

**Internship report**

Programming of a new API in Python (Django) and Javascript.

RHOULAM Mehdi

Promotion 2022

Epitech Marseille

# Appreciation

It’s never easy for a student to find an internship. So, I would like to thank **Mohamed Bijou** who made me discover this opportunity in **Crossdock** to have integrated me into their company.

I would like to thank my internship supervisor, IT project manager, for his patience and for accompanying me throughout my internship.

I would also like to thank **Abdouraman**, an IT intern who finished his master, for giving me a good example of how projects are managed, how to think about a project's architecture,

He also gave me good advice on how to learn fast with good understanding of django, python and his valuable advice really helped me to improve myself.

Finally I would like to thank all the employees like Leila and Hakim the company for their welcomeness and their professional seriousness during 6 months of internship.

# Contents

[Appreciation 2](#_Toc518237475)

[Contents 3](#_Toc518237476)

[Introduction 4](#_Toc518237477)

[The company 5](#_Toc518237478)

[Organization Chart 7](#_Toc518237479)

[Medreport 8](#_Toc518237480)

[Realized projects 10](#_Toc518237481)

[Big Report 10](#_Toc518237482)

[Holiday administration and management 11](#_Toc518237483)

[Smartphone version 15](#_Toc518237484)

[The interface of the site in tablet and smartphone version 15](#_Toc518237485)

[The technologies 16](#_Toc518237486)

[Development Environment 16](#_Toc518237487)

[Data base 16](#_Toc518237488)

[Langages 16](#_Toc518237489)

[Frameworks 17](#_Toc518237490)

[Software architecture 18](#_Toc518237491)

[The MVT 18](#_Toc518237492)

[The MVVM 19](#_Toc518237493)

[The MVVM 20](#_Toc518237493)

[The MVVM 19](#_Toc518237493)

The MVC …………………………………………………………………………………………………………………………………18

[Conclusion 20](#_Toc518237494)

[Letter of motivation 21](#_Toc518237495)

# Introduction

To validate my second year at Epitech, each student must complete an internship of 6 months in a company. I was really lucky because I worked at Crossdock as an intern for a period of 6 months from March 04, 2019 to September 1, 2019, located at 122 Allée de la Lavande, 84300 Cavaillon.

During this six-month internship I learned to work in a company and put to good use the knowledge I acquired during my two year at Epitech.

The first two weeks of the internship allowed me to learn new technologies through training under the supervision of Abdouramane Mahamane, my internship colleague. The good atmosphere that prevailed in the group gave me the motivation I needed to complete the various tasks that were assigned to me.

The main topic of this course create from scratch a new API from the help of an older one

In a technology called Filemaker 14 in order to have a consistent long term support,

And a philosophical envy to have its own api developped by its own employees.

# The company

History

Crossdock was founded in 2013 by Mr. Mohamed Bijou.

Crossdock is a recognized logistic provider, ahead in term of « tracability » and « security » for drugstore, in the field of unrestricted medical contents in France they are working for the leader of dietetic groups.

Activities

The main areas of intervention:

* Photograph packaged products at the entrance.
* Delivers products directly.
* Seeks delivery companies for customers.
* Resolves issues between delivery companies and customers or the destinators.
* Change automatic delivery process by the urge for the destinator.

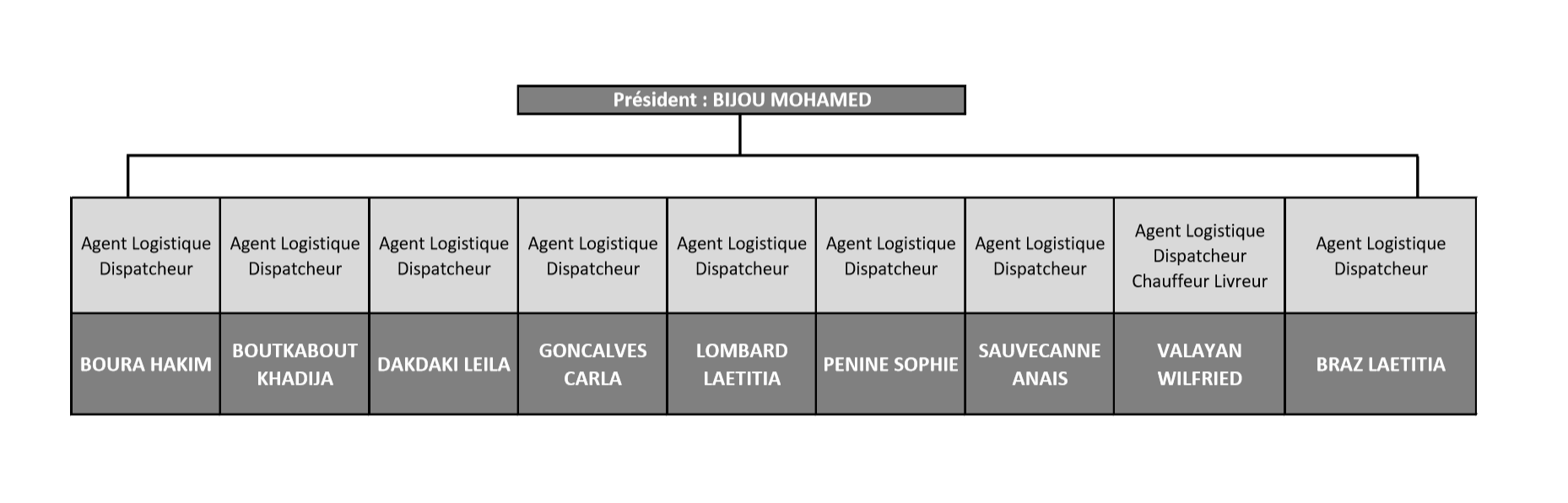
The company has about ten employees at its headquarters in the Vaucluse region and a lot of destinator in France. Each package is checked by a Logistics Agent Dispatcher. The Logistics Agent Dispatcher is the person in charge of verification, each issues must be reported to the chairman.

Organization

Crossdock has not stopped growing during these last years, in particular thanks to the organization's quality. To respond most effectively to a customer, the company has set up a quality infrastructure. Indeed Crossdock is composed of several crucial points:

* **Supply managing**: Organize the management of the service. Facilitate the different stages of the income/outcome of products from stock. Manage part of the department's files independently for each package received & sent out. Analyze situations and implement elements to solve, improve, innovate. There are different main activities such as setting up and updating supplier or product(article name…) databases, managing the production/non-production supplier panel.
* **Products Managements :**
* **Litigation/disputes :**
* **Labels / QRCodes :**

## Organization Chart



# Gestion Stock

During my internship I worked a lot on an application called **Filemaker**. **Filemaker** is a propietary application made for developping api really fast, its made up by Apple inc,

set up by Crossdock to facilitate the management of multiple packages and their tracability as for example being able to modify a package expiration’s date of the application or still consult or modify a site of building site etc…

There are two versions of this :

* **Gestion Stock v1** (Fully programmed with \***Filemaker/No-sql** Not accessible out of the network trough the firewall)
* **Gestion Stock v2** (Fully programmed with \***Django/html/jquery** Not accessible out of the network trough the firewall)

**Gestion Stock v1** will gradually disappeare by updating its functionality on **Gestion Stock v2**

We also need to be sure of its sustainability & stability

We are invited to login in order to access the **Gestion Stock** service (user and password are provided by the company’s Chairman)

On **Gestion Stock** the rights to view, modify documents or other rights vary according to the type of user connected. There are 3 types of users to be developped: Admin(Check logs…), employees(do the disptaching stuff) and clients(see what happen to their products).

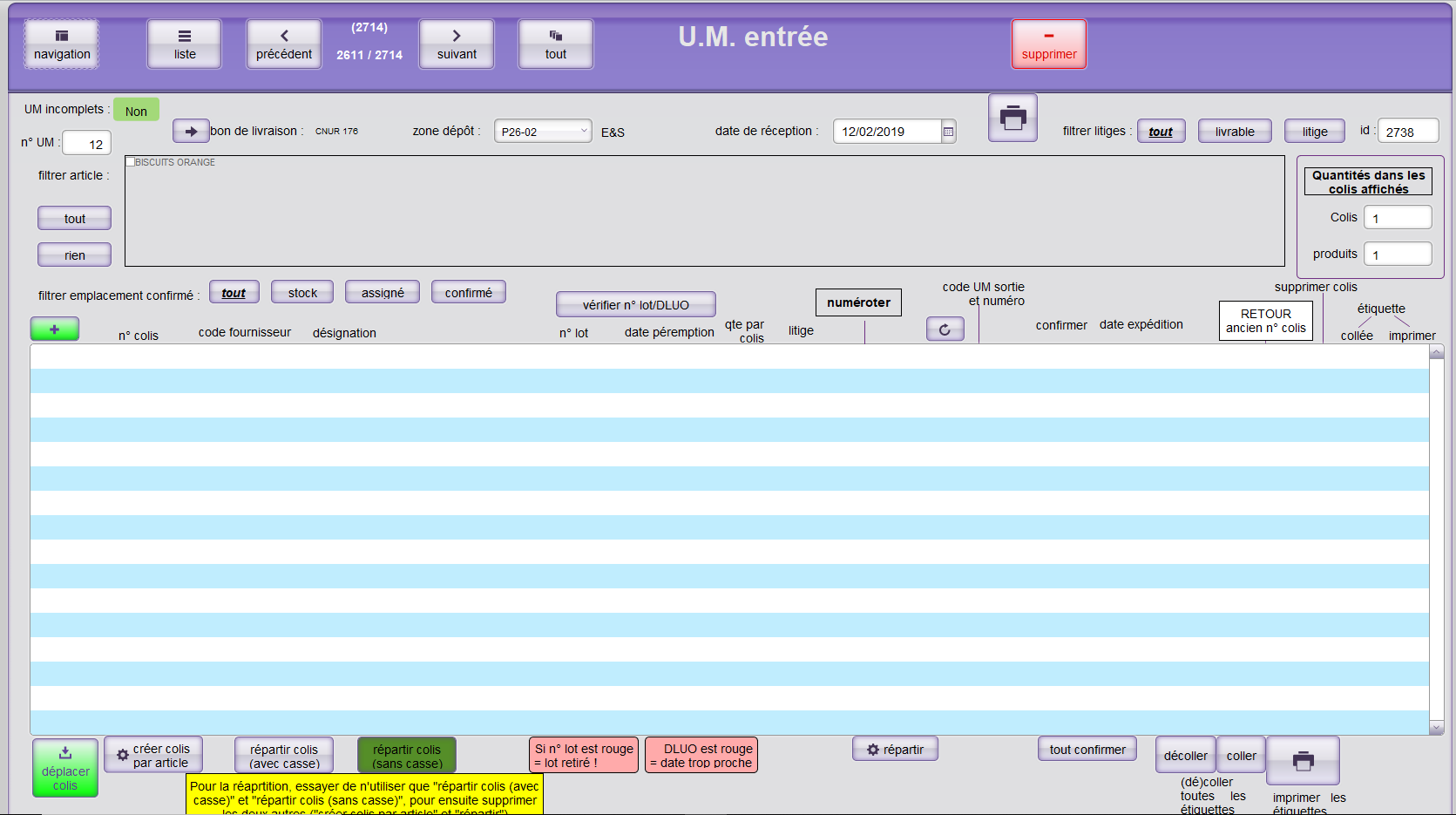
As a developer I had access to **Gestion Stock** with an internal account, which allowed me to have access to all the different **Gestion Stock** functionalities (in a Back-up).

In order to add new functionalities, there is a development process :

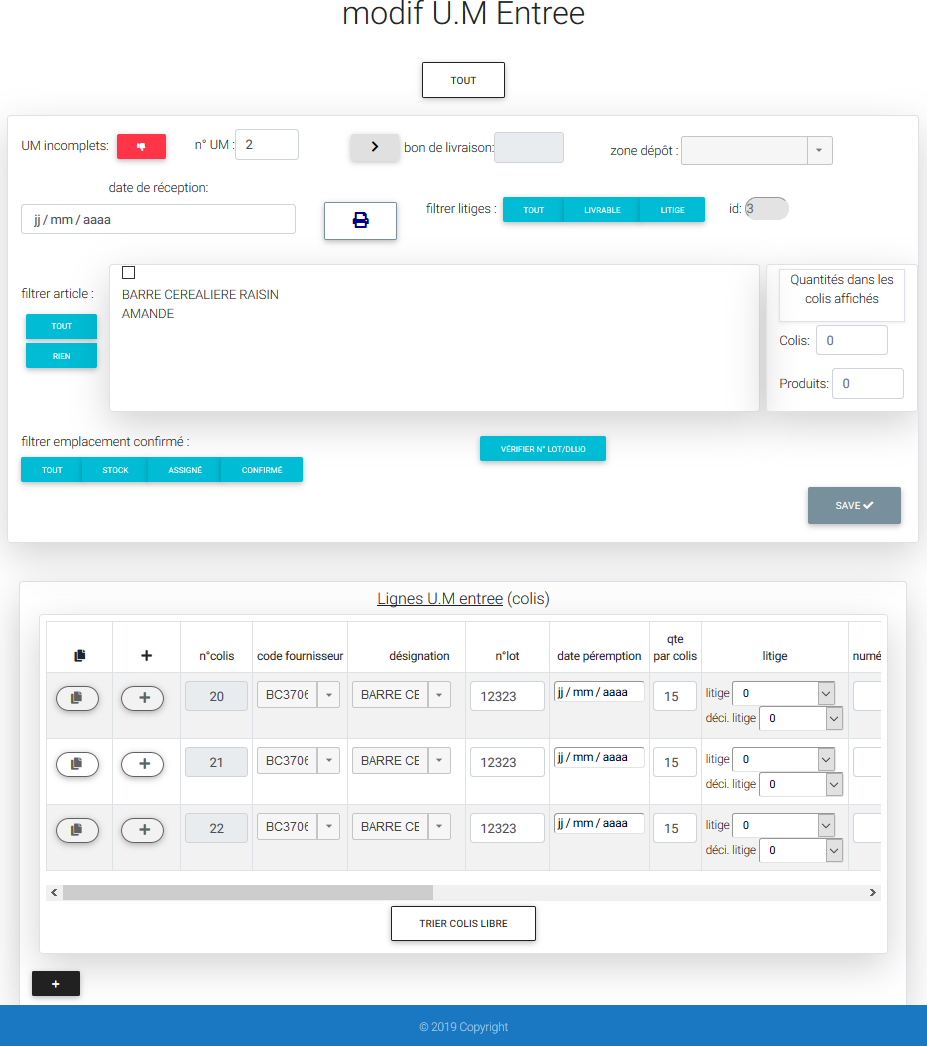
* We are developing a functionalities on « **Medreport development** » (with a different database from « **Medreport production** » which is the public mode and accessible in local network and with a different URL) This allows us not to create conflicts with « **Medreport production** »
* Once everything is set up, we first publish on « **Medreport demo »**. This one uses the database of the production version, we can test if everything works well, knowing that the database of the production version is more important in terms of content than that of the development version.
* And finally, once we have finished testing on the demo version we publish on the production version.

Here is what the different versions of Medreport (In development mode) look like.

**Gestion Stock v1**



**Gestion Stock v2**



# Realized projects

## Purchase Orders :

## 

1. This zone allows to target sites by type (for example AMU or SST), by zone (for example East or West) or by a specific name (for example "SPIE" to have only the sites which contain the field "SPIE")
2. The zone which makes it possible to define the period on which we wish to carry out our research (if the period is of a year or 12 months we add every month for a selected data) as well as to choose also the inactive sites (A site can be deactivated for works).
3. Button to pop up a new window to select the data we want to have for each site found.
4. The event after pressing this button is the export under a worksheet (Excel for example) of the result displayed on the main grid. A powerful algorithm is set up to recover all the data displayed on the grid.
5. The main grid where is displayed the result found thanks to the filter applied in the form. We can see that there is a fairly distinguished and accurate grid layout.
6. This part allows you to see the total. which means that we add each numerical data.

## Holiday administration and management

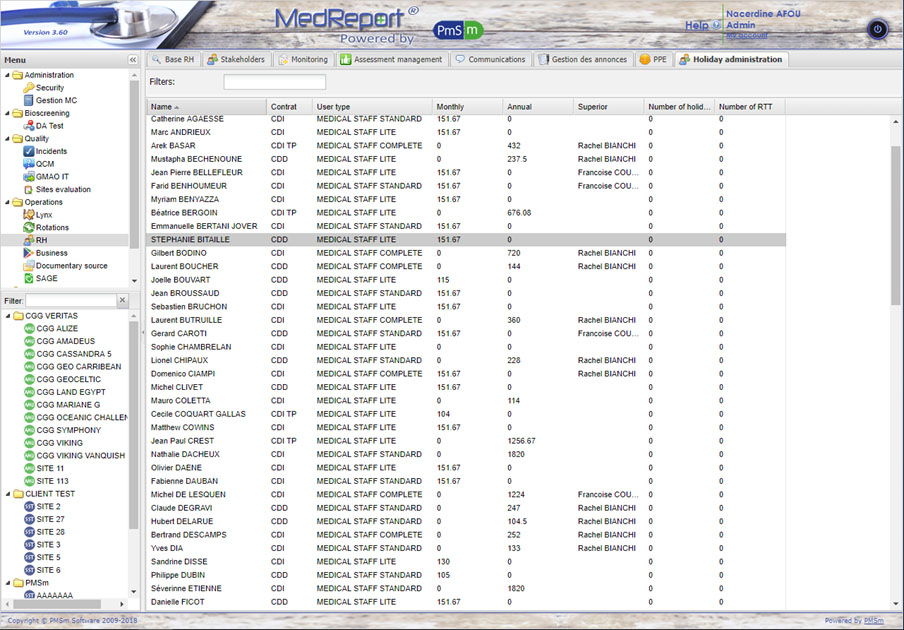
As mentioned above there are three types of user. Clients, interns and interveners, we will be interested in interveners. When a interveners requests days off, it is mandatory for him to send a leave request form to the person in charge of the site (CMO). The CMO must check the intervener's schedule to see if this person can take the leave on the date requested. This verification can be long and wastes a lot of time for the CMO.

Based on this problem, we decided to set up a leave request site. Indeed this will facilitate the work at the CMO who only needs validate the request and not following the entire verification process.

*\*CMO : The CMO is a person who is part of the "Operations FRANCE" department that takes care of a client's site. He is the hierarchical superior of the interveners and the person in charge of the site. It performs several tasks such as recruiting staff or granting days off for the site's interveners.*

The project was carried out in two parts :

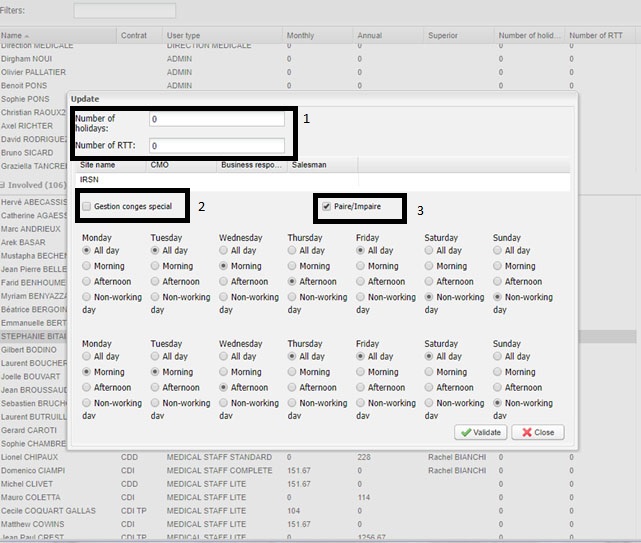
1. **Holiday administration**: This part can only be seen and modified by PmSm staff. It makes it possible to define the days worked of a user and then apply the necessary rules according to the schedule of a intervener.



As we can see on the image above, I list all the interveners with their type of contract which makes it possible to differentiate the rules to apply, the profession of the intervener (doctor or nurse), the number of hours worked over the month or the year.

When I click on a user a new window appears to modify the days worked.

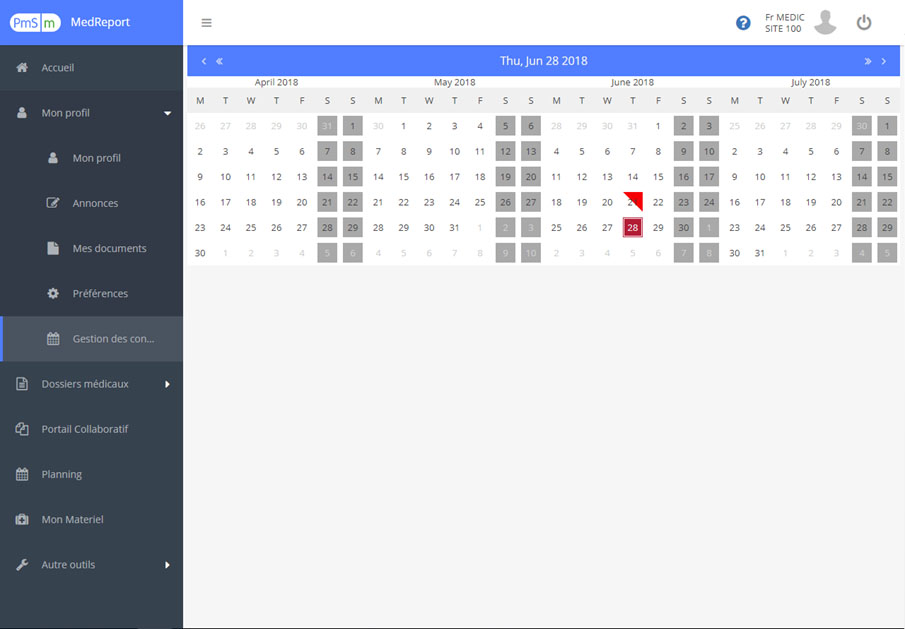
Here is an illustration:



1. These fields allow you to modify the number of days of leave available to the user (Whether annual leave granted by the state or "RTT" type vacations).
2. This checkbox allows you to specify if the user has a different type of contract than the majority of interveners. This allows the CMO to treat individual cases differently.
3. The majority of speakers have an "odd/even" type of week contract and that is why a checkbox option has been added to this platform to specify if a person is on a monthly contract with days worked that varies according to the week.

This configuration is very important for the second part. Indeed it will make it possible to see the days not worked of an intervener to apply rules ( compared to the French laws) as regards the deposits of leave.

1. **Holiday deposit management**: This part is visible and modifiable by the interveners. It allows you to deposit vacation days, several vacation deposit options are available such as the possibility to choose several days in a row, the possibility to choose only half a day, the possibility to modify or cancel a vacation request or to consult the vacation history.

****

On this illustration we can see the calendar with the following indications :

* Shaded cells for days not worked. Indeed thanks to the administration leaves we configured the days not worked of an intervener who were loaded directly on this panel in order to disable the cells of the days not worked.
* A cell half filled in red to represent a half day (the upper half to represent the morning and the lower half to represent the afternoon) of leave deposited by the user (the intervener).
* A cell filled with dark red to represent today's day

It should be noted that this part is still in progress and therefore the presentation is not complete. There will be additional features such as a leave history filed or being validated. A mobile version is planned for vacation management.

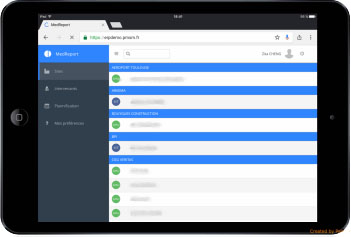
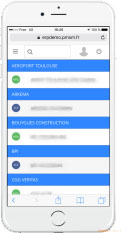
# Smartphone version

The **Gestion Stock v1** site was not designed for use on smartphones or tablets, which is very

impractical when you don't have a computer with you.

**Gestion Stock v2** uses a Boostrap. Therefore it is possible to make it compatible with the smartphone thanks to the mobile first design used by Bootstrap.

## The interface of the site in tablet and smartphone version

The presentation of the site don’t really varies according to the size of the screen of the support used. On a larger interface we have a menu which is displayed bigger.

While on an interface smaller you have to scroll on menu button to display the choices.

We designed the app to look same on phone, tablet and pc, by using simple « table » logics,

And tried to have the less interaction between api and users, most data is found by the « views »

User only have to select good value and « use » it most of the time.

# The technologies

## Development Environment

During the internship I was working under Windows on IntelliJ which is an editor for Java(good for javascript), HTML and Javascript editing by JetBrains.

JetBrains has several advantages that facilitate the use of the software such as :

* **Major frameworks supported**: IntelliJ is perfect for working with Django, WordPress, Laravel, Magento, CakePHP, codeigniter, and other frameworks.
* **All Python tools**: The editor actually 'gets' your code and deeply understands its structure, supporting all Python language features for modern and legacy projects. It provides the best code completion, refactorings, on-the-fly error prevention, and more (we got 3 lvl of Highlighting, None, Syntax, Inspections).
* **Intelligent Coding Assistance** : Hundreds of inspections take care of verifying your code as you type, analyzing the whole project. It support external documentation, code (re)arranger and formatter, quick-fixes, and other features help you write neat code that’s easy to maintain & the most important auto-indent for python hard laws on indents.
* **Fast and Safe Refactoring** : Refactor your code reliably with safe Rename, Move, Delete, Extract Method, Inline Variable, Push members Up / Pull members Down, Change Signature, and many other refactorings. Language-specific refactorings help you perform project-wide changes in a matter of a few clicks, and can​ be safely undone.

## Data base

PostgreSQL (PSQL) is a relational and object database management system administration and development. We use it for database management.

It suit the need of scalability needed in this sector of logistic

## Langages

There are two types of language used for our application:

* Client-side language (Javascript, front-end) allows to manage the user interface: display, interactions (buttons, etc).
* Server-side language (Python / Django) allows to process a request sent on the client side and returns a response in JSON format or do some redirection with some GET request in url.

## Frameworks

During this internship I discovered two Frameworks :

* Django that can be used to develop web applications and services using Python2 or 3, and provides 100% object-oriented code using a broad spectrum of language features,

Django helps you build data-intensive, cross-platform web apps for desktops, tablets, and smartphones. Take advantage of powerful features in with postegresql who can be really good at scalling data-size,

With the help of python we can do almost anything we want, with ease.

# Software architecture

## The MVT

MVT is a design patterns, mostly used to organize our code files, it follow the logic of mvc.

**Model**: Its where we create the « Database » structure,

Django can recreate database structure from models objects declarations,

We can also select wich data we send by default on object call,

We can decide of foreign-key deletion behavior.

**View**: Part that select which data is sent to a template, or the behavior to have

On the request sent by template to him,

He can do whatever he want on model, he is the « middleman »

**Template**: This part manages the logic of the code that takes decisions. In a way, it is the intermediary between the model and the view: the controller will ask the model for data, analyze them, make decisions and return the text to be displayed to view. The controller contains exclusively PHP. In particular, it determines whether the visitor has the right to see the page or not (access rights management) .

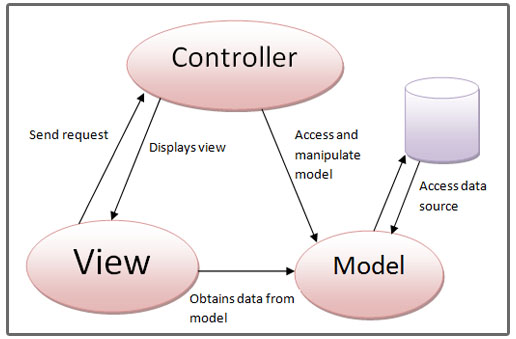


## The MVC

**Model**: Element that contains the data as well as logic related to the data: validation, reading and recording. It can, in its simplest form, contain only a single value, or a more complex data structure. The model represents the universe in which the application is embedded. For example, for a bank application, the model represents accounts, customers, and transactions such as deposits and withdrawals, and checks that withdrawals do not exceed the credit limit.

**View**: Visible part of a graphical interface. The view uses the model, and can be a diagram, a form, buttons, etc.. A view contains visual elements as well as the logic needed to display data from the model. In a typical desktop application, the view obtains the data needed to present the model by asking questions. It can also update the template by sending appropriate messages. In a web application a view contains HTML tags

**Controller**: This part manages the logic of the code that takes decisions. In a way, it is the intermediary between the model and the view: the controller will ask the model for data, analyze them, make decisions and return the text to be displayed to view. The controller contains exclusively PHP. In particular, it determines whether the visitor has the right to see the page or not (access rights management).



## The MVVM

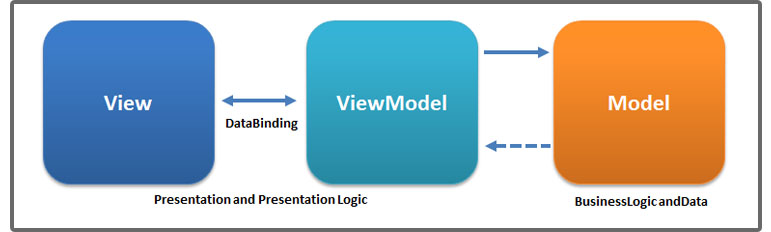
The MVVM model has been specially designed to improve the separation between the data and the view that displays it. The link between the view and the data is done by binding mechanisms.

The binding is a mechanism which makes it possible to make links between data dynamically. Which means that if A and B are related, the fact to change A will be passed on to B and vice versa.

**Model**: As for the MVC, the model contains data. Usually, this data comes from a database or an external service such as an API.

**View**: As for the MVC, the view corresponds to what is displayed. The view contains the different graphic components (buttons, links, lists) as well as the text.

**View-Model**: this component makes the link between the model and the view. It manages data links and possible conversions. This is where binding comes in.



# Conclusion

First of all I would like to thank Crossdock again for giving me a huge chance to join them.

I had the chance to have a really good project for my second year,

Working on an api from scratch on a framework i didn’t even knew, nor the knowledge of Python language, and same for filemaker, I also had to learn its syntax.

I still have 1 Month in my internship at the moment, what I was asked to do is « ready to use » of course it cannot be put in production before to have it tested a lot,

I will sustain what i did by making it more safe to have it riskless. It can therefore be said that the goal set for me at the hiring process, has been achieved !

This internship interested me a lot, it allowed me to deepen my knowledge in computer science and to acquire additional experience in the field of web api development.