EPITECH. TECHNOLOGY

TIME MANAGER

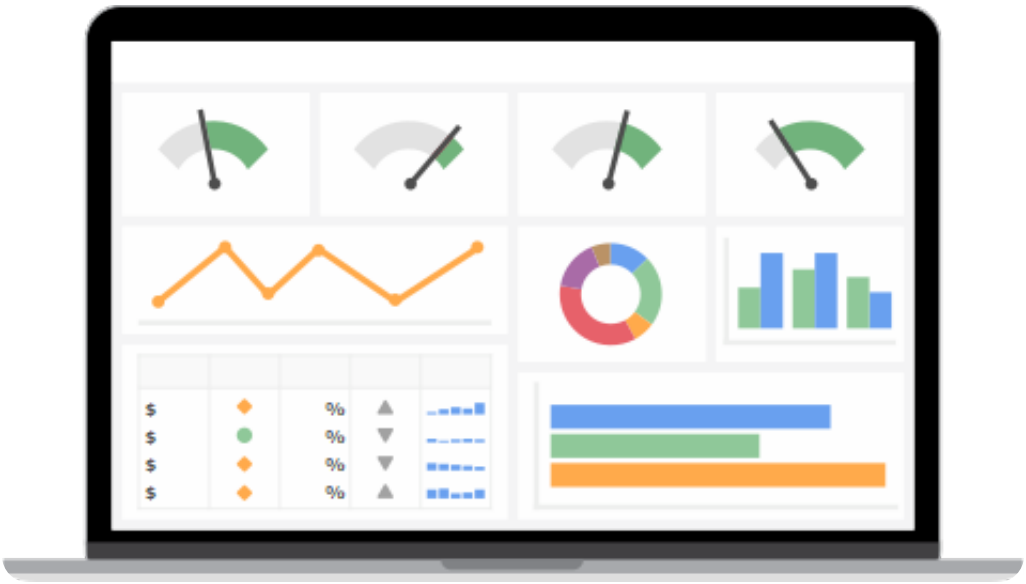
WEB INTERFACES

TIME MANAGER

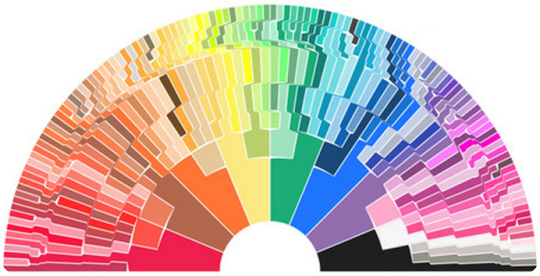
Before you start, make sure you have **finished** and **assimilated all the concepts** discussed in the Bootstrap. In addition, you will need to use the API that you developed previously, so make sure it is functional.

A meeting is planned between your manager and the Mayor of Gotham next week.
Your manager must make a first demo of the application. He asks you to put in place the employee information display so that he can give an overview of the final result.

Create a user interface that displays graphs and dashboards to visualize a person's working time.



This project is part of a problematic of *Data Visualization*, which consists in the graphic representation of figures or raw data.



Some organizational constraints have also been imposed by your project manager :

- ✓ you must use the JavaScript framework Vue.JS to create the interface.
- ✓ you can use any tool you fancy for graphics, we advise you `vue-charts` but your free to do as you want.
- ✓ for ergonomics reasons, you will only need **one and only one** view, defined in a `App.vue` file in the `/src` folder.
- ✓ all the components must be in the `/src/components` folder.

Five components are required :

- ✓ **User**
  - used to identify the current user ;
  - must be present on all pages of your web application ;
  - must implement the following methods (with self-explanatory names):
    * `createUser()` ;
    * `updateUser()` ;
    * `getUser()` ;
    * `deleteUser()`.
- ✓ **WorkingTimes**
  - used to display the working times recorded by the API ;
  - connected to the */workingTimes/:userI* route ;
  - should also have at least:
    * the `userId` and `workingTimes` data (the table summarizing the offset times) ;
    * the `getWorkingTimes()` method.
- ✓ **WorkingTime**
  - used for displaying, creating, modifying and deleting a working time ;
  - linked to the routes:
    * */workingTime/:userid* (for creation) ;
    * */workingTime/:userid/:workingtimeid* (for modification and deletion).
  - it will implement the methods:
    * `createWorkingTime()` ;
    * `updateWorkingTime()` ;
    * `deleteWorkingTime()`.
- ✓ **ClockManager**
  - used to declare hours worked ;

- connected to the */clock/:username* route ;
- it must have:
  * `startDateTime` data (is worth *null* if no work period is in progress) ;
  * `clockIn` (a boolean that is *true* if a work period is in progress) ;
  * `refresh()` and `clock()` methods (to pass from active to inactive and vice versa).

✓ **ChartManager**

- used to manage at least three graphs ;
- connected to the */chartManager/:userid* route ;
- graphs are configurable and of different types (bar, line, pie, radar, …).

You can create a specific component per chart, but you **must** have the ChartManager component. If you decide to use different components for each chart, you **must** use the Vue.JS routing system.

ALL your dates and times should be stored as follows : “YYYY-MM-DD hh:mm:ss”. Look at the *watch* side of the components.
The overall rendering of your application and its ergonomics are essential for a good user experience. A good UX and appropriate features must be your priority, otherwise your application may never be used!

v 2.0