

## RAPPORT DE STAGE

---

BARAUD Romain  
Stage de développement web au LESIA  
Du 14/04/2016 au 30/06/2016



# Informations utiles

## Stagiaire

*Nom* : BARAUD

*Prénom* : Romain

*E-mail* : romain.baraud@gmail.com

*Mission du stage* : Réaliser une application web pour la gestion du plan de charge du pôle technique

## Entreprise

LESIA - Observatoire de Paris - CNRS

*Adresse* : 5, Place Jules Janssen - 92195 Meudon CEDEX

*Tutrice* : M<sup>me</sup> HENRY Florence

*E-mail* : florence.henry@obspm.fr

*Téléphone* : 01.45.07.78.95

## Université

Institut Universitaire de Technologie de Paris XIII (Villetaneuse)

*Adresse* : 99, Avenue Jean-Baptiste Clément - 93430 Villetaneuse

*Tuteur* : M. HÉBERT David

*E-mail* : hebert.iut@gmail.com

*Téléphone* : 01.49.40.31.34



# Remerciements

## À Madame Henry

Pour son accueil, sa disponibilité, et ses précieux conseils dans la réalisation du projet. Pour ce temps que vous m'avez consacré, mais aussi pour être toujours restée à mon écoute et m'avoir appris plein de chose.

Soyez assuré de mon respect et de mes sincères remerciements.

## À Monsieur Hébert

Pour avoir accepté d'être mon tuteur de l'IUT, m'aider à la réalisation de ce rapport ainsi qu'avoir pris le temps de m'écouter lors de la soutenance. Et d'une manière plus générale, pour avoir pris du temps à m'expliquer de nombreux concepts mathématiques.

Vous trouverez ici le témoignage de ma reconnaissance.

## Au pôle plasma et technique

Merci également à toute l'équipe du pôle plasma et technique du LESIA, et particulièrement aux trois thésards : M. LION Sonny, M<sup>me</sup> GRITON Léa et M. MARTINOVIC Mihailo pour leur accueil chaleureux et pour m'avoir soutenu et supporté tout au long de mon stage.

## Et enfin

À M. DROSSART Pierre, directeur du LESIA, pour m'avoir accueilli au sein de son établissement.

À M<sup>me</sup> CARDOSO Sylvie, de sa précieuse aide pour la recherche d'un stage.

À l'ensemble des professeurs de l'IUT de Villetaneuse, pour leurs enseignements efficaces et pédagogiques.

Ainsi qu'à vous qui allez lire ce rapport.



# Table des matières

<b>Informations Utiles</b>	<b>3</b>
<b>Remerciements</b>	<b>5</b>
<b>1 Introduction</b>	<b>9</b>
<b>I Présentation de l'entreprise</b>	<b>11</b>
<b>2 L'Observatoire de Paris</b>	<b>13</b>
1 Son histoire . . . . .	13
2 L'Observatoire aujourd'hui . . . . .	13
<b>3 Le LESIA</b>	<b>17</b>
1 Son histoire . . . . .	17
2 Organisation du laboratoire . . . . .	18
<b>4 Structure de l'entreprise</b>	<b>21</b>
<b>II L'application</b>	<b>23</b>
<b>5 L'application demandée</b>	<b>25</b>
<b>6 Les différents outils</b>	<b>27</b>
1 La base de données . . . . .	27
2 Le développement . . . . .	29

3	Les frameworks . . . . .	31
<b>7</b>	<b>La création</b>	<b>33</b>
1	La base de données . . . . .	33
2	Développement avec Rails . . . . .	35
3	Le site web . . . . .	40
<b>III</b>	<b>Expérience professionnelle</b>	<b>45</b>
<b>8</b>	<b>Mon expérience</b>	<b>47</b>
1	Préambule . . . . .	47
2	L’humain . . . . .	47
3	La technique . . . . .	48
<b>9</b>	<b>Conclusion</b>	<b>49</b>
<b>IV</b>	<b>Annexes</b>	<b>53</b>
<b>A</b>	<b>Documentation</b>	<b>55</b>
<b>B</b>	<b>Schémas de la base de données</b>	<b>63</b>
<b>C</b>	<b>Les outils utilisés</b>	<b>67</b>
<b>D</b>	<b>Captures d’écran</b>	<b>71</b>



# Chapitre 1

## Introduction

Dans le cadre de mon DUT Informatique parcours PEL<sup>1</sup>, j'ai effectué un stage de onze semaines (et un jour) au sein de l'Observatoire de Paris et plus précisément dans un de ses laboratoires : le Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique (LESIA). Ainsi, pendant plus de deux mois, le service technique de ce laboratoire m'a accueilli et pris sous son aile, afin de les aider à développer une application de gestion du plan de charge des IT<sup>2</sup>.

L'objectif de ce stage est de mettre en pratique les connaissances acquises en développement logiciel et web durant les deux années d'IUT et de découvrir la réalité du monde du travail. L'informatique étant vaste, je souhaitais, personnellement, une entreprise qui me permettrait de m'enrichir dans le domaine du développement web. De plus, je souhaitais, une entreprise ou un organisme qui ne soit pas une ESN<sup>3</sup>, je voulais changer de cadre, les ESN étant orientées informatique pure et dure. C'est le cas du LESIA, qui est un laboratoire de recherche français, qui recherchait un stagiaire pour développer une application web pour ses différents services.

Durant mon stage, voici les différentes missions que l'on m'a attribuées :

- Apprentissage d'un nouveau langage : Ruby on Rails
- Implémentation des framework<sup>4</sup> : Bootstrap et HighCharts
- Finaliser les spécifications du produit
- Proposer les évolutions nécessaire à la base de données et les implémenter
- Coder, tester et déployer l'application

Mais cet organisme me donne l'opportunité de découvrir plus que de l'informatique. Je suis plongé au cœur d'un univers rempli de chercheurs et d'ingénieurs spécialisés en astronomie et plus précisément en physique des plasmas. Ainsi, j'ai pu profiter de ces onze semaines pour appréhender chaque service et enrichir ma culture de l'astronomie.

Je vous parlerai d'abord de l'histoire et de l'organisation du laboratoire, puis dans un second lieu j'aborderai mon expérience professionnelle, le développement de l'application et sa mise en place.

---

1. Poursuite Études Longues

2. Ingénieurs et Techniciens

3. Entreprise de services du numérique

4. Un ensemble de fonctions déjà implémentées prêts à l'emploi, destinées autant à faciliter le travail qu'à augmenter la productivité du programmeur qui l'utilisera



# Première partie

## Présentation de l'entreprise



# Chapitre 2

## L'Observatoire de Paris

### 1 Son histoire

Sous la pression de nombreux savants et notamment Adrien Auzout qui écrit, en 1665, une lettre à Louis XIV pour lui demander de créer sans plus attendre une compagnie des sciences et des arts, c'est en 1666 que Louis XIV et Jean-Baptiste Colbert fondent l'Académie royale des sciences.

Lors de sa première séance, le 22 décembre 1666, il est décidé la création de l'Observatoire royal, qui deviendra l'actuel Observatoire de Paris. Il devait servir de lieu de réunion et d'expérimentation pour tous les académiciens. Mais en raison de son éloignement du Paris de l'époque, seuls les astronomes l'ont utilisé.

Le 21 juin 1667 (jour du solstice d'été), les mathématiciens de l'Académie tracent sur le terrain, à l'emplacement actuel du bâtiment, le méridien et les autres directions nécessaires à l'implantation exacte de l'édifice conçu par l'architecte et médecin Claude Perrault (frère du conteur Charles Perrault qui était également secrétaire de Colbert). Le plan médian du bâtiment définira désormais le méridien de Paris, les horloges se réglant sur le midi vrai. En 1669, Colbert appelle Giovanni Domenico Cassini pour diriger l'institution, il fait effectuer des modifications du bâtiment. Louis XIV visitera pour la première fois l'Observatoire de Paris en mai 1682, soit 10 ans après la fin des travaux de l'Observatoire. Conçu comme une citadelle des sciences, le bâtiment est sobre avec une tour carrée en avant-corps (exposée plein nord) du côté de la grande avenue, et ses ailes latérales en forme de pavillons octogones.

Le site de Meudon a été établi en 1876 à l'emplacement de l'ancien château de Meudon.

### 2 L'Observatoire aujourd'hui

L'Observatoire de Paris [8] est un grand établissement, séparé en 3 zones géographiques (Paris, Meudon et Nancay) dont le fonctionnement est défini par le décret 85-715 du 10 juillet 1985. Il est dirigé par un président élu pour quatre ans, et administré par un conseil d'administration.

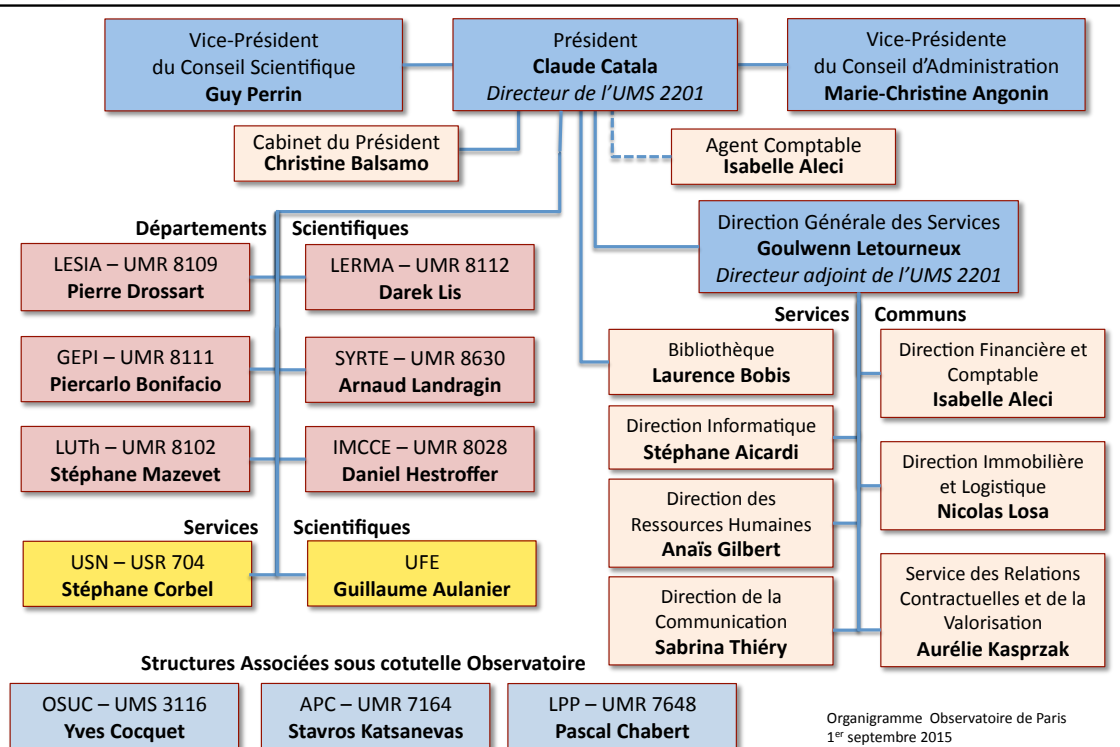


FIGURE 2.1 – L’organigramme fonctionnel de l’Observatoire de Paris (2015)

Le conseil d’administration et le président sont assistés d’un Conseil scientifique. Un Haut Comité Scientifique (HCS), composé de personnalités indépendantes, évalue régulièrement l’activité de l’établissement. L’Observatoire de Paris comporte cinq départements scientifiques, un institut, deux services scientifiques et des services communs.

En 2010, l’Observatoire compte près de 600 emplois permanents (Rapport AERES 2010) :

- 333 emplois permanents propres relevant du Ministère de la Recherche et de l’Enseignement Supérieur (dont 89 CNAP<sup>1</sup>, 10 Enseignants-Chercheurs, 2 PRAG<sup>2</sup>, 232 personnels de soutien)
- 248 titulaires du CNRS<sup>3</sup> y sont affectés
- 35 enseignants-chercheurs d’autres établissements de l’enseignement supérieur
- 9 personnels divers travaillent dans les laboratoires de l’établissement
- l’établissement emploie 42 contractuels dont 11 sur postes vacants, 19 sur contrat CNRS, le reste sur budget de l’établissement.

Établissement d’enseignement supérieur, 245 étudiants y étudient (ce chiffre intègre les étudiants suivant les cours de l’Observatoire mais inscrits dans les universités partenaires).

1. Conseil National des Astronomes et Physiciens  
 2. PRofesseurs AGrégés  
 3. Centre National de la Recherche Scientifique

L'Observatoire de Paris doit remplir 3 missions :

- Une mission de recherche
- Une mission d'enseignement
- Une mission de diffusion du savoir vers le grand public

## 2.1 Recherche

L'Observatoire de Paris est le plus grand pôle français de recherche en astronomie. Les recherches menées en son sein couvrent tous les champs de l'astronomie et l'astrophysique contemporaines, en étudiant :

- Le Soleil et ses relations avec la Terre
- Les planètes et les systèmes planétaires
- La formation des étoiles
- Le milieu interstellaire
- La formation et l'évolution des galaxies
- Les astroparticules
- La cosmologie
- La métrologie de l'espace et du temps
- L'histoire et la philosophie des sciences

Les chercheurs et ingénieurs de l'Observatoire réalisent des instruments d'observation pour les télescopes au sol ou les sondes spatiales, organisent des campagnes d'observation, mettent en place le traitement et l'analyse des données d'observation, élaborent des bases de données, ou réalisent des simulations numériques permettant de modéliser les phénomènes astrophysiques et d'en faire l'interprétation théorique.

Ils sont regroupés au sein de 7 laboratoires :

- Le LESIA (Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique)
- Le LUTH (Laboratoire Univers et THéories)
- Le GEPI (Galaxies, Étoiles, Physique et Instrumentation)
- Le LERMA (Laboratoire d'Études du Rayonnement et de la Matière en Astrophysique)
- Le SYRTE (SYstèmes de Référence Temps Espace)
- L'USN (Unité Scientifique de la station de Nançay)
- L'IMCCE (Institut de Mécanique Céleste et de Calcul des Ephémérides )

## 2.2 Enseignement

L'Observatoire dispense un enseignement supérieur de haut niveau en astronomie et astrophysique allant du master au doctorat, avec possibilité de formation à distance :

- Master 1 : Astronomie et Astrophysique, Sciences de l'Univers et Technologies Spatiales
- Master 2 Recherche : Astronomie, Astrophysique et Ingénierie Spatiale
- Master 2 Professionnel : Outils et Systèmes de l'Astronomie et de l'Espace
- École doctorale 127 : Astronomie et Astrophysique d'Île-de-France
- Formation à distance avec deux Diplômes Universitaires en ligne pouvant donner lieu à la validation d'ECTS

- Formation supérieure en présentiel avec deux Diplômes Universitaires en présentiel pouvant donner lieu à la validation d'ECTS
- Formation des professeurs
- L'Observatoire de Paris accueille aussi les étudiants du cycle pluridisciplinaire d'études supérieures (CyPES) de Paris Sciences et Lettres pour des travaux pratiques.

L'ensemble des activités d'enseignement de l'observatoire est organisé par l'Unité Formation-Enseignement (UFE) de l'Observatoire de Paris.

## 2.3 Diffusion du savoir

L'Observatoire de Paris dispose d'une direction de la Communication qui a pour mission de diffuser le savoir auprès du grand public. Pour cela différents supports et manifestations sont utilisés :

- Site internet
- Expositions itinérantes
- Visites des 3 sites de l'Observatoire et de leurs instruments
- Parcours Système Solaire sur le site de Meudon (représentation à l'échelle du système solaire)
- Parrainages de classes
- Journées portes-ouvertes
- Observations nocturnes pour le grand public une fois par an
- Évènements ponctuels selon l'actualité (année mondiale de l'astronomie en 2009, Année Le Verrier en 2011)



# Chapitre 3

## Le LESIA

### 1 Son histoire

Le Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique (LESIA) a été créé le 1<sup>er</sup> janvier 2002 [9]. Sa naissance est le fruit d'une restructuration générale des laboratoires de recherche de l'Observatoire de Paris qui fait passer leur nombre de huit à cinq.

Installé au départ sur huit bâtiments du campus de Meudon, le LESIA est le plus gros laboratoire de l'Observatoire. Il résulte principalement de la fusion de deux anciens laboratoires, le Département de Recherche Spatiale (DESPA) et le Département de Physique Solaire (DASOP). Il a aussi bénéficié de la venue de l'équipe de planétologie du Département de Radioastronomie (ARPEGES), ainsi que d'une équipe de physique stellaire du Département d'Astronomie Stellaire et Galactique (DASGAL).

Aujourd'hui, le Laboratoire est dirigé par une équipe de direction constituée de :

- Pierre DROSSART, directeur
- Sylviane CHAINTREUIL, directrice adjointe
- Yann CLÉNET, directeur adjoint
- Yann HELLO, directeur technique
- Claudine COLON, administratrice du laboratoire, remplissant également le rôle de secrétaire général et de chef du personnel.

Au 30 avril 2014, le LESIA compte 253 personnels dont :

- 69 chercheurs permanents (CNRS, Observatoire de Paris, Universités Paris VI/VII)
- 65 ITA<sup>1</sup> permanents (CNRS, Observatoire de Paris)
- 20 post-doctorants financés sur contrat (ANR, CEFIPRA, CNES, CNRS, Europe, Observatoire de Paris, Région Île-de-France, ...etc.)
- 31 doctorants financés sur contrat (CNES, DGA, ESO, Europe, Observatoire de Paris, ONERA, Région Ile de France, pays étrangers, ...etc.)
- 21 ITA financés sur contrat
- 1 apprenti

---

1. Ingénieurs, Techniciens et Administratifs

Le LESIA a participé ou initié la création de nombreux instruments, au sol ou embarqués dans différentes missions, parmi lesquels :

- NAOS, un système d'optique adaptative installé sur le Very Large Telescope (VLT) au Chili
- OEIL, une méthode d'optique adaptative pour l'analyse oculaire, actuellement en exploitation au Centre Hospitalier National d'Ophtalmologie des Quinze-Vingts
- CoRoT, un satellite d'études de sismologie stellaire et de détection de planètes extrasolaires
- L'instrument OMEGA, embarqué sur Mars Express, un spectromètre pour l'analyse de l'atmosphère et de la surface martienne
- Les instruments RPWS, CIRS, et VIMS sur la sonde Cassini
- L'instrument VIRTIS-H sur la sonde Rosetta

## 2 Organisation du laboratoire

Le LESIA (Laboratoire d'Etudes Spatiales et d'Instrumentation en Astrophysique) est un département de l'Observatoire de Paris. Il est aussi une unité mixte de recherche du CNRS (UMR-8109).

Le LESIA a quatre tutelles :

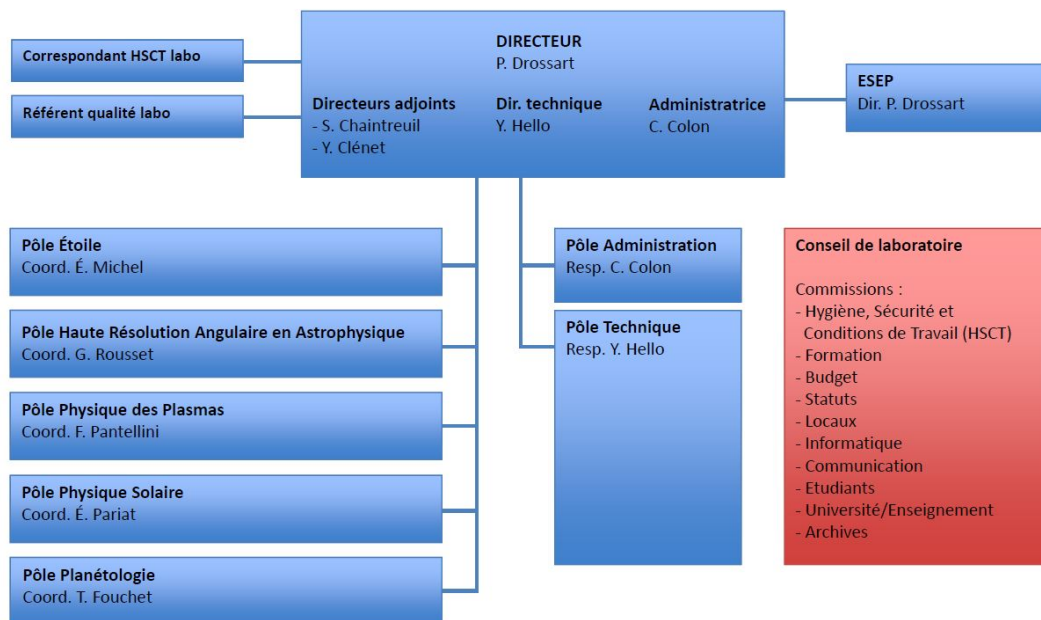
- L'Observatoire de Paris (tutelle d'hébergement)
- Le Centre National de la Recherche Scientifique (CNRS)
- L'Université Pierre et Marie Curie (Paris-6)
- L'Université Paris-Diderot (Paris-7)

Le LESIA a plusieurs pôles (cinq scientifiques, un administratif et un technique), comme le montre l'organigramme suivant :

Je suis rattaché au pôle technique du LESIA, lui-même sub-divisé en plusieurs services :

- Le C2L (Cellule Logistique du LESIA)
- Le CPR (Cellule Prévention des Risques)
- Le CQL (Cellule Qualité du LESIA)
- Le GEFL (Groupe d'Etude et de Fabrications du LESIA)
- Le GIGL (Groupe Informatique Générale du LESIA)
- Le MESPAL (Moyens d'Essais Salles Propres AIT/AIV du LESIA)
- Le SIGAL (Service Internet, Graphisme et Animations du LESIA)
- Le SII (Service Informatique Instrumentale)
- Le SIS (Service Informatique Scientifique)
- Le SPL (Soutien Projets du LESIA)
- Le SSL (Service Solaire du LESIA)

C'est pour tous ces différents services, ainsi que pour le directeur technique M. Yann HELLO, que je dois développer l'application de gestion de plan charge du personnel.



27/03/2015

FIGURE 3.1 – L'organigramme fonctionnel du LESIA (2015)



# Chapitre 4

## Structure de l'entreprise

Je travaille sur le site de Meudon. Il est délimité par l'avenue Marcellin BERTHELOT et la place Jules JANSSEN comme le montre le schéma suivant :

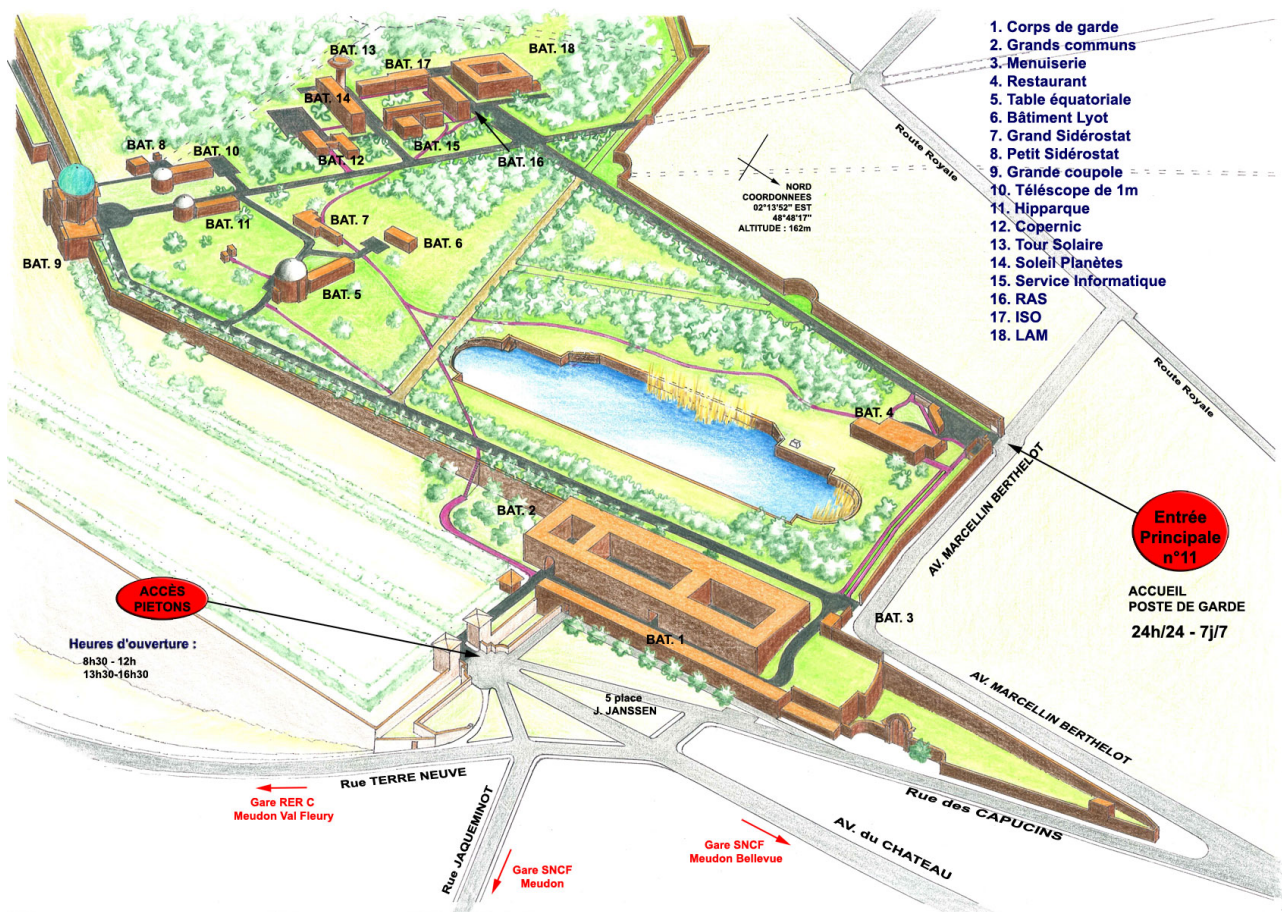


FIGURE 4.1 – Plan du site de Meudon

Chaque bâtiment à sa fonction, je suis dans le n° 16, principalement utilisé par le pôle physique des plasmas ainsi que le pôle technique et administratif du LESIA. Deuxième exemple, le bâtiment n° 14, est lui utilisé par le pôle solaire ainsi que le pôle étoiles. Mais aussi, le bâtiment n° 17 est utilisé par le pôle planétologie.



## Deuxième partie

### L'application





# Chapitre 5

## L'application demandée

L'agencement du laboratoire, vu précédemment (chap. 3 p. 17), est important pour comprendre le fonctionnement de l'application. Elle a pour but la gestion du plan de charge des personnels du pôle technique du LESIA. C'est-à-dire connaître la charge (en ETP<sup>1</sup>) d'un agent (en pourcentage) par semestre et par projet/service. La charge (en ETP) d'un projet/service en par semestre, ainsi que, la charge totale de tout le pôle technique du LESIA.

**Définition ETP :** Une charge de « n » ETP pendant une période donnée correspond à un travail qui nécessiterait l'affectation de « n » personnes à plein temps pendant toute la période considérée, pour le réaliser. Par exemple, une charge de 3 ETP pendant une semaine correspond à un travail réclamant trois personnes affectées à plein temps (typiquement, cinq jours) pendant la semaine. Dans le cadre de cette application, la période de temps est de l'ordre du semestre.

Actuellement, les plans de charge sont réalisés par les différents chefs de service et de projets dans différents fichiers excel. Puis ils sont tous centralisés par le directeur technique, Yann HELLO, qui vérifie l'intégralité des charges puis les fusionnes dans un unique fichier excel. De là, il en génère différents graphiques utiles lors de différents réunions ou assemblés. D'où l'utilité d'une application qui simplifierais toutes ces démarches. M. HELLO a donc rédigé les spécifications de l'application (cf. annexe - A)

Elle nécessite l'utilisation d'une base de données, ainsi qu'un moyen de la programmer, de l'utiliser, et de la mettre en forme. C'est ce que nous allons voir maintenant.

---

1. Équivalent temps plein



# Chapitre 6

## Les différents outils

### 1 La base de données

Mes 2-3 premières semaines ont été une phase d'apprentissage autant sur le point de vue technique, que sur le point de vue de l'organisation de l'entreprise. J'ai dû apprendre comment fonctionnait l'entreprise, et surtout quelles technologies elle utilisait. Une des premières technologies avec lequel j'ai travaillé fut la base de données.

*Mais tout d'abord, pourquoi une base de données ?*

- Il fallait gérer de nombreuses de données (charge de tous les agents, l'historique d'un agent au cours du temps, les différents statuts d'un agent ou d'un projet, la charge des projets/services, les différentes fonctions etc)
- Il fallait également que les informations "communiquent" entre-elles, par exemple, pour avoir la charge d'un projet, il faut sommer la charge de tous les agents qui sont affectés à ce projet. De même, pour les graphiques que nous allons voir plus tard, nous avons besoin d'utiliser cette base de données
- Et enfin, il fallait que n'importe qui (ayant les droits) puisse modifier ou ajouter des informations de manière simple (sans toucher au code)
- Il y avait déjà une base existante, de plus, ma tutrice avait déjà fait un schéma préparatoire (fig. [B.1](#)) pour les nouvelles tables à implémenter.

C'est pour toutes ces raisons, que la solution d'une base de données s'est imposée.

Une de mes missions a été alors de comprendre, d'interpréter ainsi que proposer des modifications à cette base de données.

## 1.1 MySQL



Pour des raisons de compatibilités avec la base de données actuelle, les nouvelles tables seront développées avec MySQL [3]. MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

A l'IUT, on nous a appris à utiliser un autre SGBDR, PostgreSQL, similaire dans les bases à MySQL. Toutefois, j'ai déjà utilisé MySQL pour le projet du semestre 3 à l'IUT, ce n'était donc pas quelque chose qu'il fallait que j'apprenne.

Pour avoir une vue d'ensemble de cette base de données, j'ai décidé d'installer sur mon ordinateur phpMyAdmin

## 1.2 phpMyAdmin



phpMyAdmin (abrégé : PMA) est un gestionnaire de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL [3]. Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base de données sous forme de fichier .sql et d'y transférer ses données, même sans connaître SQL.

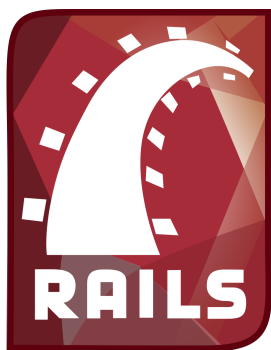
Les requêtes SQL restent possibles. Cette interface est vraiment très pratique dans le cadre de ce projet, car je pouvais visualiser rapidement, la présence ou l'absence de telles ou telles données. C'est un peu plus fastidieux (et surtout moins ergonomique) en utilisant uniquement la console MySQL.

Un screenshot de phpMyAdmin avec la base de données utilisé est en annexe (fig. C.1).

## 2 Le développement

Comme on l'a vu précédemment, le logiciel demande l'utilisation quasi permanente de la base de données. La méthode classique de création d'un site web, par des fichiers HTML, CSS, JavaScript, PHP étant fastidieuse, ma tutrice a décidé d'utiliser un framework web. Ce système permet de gérer les requêtes à la base simplement et d'une manière plus sûre (le framework s'occupant de cette partie).

### 2.1 Ruby on Rails



Ruby on Rails, également appelé RoR ou Rails, est un framework web libre écrit en Ruby. Il suit le motif de conception modèle-vue-contrôleur aussi nommé MVC.

Ce paradigme regroupe les fonctions nécessaires en trois catégories : un modèle (modèle de données, concrètement ce sont nos tables de la base de données), une vue (présentation, interface utilisateur, l'affichage du site web), un contrôleur (logique de contrôle, gestion des événements, synchronisation, c'est lui qui redirige les requêtes). En tant que framework, il propose une structure au programmeur qui lui permet de développer plus vite et plus intuitivement. Il ajoute aussi un grand niveau d'abstraction dans la programmation de l'application par un ensemble de fonctions de haut niveau qui lui offre ainsi l'économie d'écrire lui-même la plupart des routines obligatoires d'une application web.

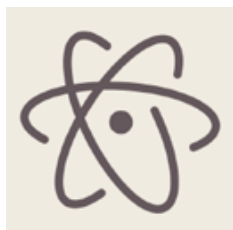
Concrètement :

- Le *modèle* représente le cœur (algorithmique) de l'application : traitements des données, interactions entre les éléments de la base de données, etc. Il décrit les données manipulées par l'application
- La *vue* est ce avec quoi l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (clic de souris, sélection d'un bouton radio, cochage d'une case, entrée de texte, de mouvements, de voix, etc.).
- Le *contrôleur* prend en charge la gestion des événements pour mettre à jour le modèle. Il reçoit tous les événements de la vue et enclenche les actions à effectuer.

L'inconvénient est qu'il fallait apprendre ce framework. De plus, il est basé sur un langage de programmation que je ne connaissais pas : Ruby. Il m'a fallu apprendre les bases algorithmiques de ce langage (qui ressemble un peu au Python), en plus d'apprendre le framework. C'était une étape assez fastidieuse, mais très enrichissante.

Pour pouvoir coder il faut un éditeur, et j'ai choisi Atom.

## 2.2 Atom



Atom est un éditeur de texte libre multiplate-forme (OS X, GNU/Linux, et Windows) développé par GitHub. Il supporte des plug-ins écrits en Node.js et implémente Git Control. La plupart des extensions sont sous licence libre et sont maintenues par la communauté. Atom est basé sur Chromium et écrit en CoffeeScript. Il est aussi utilisé en tant qu'IDE.

Il est très utile car il supporte de nombreux langages (dont Rails), permet l'auto-complétion (pratique lorsque l'on code pendant de nombreuses heures), et possède un plug-in pour l'intégration de SVN (colore en jaune les fichiers modifiés et en vert les fichiers non-ajoutés au dépôt).

## 2.3 SVN



Subversion (en abrégé svn) est un logiciel de gestion de versions, distribué sous licence Apache et BSD [3]. Il a été conçu pour remplacer CVS.

Subversion fonctionne donc sur le mode client-serveur. Soit, un serveur informatique centralisé avec le logiciel Subversion tournant en « tâche de fond », avec les fichiers constituant la référence (le « dépôt » ou « référentiel », ou « repository » en anglais). Puis un ou des postes clients sur lesquels se trouvent les fichiers recopiés depuis le serveur, éventuellement modifiés localement, pour ensuite, mettre à jour le serveur avec sa version.

À première vue, on peut se dire que cela ressemble à un Google Drive ou encore un Dropbox, mais c'est bien plus poussé que cela, et ce pour deux principales raisons :

- SVN garde une trace des modifications, on peut très facilement revenir en arrière de plusieurs modifications sur les fichiers
- SVN permet de travailler facilement à plusieurs sur un même fichier

J'ai donc travaillé avec SVN dans le cadre du développement du logiciel web, pour faciliter le codage avec ma tutrice et aussi pour avoir un historique de ses modifications (très pratique lors d'une erreur de code).

## 3 Les frameworks

Deux outils, ont été nécessaires pour poursuivre le projet, je vais les détailler, et pourquoi ont ils été nécessaires.

### 3.1 Bootstrap



Bootstrap est une collection d'outils utiles à la création de sites et d'applications web [2]. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript. Il permet de réduire considérablement le temps passé à l'élaboration du CSS et est conçu pour être responsive<sup>1</sup>.

Cette plate-forme a été conçue par deux développeurs faisant partie de la mouvance de développeurs qui gravitent autour de Twitter, Mark Otto et Jacob Thornton<sup>2</sup> et avait le nom de Twitter Blueprint en 2010

Dans le cadre de cette mission, il permettait de faire facilement un site web élégant. J'avais déjà utilisé Bootstrap pour un projet de l'IUT (en semestre 3), j'ai donc réutilisé mes connaissances pour cette application. Bootstrap est une bonne idée pour réduire le travail du design web au maximum tout en ayant un bon rendu visuel final. Il est important à noter, que j'avais libre court à mes idées pour le design (sous réserve de validation par ma tutrice).

### 3.2 Highcharts



Highcharts est une bibliothèque JavaScript qui permet de créer des graphiques interactifs [7]. Cette librairie possède des caractéristiques qui font d'elle un outil indispensable pour la création de graphiques.

Highcharts est simple d'utilisation, compatible avec tous navigateurs et responsive. C'est un outil modulable et interactif proposant différents types de graphiques basés sur une structure en HTML5 (fig. C.3).

C'est très utile dans notre application, car elle permet d'afficher visuellement, le plan de charge du service, d'un agent ou bien d'un projet. Un chef de service pourra instantanément voir au sein de son service ce qui va, de ce qui ne va pas. De manière totalement modulaire et esthétique. C'est la plus grosse attente de cette application.

---

1. S'adapte à n'importe quelle taille d'écran





# Chapitre 7

## La création

### 1 La base de données

Un de mes premiers travaux a été de concevoir, ou plutôt d'apporter des modifications à la base de données existante. En effet, j'ai dû m'appuyer sur la base de données du personnel, qui comporte plusieurs dizaine de tables, pour concevoir l'application. Dans les faits, sur toutes ces tables j'en ai utilisé moins d'une dizaine et principalement la table "agent".

#### 1.1 La base de données initiale

Pour réaliser cette application, je devais m'appuyer sur une base de données déjà existante, notamment une table "agent" qui comporte toutes les informations nécessaire à un agent (nom, prénom, date de naissance, genre, pôle, sa date d'arrivée, son statut, son échelon etc).

Cette table est également relié à d'autres : "Organisme", "Statut", "Corps", "BAP" qui complète les informations d'un agent.

A partir de là, ma tutrice à élaborer un schéma préliminaire de la base (fig. B.1) qui répond aux besoins de l'application. A savoir la gestion des charges d'un agent ou d'un responsable pour un service ou un projet pour un semestre donné, lister les différents services et projets, les compétences et les fonctions d'un agent. C'était donc assez simple pour moi de comprendre la logique et ce que l'on attendait de l'application.

#### 1.2 Les modifications

Toutefois, j'ai dû à plusieurs reprise, rectifier le schéma de la base (et par conséquent la base elle-même) pour intégrer quelques modifications (fig. B.2). Toutes les modifications de cette bases sont intervenues, tout au long du projet, pour aboutir à la version finale (fig. B.3).

## La table "Semestre"

La table semestre contient : un id, une date de début et une date de fin.

Au départ, je me suis posé la question de quelle granularité prendre pour la charge (chap. 5), on a conclu avec ma tutrice qu'une granularité de l'ordre du semestre était suffisant pour le laboratoire. La table comporte donc un id, et deux colonnes pour la date de début et date de fin.

Cette table a posé quelques problèmes, car on voulait absolument que son id (l'identifiant unique de chaque données), soit un string<sup>1</sup> (ex : "2016\_S1"), pour qu'il soit le plus clair possible lors de son utilisation avec d'autres tables et Rails. Par défaut, pour Rails, cet identifiant doit être un entier positif.

## Les responsables

Au départ, il y avait deux tables bien distinctes : "ResponsableProjet" et "ResponsableService". En fait la seule chose qui différenciait les tables était un champ "id\_projet" ou "id\_service". De plus, on s'est rendu compte qu'un projet était *toujours* associé à un ou plusieurs services.

J'ai donc réuni les deux tables en une seule : "responsables". Pour avoir un responsable service, il faut tout rentrer sauf le champ "projet\_id". Pour avoir un responsable projet, il faut simplement remplir tous les champs (fig. D.2).

Par la même occasion, le champ "type\_responsable" des tables originelles ("ResponsableProjet" et "ResponsableService") a été renommée en "responsabilite\_id" et pointe désormais vers une nouvelle table "responsabilites" qui liste ces responsabilités.

## Les agents

Dans la même idée que pour les responsables, les agents affectés à un projet ou un service était séparé dans deux tables ("AgentProjet" et "AgentService"), on s'était dit qu'il y allait avoir deux tables similaires et qu'une conception plus propre de la base de données serait de réunir les deux tables.

La table "affectations" a donc été créée.

## L'historique

Une question est venue à nous *"Si un agent, change de statut, de projet ou de service au cours de sa carrière qui pourra le savoir ? De même pour un projet qui évolue au cours du temps ?"*, la base de données dans l'état ne permettait pas d'avoir ces informations.

J'ai donc créé plusieurs tables : "AgentSemestres" pour avoir l'historique d'un agent tout du long de sa carrière, "ProjetSemestre" pour avoir aussi l'historique d'un projet. On peut

---

1. Chaîne de caractère

remarquer qu'il n'y a pas de "ServiceSemestre", en effet un service n'a que deux états : actif ou non utilisé. Et dans ce dernier cas il n'est pas nécessaire d'avoir l'historique de ce service.

## Statut d'un agent

Dernier cas, on s'est rendu compte que l'on n'avait pas pris en considération le statut d'un agent (permanent, non-permanent) dans la base de données. C'est chose fait, en utilisant la table "statutagent".

## 2 Développement avec Rails

Nous avons vu précédemment comment fonctionne un framework MVC (chap.6 §2.1, p.29), maintenant, je vais montrer au travers d'un exemple comment fonctionne Rails, et concrètement tout ce que j'ai pu faire. Rails met à disposition ce que l'on appelle des générateurs. Ils portent bien leur nom : ils génèrent du code. Du code à trous, prêt à accueillir le nôtre.

### 2.1 Modèle

Le modèle est la ressource que le va utiliser. Typiquement dans notre cas ce sont les tables, prenons un exemple simple "Projet", tout d'abord il faut créer cette table dans la console (un projet comporte un nom, un ou plusieurs poles et un statut) :

```
Rails generate model Projet nom:string pole_id:integer statut_id:integer
```

Cela va créer le modèle "Projet" avec lequel on va paramétrer les associations entre tables, les conditions de validations ainsi que la création de méthode de classe ou d'objet. Cette commande va aussi créer une migration. Une migration est fichier qui (une fois validé) va crée la table dans la base de données. On lance la migration avec cette commande :

```
rake db:migrate
```

Voilà, la table est créée, il faut maintenant la paramétrer.

### Les associations

Ensuite, il faut personnaliser le modèle (notamment pour les liaisons entre différentes tables). Pour cela, Rails utilise des associations : *has\_many* ou *has\_one*, désigne une référence à une autre table (de 1 ou plusieurs) et *belongs\_to* désigne quant à lui les autres tables qui utilisent cette table. Dans notre cas nous avons quelque chose de ce genre :

```
belongs_to :pole # Un projet appartient à un pole
belongs_to :statut_projet
has_many :responsables # Un projet à plusieurs responsables
has_many :domaine_projets # Un projet à plusieurs domaines
has_many :domaines, through: :domaine_projets
has_many :affectations
has_many :projet_semestres
```

## Les validations

Il faut également faire attention à ne pas rentrer n'importe quoi dans la base de données. Pour cela on utilise les validations, on peut faire énormément de choses avec les validations parmi lesquels :

- Vérifier la présence d'un champ
- Vérifier que la valeur d'un champ n'existe pas déjà dans la base de données
- Vérifier que la valeur soit valide
- Vérifier l'unicité d'une paire (par exemple : Il ne peut y avoir qu'un seul statut d'un agent pour un semestre donné)

Pour le cadre d'un projet cela donne :

```
# On vérifie donc la présence du nom
validates_presence_of :nom, message: ": non renseigné ou invalide"
# On vérifie si le nom est unique
validates_uniqueness_of :nom, message: ": \"%{value}\" existe déjà"
# On vérifie que l'on a bien sélectionné un pôle
validates :pole_id, inclusion: {in: Pole.all.map {|p| p.id},
  message: ": non renseigné ou invalide"}
# On vérifie qu'il y a bien au moins un domaine coché
validates_each :domaine_ids do |record, attr, value|
  record.errors.add(attr, ": il faut choisir au moins un domaine") if value.empty?
end
# On vérifie que l'on a bien sélectionné un statut pour le projet
validates :statut_id, inclusion: {in: StatutProjet.all.map {|sp| sp.id},
  message: ": non renseigné ou invalide"}
```

J'ai effectué dans cette partie 526 lignes de code pour 26 modèles différents.

## Les méthodes

Une partie optionnelle est de définir des méthodes de classe ou d'objet, pour avoir des informations spécifiques

Par exemple, cette méthode de classe, donne la charge pour un statut donnée sur une affectation (et on voit que l'on utilise ici du SQL) :

```

def Affectation.charge_par_statut(statut)
  Affectation.joins("LEFT OUTER JOIN agent_semestres
ON affectations.agent_id = agent_semestres.agent_id
  AND affectations.semestre_id = agent_semestres.semestre_id
LEFT OUTER JOIN statutagent
ON agent_semestres.statut_id = statutagent.IDstatut").
  where("statutagent.permanent" => statut).
  order("affectations.semestre_id").
  group("affectations.semestre_id").sum(:charge)
end

```

## 2.2 Le contrôleur

C'est lui qui va gérer les actions (quoi faire) :

- Index : Liste tous les ressources du contrôleur
- Show : Affiche une ressource particulière (définie par son ID)
- New : Débute la création d'une ressource (typiquement, cela amène vers le formulaire de création)
- Create : Valide la création d'un ressource (la base de données est alors modifiée)
- Edit : Débute une édition d'une ressource (typiquement, cela amène vers le formulaire d'édition)
- Update : Valide l'édition de la ressource (la base de données est alors modifiée)
- Destroy : Supprime une ressource.

Voici un exemple du contrôleur du projet :

```

def index
  @projets = Projet.all.order(:nom) // On liste tous les projets trié par leurs noms
end

def show
  @projet = Projet.find(params[:id]) // On récupère un projet par son id
end

def new
  @projet = Projet.new // On crée un nouveau projet
end

def create
  @projet = Projet.new(projet_params) // On instancie un nouveau projet avec des
  paramètres autorisé.
  if @projet.save // On sauvegarde et si le projet est bien sauvegarder alors
    redirect_to @projet // On redirige vers la vue du projet
  else
    render 'edit' // Sinon on remet le formulaire
  end
end
end

```

```

def edit
  @projet = Projet.find(params[:id]) // On édite un projet spécifique
  // (par le biais de son id)
end

def update
  @projet = Projet.find(params[:id]) // On sélectionne le projet par son ID
  if @projet.update(projet_params) // Si le projet comporte des paramètres autorisés
    redirect_to @projet // On redirige vers la vue du projet
  else
    render 'edit' // sinon on remet le formulaire
  end
end

def destroy
  @projet = Projet.find(params[:id]) // On sélectionne un projet par son ID
  @projet.destroy // On le supprime
  redirect_to projets_path // Et on redirige vers la liste de tous les projets.
end

private
def projet_params # Autorise les paramètres à modifier pour un projet
  params.require(:projet).permit(
    :nom,
    :pole_id,
    :statut_id,
    domaine_ids: [])
end
end

```

J'ai effectué dans cette partie 557 lignes de code pour 13 contrôleurs différents.

## 2.3 Les routes

Rails fonctionne sur un principe de route. Il sait grâce à l'URL plus la méthode envoyée (principalement post/get/patch/put/delete) quoi faire.

Par exemple : GET http://mon-site/service/5

Va renvoyer sur le contrôleur service avec la méthode show et l'ID 5.

## 2.4 Les vues

Une fois que le modèle et le contrôleur sont faits, on peut passer aux vues. La vue c'est tout simplement la page web, du HTML mélangé avec un peu de Rails (délimité par <%= %>) et éventuellement du Javascript.

Comme vu précédemment, en fonction de l'URL (et de la méthode), nous allons avoir une vue différente.

Voici une partie de la vue "index" pour le projets :

```
<h2 class="sub-header">Liste des projets</h2>

<!-- Bouton pour créer un nouveau projet -->
<%= link_to "Nouveau Projet", new_projet_path, class: "btn btn-success" %>

<!-- Un simple tableau pour lister tous les projets -->
<div class="table-responsive espace_tableau">
  <table class="table table-striped">
    <thead>
      <tr>
        <th>Nom du projet</th>
        <th>Domaine</th>
        <th>Statut</th>
        <th colspan="2">Actions</th>
      </tr>
    </thead>
    <tbody>
      <% @projets.each do |projet| %>
      <tr>
        <td>
          <%= link_to projet.nom, projet_path(projet), class: "btn btn-link" %>
        </td>
        <td>
          <%= (projet.domaines.map {|d| d.nom}).join(' & ') %>
        </td>
        <td>
          <%= projet.statut_projet.nil? ? '' : projet.statut_projet.nom %>
        </td>
        <td>
          <%= link_to 'Modifier', edit_projet_path(projet), class: "btn btn-link" %>
        </td>
        <td>
          <%= link_to 'Supprimer', projet_path(projet), class: "btn btn-link",
            method: :delete, data: { confirm: 'Êtes-vous sûr ?' }%>
        </td>
      </tr>
      <% end %>
    </tbody>
  </table>
</div>
```

C'est bien ici, où j'ai passé beaucoup de temps à développer : 3117 lignes de codes pour 79 vues différentes.

### 3 Le site web

J'ai décidé de décomposer le site web en trois parties : les agents, les projets et les services. Cela me semblait logique au vu des besoins de l'application.

#### 3.1 Agent

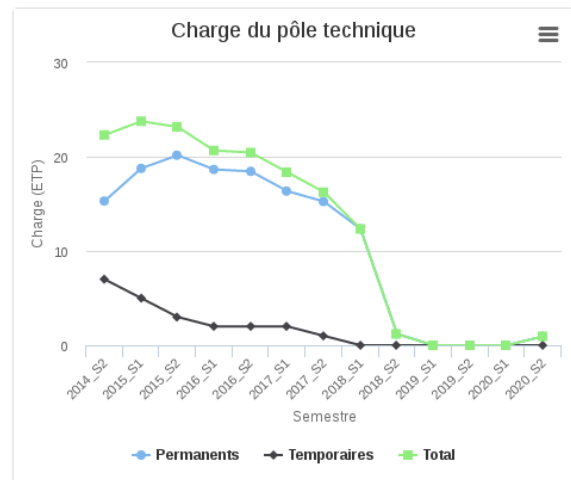
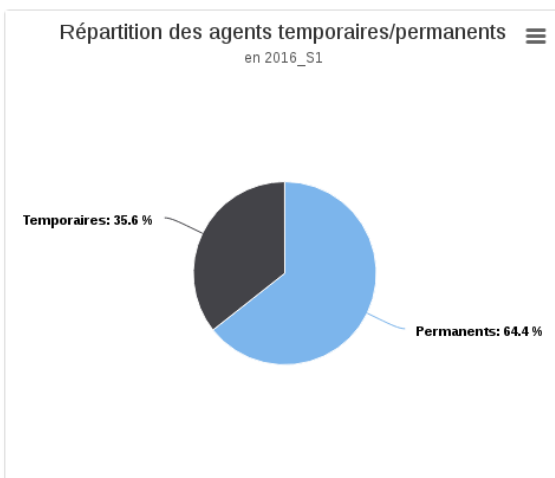
Dans cette partie, j'ai :

Agent ▾

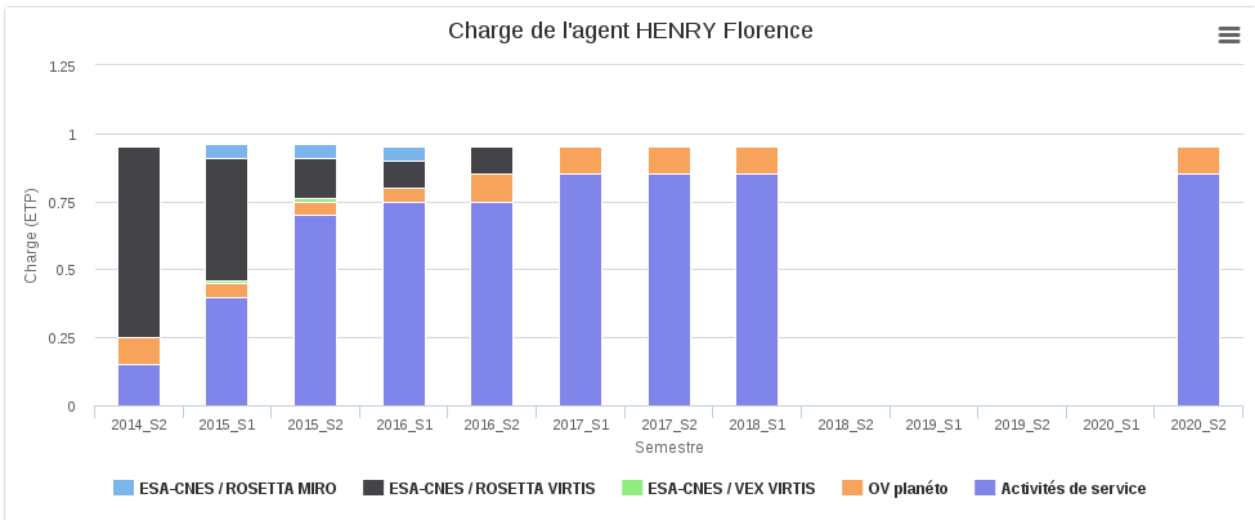
- Liste des agents
- Affecter un agent à un projet/service
- Liste des compétences
- Nouvelle compétence
- Liste des responsabilités
- Nouvelle responsabilité

- Lister tous les agents du pôle technique
- Par le biais de cette liste, on peut, accéder au plan de charge d'un agent (fig. D.1)
- Affecter un agent à un projet ou à un service (fig. D.2)
- On peut lister toutes les compétences possibles d'un agent et attribuer des compétences à tel ou tel agent, et bien sûr en créer (fig. D.3)
- Et enfin, on peut faire de même avec les responsabilités (fig. D.4)

De même j'ai intégré les différents graphiques : Répartition des agents permanents et temporaires, la charge totale du pôle technique, la charge d'un agent.







### 3.2 Service

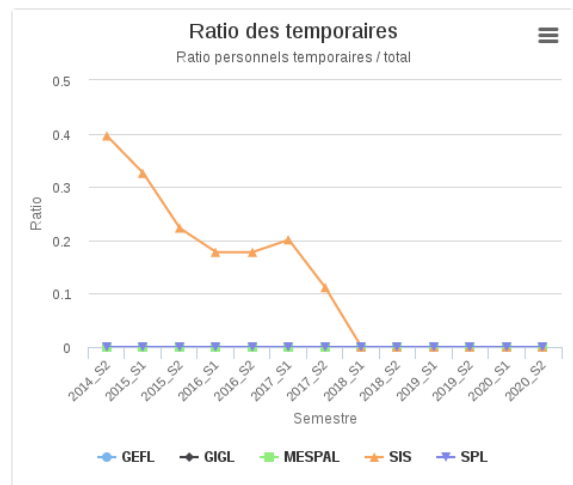
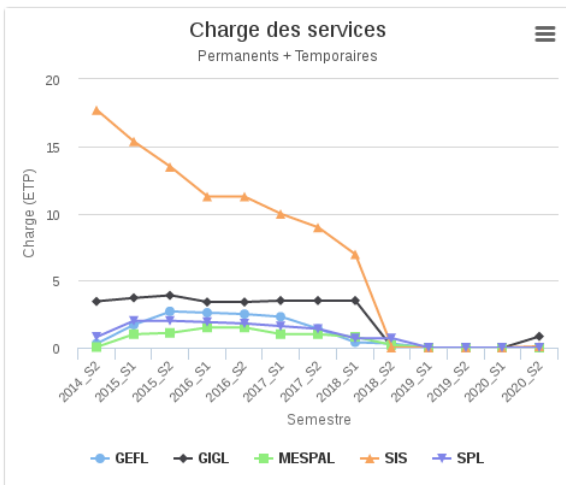
Pour un service c'est plus succinct :

Service ▾

- Liste des services
- Nouveau Service
- Affecter un responsable à un service

- Lister tous les services (fig. D.5)
- Créer un nouveau service
- Affecter à un responsable à un service

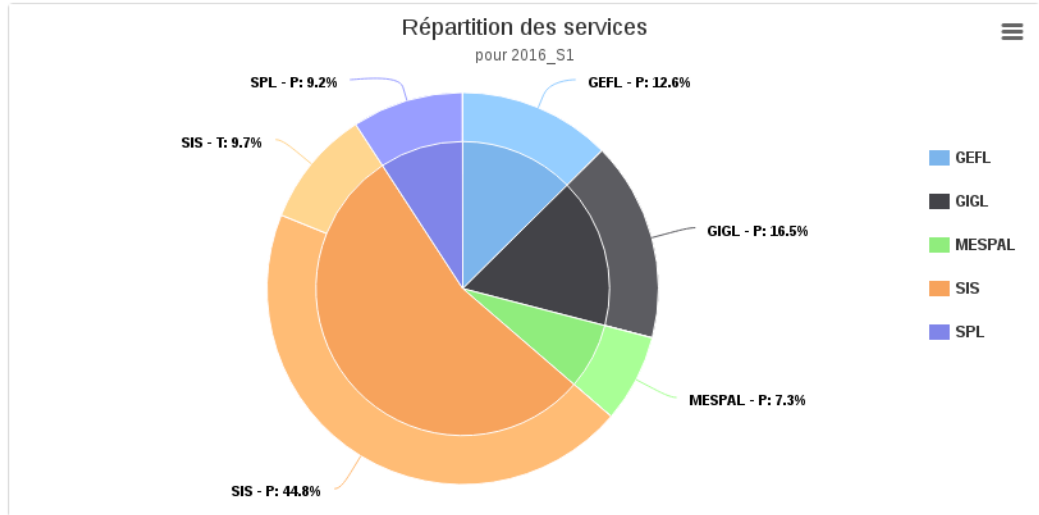
Bien sûr des graphiques sont également là : La charges des services pour les permanents ou temporaires ou le total. Le ratio des temporaires par service. La répartition des personelles permanents et temporaire. Et enfin la répartition des charges par services (et pour un semestre donné).



Choisir un semestre

2016\_S1

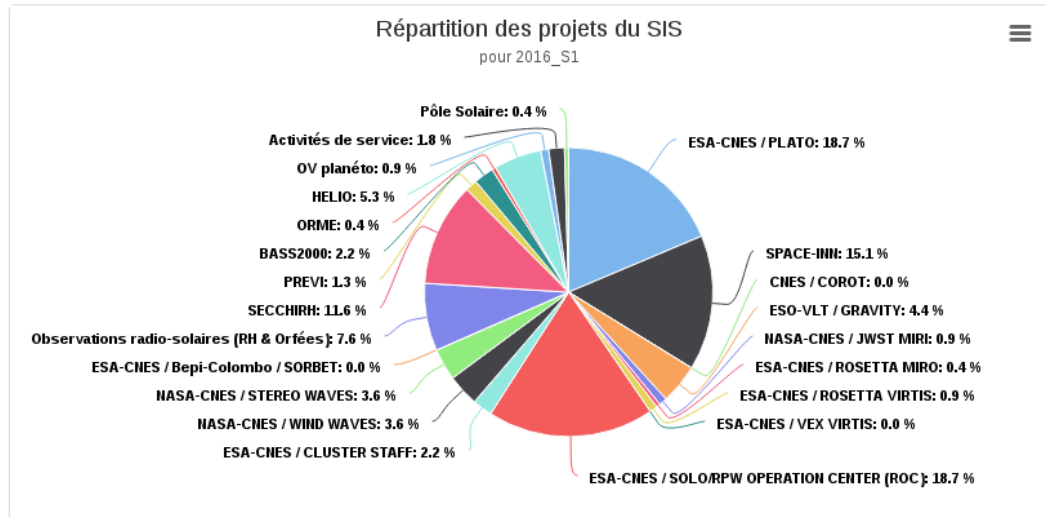
Afficher



Choisir un semestre

2016\_S1

Afficher



### 3.3 Projet

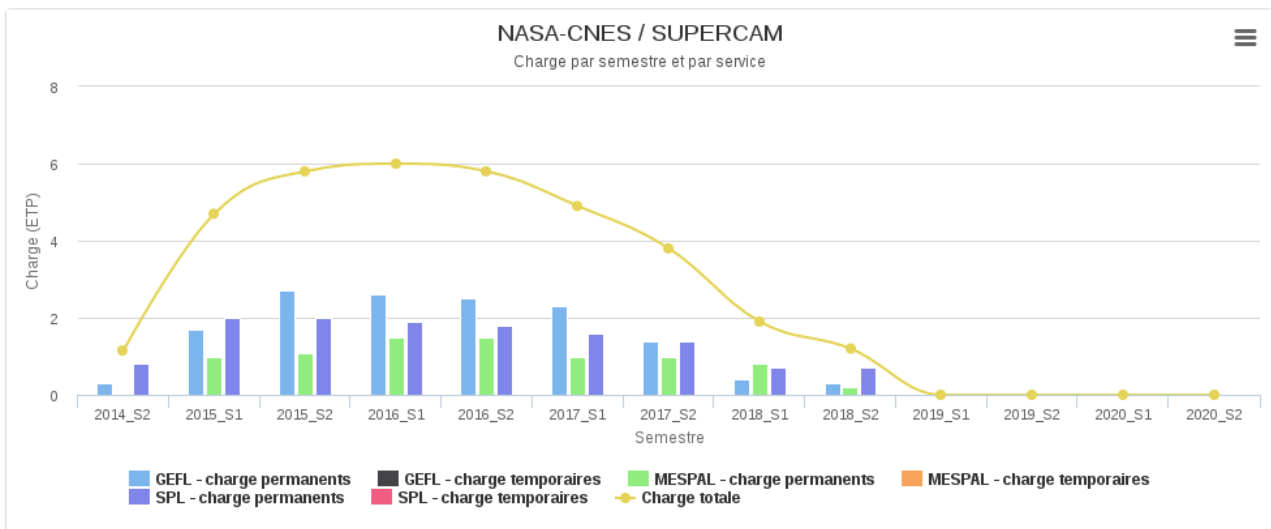
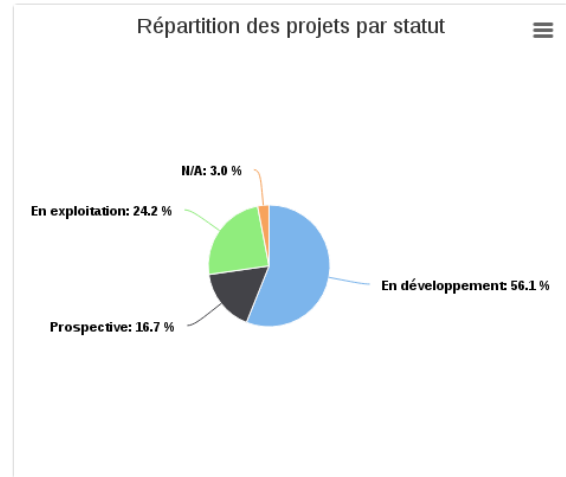
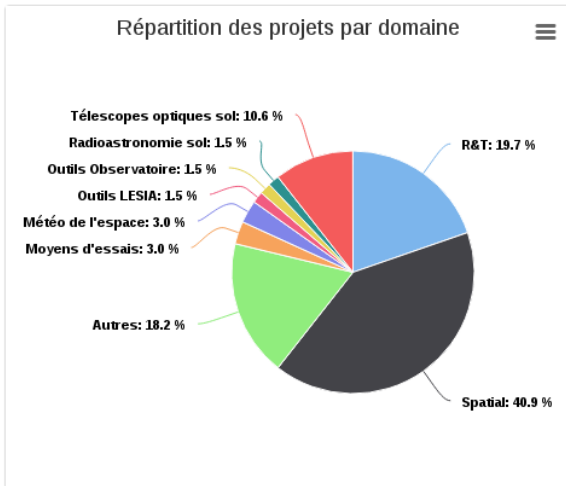
Projet ▾

- Liste des projets
- Nouveau Projet
- Affecter un responsable projet
- Listes des domaines
- Nouveau Domaine

Le menu du projet est sur la même structure qu'un service c'est-à-dire :

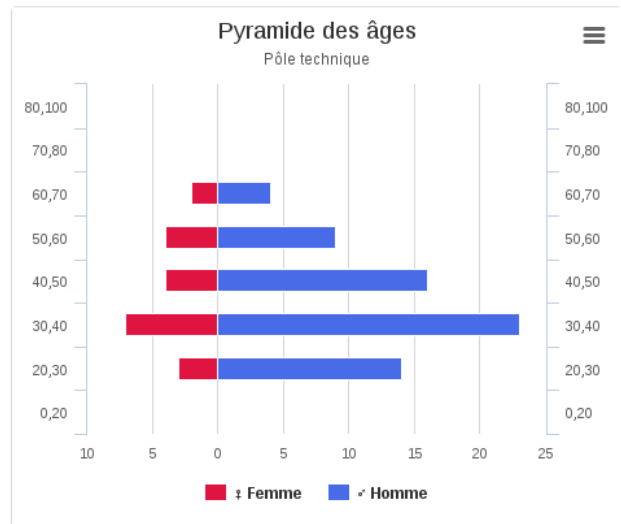
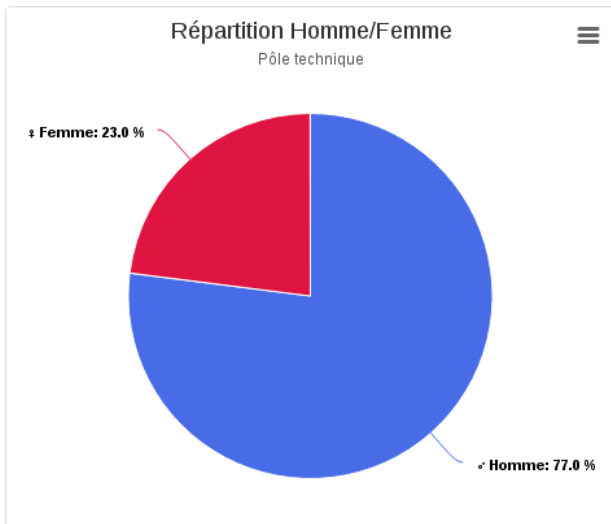
- Lister tous les projets (fig. D.6)
- Créer un nouveau projet (fig. D.7)
- Affecter à un responsable à un projet
- Mais aussi, créer un nouveau domaine et les lister (un projet pouvant avoir plusieurs domaines)

Et toujours des graphiques : la répartition des projets par domaine, par statut, et la charge par semestre et par service d'un projet.



### 3.4 Statistique

Et enfin, une dernière partie, pour les statistiques du laboratoire. Le ratio homme/femme, ainsi que la pyramide des âges.



## Troisième partie

### Expérience professionnelle



# Chapitre 8

## Mon expérience

### 1 Préambule

Ce qui m'aura marqué durant ce stage, c'est la différence entre le monde du travail et celui des études. En effet, je suis dans une filière que je trouve très aseptisée, sans réellement de contact humain. Et j'ai découvert, non sans joie, un environnement avec beaucoup de contact, réglementé certes, mais où l'Humain possède une place importante. Même dans mon domaine, l'informatique, on cherche à savoir précisément ce qu'attend le client, ce qu'il veut, et ce que l'on peut faire. Et il est difficile d'apprendre ça en cours, bien que les cours de communication et de PPP<sup>1</sup> tende à nous introduire le sujet, cela vient avec l'expérience et l'habitude.

Mais c'est justement ce côté humain et singulier qui m'a plu au LESIA. Ce qui m'a également intéressé est le côté polyvalent du stage. Comme je l'ai dit au début de mon rapport, je voulais un poste qui me permettait d'en apprendre d'avantage sur les différents domaines du développement web. J'ai pu atteindre cet objectif grâce au LESIA. En effet, ce laboratoire, est une petite structure de 248 personnes. Il est cependant, le deuxième plus gros laboratoire en astrophysique de France. Ainsi, j'ai pu appréhender différentes missions tels que la conception de la base de données, l'interaction avec les clients directs, le développement du site web, l'implémentation de différents framework, et l'apprentissage d'un nouveau langage.

Mais avant tout, ce stage m'a permis de découvrir le monde du travail. J'appréhendais un peu le coté développement et le fait de rester assis derrière un ordinateur toute la journée. Mais finalement les heures passaient plus vite qu'en cours car durant ce stage j'avais de nombreuses tâches à réaliser ce qui fait que mon temps était occupé et je ne voyais pas la journée passer.

### 2 L'humain

De plus, j'ai pu découvrir des nouveaux aspects de l'entreprise que je ne soupçonnais pas. Par exemple, la notion d'ETP était pour moi inconnue. Pour développer cette application, j'ai dû me mettre dans la peau d'un chef de service, qui doit gérer son service. J'ai donc pour cela, beaucoup discuté avec ma tutrice pour savoir comment organiser au mieux l'application.

---

1. Projet Personnel Professionnel

Étant presque le seul informaticien de tout le bâtiment, j'ai pu m'imprégner d'un autre domaine, l'astronomie. Ici, beaucoup de personnes travaillent pour la recherche en physique des plasmas. C'était pour moi une occasion inouï de les questionner sur plein de détails techniques ou des phénomènes physique qui m'intriguaient.

Dans mon bureau, se trouvaient aussi plusieurs thésards dont une, M<sup>me</sup> Léa GRITON, représentante des étudiants. Elle ma permis de voir un peu tout ce que sa fonction imposait (beaucoup de responsabilité) ainsi que d'une manière plus globale de comprendre un peu mieux le fonctionnement du laboratoire.

### 3 La technique

J'ai appris énormément au sein de ce laboratoire, bien plus que je ne l'aurais imaginé. J'ai notamment appris l'utilisation d'un nouveau langage Ruby on Rails (chap.6 §2.1, p.29), qui fut une véritable révélation pour moi, dans le cadre d'un site web utilisant beaucoup une base de données. C'est pratiquement obligatoire à mon sens d'utiliser un framework MCV.

J'ai aussi appris (ou réappris) l'utilisation de nombreux framework tels que HighCharts ou bien BootStrap. Ce dernier, est réellement très puissant pour faire facilement de beaux sites web, sans pour autant être un web designer. Plus précisément, c'est toute une réflexion sur comment intégrer telle ou telle chose que j'ai appris.

D'une manière plus générale, j'ai appris à travailler en profondeur avec les documentations de chaque framework (le plus souvent en anglais), ce qui m'a rendu beaucoup plus autonome qu'à mes débuts.

Et enfin, avec les conseils de ma tutrice (et de mon tuteur), j'ai appris à utiliser L<sup>A</sup>T<sub>E</sub>X, pour l'élaboration de ce rapport ainsi que des bulletins d'avancement, c'est vraiment très pratique pour la rédaction.

De plus, je sais que ces connaissances me serviront pour mon avenir professionnel car il faudra un jour choisir dans quel domaine travailler. Et, je trouve, qu'il est important d'avoir un panel large de compétences dès le début.



# Chapitre 9

## Conclusion

De façon générale, ce stage m'a permis d'appréhender le monde du travail et de cerner les attentes des différents postes du service informatique. Grâce à cette expérience professionnelle, j'ai pu confirmer mon souhait de continuer mes études dans ce domaine et approfondir davantage mes connaissances. Néanmoins je suis désormais partagé entre deux voies :

- D'un côté, j'ai réalisé que la pratique contribuait énormément à l'assimilation des connaissances. En effet, cela permet d'illustrer et de concrétiser les notions acquises en cours. Une licence professionnelle en apprentissage ou une licence générale suivit d'un master professionnel est donc idéale pour cette situation.
- Mais d'un autre côté, après avoir côtoyé de nombreux chercheurs et ingénieurs très compétents, cela me donne envie aussi de renforcer mon côté théorique et de faire une école d'ingénieur.

Ces onze semaines passées au sein du LESIA m'ont beaucoup apporté et pas seulement au niveau informatique. En effet, c'est tout un monde qui s'est ouvert à moi et j'ai pris un réel plaisir à le découvrir. De par son activité particulière, j'ai pu parfaire ma culture générale sur l'astronomie, la physique des plasmas et l'astrophysique.

Nous arrivons à la fin de ce stage. C'est avec un certain regret que je regarde une dernière fois ce magnifique parc de Meudon et ses nombreux télescopes. Avant de partir, je jette un ultime regard au château de Meudon, où l'on peut facilement s'imaginer les gens aller et venir au cours des siècles. Plus qu'une expérience professionnelle, le LESIA m'aura apporté une expérience humaine.



# Bibliographie

- [1] Api de ruby on rails, 2016. [api.rubyonrails.org](http://api.rubyonrails.org).
- [2] Bootstrap, 2016. [getbootstrap.com](http://getbootstrap.com). 3.1
- [3] Encyclopédie libre, 2016. [fr.wikipedia.org](http://fr.wikipedia.org). 1.1, 1.2, 2.3
- [4] Google, 2016. [www.google.fr](http://www.google.fr).
- [5] Guide pour ruby, 2016. [guides.rubyonrails.org](http://guides.rubyonrails.org).
- [6] Guide pour ruby on rails, 2016. [www.techotopia.com/index.php/Ruby\\_Essentials](http://www.techotopia.com/index.php/Ruby_Essentials).
- [7] Highcharts, 2016. [www.highcharts.com](http://www.highcharts.com). 3.2
- [8] Site web de l'observatoire, 2016. [www.obspm.fr](http://www.obspm.fr). 2
- [9] Site web du lesia, 2016. [lesia.obspm.fr](http://lesia.obspm.fr). 1
- [10] Stackoverflow, 2016. [stackoverflow.com](http://stackoverflow.com).
- [11] Benjamin Bayart. *Joli manuel pour Latex*. ESIEE, Dec 1995.
- [12] Yann Hello. Spécifications pour un outil de gestion des it. Technical report, LESIA, Jan 2016. A
- [13] Matthieu Nebra. Open class rooms, 2016. [www.openclassrooms.com](http://www.openclassrooms.com).



# Quatrième partie

## Annexes



# Annexe A

## Documentation

Cette partie comporte les spécifications pour l'outil de gestion des IT [\[12\]](#)

## Spécifications pour un outil de gestion des IT

### RÉSUMÉ

Ce document expose les caractéristiques de l'outil informatique qui sera utilisé pour le calcul du plan de charge IT du LESIA

	Nom et Fonction	Date et signature
Préparé	Y. HELLO - DT	
Vérifié	Directeur technique du LESIA - Référent qualité du LESIA	
Approuvé	Directeur du LESIA	



Revision	Date	Pages affectée	Description des changements
Draft 0	4/10/2015	Tout	Première version
Draft 1	29/10/2015	Titre	Changement du titre
		Chapitre 1	Clarification
		Chapitre 2	Ajout du besoin "métiers"
		Chapitre 3	Suppression de la solution table "echelons de temps" et précision sur la période de 4 ans. Demande de champs supplémentaires sur la base existante.
		Chapitre 3.1	Suppression des solutions. Ajout des adjoints. Précision sur les fonctions des informaticiens. Ajout de la possibilité de se référer aux métiers officiels du CNRS ou MEN. Transfert des demandes d'ajouts de champ sur la base existante au chapitre precedent.
		Chapitre 3.2	Suppressions des solutions
		Chapitre 3.3	Suppression du chapitre
		Chapitre "Autorisations"	Réécriture
		Chapitre "Sorties"	Ajouts des listes de responsables et adjoints
Draft 2	1/12/2015	Titre page de garde	Mise en conformité avec titre entête
		Chapitre 3	Ajout des formations
1	14/1/2016	Chapitre 3	Historique des champs : statuts, corps, échelons et BAP Champ "responsable d'entretien" et "emploi type" Simplification de la description du besoin de formation de l'unité
		Chapitre 3.1	Ajout d'un champ compétences
		Chapitre 3.2	Enlèvement de "autres" dans les domaines Ajout "de la mise à jour" pour la fiche projet Suppression du champ date du plan de charge. L'information de la date de mise à jour du plan de charge se retrouve dans les sorties de la base Historique des statuts projets
		Chapitre "sorties"	Ajout d'un deuxième temps pour la definition des besoins
		Ajout d'un chapitre avant "autorisation" qui s'appelle "formation et competences"	

## SOMMAIRE

<b>1</b>	<b>BUT DU DOCUMENT</b>	<b>4</b>
<b>2</b>	<b>INTRODUCTION</b>	<b>4</b>
<b>3</b>	<b>BESOINS</b>	<b>4</b>
3.1	SERVICES	5
3.2	PROJETS	6
3.3	FORMATIONS ET COMPETENCES	6
3.4	AUTORISATIONS	7
3.5	SORTIES	7

 <p>Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique</p>	<h1>Organisation du LESIA</h1> <h2>Pôle technique</h2> <hr/> <p>Spécifications pour un outil de gestion des IT</p>	<p>Ref : Version :1</p> <p>Date : 14/1/2016 Page 4 sur 7</p>
---	--	--

## 1 But du document

Ce document expose les caractéristiques de l'outil informatique qui sera utilisé pour aider à gérer les IT du LESIA (notamment le calcul du plan de charge).

## 2 Introduction

Le LESIA compte environ 60 projets sur lesquels les 70 IT du laboratoire sont impliqués à des degrés divers en fonction du temps. La direction du LESIA a besoin, le plus souvent possible (une granularité au semestre semble raisonnable par rapport à la durée moyenne des projets du LESIA de quelques années), d'avoir un état de l'activité de ses agents IT afin d'établir une stratégie d'implication technique pour les objectifs scientifiques du laboratoire.

Afin de prévoir l'évolution des métiers en fonction des projets qui se présentent et définir une stratégie de besoin en personnel, il est nécessaire de faire un état des lieux des spécialités en activité au LESIA.

Une base de donnée du personnel du LESIA est déjà disponible qu'il faudrait étoffer pour qu'elle comporte les informations utiles au calcul de ce plan de charge.

## 3 Besoins

L'outil envisagé serait donc une extension de la base de donnée personnels existant déjà au LESIA.

Cette base de donnée comporte les champs suivants :

- Nom de famille ;
- Prénom ;
- Date de naissance ;
- Genre ;
- Organisme (MEN, CNRS,...) ;
- Statut (Chercheur, ITA,...) ;
- Corps (DR, CR, IR, IE,...) ;
- Echelon ;
- BAP ;
- Pôle ;
- HDR ;
- Date HDR ;
- Quotité ;
- Date d'arrivée ;
- Date de départ ;
- Commentaires.

 <p>Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique</p>	<h1>Organisation du LESIA</h1> <h2>Pôle technique</h2> <hr/> <p>Spécifications pour un outil de gestion des IT</p>	<p>Ref : Version :1</p> <p>Date : 14/1/2016 Page 5 sur 7</p>
---	--	--

Pour les IT, on a besoin d'avoir, en plus, d'autres informations pour avoir les indicateurs de charge du pôle technique.

Les indicateurs qu'on rajoutera à cette base dépendront du temps, un découpage en semestres avec une prévision sur une période de 4 ans semble raisonnable.

Il faudrait garder un historique des champs : statuts, corps, échelons et BAP.

On devra ajouter un champ de la base existante du LESIA pour préciser le responsable hiérarchique, le responsable d'entretien et l'emploi-type. La donnée sera rentrée manuellement et non pas automatiquement calculée à partir des quotités (liste de choix à partir des agents du LESIA).

On devra ajouter aussi l'information si l'agent est un prestataire et quelle est sa société.

Il serait intéressant de pouvoir rentrer des agents virtuels correspondants à des besoins déclarés par des services mais qui ne sont pas financés (financement futur par CDD, prestation ou poste fixe).

Cette base devra permettre de pouvoir gérer le plan de formation de l'unité.

### 3.1 Services

Chaque agent est affecté à un ou plusieurs services à des quotités variables dans le temps. Si un agent est affecté à plusieurs services, il faudra définir le service prioritaire pour gérer les conflits potentiels. La responsabilité hiérarchique de l'agent sera donnée au responsable du service pour lequel l'agent a la quotité la plus forte.

La nouvelle base de donnée devra donc comporter la liste des services, leurs responsables et adjoints provenant de la liste des IT du LESIA.

Il serait intéressant pour le LESIA (estimation des métiers les plus sollicités) d'avoir pour chaque agent sa fonction dans le service correspondant aux compétences des IT du LESIA. Un agent peut avoir plusieurs fonctions au sein d'un service (ex : chef de projet et opticien). Une liste des fonctions du LESIA pourrait être la suivante (noms de la table « fonctions ») (à compléter et voir si on se réfère au portail des métiers officiel CNRS ou MEN) :

- direction ;
- scientifique ;
- chef de projet ;
- système ;
- qualité ;
- mécanicien BE ;
- mécanicien atelier ;
- mécanicien AIT/AIV ;
- opticien ;
- opticien AIT/AIV ;
- ASR (Administration, Système et Réseaux)
- Développement logiciel

 <p>Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique</p>	<h1>Organisation du LESIA</h1> <h2>Pôle technique</h2> <hr/> <h3>Spécifications pour un outil de gestion des IT</h3>	<p>Ref : Version :1</p> <p>Date : 14/1/2016 Page 6 sur 7</p>
---	--	--

- Informaticien chef de projet / architecte
- Informaticien développeur
- Informaticien spécificateur / valideur
- Informaticien ingénieur qualité
- électronicien développement ;
- électronicien AIT/AIV ;
- électronicien réalisation ;
- instrumentaliste ;
- logisticien ;
- graphiste ;
- documentaliste.

La charge service pour chaque agent en fonction du temps devra pouvoir être rentrée dans la base. Cette charge sera la charge service pure, elle ne tient pas compte de la charge projet (qui sera comptée dans la charge projets).

On devra pouvoir aussi rentrer les compétences de l'agent dans son service par une liste qui grandira automatiquement en fonction des entrées.

### 3.2 Projets

Le LESIA mets en place une soixantaine de projets instrumentaux qui utilisent les ressources IT du pôle technique (organisé en services). La nouvelle base de donnée devra comporter la liste des projets instrumentaux. Elle devra être renseignée pour chaque projet par :

- Le pôle scientifique de rattachement
- le domaine (outils LESIA, spatial, R&T, télescopes optiques sol, radioastronomie sol, météo de l'espace, outils OP, moyens d'essais, outils SII) ;
- le statut (prospective, en développement, en exploitation, fini) ;
- le responsable scientifique ;
- le responsable technique ;
- la date de la mise à jour de la fiche projet ;
- la date de la dernière RSP.

Chaque agent impliqué dans les projets aura une charge et une fonction (et donc un service) qui pourront varier avec le temps à renseigner dans la base. Un agent pourra être impliqué dans un projet avec des fonctions différentes (et donc des services différents).

Il faut garder un historique des statuts projets.

### 3.3 Formations et compétences

Pour chaque agent, on doit pouvoir avoir les informations suivantes :

- la date de l'entretien annuel
- les compétences (à partir d'une table qui s'auto-complètera avec les entrées)

 <p>Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique</p>	<h1>Organisation du LESIA</h1> <h2>Pôle technique</h2> <hr/> <p>Spécifications pour un outil de gestion des IT</p>	<p>Ref : Version :1</p> <p>Date : 14/1/2016 Page 7 sur 7</p>
---	--	--

- le besoin en formation

### 3.4 Autorisations

L'écriture et la lecture dans les tables devra être limitée à un nombre restreint de personnes principalement la direction et celles qui ont déjà été accordées pour la base existante LESIA. Chaque chef de service pourra avoir accès à l'écriture de la charge de ses agents.

La lecture des renseignements de charge et métiers pour tout le LESIA pourra se faire en plus par les chefs de services.

### 3.5 Sorties

Dans un premier temps, les informations qu'il faudra sortir de cette base devront être :

- la charge d'un agent en fonction du temps par projet ou par service ;
- la charge en ETP d'un projet en fonction du temps (séparable en permanents et temporaires CDD ou prestataires) ;
- les fonctions professionnelles actives d'un projet en fonction du temps ;
- la répartition des ETP par services (séparable en permanents et temporaires CDD ou prestataires) ;
- les fonctions professionnelles d'un service quantifiées en ETP et en fonction du temps (séparable en permanents et temporaires CDD ou prestataires) ;
- la charge totale du pôle technique en ETP en fonction du temps (séparable en permanents et temporaires CDD ou prestataires) ;
- les fonctions professionnelles du pôle technique quantifiées en ETP et en fonction du temps (séparable en permanents et temporaires CDD ou prestataires)
- la liste des chefs de services
- la liste des adjoints aux chefs de services
- la liste des responsables scientifiques
- la liste des chefs de projets
- l'état de fraîcheur de la mise à jour des données ou d'un service

Les sorties étant indépendantes de la fabrication de la base de donnée, on pourra envisager de redéfinir le besoin dans un deuxième temps.

# Annexe B

## Schémas de la base de données

Cette annexe présente les schémas de la base de données au fur et à mesure de sa conception et de son évolution.

Pour toutes les figures, les cases sont des tables ; les lignes sont les relations entre les tables. La couleur jaune correspond à des entiers, le vert à une date et le rouge à une chaîne de caractère.

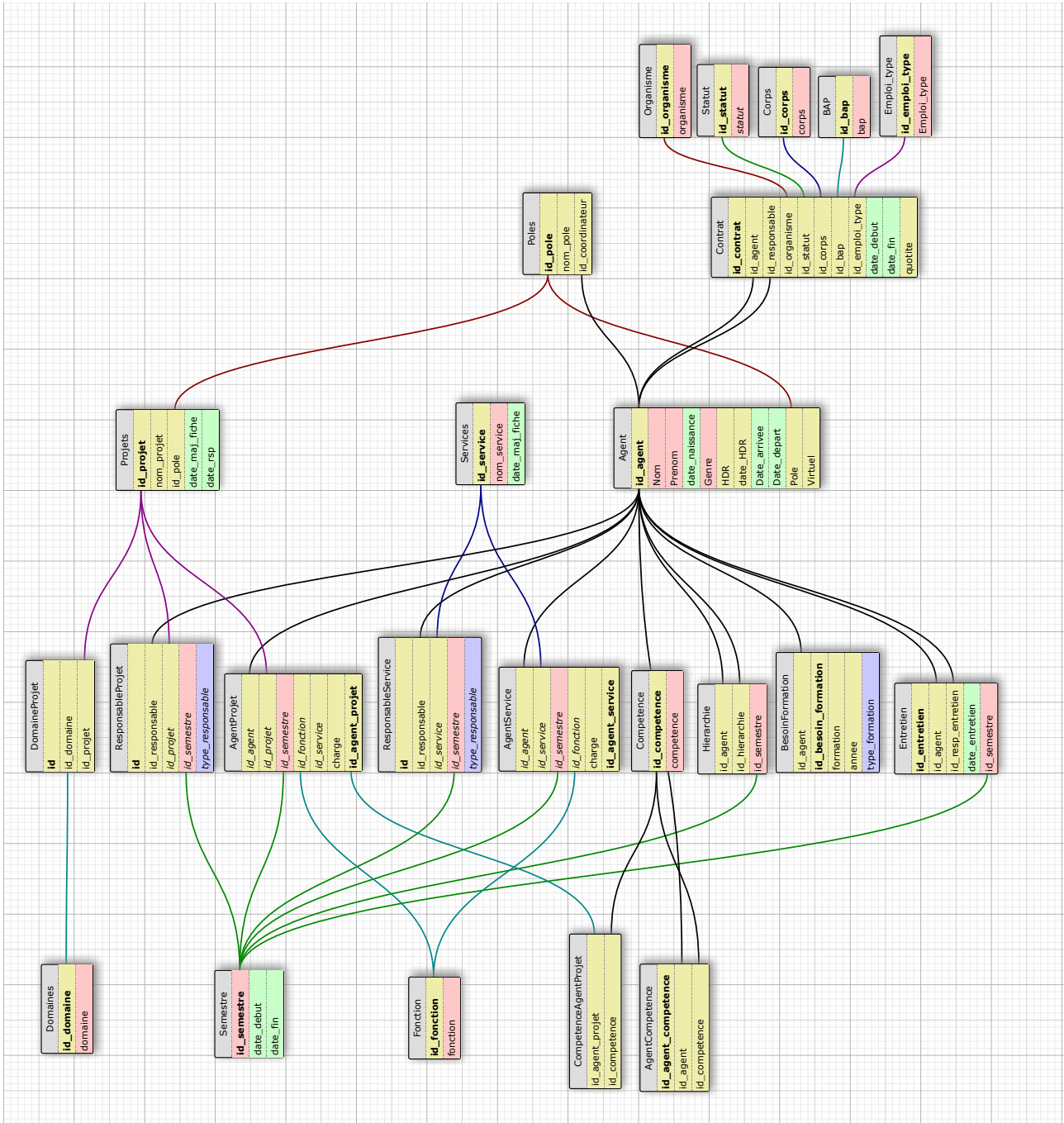


FIGURE B.1 – Base de donnée initiale



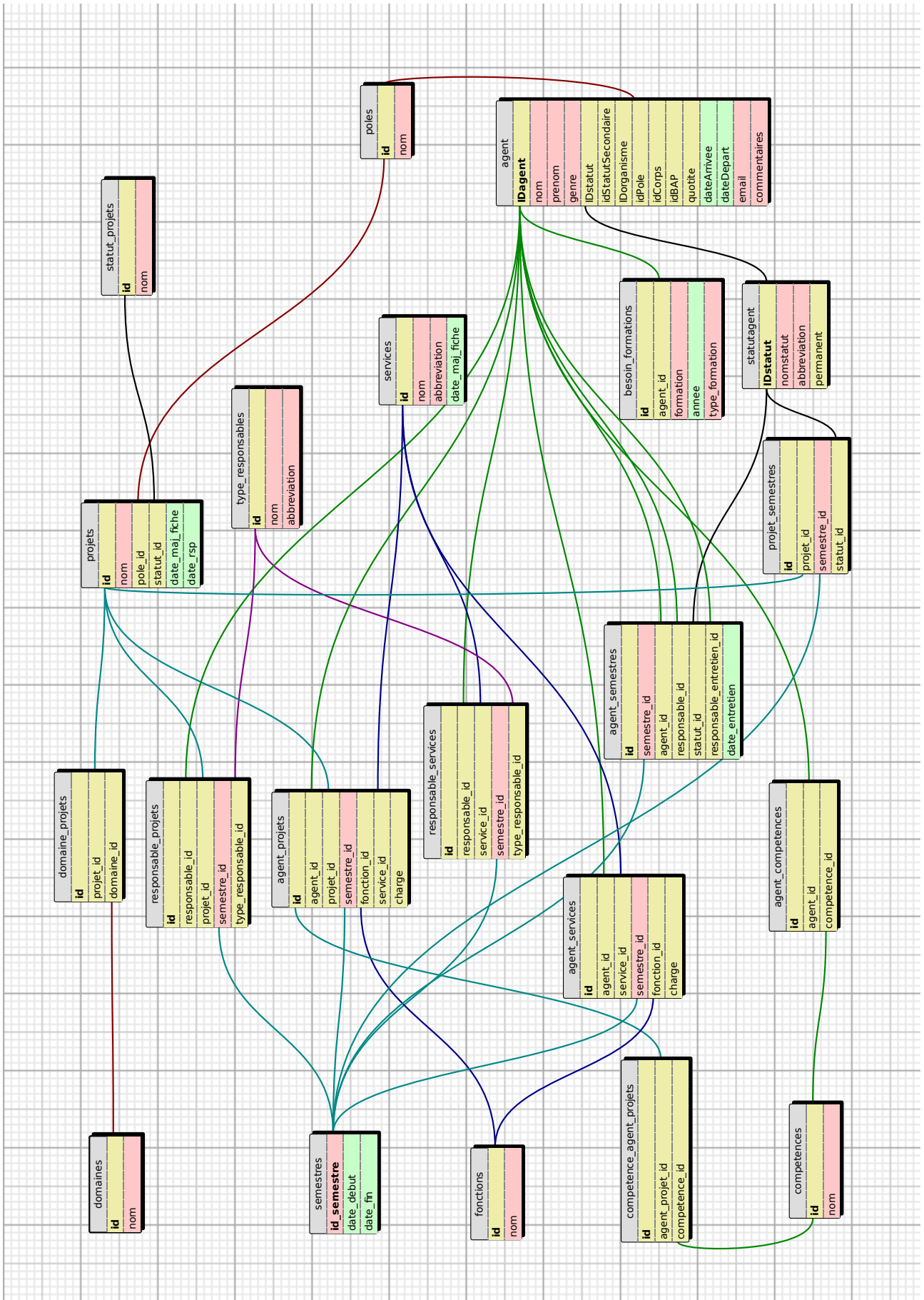


FIGURE B.2 – Base de donnée intermédiaire

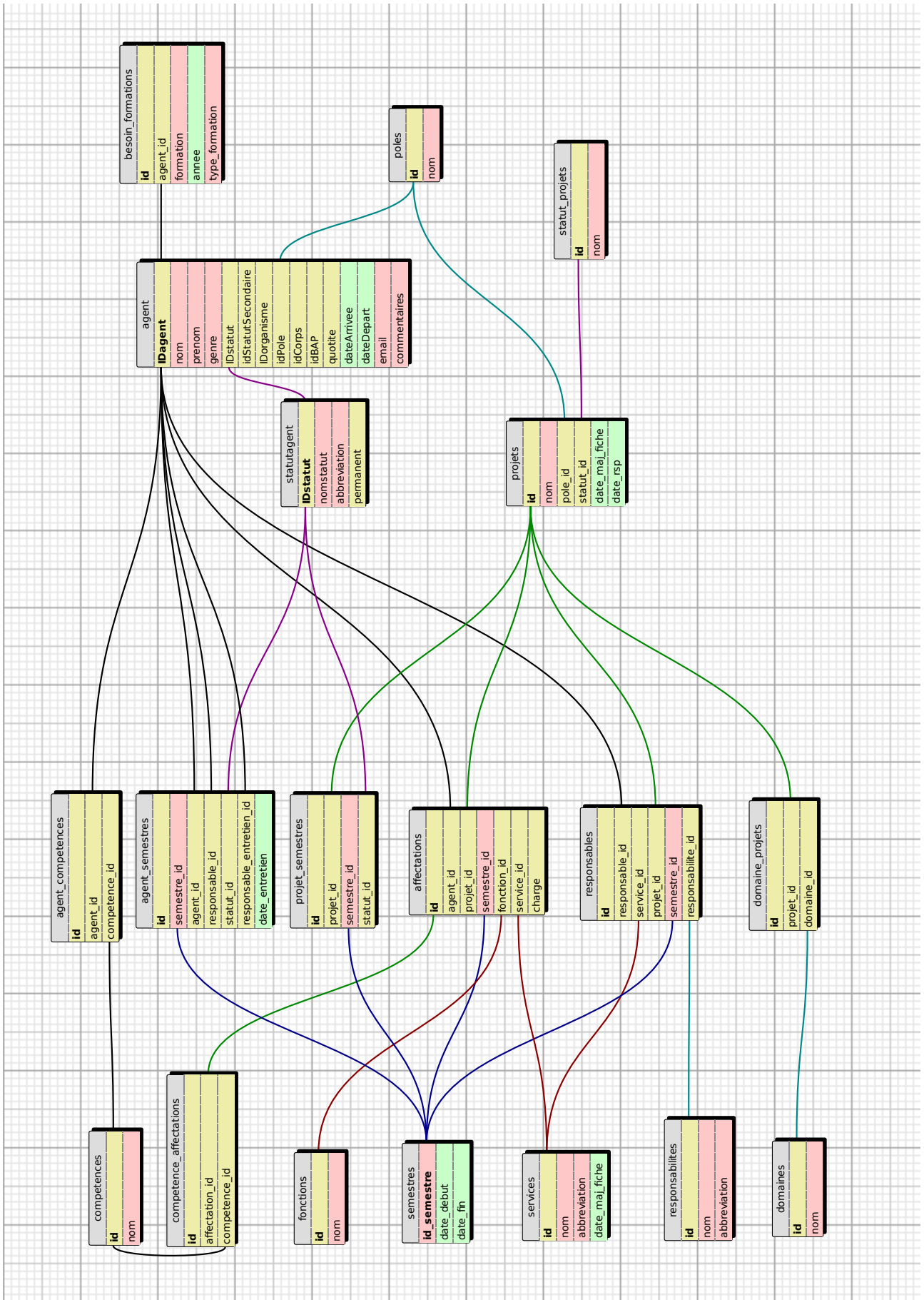


FIGURE B.3 – Base de donnée finale

# Annexe C

## Les outils utilisés

Dans cette annexe, je vais présenter différents outils utilisés tout au long de mon stage.

phpMyAdmin

localhost

Base de données: personnels\_rails » Table: services

Affichage des lignes 0 - 10 (total de 11, Traitement en 0.0057 secondes.) [date\_maj\_fiche: -]

SELECT \* FROM `services` ORDER BY `date\_maj\_fiche` ASC

Profilage [ En ligne ] [ Modifier ] [ Expliquer SQL ] [ Créer source PHP ] [ Actualiser ]

Nombre de lignes : 25 Filtrer les lignes: Rechercher dans cette table

Trier sur l'index: Aucune

+ Options

		id	nom	date_maj_fiche	abbreviation	created_at	updated_at
<input type="checkbox"/>			1	Effacer	1	2016-06-09 10:01:13	2016-06-09 10:01:13
					2	2016-06-09 10:01:13	2016-06-09 10:01:13
					3	2016-06-09 10:01:14	2016-06-09 10:01:14
					4	2016-06-09 10:01:14	2016-06-09 10:01:14
					5	2016-06-09 10:01:14	2016-06-09 10:01:14
					6	2016-06-09 10:01:14	2016-06-09 10:01:14
					7	2016-06-09 10:01:14	2016-06-09 10:01:14
					8	2016-06-09 10:01:14	2016-06-09 10:01:14
					9	2016-06-09 10:01:14	2016-06-09 10:01:14
					10	2016-06-09 10:01:14	2016-06-09 10:01:14
					11	2016-06-09 10:01:14	2016-06-09 10:01:14

Tout cocher Pour la sélection: Modifier Effacer Exporter

Nombre de lignes : 25 Filtrer les lignes: Rechercher dans cette table

Opérations sur les résultats de la requête

Version imprimable Version imprimable (avec textes complets) Exporter Afficher le graphique Créer une vue

Conservation de cette requête SQL dans les signets

Signet visible pour les autres utilisateurs

Récentes | Préférées

- organisme
- paramètre
- pers\_bap
- pers\_contrats
- pers\_convention
- pers\_corps
- pers\_echelon
- pers\_etablissement
- pers\_formation
- pers\_niveau
- pers\_partenaire
- pers\_payestagiaire
- pers\_typecontrat
- pers\_xlabentite
- pers\_xlaborig
- poles
- projets
- projet\_semestres
- reliquat
- responsabilites
- responsables
- schema\_migrations
- semestres
- services
- stagiaire
- statutagent
- statut\_projets
- statut\_secondaire
- theses\_directeurs
- theses\_directeurs\_theses
- theses\_docteurs
- theses\_docteurs\_theses
- theses\_ecoles
- theses\_ecoles\_theses

FIGURE C.1 – Table "services" avec phpMyAdmin

```

8 belongs_to :service
9 belongs_to :agent_semestre,
10   -> { |aff| { where("semestre_id = ?", aff.semestre_id) },
11   foreign_key: "agent_id",
12   primary_key: "agent_id"
13
14 has_many :competence_affectations
15 has_many :competences, through: :competence_affectations
16
17 # VERIFICATIONS
18 validates :agent_id, inclusion: {in: Agent.all.map {|a| a.idagent},
19   message: ": non renseigné ou invalide"}
20 validates :projet_id, inclusion: {in: Projet.all.map {|p| p.id},
21   message: ": invalide"}, allow_nil: true
22 validates :service_id, inclusion: {in: Service.all.map {|s| s.id},
23   message: ": non renseigné ou invalide"}
24 validates :semestre_id, inclusion: {in: Semestre.all.map {|s| s.id_semestre},
25   message: ": non renseigné ou invalide"}
26 validates :fonction_id, inclusion: {in: Fonction.all.map {|f| f.id},
27   message: ": invalide"}, allow_nil: true
28 validates :charge, numericality: {only_integer: true, greater_than: 0,
29   message: ": doit être un entier > 0"}
30
31 # il ne faut pas accepter une affectation pour un semestre
32 # si l'agent n'a pas été mis à jour avec agent_semestre
33 validates_each :semestre_id do |record, attr, value|
34   unless (record.semestre.nil?) then
35     agent_semestre = AgentSemestre.where(semestre_id: value, agent_id: record.agent)
36     if agent_semestre.empty? then
37       record.errors.add(attr, ": le statut de cet agent n'a pas été renseigné pour le semestre #{value}")
38     end
39   end
40 end
41
42 #METHODES D'INSTANCES
43 def permanent
44   agent_semestre.statutagent.permanent
45 end
46
47 # METHODES DE CLASSE
48 # Sommer les charges des affectations totales
49 def Affectation.charge_totale
50   Affectation.joins("LEFT OUTER JOIN agent_semestres
51     ON affectations.agent_id = agent_semestres.agent_id
52     AND affectations.semestre_id = agent_semestres.semestre_id").
53     order("affectations.semestre_id").
54     group("affectations.semestre_id").sum(:charge)
55 end

```

FIGURE C.2 – Screenshot d'Atom

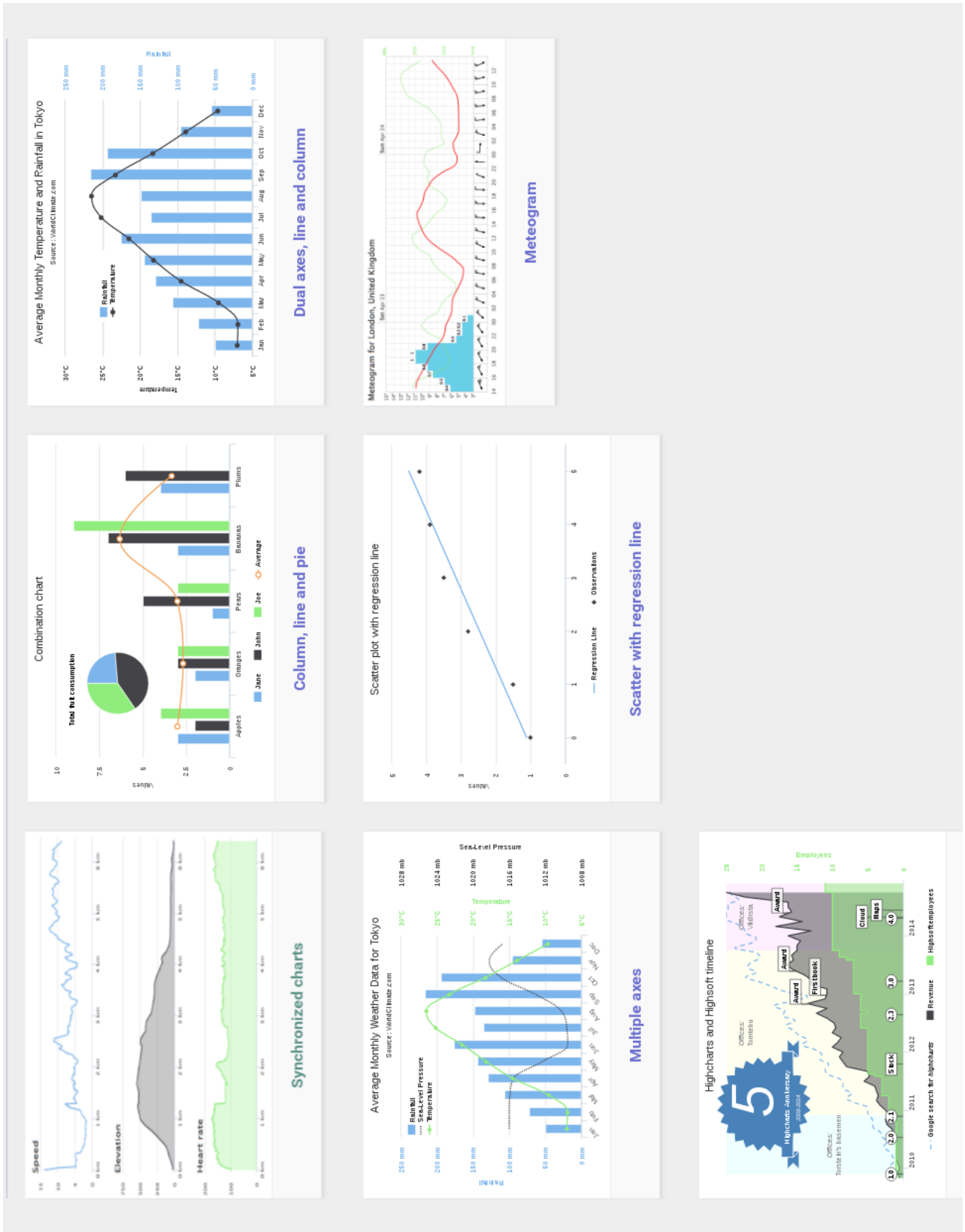


FIGURE C.3 – Exemples de graphique possible avec HighCharts

# Annexe D

## Captures d'écran

Cette annexe présente différents capture d'écran issu de l'application web.

## HENRY Florence

Accueil

Agent ▾

Service ▾

Projet ▾

## Fiche de l'agent

**ID :** 99  
**Genre :** Féminin  
**Statut :** ITA (Permanent)  
**Organisme :** CNRS  
**Pole :** Technique  
**Corps :** Ingénieur de recherche 1  
**BAP :** E - Informatique, Statistique et Calcul Scientifique  
**Quotité :** 100%  
**Date arrivée :** 1990-01-01  
**Date départ :** 2024-12-31  
**E-mail :** [Florence.Henry@obsspm.fr](mailto:Florence.Henry@obsspm.fr)

## Commentaires :

## Compétences :

## Compétence d'un agent

Saisissez les compétences séparées par des virgules ( , ) :

Enregistrer

## Responsabilité d'un service

Semestre	Service	Type	Actions
2020_S2	GIGL	Chef de service	Modifier Supprimer
2020_S1	GIGL	Chef de service	Modifier Supprimer
2019_S2	GIGL	Chef de service	Modifier Supprimer
2019_S1	GIGL	Chef de service	Modifier Supprimer
2018_S2	GIGL	Chef de service	Modifier Supprimer
2018_S1	GIGL	Chef de service	Modifier Supprimer
2017_S2	GIGL	Chef de service	Modifier Supprimer

## Responsabilité d'un projet

Semestre	Projet	Type	Actions
2016_S1	ESA-CNES / ROSETTA MIRO	Chef de projet	Modifier Supprimer
2016_S1	ESA-CNES / ROSETTA VIRTIS	Chef de projet	Modifier Supprimer
2016_S1	OV planéto	Chef de projet	Modifier Supprimer
2016_S1	POP	Chef de projet	Modifier Supprimer

FIGURE D.1 – Un agent



## Affectation d'un agent à un projet et/ou service

<b>Nom de l'agent *</b>	Choisir un agent ▼
<b>Nom du projet</b>	Choisir un projet ▼
<b>Nom du service *</b>	Choisir un service ▼
<b>Choisir un semestre *</b>	Choisir un semestre ▼
<b>Choisir une fonction</b>	Choisir une fonction ▼
<b>Charge *</b>	Ex: 55 %
	<input type="button" value="Envoyer"/>

\* Ce champ est obligatoire.

FIGURE D.2 – Affectation d'un agent à un projet ou un service

# Liste des compétences

Nouvelle compétence	
Compétences	Actions
3 C	Modifier Supprimer
2 C++	Modifier Supprimer
3 CSS	Modifier Supprimer
2 Excel	Modifier Supprimer
1 Gestion de projet	Modifier Supprimer
1 HTML	Modifier Supprimer
0 IDL	Modifier Supprimer
1 Java	Modifier Supprimer
1 Javascript	Modifier Supprimer
0 Latex	Modifier Supprimer
0 Photoshop	Modifier Supprimer

FIGURE D.3 – Liste de diverse compétence, le badge à côté du nom représente le nombre d’agent possédant cette compétence

## Liste des responsabilités

Nouvelle responsabilité		
Abbréviation	Nom	Actions
CP	Chef de projet	Modifier Supprimer
CS	Chef de service	Modifier Supprimer
CSad	Chef de service adjoint	Modifier Supprimer
RS	Responsable scientifique	Modifier Supprimer

FIGURE D.4 – Liste de diverse responsabilité

# Liste des services

Nouveau Service

Abbréviation	Nom du service	Actions	
C2L	Cellule Logistique du LESIA	Modifier	Supprimer
CPR	Cellule Prévention des Risques	Modifier	Supprimer
CQL	Cellule Qualité du LESIA	Modifier	Supprimer
GEFL	Groupe d'Etude et de Fabrications du LESIA	Modifier	Supprimer
GIGL	Groupe Informatique Générale du LESIA	Modifier	Supprimer
MESPAL	Moyens d'Essais Salles Propres AIT/AIV du LESIA	Modifier	Supprimer
SIGAL	Service Internet, Graphisme et Animations du LESIA	Modifier	Supprimer
SII	Service Informatique Instrumentale	Modifier	Supprimer
SIS	Service Informatique Scientifique	Modifier	Supprimer
SPL	Soutien Projets du LESIA	Modifier	Supprimer
SSL	Service Solaire du LESIA	Modifier	Supprimer

FIGURE D.5 – Liste des services

## Liste des projets

[Nouveau Projet](#)

Nom du projet	Domaine	Statut	Actions
4Q++	R&T	En développement	Modifier Supprimer
Activités de service	Autres	En exploitation	Modifier Supprimer
BASS2000	Autres	En exploitation	Modifier Supprimer
CNES / COROT	Spatial	En exploitation	Modifier Supprimer
CNES / NOIRE	Spatial	Prospective	Modifier Supprimer
CNES / R&T Radio	R&T	En développement	Modifier Supprimer
CNES / R&T UVMAG	R&T	En développement	Modifier Supprimer
CNES / SPIM (BESTIAL)	R&T	En développement	Modifier Supprimer
CNES / UVMAG Phase 0	Spatial	En développement	Modifier Supprimer
CNRS-GEMaC / uPPI	R&T	En développement	Modifier Supprimer
CTS (SIMENOM-CERES)	Moyens d'essais	En développement	Modifier Supprimer
Détecteurs APD	R&T	Prospective	Modifier Supprimer
DGA / METEOSPACE	Météo de l'espace	En développement	Modifier Supprimer

FIGURE D.6 – Liste de quelques projets

## Ajout d'un projet

**Nom du projet \***

**Pôle \***

**Domaine \***  Autres  Météo de l'espace  Moyens d'essais  Outils LESIA  Outils Observatoire  R&T  
 Radioastronomie sol  Spatial  Télescopes optiques sol

**Statut \***

**Date RSP**

**Date MAJ Fiche**

\* Ce champ est obligatoire.

FIGURE D.7 – Formulaire d'ajout d'un projet