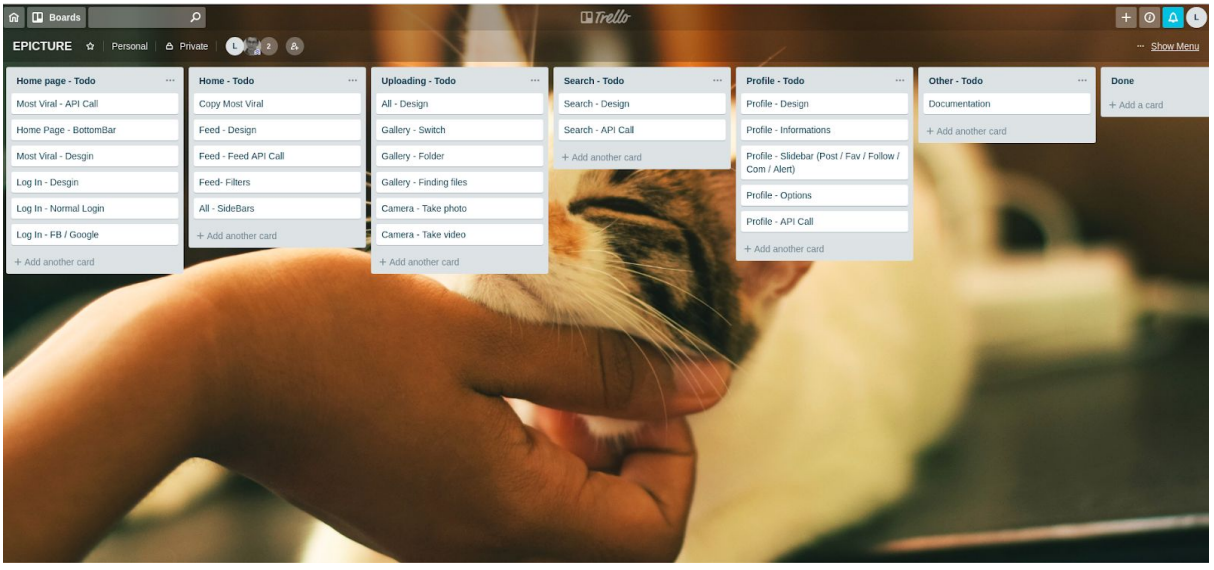

Figure 4
Start of Project

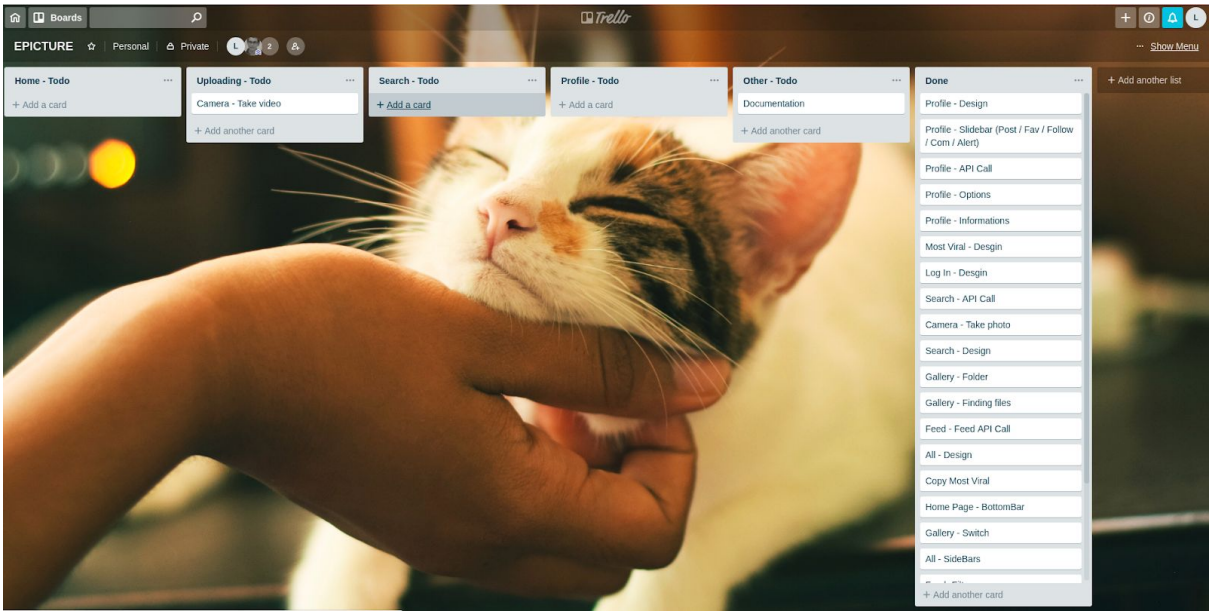

Figure 5
End of Project

note: As you can see we did not do the Camera video implementation, this is mainly because the Imgur Api does not have a method to upload video.

To not get *lost* in our codes and to make it *scalable* we made re-usable Component, they allow us to:

- Parse API response and map it.
- Display Images
- Display Videos
- Display Comments
- Display Tags
- Display Galleries Information

They can be found in *src/ParseContent* folder, here a little example:

JS ParseContent.js ×

```
import React from 'react';
import ImageComponent from './ImageComponent/ImageComponent';
import VideoComponent from './VideoComponent/VideoComponent';

export default class ParseContent extends React.Component {
    render() {
        return this.props.apiRes.map((data, index) => {
            if (data.images) {
                if (data.images[0].link.match(/\.(jpg|png|gif)/g)) {
                    return (
                        <ImageComponent
                            image={data.images[0]}
                            data={data}
                            key={'data' + index}
                            token={this.props.token}/>
                    )
                } else {
                    return (
                        <VideoComponent
                            video={data.images[0]}
                            data={data}
                            key={'data' + index}
                            token={this.props.token}/>
                    )
                }
            } else {
                if (data.link.match(/\.(jpg|png|gif)/g)) {
                    return (
                        <ImageComponent
                            image={data}
                            data={data}
                            key={'data' + index}
                            token={this.props.token}/>
                    )
                } else {
                    return (
                        <VideoComponent
                            video={data}
                            data={data}
                            key={'data' + index}
                            token={this.props.token}/>
                    )
                }
            }
        })
    }
}
```

Figure 6
ParseContent.js