# REMERCIEMENTS

Avant toute chose, je tiens à remercier au Dieu Tout Puissant. Une fois encore il a montré son pouvoir et sa force car il m’a donné la force pour achever ce stage et mener à terme ce présent mémoire.

Ensuite, je tiens à remercier les personnes suivantes car sans leur soutien et leur précieuse collaboration, je n’aurais pas pu arriver à bout de ce mémoire de fin d’étude.

* Le professeur RABOANARY Julien Amédée, Recteur et fondateur de l’Institut Supérieur

Polytechnique de Madagascar ;

* Monsieur Paul Pierre ARDILE, le Président Directeur Général de la société eTech qui nous a reçu en tant que stagiaire ;
* Monsieur Andry RAMANAMBAHOAKA, Directeur de projet Web-PHP ;
* Monsieur Clarel RAKOTONDRAHAJA, Directeur de projet et aussi enseignant à l'Institut Supérieur Polytechnique de Madagascar ;
* Monsieur Rija RAOBIVELO RABETOKOTANY, Directeur de projet (.Net – Java - Mobile) ;
* Madame Eva RALALAHARIMANANA, Responsable Administratif et Financier ;
* Monsieur RAVELOSON Hery Nandrianina, chef de projet Web-PHP et aussi mon encadreur professionnel ;
* Monsieur RASANDIMANANA Tanjona Tsioharana, mon encadreur pédagogique et aussi enseignant à l'Institut Supérieur Polytechnique de Madagascar ;
* Tout le corps enseignant de l'Institut Supérieur Polytechnique de Madagascar pour son assistance et son engagement pendant ces cinq années d'études ;
* Tout le personnel administratif de l'Institut Supérieur Polytechnique de Madagascar ;
* Tous mes collaborateurs chez eTech qui se sont aussi donnés la peine de transmettre leur savoir-faire et leur expérience durant ces quelques mois de stage.
* Toute la famille et tous nos amis respectifs, ainsi que tous ceux qui ont contribué de près ou de loin à l'élaboration de ce présent mémoire par leur affection, leurs prières, leurs précieux conseils, leur soutien financier et matériel.

# AVANT-PROPOS

Selon le programme d’enseignement à l’ISPM, pour obtenir son diplôme de master II, il est indispensable d’effectuer un stage de fin d’étude. Cela nous permet d’avoir un aperçu du travail qui nous attend et d’acquérir un peu plus d’expérience avant de se lancer dans le monde professionnel.

En effet, j’ai fait mes 6 mois de stage au sein de la société eTech, qui était un stage très enrichissant, socialement, et professionnellement. De ce fait, on m’a affecté à un projet très intéressant et que j’ai pris comme thème pour ce présent mémoire. Il s’agit de **e-Asa**.

Je tiens à préciser qu’avec l’aide de mes collaborateurs chez eTech, j’ai bien pu réaliser le ce projet avec brio, j’ai pu mettre en pratique mes acquis théoriques et j’ai pu améliorer mon savoir-faire en tant que développeur informatique.

# LISTE DES ABREVIATIONS

SOAP

# LISTE DES FIGURES

# LISTE DES TABLEAUX

# SOMMAIRE

# BIBLIOGRAPHIE

**COURS DISPENSÉS À L’ISPM**

[1] RABOANARY Julien Amédée, **Gestion de qualité*,*** (support de cours, 2012-2013)

[2] RABOANARY Andry Heriniaina**, Algorithme avancée** (support de cours 2012-2013)

[3] RAKOTOARIMANANA Andriambao Johnson**, UML 2,** (support de cours 2013-2014-2015)

[4] RAKOTONDRAHAJA Chantal**, AOO,** (support de cours 2013-2014)

[5] RAKOTONDRAHAJA Clarel**, COO,** (support de cours 2014-2015)

[6] RAKOTONDRAHAJA Clarel**, SGBDR,** (support de cours 2014-2015)

[7] RANAIVOHARINIRIANA Tanjona, **JavaScript,** (support de cours, 2012-2013)

[8] RANDRIANTSOA Lalaina Thierry, **PHP,** (support de cours, 2012-2013)

[9] RASANDIMANANA Tanjona Tsioharana, **MySQL,** (support de cours, 2010-2011)

**OUVRAGES**

[10] B.M. Harwani, **jQuery Recipes**, Editions Apress, 2010

[11] David Schultz, Craig Cook, **Beginning HTML with CSS and XHTML**, Editions Apress, 2007

[12] Eric Cariou, **Introduction à UML**, Université de Pau et des Pays de l'Adour, 2013

[13] Laurent GUÉDON, Damien HEUTE, Thomas HEUTE, Pierre-Emmanuel MULLER, **la bible PHP 5**, Micro Application, 2005

[14] Olivier Gerbe, **Cours UML**, IFT, 2000

[15] Pierre-Alain Muller, **Modelisation obbet avec UML**, Eyrolles, 1997

**WEBOGRAPHIE**

[a] <https://symfony.com/doc/current/cookbook/index.html> **pour tous les cours de Symfony2,** consulté en novembre 2015

[b] <https://www.stackoverflow.com>  **pour la résolution des bugs recontrés dans le projet**, consulté en novembre 2015

[c] <https://github.com> **pour la documentation des différents bundles utilisés en Symfony2,** consulté en novembre 2015 et en début 2016

RATSIMANDEFITRA Miora Manitra

Lot IAB 39 A Andrononobe

103 - Antananarivo

Tel : +261331111079

E-mail : miora.ratsimandefitra@gmail.com

## ETAT CIVIL

|  |  |
| --- | --- |
| * ***Nationalité*** | Malagasy |
| * ***Situation matrimoniale*** | Célibataire |
| * ***Date de naissance*** | 05 Février 1993 |
| * ***Lieu de naissance*** | Anjanahary |

## DIPLOMES OBTENUS

|  |  |
| --- | --- |
| * ***2015*** | ***Diplôme d’ingéniorat*** (Master) en Informatique de Gestion, Génie Logiciel et Intelligence Artificielle à l’ISPM **(*en cours)*** |
| * ***2013*** | ***Diplôme de licence*** en Informatique de Gestion, Génie  Logiciel et Intelligence Artificielle à l’ISPM |
| * ***2010*** | ***Baccalauréat*** - série C |

## COMPETENCES TECHNIQUES

|  |  |
| --- | --- |
| * ***Février 2016 - Actuellement*** | **Poste actuel**: Ingénieur Développeur Junior (eTech) |
| * ***Depuis Mars 2016*** | **Projet** : Smallabe et Ivalis |
| * ***Depuis Novembre 2015*** | **Projet** : Ivalis : Myspace, WeeklyPlanning, EspaceJob |

## COMPETENCES INFORMATIQUES

|  |  |
| --- | --- |
| * ***Analyse Informatique*** | *MERISE I/II, AOO avec UML* |
| * ***Base de données et SGBD*** | *MySQL* |
| * ***Framework PHP*** | *Symfony2* |
| * ***SCM*** | *Git,SVN* |
| * ***Autres framework*** | *JQuery, Bootstrap, AngularJs* |
| * ***Informatique bureautique*** | *World, Excel, PowerPoint* |
| * ***Langage de modélisation*** | *UML* |
| * ***Langage informatique*** | *C#, Java, PHP7, PHP5, Javascript, Html/Css,* |

## CONNAISSANCE LINGUSTIQUE

|  |  |
| --- | --- |
| * ***Malagasy*** | Langue maternelle |
| * ***Français*** | Lu et parlé |
| * ***Anglais*** | *Lu et parlé* |

## INFORMATIONS COMPLEMENTAIRES

|  |  |
| --- | --- |
| * ***Sport*** | Football, Natation |
| * ***Loisirs*** | Musique, Livres, Film, Reportages |
| * ***Permis*** | Catégorie B |

# INTRODUCTION

Depuis, ces dernières décennies, le monde du numérique n’a jamais cessé d’évoluer. On peut très bien remarquer que du temps de nos parents, on envoyait encore des lettres ou des télégrammes pour communiquer. Actuellement, grâce à cette révolution technologique, on peut très bien se communiquer à travers les réseaux sociaux, autres application de messagerie, et par email. Maintenant on ne peut plus s’en passer car la technologie entre peu à peu dans la vie quotidienne. Dans le domaine professionnel aussi, le numérique prend de plus en plus d’ampleur, d’où l’existence de nombreuse plate-forme de recherche d’emploi par exemple.

De ce fait, **e-Asa** est une plate-forme permettant de rechercher du travail parmi les missions existantes proposées. Il permet aussi d’avoir accès à un espace où l’on peut consulter ses missions, modifier ses informations personnelles, voir le calendrier de ses missions… **e-Asa** permet de voir le planning des missions pour une semaine, 1 mois, ou les années à venir. Il faut savoir que **e-Asa** est une très grande plate-forme qui est divisé alors en 3 sous projets : **e-Asako**, **e-Asa-tinady** et **Lamin’Asa** ,

Pour ce mémoire ci, nous présenterons de façon générale le projet **e-Asa**. Mais notre étude se portera surtout sur le premier sous projet **e-Asako**. En premier lieu, nous présenterons en première partie, l’ISPM puis la société eTech. En deuxième partie nous expliquerons le projet avec les divers outils de réalisation ;

# PARTIE I : PRESENTATION GENERALE

# CHAPITRE 1 : PRESENTATION DE L’INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR

## Historique et présentation de l’ISPM

### Historique de l’ISPM

L’ISPM ou Institut Supérieur Polytechnique de Madagascar a été créé le 23 janvier 1993 par le Professeur RABOANARY Julien Amédée.

C’est une institution privée à vocation académique et professionnelle agréée par L’Etat suivant l’arrêté N° 3725 du 19 Août 1994 du Ministère de l’Enseignement Supérieur et de la Recherche scientifique.

A cette époque, son siège se trouvait à Ankadindramamy, il était connu sous le nom d’ESSTIM ou École Supérieure des Sciences et de la Technologie Informatique à Madagascar et ne comportait que le département Informatique: la filière Informatique de Gestion génie Logiciel et Intelligence Artificiel ou I.G.G.L.I.A et la filière Maintenance des Systèmes Informatiques ou M.S.I C’était le seul institut privé polytechnique à Madagascar. Son siège se trouvait à Ankadindramamy.



Figure : ESSTIM (source ISPM)

Actuellement, l'ISPM et son bureau administratif se trouve à Ambatomaro Antsobolo, dans un cadre plus étendu, propice aux études.



Figure : ISPM Antsobolo Ambatomaro (Source:ISPM)

Au cours du temps, plusieurs départements et filière sont nés:

* 1994: Création du Département Informatique avec les deux filières: I.G.G.L.I.A et E.S.I.I.A;
* 1995: Création du Département Biotechnologique comportant deux filières: I.A.A et P.I.P;
* 1996: Création du Département tertiaire avec la filière C.A.A;
* 1998: Création du Département Génie Civil et Génie Industriel et aussi de la filière E.M.I.I;
* 1999: Mise en place de la filière G.C.A;
* 2000: Ouverture de la formation en troisième cycle;
* 2004: Création et ajout de la filière F.I.C dans le Département tertiaire;
* 2004: Intégration des deux filières: I.M.T.I.C.I.A dans le Département Informatique et A.E.E dans le Département Biotechnologie;
* 2009: Ajout des filières D.T.J.A, E.M.P et I.C.M.P;
* 2009: Tourisme et hôtellerie dans le Département Technique Tourisme;
* 2010: Création de la nouvelle filière I.S.A.I.A.

### Présentation des différentes mentions de l’ISPM

Actuellement, l’ISPM offre un large choix de filières pour les nouveaux bacheliers et les gens des entreprises qui souhaitent continuer leur étude supérieure; car dans cette institution il existe cinq (5) mentions qui, à leurs tours comportent au moins deux filières chacun, à savoir:

* La mention Informatique et télécommunications;
* La mention Droit et Techniques des Affaires;
* La mention Biotechnologique et Agronomie;
* La mention Génie industriel ;
* La mention Génie Civil;
* La mention Techniques du tourisme.

#### Mention informatique et télécommunication

C’était en 1994 que cette mention a été créée et il comporte quatre (4) filières. Ce sont les filières:

* IGGLIA;
* ESIIA;
* IMTICIA;
* ISAIA.

#### Mention droit et affaire

Cette mention a été créée en 1996. Il propose les quatre filières suivantes:

* CAA;
* FIC;
* DTJA;
* EMP.

#### Mention Biotechnologique et Agronomie

Cette mention a été créée en 1995, il comporte aussi trois (3) filières. Ce sont les suivantes:

* IAA;
* PIP;
* AEE.

#### Mention Génie Industriel

Cette mention se divise en trois (3) filières:

* EMII;
* ICMP.

#### Mention Génie Civil

Cette mention ne comporte qu’une seule filière :

* GCA.

#### Mention Techniques du Tourisme

Cette mention se divise en deux (2) filières:

* TEE;
* TEH.

Si tels sont l’histoire et les mentions au sein de l’ISPM, il est important de connaître les études et les formations offertes par cet institut.

### Description Physique de l’I.S.P.M

#### Les infrastructures

L’I.S.P.M possède différentes infrastructures composées de :

* Plus de Soixante-dix salles de cours ;
* Une grande salle pouvant contenir jusqu’à 1500 étudiants ;
* Un amphithéâtre en plein air ;
* Des bureaux pour l’administration ;
* Des laboratoires d’informatique ;
* Un laboratoire de langes ;
* Un laboratoire de chimie, de biologie et de pharmacologie ;
* Un atelier de mécanique ;
* Deux cafétérias ;
* Deux salles de soutenance de mémoire ;
* Un terrain de Basket-ball ;
* Un terrain de football (Mini-Foot) ;
* Un terrain de Volley-ball.

### Les Description de l’I.S.P.M

#### La devise de l’I.S.P.M

L’I.S.P.M a sa propre devise : **FAHAIZANA – FAMPANDROSOANA – FIHAVANANA**

#### Le logo de l’I.S.P.M



**I**NSTITUT **S**UPERIEUR **P**OLYTECHNIQUE

DE **M**ADAGASCAR

Figure : logo de l’I.S.P.M (source : ISPM)

Explications :

* La toque noir signifie que l’I.S.P.M forge des cadre supérieurs d’où le

«FAHAIZANA».

* Le globe terrestre montre que l’I.S.P.M s’ouvre au monde entier d’où le «FAMPANDROSOANA».
* Les Mains qui se serrent nous indiquent que l’I.S.P.M est ouvert à toutes les nationalités d’où le «FIHAVANANA».

#### L’Hymne de l’I.S.P.M

‘’ Ilay I.S.P.M tena maminay ‘’

Ilay I.S.P.M tena maminay

Tsy mba foinay tokoa hatrizay

Toera-nanabeazana ny tenanay

Mba ho tena olom-banona mahay

*Ref: Ho mendrika tokoa,*

*Ho mafy orina avokoa*

*Ny fahaizana, fampandrosoana, fihavanana*

*Ho andry sy tokin’ny Tanindrazanay*

Ny fahaizana no apinga ho enti-miady

Nampitaina,nozaraina sy nomena

Tsy andrarak’ilo fa ho tena kiady

Hoenti-mampandroso ny firenena

Ka ny fihavanana, firaisankina

No asandratray hatrany hatrany

Manoloana fifaliana, fijaliana

Mandra-piala anay ety an-tany

#### Adresse et contact

L’I.S.P.M a désormais son siège à: Ambatomaro – Antsobolo – Antananarivo 101

|  |  |
| --- | --- |
| * Ses contacts | : 034 17 583 69 / 033 12 171 60 / 032 84 509 39 / 020 26 384 08 |
| * Son adresse e-mail | : [*ispm@ispm-edu.com*](mailto:ispm@ispm-edu.com) |
| * *Son site web* | : [*www.ispm-edu.com*](http://www.ispm-edu.com) |



Figure : site web de l’I.S.P.M (source : ISPM)

## HIERARCHIE ET ORGANIGRAMME DE L’I.S.P.M

### Hiérarchie de l’I.S.P.M

L’I.S.P.M possède un niveau organigramme bien structuré où chaque individu a un rôle bien déterminé:

#### Le Rectorat

Le Professeur RABOANARY Julien Amédée est le recteur de l’I.S.P.M, Il est le premier responsable administratif à l’I.S.P.M.

#### La Direction Administratif et financière

Dirigée par Madame RAZAFINDRAHETY Aimée qui s’occupe de la finance et par Monsieur RABOANARY Gérard pour l’Administration.

#### La direction des études

Chaque département possède un chef de département qui est nommés par le recteur gère la qualité de l’enseignement.

### Organigramme de l’I.S.P.M

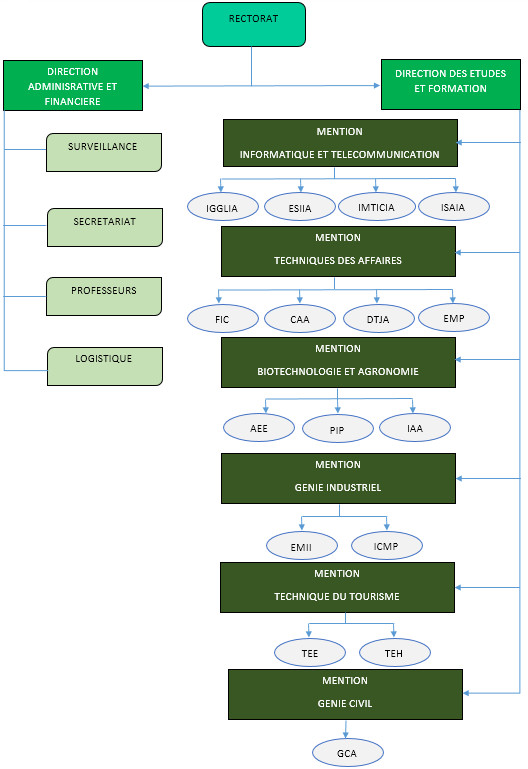


Figure : Organigramme de l’I.S.P.M (source: ISPM)

## CURSUS D’ETUDE A L’ISPM

### Le Cadre physique de l’ISPM

L’entrée en première année à l’ISPM se fait par voie de concours, et comme nous l’avons évoqué, l’institut est ouvert aux bacheliers qu’ils soient Malagasy ou Etrangers, et même qu’ils soient nouveaux ou ayant déjà travaillé en entreprises et voulant se spécialiser dans l’une des filières existantes.

Actuellement l’ISPM offre trois (3) cycles d’études et de formation. C’est-à-dire il existe dans cet institut la formation pour le :

⮳ Premier cycle (de 1ère année à la 3ème année);

⮳ Second cycle (4ème année et 5ème année);

⮳ Troisième cycle (6ème année et plus).

### Les cycles d’études

#### Le premier cycle

Pour l’étude en premier cycle, les étudiants doivent suivre les trois années d’études car cette première formation est sanctionnée d’un Diplôme de Licence. Ces trois années d’études sont suivies d’un stage de trois mois en entreprise, et d’une soutenance de mémoire sur ce stage afin d’obtenir ce diplôme.

#### Le second cycle

Ce second cycle se déroule en deux années d’études, cela veut dire que seuls les étudiants qui ont le Diplôme de Licence, étudiants qui ont bien terminé le premier cycle, peuvent continuer cette formation.

Après ces deux années d’études théoriques, les étudiants doivent faire six mois de stage en entreprise ou en bureau d’études à l’issue duquel ils soutiennent un mémoire de fin d’études dont la réussite leur octroie le diplôme d’Ingénieur.

#### Formation du troisième cycle

La formation du troisième cycle a été ouverte en l’an 2000 pour obtenir de Diplôme d’Etudes Approfondies (DEA) et du Doctorat en Informatique. Seuls les étudiants qui ont le diplôme d’ingénieur peuvent suivre cette formation.

Actuellement, les enseignants sont impliqués profondément dans la recherche. Et le système LMD étant d’actualité, l’ISPM se veut être un *leader* dans ce domaine d’innovation.

Présentation graphique du CURSUS :

**

CONCOURS

D’ENTREE

BACCALAUREAT

MASTER (INGENIORAT)

BAC + 5

ACTIVITES

PROFESSIONNELLES

Figure : graphique du cursus (source : ISPM)

## Le régime de l’I.S.P.M

L’ISPM est un Institut à vocation, à la fois académique et professionnelle. En effet, il a pour but d’assurer l’avenir des futurs Techniciens Supérieurs et des Ingénieurs.

L’ISPM dispense des cours huit (08) heures par jour du lundi au vendredi et quatre heures pour le mercredi et le samedi.

Toutefois, les étudiants sont libérés le mercredi après-midi pour des travaux personnels ou des travaux en groupe.

Ils doivent effectuer des examens de contrôle (*tests ou contrôles continus*) et deux examens semestriels.

Et les étudiants en cinquième année doivent passer un examen de fin d’études (ou examen clinique) avant d’entamer leurs stages.

Chaque année l’ISPM réalise des *journées portes ouvertes* pour présenter aux visiteurs les recherches effectuées par les étudiants.

En plus, I S P M présente une spécificité dans le cadre de la formation qui n’est autre que « le Projet » qui est obligatoire pour chaque étudiant. Tous les projets sont exposés pendant les journées portes ouvertes organisées par l’I S P M généralement au mois de juin, les meilleurs iront ensuite au salon de l’I S P M au Madagascar Carlton.

## Performance de l’I.S.P.M

Les étudiants formés par l’ISPM sont embauchés à hauteur de 95%, les 5% restants ont continué leurs études ou ont créé leurs propres entreprises.

Depuis des années, I.S.P.M participe à des différant concours où il a gagné plusieurs primes.

Voici quelques prix obtenus par l’ISPM depuis son existence.

Tableau : prix obtenus par l’I.S.P.M depuis son existence (source: ISPM)

|  |  |  |  |
| --- | --- | --- | --- |
| **Rang** | **PROJET** | **ANNEE** | **ORGANISATEUR** |
| 1er | Logiciel de Fanorona | 1997 | Ambassade de France |
| 3ème | SRI (Repiqueuse) | 2003 | FIERMADA |
| 3ème | Tracteur et Repiqueuse | 2003 | Forum de la Recherche MENRS |
| 1er | Analyse de la structure biotechnologique du riz | 2006 | Salon de la Créativité et de l’invention |
| 2ème | Etuve phytotronique assistée par ordinateur | 2006 | Salon de la Créativité et de l’invention |
| 1er | Réchaud à alcool | 2007 | Fondation Tany Meva |
| 1er | Valorisation de la jacinthe d’eau pour la fertilisation en aquaculture | 2008 | Fondation Tany Meva |
| 1er | Production d’un plastique biodégradable | 2009 | Fondation Tany Meva |
| 1er | Smart planning managing | 2012 | Forum de la Recherche MESUPRES |
| 2ème | Intelligent House | 2012 | Forum de la Recherche MESUPRES |
| 4ème | Energies renouvelable | 2012 | Forum de la recherche MESUPRES |
| 1er | Olympiade Informatique de Madagascar | 2012 | Madagascar au temps TIC |
| 2è | Concours de développement d'application malagasy : « Flash Credit » | 2014 | Orange Madagascar |
| 2è | Concours de développement d'application malagasy : « Safe sms » | 2014 | Orange Madagascar |
| 1er | WEBCUP | 2015 | GOTICOM & Office du Tourisme |
| 1er | WEBCUP VIRALITE | 2015 | WEBCUP Association |

# CHAPITRE 2 : PRESENTATION DE L’ENTREPRISE « eTech »

## Historique de l’entreprise « eTech »

eTech est une Société de Services en Ingénierie Informatique (SSII) implantée à Madagascar depuis 2006. Son local se trouve dans l’immeuble Master Trade à Anosivavaka Ambohimanarina, route du Pape. Elle est née d’une collaboration entre Patrick Trebalag, Dominique et Fred Félix en tant que société spécialisée dans le développement de solutions informatiques.

En juin 2008, suite à des problèmes commerciaux, les actionnaires d’eTech se sont rapprochés de Monsieur Pierre-Paul Ardile, Président Directeur Général d’un cabinet de conseil expert des solutions professionnelles mobiles appelé : « Mov’age», pour un transfert d’actions.

En Septembre 2008, sous la direction de Monsieur Pierre-Paul Ardile, de nouveaux collaborateurs, associant compétences malagasy et française, reprennent les activités de la société. Depuis, eTech travaille en étroite collaboration avec la société de services informatiques française « ArkeUp » qui la représentent en France.

eTech, spécialiste des solutions digitales web, mobile et e-business en offshore fait partie du Groupe ArkeUp avec une représentation en France et au Maroc.

## Position géographique

L’entreprise eTech est située à Antananarivo I, route du Pape Jean Paul II, Anosivavaka Ambohimanarina, dans la capitale de Madagascar.

## Logo de l’eTech



Figure : logo de l’eTech (source eTech)

## Caractéristiques de l’entreprise « eTech »

### Site et location

|  |  |
| --- | --- |
| **Adresse :** | Immeuble Master Trade, route du Pape Jean Paul II, Anosivavaka Ambohimanarina Antananarivo |
| **Téléphone :** | +261 20 22 536 04 |
| **Boîte postale :** | 8721 |
| **Code postal** : | 101 |
| **Courriel :** | etech@etechconsulting-mg.com |
| **Site web :** | http://www.etechconsulting-mg.com |

### Activité principale

La société eTech se spécialise dans le domaine du développement d'applications informatiques (sites web, applications mobiles) et utilise un certain nombre de technologies afin de répondre à tous les besoins et exigences de ses clients et collaborateurs.

Entre autre, l’entreprise effectue :

* L’audit et l’analyse des besoins ;
* Les spécifications et la conception de base de données ;
* La conception graphique : de la création à l’intégration ;
* Le développement ;
* Le déploiement ;
* La maintenance et le support des applications.

### Effectif du personnel

Son centre de compétences et de développement situé à Madagascar regroupe plus de 160 ingénieurs et experts dans différentes technologies. Dans le cadre de la forte croissance de son activité, la société eTech a recruté plus de 30 personnes sur le début d’année 2015 et souhaite recruter encore 70 personnes avant la fin de l’année 2016.

### Organigramme de l’entreprise eTech

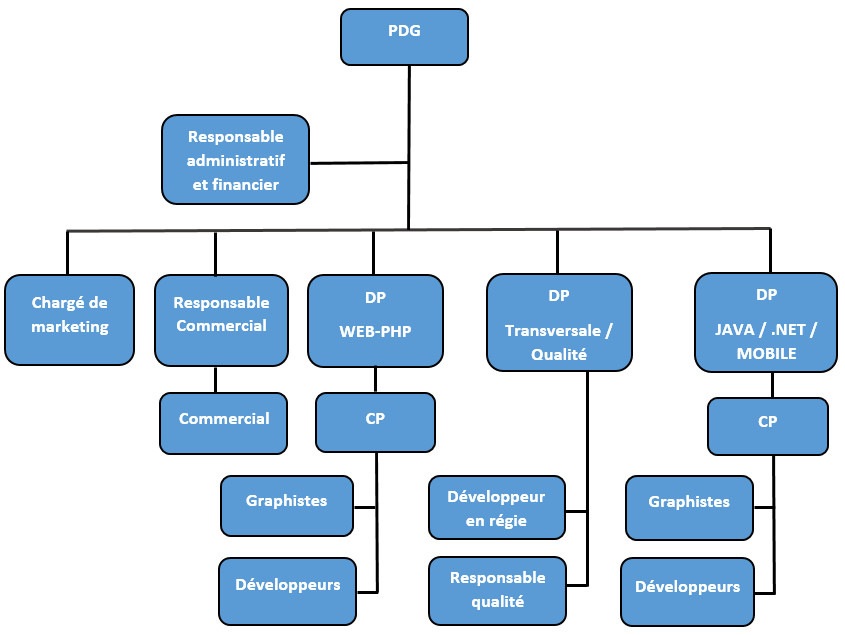


Figure : Organigramme de l’eTech (source eTech)

* PDG : Monsieur Pierre Paul ARDILE;
* Responsable Administratif et Financier : Madame Eva RALALAHARIMANANA ;
* Responsable Commercial : Madame Tiana NOHAVY ;
* Chargé de Marketing : Karen Radrigan ;
* Directeur de projet (.Net - Java - Mobile) : Monsieur Rija RAOBIVELO RABETOKOTANY;
* Directeur de projet (Transversale et Qualité): Monsieur Clarel RAKOTONDRAHAJA;
* Directeur de projet (Web-PHP) : Monsieur Andry RAMANAMBAHOAKA.

# PARTIE II : METHODE ET OUTILS

# CHAPITRE 3 : ETUDE DE L’EXISTANT

Le projet e-Asako est l’un des sous projet de e-Asa, qui est une plateforme de recherche de travail à temps partiel, pour des travails de vacances par exemple. Elle permet aussi de gérer les interventions des travailleurs.  
  
e-Asako est donc la partie qui permet au travailleur de mieux gérer les missions et les interventions. Cette partie permet donc  de:

* Voir le calendrier de mission
* Voir une carte où la mission aura lieu
* Voir le calendrier de congé
* Créer les absences ou congés des travailleurs
* Voir une les statistiques et les performances du travailleur
* Modifier les informations concernant le travailleur

Pour ce genre d’application, il faut prévoir d’ailleurs une solution afin gérer des données volumineuses. C’est d’ailleurs l’une des causes de la refonte de la plateforme elle-même.

Symfony2 est donc une solution pour le développement du projet et pour les données, on a utilisé la solution cloud Salesforce. De ce fait, les données sont gérées par cette dernière et notre projet utilise donc les données retournées par le web-service (SOAP).

# CHAPITRE 4 : METHODE D’ANALYSE ET DE CONCEPTION

## Présentation et justification du choix d'UML

La réalisation d'un site web doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un site afin de rendre ce développement plus fidèle aux besoins du client.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalité. La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

### Définition

Unified Modeling Language est un langage unifié de modélisation objets. Ce n'est pas une méthode, il ne donne pas de solution pour la mise en oeuvre d'un projet. C'est avant tout un **formalisme graphique** issu de notations employées dans différentes méthodes objets.

### Utilité d'UML

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

* Activité d'un objet/logiciel
* Acteurs
* Processus
* Schéma de base de données
* Composants logiciels
* Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple en langage Java, à partir des divers documents réalisés.

## L’Analyse Orientée Objet

L’AOO est une approche de l’ingénierie logicielle qui consiste à modéliser un système d’information comme un ensemble d’objets qui interagissent entre eux. Chaque objet représente une entité dans le système modélisé et est caractérisé par sa classe, ses attributs et ses méthodes.

L’AOO modélise le système indépendamment de la mise en œuvre. Différents modèles peuvent être schématisés pour décrire les structures statiques, dynamiques et temporelles du système. On propose ensuite une ou des solutions techniques pour représenter les éléments définis dans le système informatique et modélisés par l’AOO.

C’est une réflexion faite en amont de la création d'une application ou d'un logiciel. Elle a pour but de modéliser l'ensemble des besoins de l'utilisateur pour décrire ce qu'il faut faire. Elle définit les objets et les liens entre eux dans un langage humain avec une documentation à la clef.

### Une description du système logiciel

Durant l'analyse, tout ce qui concerne l'implémentation concrète de l'application (ou plus globalement du système logiciel) est oublié. On reste totalement hors de tout langage spécifique. On décrit ce que le logiciel doit faire, les classes et leurs attributs dans un langage humain avec des mots du domaine de l'utilisateur. L'utilisateur étant ici la personne qui va utiliser le logiciel.

|  |
| --- |
|  |

L'analyse orientée objet utilise souvent la notation d'UML (pour *Unified modeling language* en Anglais) afin de définir et d'illustrer la structure du logiciel en termes de classes et de composants ainsi que pour modéliser sa dynamique à l'aide de diagrammes d'interactions ou d'états/transitions. L'UML est devenu le langage de modélisation standard utilisé dans l'analyse et la conception orienté objet pour illustrer graphiquement les concepts du système.

### Le but de l’analyse orienté objet

Le modèle d'un système, composé d'objets combinés entre eux, peut être divisé en de multiples domaines lesquels sont analysés séparément. Un modèle d'analyse ne prendra pas en compte les contraintes d'implémentation comme la distribution, la persistance ou l'héritage ni comment le système sera construit.

La source de l'analyse est la déclaration des besoins nécessaires, et/ou des cas d'utilisation. Et le résultat d'une analyse orientée objet est une description de ce qui doit être construit, utilisant des concepts et des relations entre concepts, souvent décrites comme un modèle conceptuel. Dans le résultat de l'analyse est inclut une documentation qui est utilisée pour décrire ce qui doit être construit. Ce qui peut inclure une interface d'utilisation détaillée. Les contraintes d'implémentation, c'est-à-dire le "comment ", sont décidées pendant le processus de conception orientée objet (OOD).

La plupart des méthodologies de conception et d'analyse orientées objet modernes sont des " cas d'utilisation " conduits au travers des demandes, de la conception, de l'implémentation du test et du déploiement.

Pour conclure, il est intéressant d'ajouter qu'une des raisons de l'analyse objet réside dans son utilité pour le développement de programmes qui auront une longue durée de vie.

### La démarche UP

UP est un processus de développement d’un logiciel, centré sur l’architecture logicielle, piloté par les cas d’utilisation et focalisé sur la prévention des risques.

UP est une méthode de prise en charge du cycle de vie d’un logiciel et donc du développement pour les logiciels orientés objet. C’est une méthode générique, itérative et incrémentale. UML ne prend pas en charge le cycle de vie du logiciel, le processus de création et même la conception des modèles. Ainsi, pour transformer les besoins des utilisateurs en logiciel, nous devons présenter les caractéristiques suivantes :

* UP utilise UML ;
* UP est piloté par le cas d’utilisation ;
* UP est centré sur l’architecture ;
* UP est itératif et incrémental.

#### Caractéristiques

* UP est itératif et incrémental

Les méthodes de conception et de réalisation sont découpées en quatre phases :

* L’analyse des besoins : recensement des besoins, de l’étendue du projet ainsi que de la capacité du futur système ;
* L’élaboration : spécification fonctionnelle et technique, définition de l’architecture de base ;
* La construction du produit ;
* La transition : transition du produit vers les utilisateurs.
* UP est centré sur l’architecture logicielle

UP insiste sur le fait que l’architecture du futur système doit être au centre des efforts de l’équipe du projet pour maîtriser le système à développer. Il adopte et supporte plusieurs modèles et vues architecturaux.

* UP est conduit par le cas d’utilisation

Le processus de développement sera axé sur l’utilisateur. Les diagrammes de cas d’utilisation permettent d’illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels et leur ensemble constitue le modèle de cas d’utilisation qui dicte les fonctionnalités du système.

* UP est focalisé sur la prévention des risques

UP exige aux développeurs de développer en premier les domaines à haut risque. Et pour chaque itération, les tests et les validations occupent une place importante dans la phase de la transition.

#### Les activités

* L’expression des besoins

Comme son nom l’indique, elle permet de définir les différents besoins :

* Inventorier les besoins principaux et fournir une liste de leur fonction ;
* Recenser les besoins fonctionnels qui conduisent à l’élaboration du modèle de cas d’utilisation ;
* Appréhender les besoins non fonctionnels et livrer une liste des exigences.

Le model de cas d’utilisation présente le système du point de vue de l’utilisateur et représente sous forme de cas d’utilisation et d’acteurs les besoins du client.

* L’analyse

L’objectif de l’analyse est d’accéder à une compréhension des besoins et des exigences du client. Il s’agit de livrer des spécifications pour permettre de choisir la conception de la solution.

* La conception

Elle permet d’acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l’utilisation des composants et au système d’exploitation.

* L’implémentation

Elle est le résultat de la conception pour implémenter le système.

Ses objectifs principaux sont de planifier les intégrations pour chaque itération et de produire les classes et les sous-systèmes sous forme de code.

* Les tests

Ils permettent de vérifier les résultats de l’implémentation en testant la construction.

Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de test et prendre en compte le résultat de chacun.

#### Les phases

* L’analyse des besoins

Donne un point de vue du projet sous forme de produit fini. Cette phase porte essentiellement sur les besoins principaux, l’architecture générale du système, les risques majeurs, les délais et les coûts.

* L’élaboration

Elle reprend les éléments de la phase d’analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre.

Les taches qu’on y effectue sont les suivantes :

* La création d’une architecture de référence ;
* L’identification des risques ;
* La définition du niveau de qualité à atteindre ;
* La formulation des cas d’utilisation pour couvrir les besoins fonctionnels et planifier la phase de construction.
* La construction

C’est le moment où l’on construit le produit. L’architecture de référence se métamorphose en produit complet.

Le produit contient tous les cas d’utilisation que les chefs de projet, en accord avec les utilisateurs, ont décidé de mettre au point pour cette version.

* La transition

Le produit est en version béta, un groupe d’utilisateurs essaye le produit et détecte les anomalies et défauts.

Cette phase suppose des activités comme la formation des utilisateurs, la mise en œuvre d’un service d’assistance et la correction des anomalies constatées.

#### Les diagrammes utilisés

Comme cité auparavant, UP utilise les diagrammes UML à ceux-ci viennent s’ajouter quelques diagrammes propres à l’AOO.

Au final, les diagrammes utilisés sont :

* Le diagramme de cas d’utilisation ;
* L’inventaire des objets-métiers ;

La méthode standard pour faire l’inventaire des objets-métiers est de lire les spécifications de l’application à développer, de considérer si chaque nom peut être un objet-métier ou non et enfin de déterminer la responsabilité de chaque verbe pour un objet métier. La responsabilité d’un objet est soit la connaissance d’une information, soit l’action qu’il fait.

* Les CRC-Card (Classe, Responsabilité, Collaboration) ;

Elles sont utilisées pour passer de l’inventaire des objets-métier au diagramme de classe. Une CRC-Card est une petite carte de type carte d’indexation divisée en trois parties :

* Le nom de la classe ;
* La responsabilité ;
* Les collaborateurs.

Tableau 2 : Représentation d’une CRC-Card

|  |  |
| --- | --- |
| Nom de la classe | |
| Responsabilités | Collaborateurs |

* Le diagramme de classe métier ;

Selon le cas et selon les besoins, des diagrammes de vue dynamiques peuvent être ajoutés pour nous aider à matérialiser les actions et l’état des objets concernés à un instant donné par la méthode concernée.

Ces diagrammes sont :

* Le diagramme de collaboration ;
* Le diagramme de séquence ;
* Le diagramme d’état transition ;
* Le diagramme d’activité.

En conclusion, UML est un outil précieux, mais, pour bien l'utiliser et en faire un instrument de lisibilité, il nous faut l'accompagner d'un mode d'emploi pour l'élaborer, il nous faut reprendre les questions dans la tradition du génie logiciel et suivre les démarches de conception et d'analyse :

* Analyse de problème en utilisant processus unifié UP
* Utilisation d'un langage de modélisation UML
* Etude préalable, construction ainsi tests et mise au point.

## La conception orienté objet

### Notion

Au cours de la conception orienté objet, le concepteur applique les contraintes techniques et fonctionnelles sur le model conceptuel produite par l’analyse orienté objet (diagramme de classe). De telle contrainte peut inclure non seulement des contraintes imposées par architecture choisie mais aussi des contraintes fonctionnelles, technologiques ou environnementales telles que les débits de transaction, les temps de réponse, la plateforme d’exécution, l’environnement de développement, le langage de programmation.

Il en résulte un modèle de domaine, c’est-à-dire la description détaillé de la façon dont le système doit être construit.

### Les principes de la COO

Les principes de conception représentent un ensemble de « Best practices » qui nous aide surtout à ne pas avoir une mauvaise conception et donc un mauvais système. Il y a 3 principales caractéristiques d’un mauvais système :

* **Rigidité :** il est difficile d’appliquer des changements car chaque changement affecte plusieurs parties du système
* **Fragilité :** on veut faire des changements sur une partie du système et d’autres parties deviennent buggées
* **Immobilité :** il est difficile de réutiliser une brique ou module que vous avez conçu dans un système, car il est difficile de l’en extraire.

Pour résoudre ce problème, voici quelques principes qu’il faut suivre. Il s’agit des principes SOLID

* [**Responsabilité unique (Single Responsibility Principle)**](https://fr.wikipedia.org/wiki/Principe_de_responsabilit%C3%A9_unique)
* [**Ouvert/fermé (Open/closed principle)**](https://fr.wikipedia.org/wiki/Principe_ouvert/ferm%C3%A9)
* [**Substitution de Liskov (Liskov substitution Principle)**](https://fr.wikipedia.org/wiki/Principe_de_substitution_de_Liskov)
* [**Ségrégation des interfaces (Interface segregation principle)**](https://fr.wikipedia.org/w/index.php?title=Principe_de_s%C3%A9gr%C3%A9gation_des_interfaces&action=edit&redlink=1)
* [**Inversion des dépendances (Dependency Inversion Principle)**](https://fr.wikipedia.org/wiki/Inversion_des_d%C3%A9pendances)

#### Single Responsibility Principle

Ce principe stipule que si nous avons 2 raisons de changement pour une classe, nous devrions diviser la fonctionnalité en 2 classes distinctes.

#### Open/closed principle

Une conception intelligent doit prendre en compte les fréquents changements durant les phases de développement et de maintenance d’une application essentiellement l’ajout de nouvelle fonctionnalité. La règle d’or est qu’on ne doit toucher qu’un minimum de codes.

Le principe OPEN/CLOSE stipule donc que les entités logicielles comme les classes, modules, fonctions doivent être ouvertes pour une extension mais fermées pour une modification.

#### Liskov segregation principle

Quand nous concevons un programme informatique, nous manipulons souvent des hiérarchies entre les classes. En créant des classe dérivées à partir des classe existantes, nous devons nous assurer que la nouvelle classe fasse juste une extension sans remplacer les fonctionnalités de la classe mère sinon cette classe dérivée produira certainement des effets indésirables. Le principe de substitution de Liskov stipule que si nous utilisons une classe de base, la référence à cette classe peut être remplacée par la classe dérivée sans affecter les fonctionnalités du module ou de l’application.

#### Interface segregation principle

Quand nous concevons une application, nous devons prendre en compte le fait de mettre abstrait un module qui contient plusieurs sous modules. Le principe de ségrégation d’interface stipule que les classes clientes ne devront pas être forcées à implémenter des interfaces qu’elles n’utilisent pas. Au lieu d’une seule et grande classe d’interface, il vaudrait mieux utiliser plusieurs petites classes.

#### Dependency Inversion principle

Quand nous faisons la conception d’une application, nous pouvons considérer les classes de bas niveau comme des classes qui implémentent des opération basiques (accès au fichier, …) et les classes de haut niveau comme des classe qui encapsule les opérations complexes (logique métier) et ces dernières utilisent souvent des classes de bas niveau.

## Les Designs Pattern

Les designs patterns sont des architectures réutilisables mis en place en réponse à un problème de conception donné.

Ils aident dans l’amélioration de la qualité de développement et dans le raccourcissement de la durée de celui-ci.

On distingue trois grandes catégories de design patterns :

* Les design patterns créationnels : qui définissent les mécanismes pour l’instanciation et/ou l’initialisation des objets ;
* Les design patterns structuraux : qui organisent les interfaces/classes entre elles ;
* Les design patterns comportementaux : qui définissent la communication entre les classes et leur responsabilité.

Les designs patterns ont pour principal objectif de :

* Réduire le couplage entre les différentes classes d’une application ;
* Réduire le temps de développement d’une application ;
* Faciliter la maintenance d’une application ;
* Faciliter la communication entre les différents développeurs.

Cependant, l’utilisation des designs patterns présente quelques inconvénients :

* Difficulté à identifier quand un pattern s’applique nécessite un apprentissage et de l’expérience ;
* Les patterns peuvent dépendre d’autres patterns ;
* Les patterns sont nombreux (23 patterns répertoriés dans un ouvrage de référence : le GoF).

Selon leur rôle, les patterns sont répartis en trois groupes :

* **Création:** AbstractFactory, Factory, Builder, Prototype, Singleton;
* **Structuration:** Adapter, Bridge, Composite, Facade, Proxy, Flyweight, Decorator;
* **Comportement:** Interpreter, Command, Chain of Responsability, Iterator, Mediator, Template, Memento, Observer, State, Strategy, Visitor.

Pour les besoins du projet, les principaux designs patterns utilisés seront le pattern Singleton et le Pattern Factory lors de la création des objets pour réduire le couplage entre les classes et pour un souci d’économie des espaces alloués en mémoire étant donné que nous travaillerons sur des appareils avec des capacités limitées.

**Description d’un pattern**

Les Design Patterns sont représentés par :

Nom : qui permet de l’identiﬁer clairement ;

Problématique : description du problème auquel il répond ;

Solution : description de la solution souvent accompagnée d’un schéma UML.

**Nom** : Singleton

**Problématique :** La classe ne doit avoir qu'une seule instance.

**OBJECTIFS** :

* Restreindre le nombre d'instance d'une classe à une et une seule ;
* Fournir une méthode pour accéder à cette instance unique.

**RESULTAT :** Le Design Pattern permet d'isoler l'unicité d'une instance.

**EXEMPLE :**

* Connexion à une base de données.
* Gestion de la mémoire.

**DIAGRAMME DE CLASSE :**

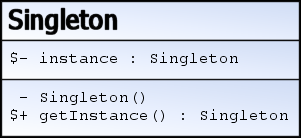


Figure : Diagramme de classe d’une classe de type Singleton

**Nom**: [Fabrique (FactoryMethod ou Virtual Constructor)](http://rpouiller.developpez.com/tutoriel/java/design-patterns-gang-of-four/?page=page_2#LIV-C)

**Problématique :** Le système utilise des objets qui sont regroupés en famille. Selon certains critères, le système utilise les objets d'une famille ou d'une autre. Le système doit utiliser ensemble les objets d'une famille.

**OBJECTIFS** : Fournir une interface pour créer des objets d'une même famille sans préciser leurs classes concrètes.

**RESULTAT :** Le Design Pattern permet d'isoler l'appartenance à une famille de classes.

**DIAGRAMME DE CLASSE :**

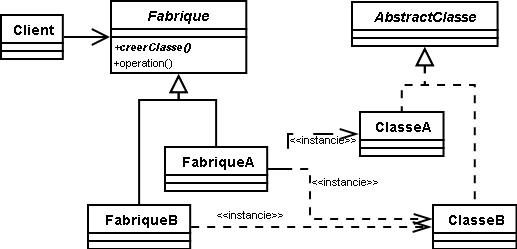


Figure : Diagramme de classe d’un Factory

**Nom:** Model-View-ViewModel (MVVM)

**Problématique :** La vue ne doit pas effectuer aucun traitement

**OBJECTIFS** **:** vise à architecturer efficacement une application afin d’en faciliter la maintenabilité et la testabilité.

**RESULTAT :** Le but de MVVM est de faire en sorte que la vue n’effectue aucun traitement, elle ne doit faire qu’afficher les données présentées par le view-model. C’est le view-model qui a en charge de faire les traitements et d’accéder au modèle.

## Les technologies utilisés

### Le framework Symfony2

#### Histoire

L'agence web française SensioLabs est à l'origine du [framework](https://fr.wikipedia.org/wiki/Framework) *Sensio Framework*[3](https://fr.wikipedia.org/wiki/Symfony#cite_note-3). À force de toujours recréer les mêmes fonctionnalités de gestion d'utilisateurs, gestion [ORM](https://fr.wikipedia.org/wiki/Mapping_objet-relationnel), etc., elle a développé ce [framework](https://fr.wikipedia.org/wiki/Framework) pour ses propres besoins. Comme ces problématiques étaient souvent les mêmes pour d'autres développeurs, le code a été par la suite partagé avec la communauté des développeurs [PHP](https://fr.wikipedia.org/wiki/PHP).

Le projet est alors devenu *Symfony* (conformément à la volonté du créateur de conserver les initiales *S* et *F* de *Sensio Framework*), puis *Symfony2* à partir de la version 2. La version 2 de *Symfony* casse la compatibilité avec la branche 1.x.

Ce [framework](https://fr.wikipedia.org/wiki/Framework) et ses fonctionnalités seront implémentés comme base de la version 8 du [système de gestion de contenu](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu) [Drupal](https://fr.wikipedia.org/wiki/Drupal).

#### Utilisation

Symfony est utilisé par [Askeet](https://fr.wikipedia.org/w/index.php?title=Askeet&action=edit&redlink=1) et beaucoup d'autres applications, dont [Delicious](https://fr.wikipedia.org/wiki/Delicious) et les vingt millions d'utilisateurs de Yahoo! Bookmarks. En 2009, [Dailymotion](https://fr.wikipedia.org/wiki/Dailymotion) a porté son code sur Symfony et a continué la transition[9](https://fr.wikipedia.org/wiki/Symfony#cite_note-9).

Les projets les plus connus utilisant Symfony sont eZPublish (utilisation de Symfony 2 en *full stack*), [Drupal](https://fr.wikipedia.org/wiki/Drupal) (utilisation de composants Symfony), [phpBB](https://fr.wikipedia.org/wiki/PhpBB), [Laravel](https://fr.wikipedia.org/wiki/Laravel), [Joomla!](https://fr.wikipedia.org/wiki/Joomla%21), [Composer](https://fr.wikipedia.org/wiki/Composer_%28logiciel%29), [Magento](https://fr.wikipedia.org/wiki/Magento), [Piwik](https://fr.wikipedia.org/wiki/Piwik).

#### Fonctionnalité

Symfony utilise la version 5 de PHP (PHP ≥ 5.2.4 pour Symfony 1.x, PHP ≥ 5.3.3 pour Symfony 2.x et PHP ≥ 5.5.9 pour Symfony 3.x).

Symfony (1.x) propose entre autres :

* une séparation du code en trois couches, selon le modèle MVC, pour une plus grande [maintenabilité](https://fr.wikipedia.org/wiki/Maintenabilit%C3%A9) et évolutivité ;
* des performances optimisées et un système de cache afin d'assurer des temps de réponse optimaux ;
* une gestion des [URL](https://fr.wikipedia.org/wiki/Uniform_Resource_Locator) parlante, permettant à une page d'avoir une URL distincte de sa position dans l'arborescence ;
* un système de configuration en cascade utilisant pleinement le langage [YAML](https://fr.wikipedia.org/wiki/YAML) ;
* un générateur de [back-office](https://fr.wikipedia.org/wiki/Application_%28informatique%29#Application_de_back_office) et un lanceur de module ([*scaffolding*](https://fr.wikipedia.org/wiki/%C3%89chafaudage_%28programmation%29)) ;
* l'[internationalisation](https://fr.wikipedia.org/wiki/Internationalisation_de_logiciel) native ;
* une couche de [mapping objet-relationnel](https://fr.wikipedia.org/wiki/Mapping_objet-relationnel) (ORM) et une d'abstraction de données ;
* le support d'[AJAX](https://fr.wikipedia.org/wiki/Ajax_%28informatique%29) ;
* une architecture extensible permettant créations et utilisations de plugins.

Symfony fournit une interface en ligne de commande pour améliorer la productivité en créant un code de base modifiable à volonté.

### Langage H.T.M.L

#### Définition

L’Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C’est un langage de balisage permettant d’écrire de l’hypertexte, d’où son nom.

HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques.

Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l’accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

#### Historique

* 1989-1992: Origine

HTML est une des trois inventions à la base du World Wide Web, avec le Hypertext Transfer Protocol (HTTP) et les adresses web. HTML a été inventé pour pouvoir écrire des documents hypertextuels liant les différentes ressources d’Internet avec des hyperliens. Aujourd’hui, ces documents sont appelés «page web».

En août 1991, lorsque Tim Berners-Lee annonce publiquement le web sur Usenet, il ne cite que le langage SGML, mais donne l’URL d’un document de suffixe .html.

Dans son livre Weaving the web, Tim Berners-Lee décrit la décision de baser HTML sur SGML comme étant aussi « diplomatique » que technique : techniquement, il trouvait SGML trop complexe, mais il voulait attirer la communauté hypertexte qui considérait que SGML était le langage le plus prometteur pour standardiser le format des documents hypertexte.

En outre, SGML était déjà utilisé par son employeur, l’Organisation européenne pour la recherche nucléaire (CERN).

Les premiers éléments du langage HTML comprennent le titre du document, les hyperliens, la structuration du texte en titres, sous-titres, listes ou texte brut, et un mécanisme rudimentaire de recherche par index.

La description de HTML est alors assez informelle et principalement définie par le support des divers navigateurs web contemporains. Dan Connolly a aidé à faire de HTML une véritable application de SGML.

* 2000-2006: XHTML

Le développement de HTML en tant qu’application du Standard Generalized Markup Language (SGML) est officiellement abandonné au profit de XHTML, application d’Extensible Markup Language (XML).

Cependant, en 2004, des fabricants de navigateurs web créent le web Hypertext Application Technology Working Group (WHATWG) dans le but, notamment, de relancer le développement du format HTML et de répondre aux nouveaux besoins sur une base technologique jugée plus aisément implémentable que celle du XHTML 2.0 en cours de conception. Ceci s’inscrit dans le contexte d’une contestation plus générale du mode de fonctionnement du W3C, réputé trop fermé par une partie des développeurs et designers web.

* 2007 à nos jours: HTML5 et abandon du XHTML 2

En mars 2007, tirant la conséquence des réticences d’une partie de l’industrie et des concepteurs de contenus web face à XHTML 2.0, le W3C relance le développement de HTML et crée un nouveau groupe de travail encadré par Chris Wilson (Microsoft) et initialement Dan Connolly (W3C), maintenant Michael Smith (W3C).

Il s’agit notamment de faire évoluer HTML pour décrire la sémantique des documents mais aussi les applications en ligne;

* de parvenir à un langage extensible via XML tout en maintenant une version non XML compatible avec les analyseurs syntaxiques (*parsers*) HTML des navigateurs actuels;
* et d’enrichir les interfaces utilisateurs avec des contrôles spécifiques: barres de progrès, menus, champs associés à des types de données spécifiques.

#### Structure et Syntaxe

À l’origine, HTML a été conçu pour baliser (ou marquer) simplement le texte, notamment pour y ajouter des hyperliens. On utilisait un minimum de balises, comme dans le document HTML suivant :

<!DOCTYPE html>

<html>

<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

<div>TODO write content</div>

</body>

</html>

### Language P.H.P

#### Définition

PHP: Personal Home Page ou Hypertext Preprocessor, plus connu sous son sigle PHP est un langage de programmation compilé à la volée libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes.

#### Historique

Le langage PHP fut créé en 1994 par Rasmus Lerdorf pour son site web. C'était à l'origine une bibliothèque logicielle en Perl dont il se servait pour conserver une trace des visiteurs qui venaient consulter son CV.

Au fur et à mesure qu'il ajoutait de nouvelles fonctionnalités, Rasmus a transformé la bibliothèque en une implémentation en langage C, capable de communiquer avec des bases de données et de créer des applications dynamiques et simples pour le Web.

Rasmus décida alors en 1995 de publier son code, pour que tout le monde puisse l'utiliser et en profiter. PHP s'appelait alors PHP/FI (pour Personal Home Page Tools/Form Interpreter).

En 1997, deux étudiants, Andi Gutmans et Zeev Suraski, redéveloppèrent le cœur de PHP/FI. Ce travail aboutit un an plus tard à la version 3 de PHP, devenu alors PHP: Hypertext Preprocessor. Peu de temps après, Andi Gutmans et Zeev Suraski commencèrent la réécriture du moteur interne de PHP. Ce fut ce nouveau moteur, appelé Zend Engine - le mot Zend est la contraction de ZEev et aNDi - qui servit de base à la version 4 de PHP.

#### Structure et Syntaxe

* Quelques exemples du traditionnel:

<?php

// forme la plus simple, recommandée

echo 'Hello World';

?>

***echo*** étant une structure du langage, il est possible et même recommandé de ne pas mettre de parenthèses, ce qui lui permet en plus d'accepter plusieurs paramètres :

<?php

echo 'Hello ', 'World';

?>

Il est aussi possible d'utiliser la version raccourcie :

<?='Hello World'?>

Résultat affiché :

Hello World

* Programmation orientée objet (POO)

Comme en C++, les versions actuelles de PHP (à partir de PHP 5) permettent de programmer en orienté objet, en créant des classes contenant des attributs et des méthodes, des instances de classes. L'héritage entre les classes existe aussi. Voici un exemple de création d'une classe :

<? php

class Control\_Etat extends CI\_Controller {

function epuiser\_control ()

{

$data ['title'] ='Epuiser';

$data ['content']='D\_etat/epuiser';

$this->load->view ('template/template',$data);

}

function perimer\_control ()

{

$data['title']='Perimer';

$data['content']='D\_etat/perimer';

$this->load->view('template/template',$data);

}

}

?>

### Langage JavaScript

#### Définition

Le JavaScript (JS) est un langage dit "client" qui est exécuté par le navigateur Web (il n'est pas nécessaire de lire ses pages depuis un serveur Web comme pour PHP). Il est desactivable sur la plupart des navigateurs Web en modifiant le menu "options». Ce langage permet entre autres de développer l'interactivité d'un site (pop-up, pop-under, contrôle de formulaire, chargement d'images...).

C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés.

#### Historique

Le langage a été créé en 1995 par Brendan Eich (Brendan Eich étant membre du conseil d'administration de la fondation Mozilla) pour le compte de Netscape Communications Corporation. Le langage, actuellement à la version 1.8.2 est une implémentation de la 3e version de la norme ECMA-262 qui intègre également des éléments inspirés du langage Python. La version 1.8.5 du langage est prévue pour intégrer la 5e version du standard ECMA.

Le développement d'une version 2 de JavaScript basée sur les spécifications du langage ECMAScript Édition 4 en collaboration avec Adobe, le projet Tamarin, a aujourd'hui été abandonné. La norme un temps rédigée par l'ECMA dans le document ECMA-262 4e édition est en effet obsolète et a été remplacée par ES5.

#### Structure et Syntaxe

Dans le cas no1, on doit inclure le fichier .js directement dans notre page web et toujours entre les balises <head> et </head> comme ceci:

...

<head>

<script type="text/javascript" src="./dossier\_javascript/fichier.js"></script>

</head>

...

Dans le cas no 2, le code doit se tenir entre les balises <script> et </script> comme ceci :

...

<head>

<!-- (déconseillé) -->

<script>

// Votre script

</script>

<!-- OU -->

<script language="javascript">

// Votre script

</script>

<!-- OU (conseillé) -->

<script type="text/javascript">

// Votre script

</script>

</head>

...

Dans le dernier cas, voici un exemple très simple :

<input type="button" onClick="..." />

Voyons comment écrire un texte à l'écran :

// Affichons le message "Bonjour tout le monde !"

document.write("Bonjour tout le monde !");

// On peut ajouter du html

document.write("<b>Bonjour</b><br /><br /><img src=\"mon\_image.jpg\" />

<br /><br /><u>tout le monde !</u>");

### Salesforce

#### Historique

Salesforce a été créé en 1999 par [Marc Benioff](https://fr.wikipedia.org/wiki/Marc_Benioff), Parker Harris, Dave Moellenhoff et Frank Dominguez et a fait ses débuts en tant que société spécialisée dans les logiciels [SaaS](https://fr.wikipedia.org/wiki/SaaS) (Software as a service). En juin 2004, la société a été introduite au [NASDAQ](https://fr.wikipedia.org/wiki/NASDAQ) sous le symbole boursier CRM, apportant ainsi 110 millions de dollars.

#### Localisation

Outre son [siège social](https://fr.wikipedia.org/wiki/Si%C3%A8ge_social) à [San Francisco](https://fr.wikipedia.org/wiki/San_Francisco) en [Californie](https://fr.wikipedia.org/wiki/Californie) sur la [côte Ouest des États-Unis](https://fr.wikipedia.org/wiki/C%C3%B4te_Ouest_des_%C3%89tats-Unis), Salesforce a plusieurs sièges régionaux à [Dublin](https://fr.wikipedia.org/wiki/Dublin) (pour l'[Europe](https://fr.wikipedia.org/wiki/Europe), le Moyen-Orient et l’Afrique), à [Singapour](https://fr.wikipedia.org/wiki/Singapour) (pour l'Asie-Pacifique) et à [Tokyo](https://fr.wikipedia.org/wiki/Tokyo) (pour le Japon). Elle a également des implantations dans de nombreux pays européens comme la France, le Royaume-Uni ou l’Allemagne

En France, Salesforce dispose de deux implantations, à Paris et à [Grenoble](https://fr.wikipedia.org/wiki/Grenoble) où l’entreprise a ouvert un centre de recherche et développement en mai 2013.

Par ailleurs, l'entreprise centralise ses données dans plusieurs centres d'hébergement : aux [États-Unis](https://fr.wikipedia.org/wiki/%C3%89tats-Unis), à [Tokyo](https://fr.wikipedia.org/wiki/Tokyo) et dans un centre ouvert en 2014 au Royaume Uni[7](https://fr.wikipedia.org/wiki/Salesforce.com#cite_note-7). Salesforce a annoncé le lancement de deux nouveaux centres de données en Europe, qui ouvriront en France et en Allemagne courant 2015.

#### Produits et services

Voici donc des services que propose Salesforce. Basé sur la [base de données](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es) *database.com* et une place de marché d'applications d'entreprise *appexchange*, les solutions de Salesforce.com sont regroupées en plusieurs grandes catégories : Sales Cloud, Services Cloud, Force.com et Appexchange, Chatter et Community Cloud, Marketing Cloud et Analytics Cloud.

* **Sales Cloud :** Sales Cloud est une application dédiée à la [productivité](https://fr.wikipedia.org/wiki/Productivit%C3%A9) d'une [force de vente](https://fr.wikipedia.org/wiki/Force_de_vente) en ligne accessible à travers une connexion Internet par mobile ou ordinateur. Elle permet de consulter à tout moment le profil de clients ou de prospects, de gérer des campagnes marketing et de rassembler toutes les informations essentielles pour optimiser la force de vente d'une entreprise.
* **Services Cloud :** Services Cloud est une plate-forme permettant d'optimiser la [gestion de la relation client](https://fr.wikipedia.org/wiki/Gestion_de_la_relation_client) d’une entreprise. Elle offre divers services comme la centralisation d'informations clients pertinentes et l’historique provenant de centre d’appels ou encore la possibilité pour chaque utilisateur de participer aux conversations sur leur entreprise au sein des réseaux sociaux. Services Cloud fournit aussi des outils d’analyse, des services d'[e-mail](https://fr.wikipedia.org/wiki/E-mail) et de [tchat](https://fr.wikipedia.org/wiki/Tchat) pour permettre aux sociétés de communiquer en temps réel avec leurs clients et ainsi optimiser leur service client.
* **Appexchange et Force.com :** Force.com est une plate-forme permettant de créer des applications d’entreprise hébergées sur le web et ne nécessitant aucun logiciels ni matériels. Ainsi toute entreprise peut créer son site web et ses propres applications pour diverses fonctions comme les ressources humaines, la gestion de projet, la logistique… Avec le lancement de la Salesforce1 Platform, les fonctionnalités de Force.com se sont enrichies : les applications métiers développées par les entreprises sur la plate-forme peuvent désormais être déployées sur tout type de terminal mobile.  
  En 2014, Appexchange est une [place de marché](https://fr.wikipedia.org/wiki/Place_de_march%C3%A9_%28commerce_%C3%A9lectronique%29) ([App Store](https://fr.wikipedia.org/wiki/App_Store) pour entreprise) qui offre d'acheter l'une des 2 800 solutions d'[éditeurs de logiciels](https://fr.wikipedia.org/wiki/%C3%89diteurs_de_logiciels) partenaires ou de vendre des [logiciels](https://fr.wikipedia.org/wiki/Logiciels) à plus de 100 000 clients salesforce.com.
* **Heroku :** Heroku est un service de cloud computing de type plate-forme en tant que service ([PaaS](https://fr.wikipedia.org/wiki/Plate-forme_en_tant_que_service)). Il permet le déploiement très rapide d'applications dans le cloud, avec une gestion souple du scaling horizontal au travers d'un modèle de gestion des processus emprunté à Unix et adapté au Web. Il supporte différents langages comme Python, Ruby, PHP, Java. Facebook le propose par défaut aux développeurs qui veulent concevoir des applications sur le réseau social. Heroku est également intégré à Force.com et permet ainsi le transfert de données entre ces deux plates-formes.
* **Chatter et Community Cloud :** Créé en juin 2010, Chatter est une plate-forme collaborative en temps-réel. Elle permet à tous les collaborateurs d’une même entreprise de mettre à jour leur profil et ainsi échanger et partager tout type d’informations sur leurs activités professionnelles du quotidien. Ce service de [réseau social d'entreprise](https://fr.wikipedia.org/wiki/R%C3%A9seau_social_d%27entreprise) fait partie de la plateforme Salesforce1. Son point fort, selon Salesforce, est qu’il intègre des process métiers avec lesquels les employés peuvent collaborer, comme la validation d’une note de frais, d’une commande… Depuis 2014, Community Cloudpermet de créer des plateformes communautaires personnalisées en ligne et d’optimiser la collaboration au sein des entreprises. Accessibles en mobilité, reliés aux réseaux sociaux et directement connectés aux processus métiers, elles permettent de connecter directement les clients, partenaires et collaborateurs.
* **Marketing Cloud :** Lancée en 2012, Marketing Cloud est une plateforme marketing permettant d’établir des relations personnalisées avec les clients, à travers tous les canaux, et en mobilité : e-mails, web, réseaux sociaux, objets connectés… Agrégeant les fonctionnalités de Radian6 (écoute et captation d'informations), buddy media (publication de contenus), ExactTarget (marketing multicanaux et parcours client) et social.com (publicité), la plateforme permet d’organiser et de suivre les campagnes sur tous les canaux. Elle permet également d’extraire et d’exploiter l’ensemble des données recueillies lors des interactions avec les clients, pour leur proposer du contenu d’autant plus ciblé, personnalisé et en adéquation avec leurs besoins.
* **Analytics Cloud :** Analytics Cloud est la 1re plateforme cloud analytique, reposant sur la technologie Wave. Elle est accessible depuis ordinateurs, smartphones et tablettes. La plateforme permet aux entreprises d’analyser leurs données, de partager les analyses entre collaborateurs et d’en tirer les informations nécessaires, afin d’identifier des opportunités commerciales.

### Les services web

De plus en plus d’entreprises se tendent vers une architecture dont les données sont déportées des applications pour une meilleure distribution d’informations.

Cette approche permet de ne pas recréer les services de consommation de la base de données. Nombreux sont ceux qui rendent  leurs applications accessibles sur le web d’où l’appel à des Web Services.

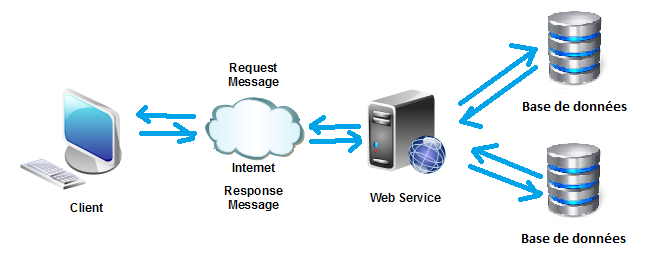


Figure : Schéma du fonctionnement des WS

Un **Web Service** est un protocole applicatif de haut niveau sur le web d’échange d’informations à distance avec un grand nombre de clients. Les aspects non-fonctionnels comme le transport de l’information sont délégués aux couches inférieures comme TCP/IP.

En simplifié, c’est une interface de distribution de données, qui peut répondre à plusieurs demandes HTTP et exposer différentes données venant de différentes sources (plusieurs bases de données, lire des fichiers etc…).

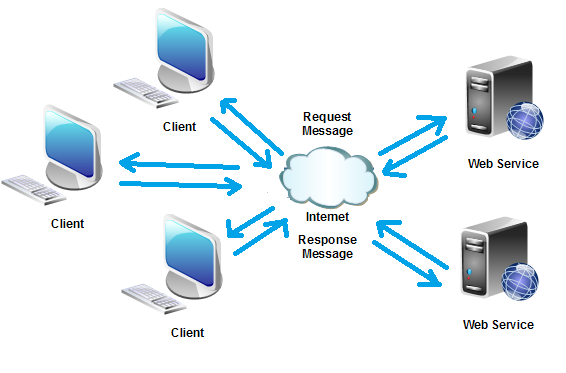


Figure : Schéma explicatif du WS

Nous ne nous attarderons pas sur les différents types d’architecture des Web Services mais uniquement sur leurs natures.

Il existe deux grandes familles de Web Services :

* Les Web Services de type **SOAP** (protocole)
* Les Web Services de type **REST** (style d’architecture)

#### Les WS de type SOAP

**SOAP,** soit l’acronyme *Simple Object Access Protocol*, est un protocole applicatif qui est composé de deux parties :

* Une **enveloppe** contenant des informations sur le message et son acheminement
* Un **modèle de donnée**s avec son format

Sa spécification décrit l’ensemble des types d’échange possibles entre applications.

Les échanges se font par des communications client / Serveur, dans l’ordre suivant :

* Le serveur expose son descripteur : WSDL
* Le client récupère le descripteur pour enrichir et comprendre les données à envoyer
* Le serveur reçoit la requête du client pour envoyer une réponse.

Ce qui implique un fort couplage entre client/serveur (une modification de l’API implique ainsi une évolution côté client).



Figure : Exemple de WSDL

Un protocole orienté sur du XML, le descripteur permet de voir les définitions des objets et services pouvant être consommés.

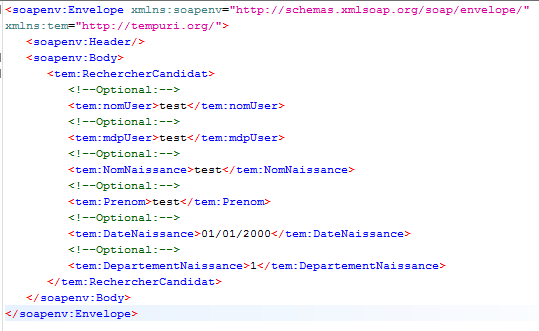


Figure : Exemple de requête

La requête consiste à « Rechercher un candidat » avec les différents paramètres en entrée.



Figure : Exemple de réponse

La réponse est structurée avec les paramètres de retour.

# PARTIE III : RESULTAT OBTENU

# CHAPITRE 5 : INTERFACE GRAPHIQUE

## Page d’authentification

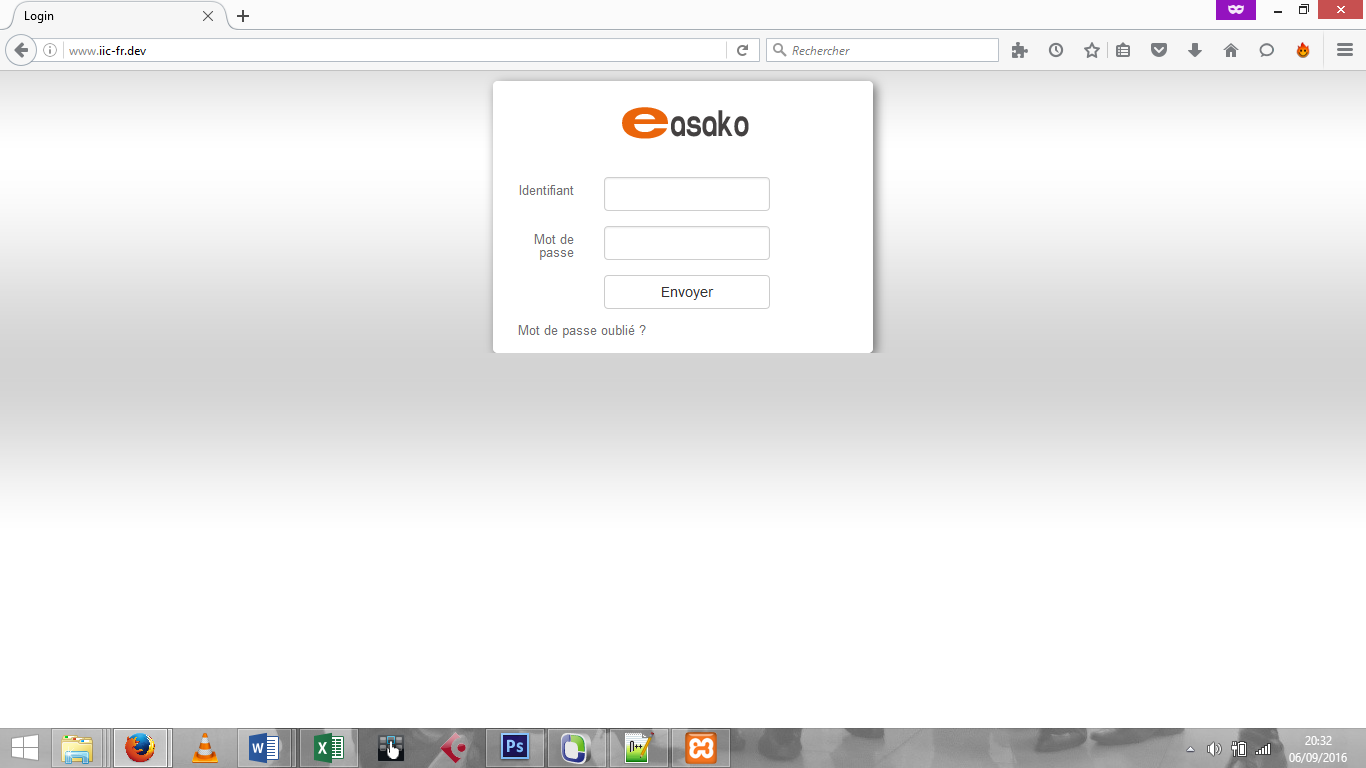


Figure : Page d'authentification

C’est sur cette page que les travailleurs et employés doivent s’identifier pour accéder à un espace où ils peuvent consulter le calendrier de ses missions et ses congés, modifier leurs informations personnelles….

## Page d’accueil

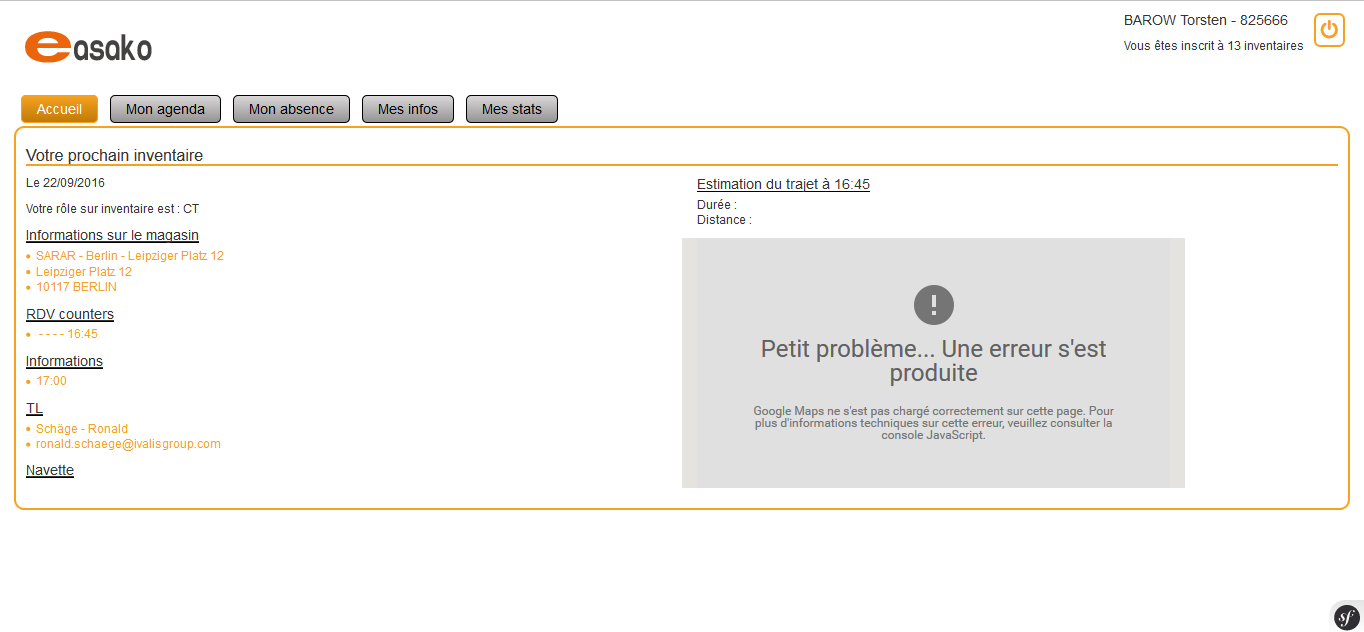


Figure : page d'accueil

On peut constater le nombre d’inventaire dont l’employé y est inscrit. Ici, notre employé est inscrit à 13 inventaires.

La page d’accueil montre aussi les détails de la prochaine mission de l’employé. Une carte google map est disponible pour situer l’endroit où la mission aura lieu.

## Page de calendrier ou Agenda

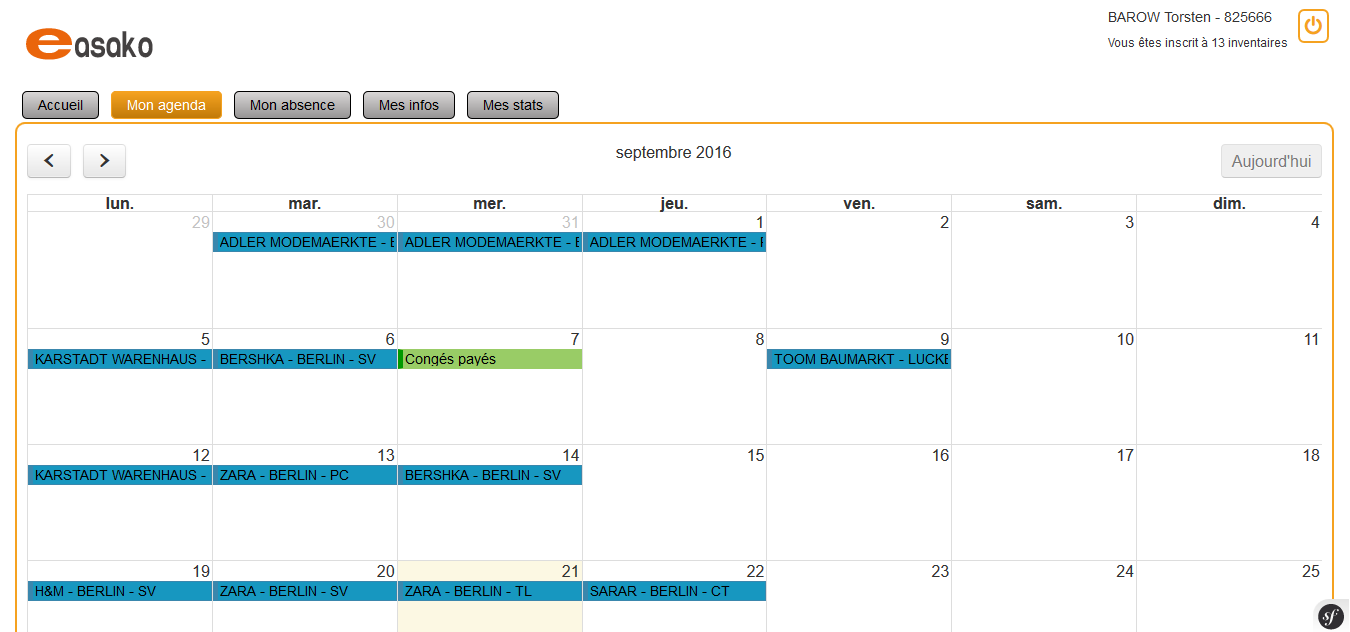


Figure : Page de calendrier ou Agenda

L’employé ou collaborateur peut donc consulter les calendriers des inventaires dont il s’est inscrit. Les missions ou inventaires sont les cases en bleu. Les congés sont donc en verts.

On peut d’ailleurs exporter ce calendrier en un classeur Excel, que le travailleur pourra consulter lorsqu’il ne sera pas en ligne, ou qu’il pourra imprimer aussi.

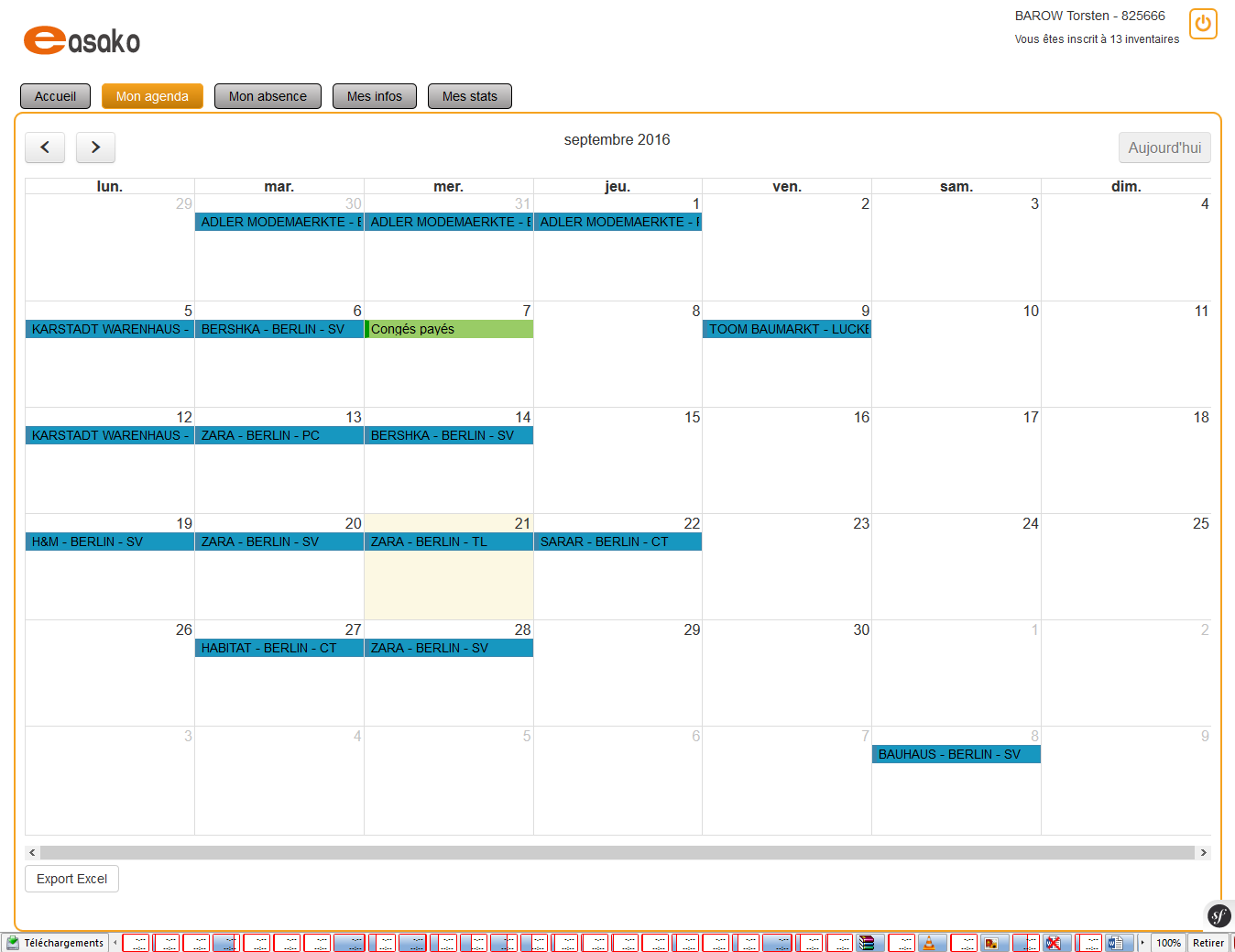


Figure : Agenda avec le bouton Exporter Excel

Le travailleur peut exporter un fichier Excel de son calendrier d’inventaire en cliquant sur le bouton Export Excel.

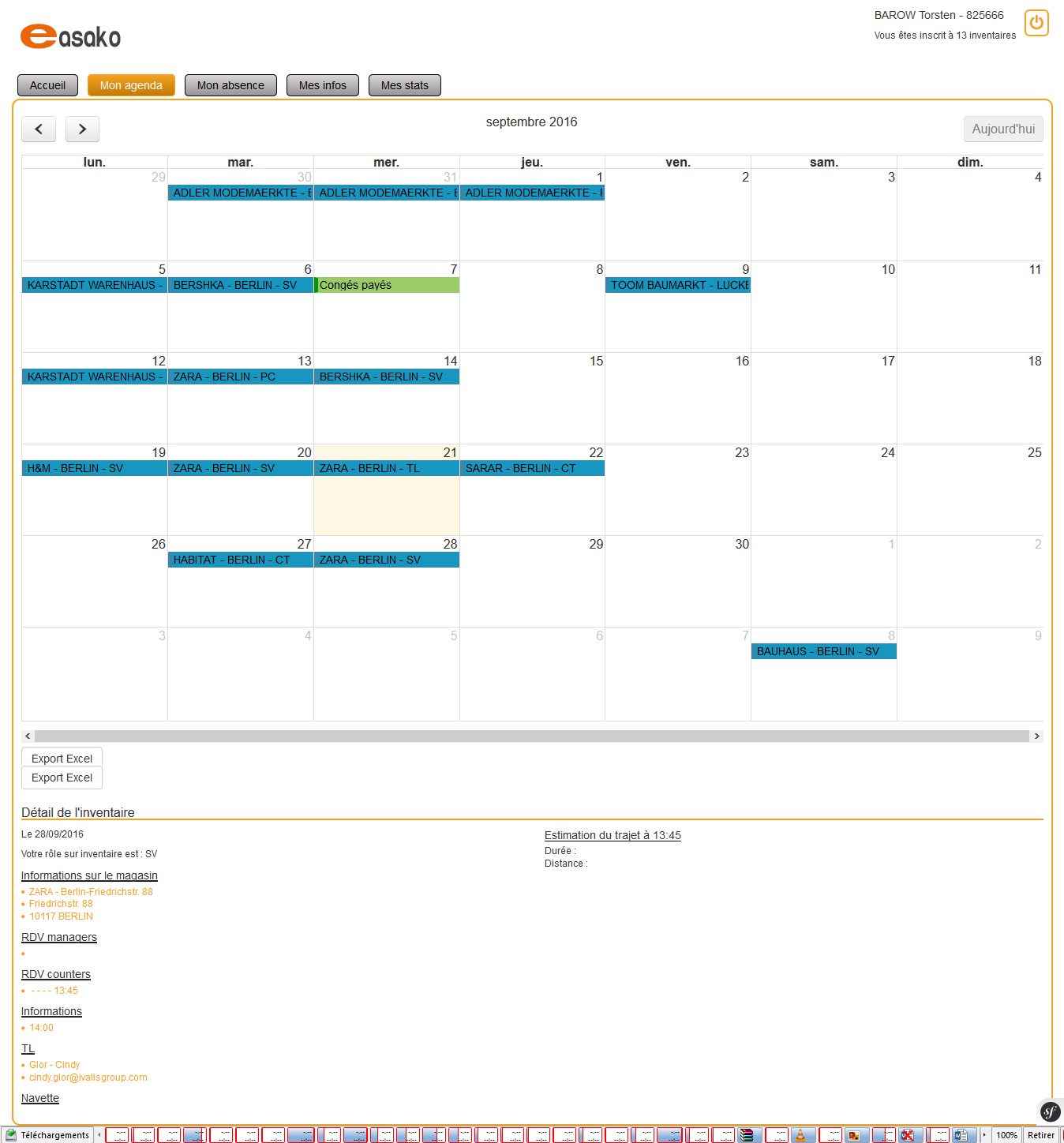


Figure : Détails inventaires

On peut aussi accéder aux détails de la mission en cliquant sur la date correspondant du calendrier.

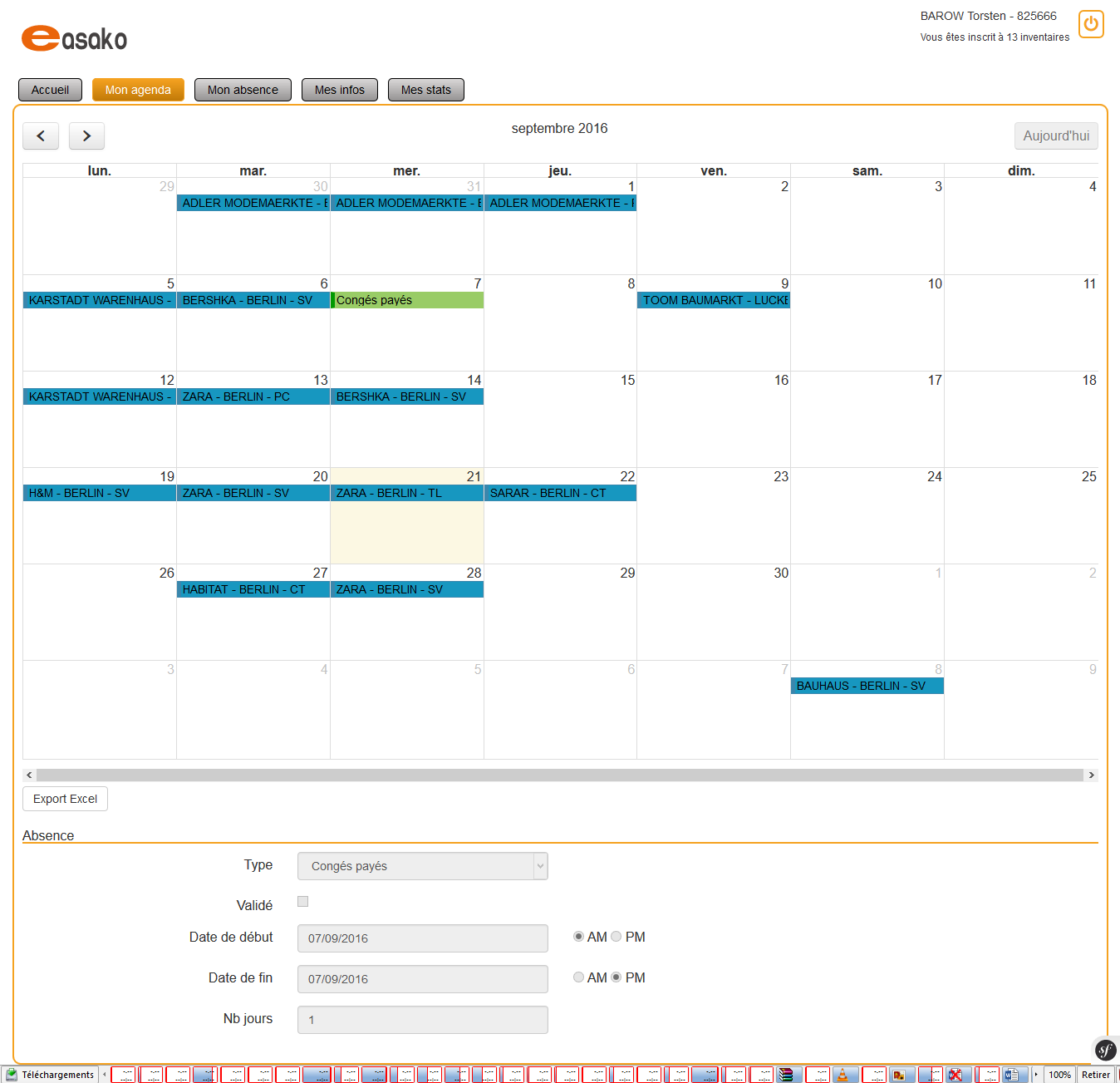


Figure : Détails absence

En cliquant sur absence, on accède aux détails d’absence. Sur cette page, l’employé ne peut que lire uniquement les informations affichées.

## Page Absence

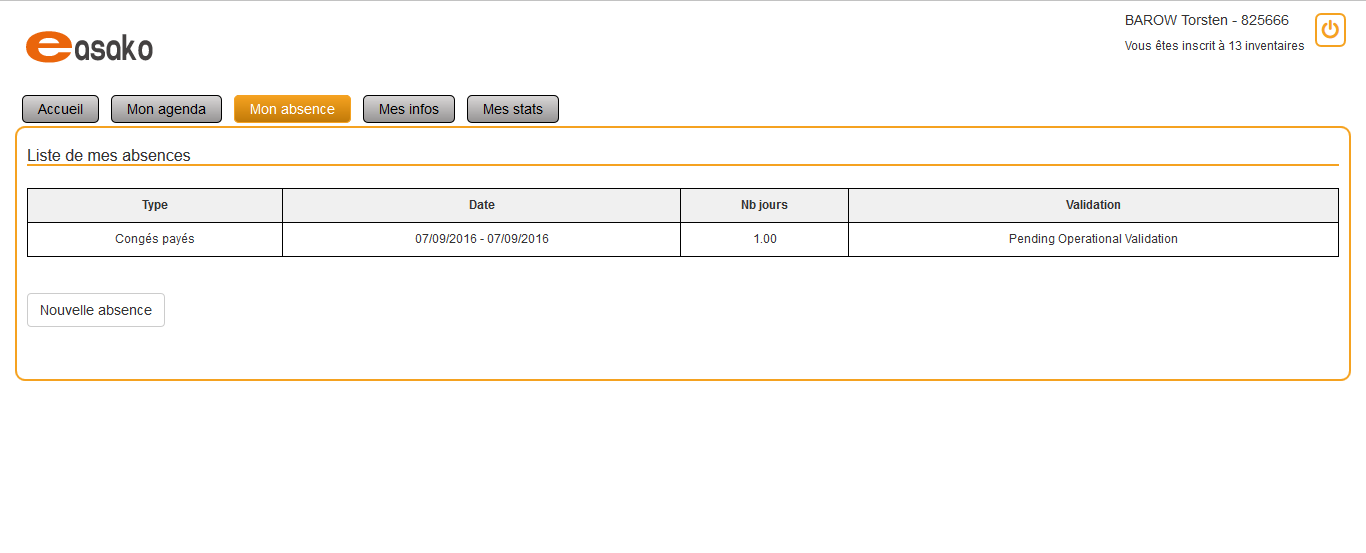


Figure : Page Absence

Sur cette page, le collaborateur peut consulter la liste et historique de ses absences et il peut aussi éditer une nouvelle.

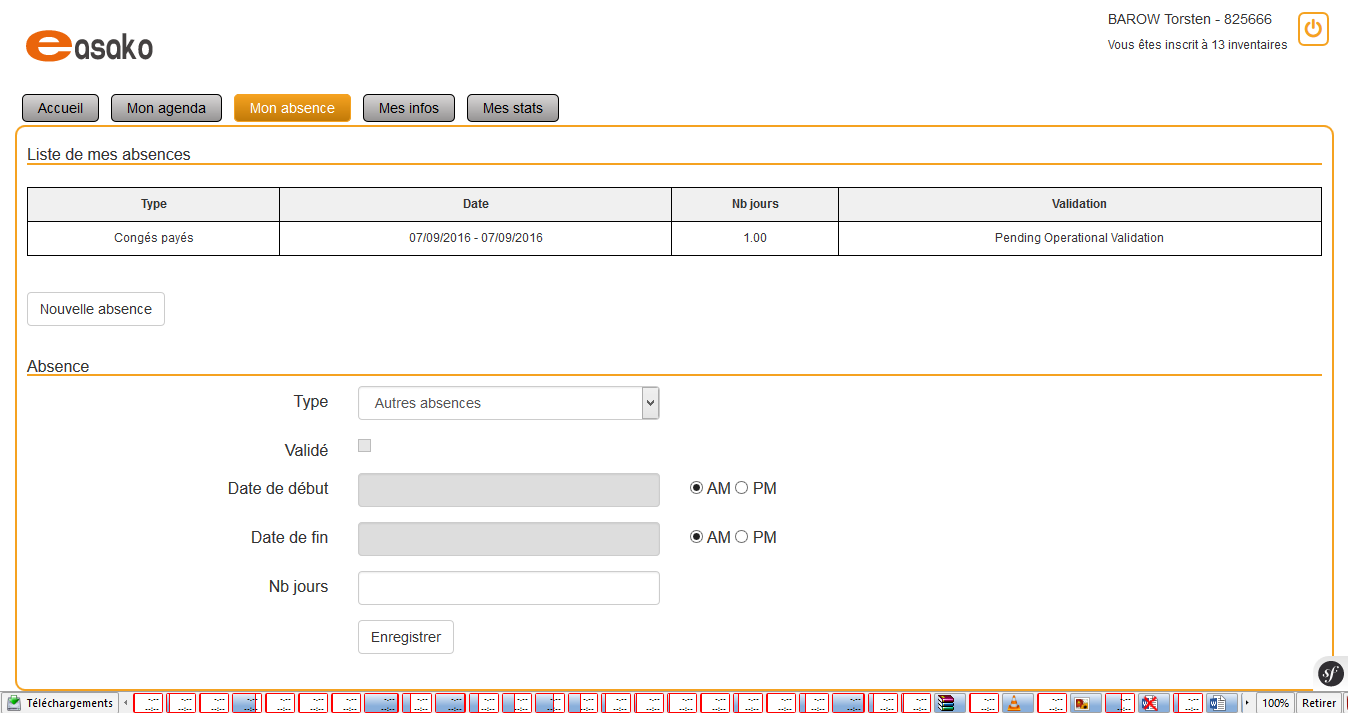


Figure : Insérer une nouvelle absence

En cliquant sur le bouton Nouvelle absence, le collaborateur peut créer une nouvelle absence en définissant le type de son absence (congé, maladie...) Il définira aussi dont la date de début et la date de fin de son absence. Le nombre de jours est calculé automatiquement en tenant compte des week-ends aussi.

## Page Mes infos ou profil

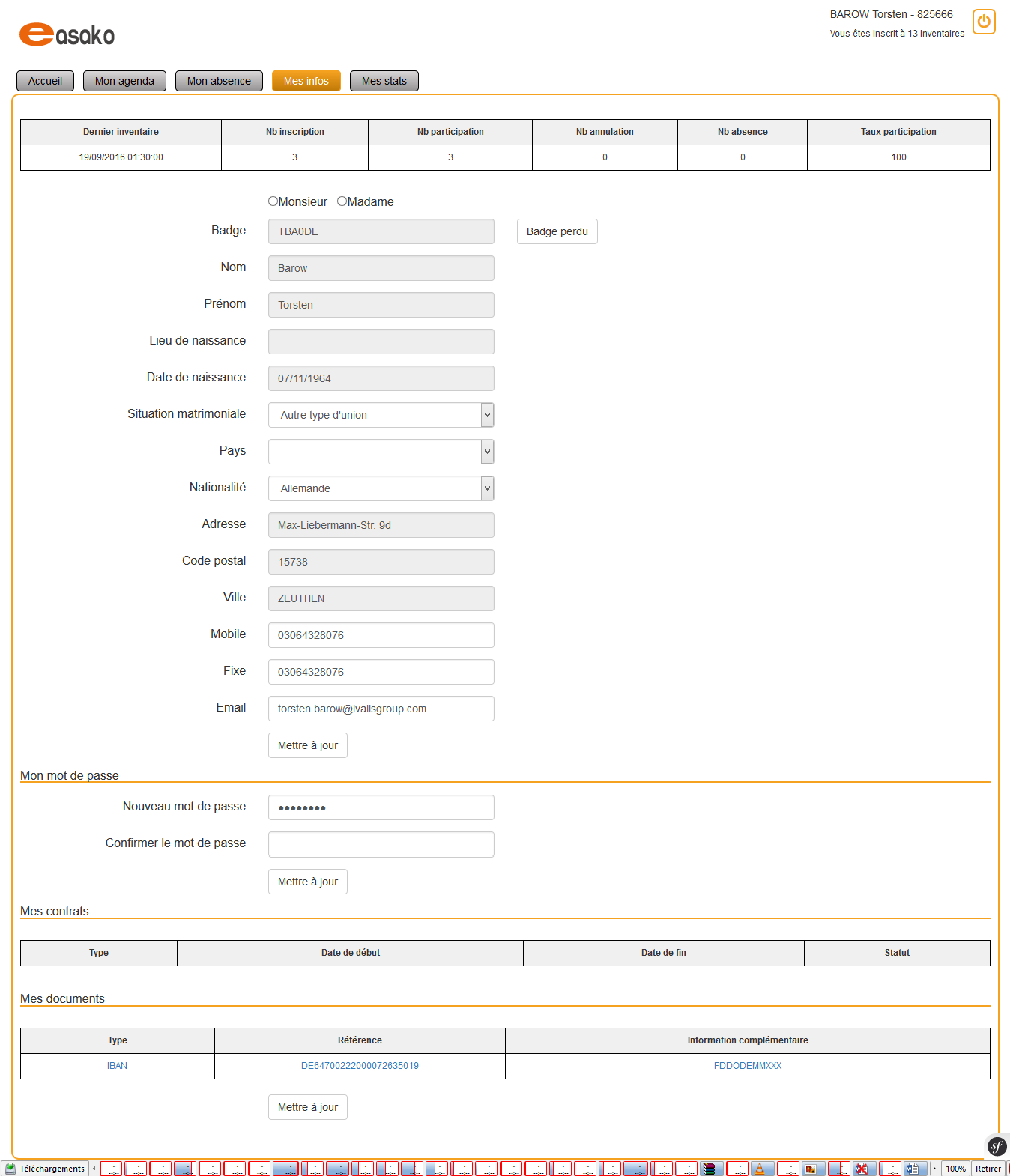


Figure : Page Mes infos

Cette page concerne sur tous les informations du collaborateur. Il peut modifier ses informations personnelles (Nom, prénom, date de naissance, email, pays d’origine…).

## Page Mes stats ou statistiques

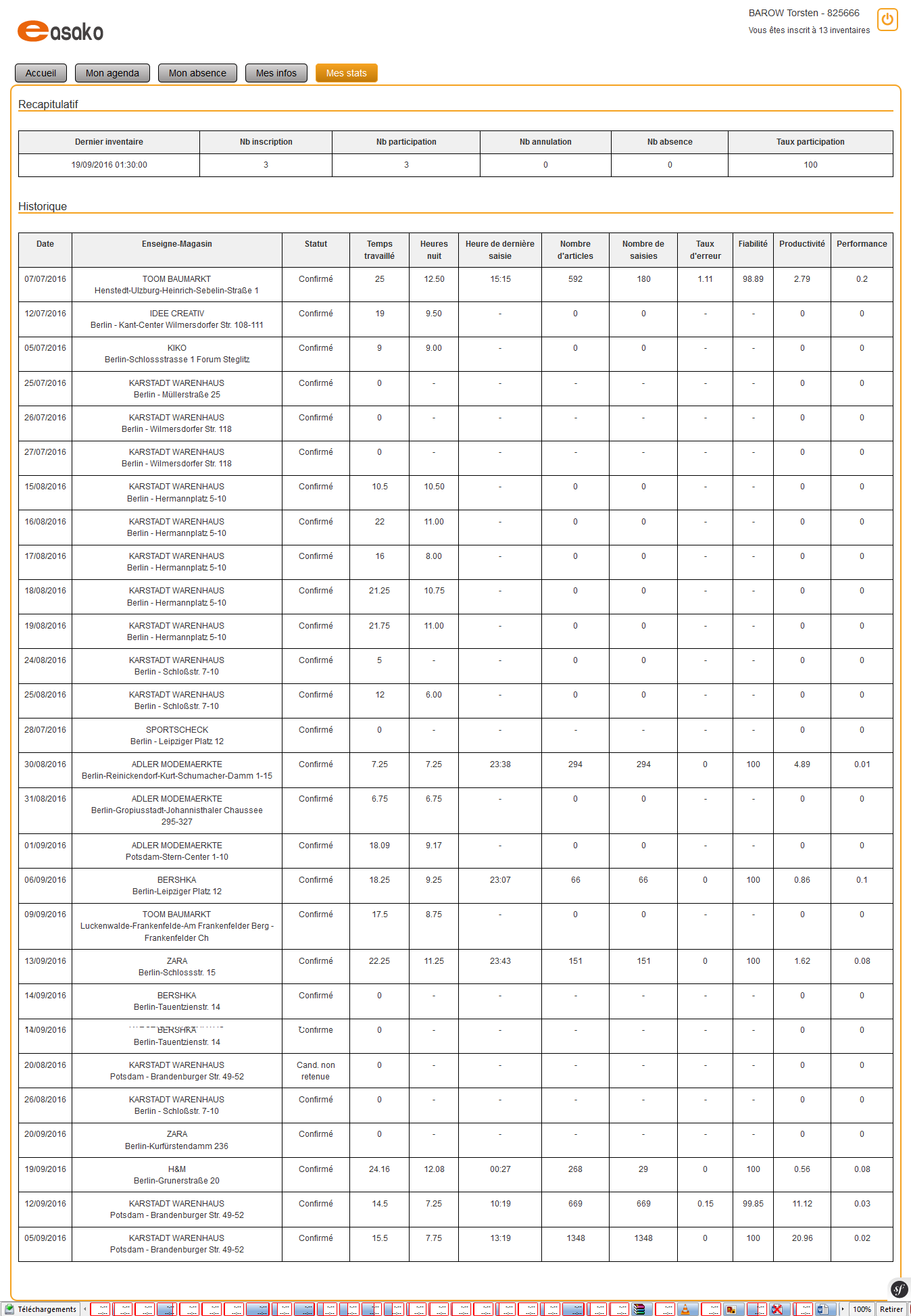


Figure : Page mes stats

Cette page renferme donc l’historique de missions effectuées par le collaborateur, un récapitulatif des dernières missions.

L’historique comprend donc le total des heures travaillé sur une mission, la fiabilité et performance du collaborateur sur une mission donnée…

# CHAPITRE 6 : EXTRAIT DE CODE

## SoapUtils.php

class SoapUtils

{

var $client = null;

var $wsdl\_sf;

var $wsdl;

var $USERNAME;

var $PASSWORD;

var $SECURITY\_TOKEN;

var $WS\_NAMESPACE;

var $user;

var $sfc = null;

var $host;

private static $sessionId = null;

protected $request;

private static $session\_file = "sessionfile.txt";

private $sessionId\_key = "sessionId";

/\*\*

\* Definir les parametres dans Bundle/Ressources/Config/Service.yml

\* @param Kernel $kernel

\*/

function \_\_construct(Kernel $kernel){

$this->wsdl\_sf = $kernel->getContainer()->getParameter('soap.wsdl\_sf');

$this->wsdl = $kernel->getContainer()->getParameter('soap.wsdl');

$this->USERNAME = $kernel->getContainer()->getParameter('soap.user');

$this->PASSWORD = $kernel->getContainer()->getParameter('soap.pwd');;

$this->SECURITY\_TOKEN = $kernel->getContainer()->getParameter('soap.token');

$this->WS\_NAMESPACE = $kernel->getContainer()->getParameter('soap.name\_space');

$this->host = $kernel->getContainer()->getParameter('soap.host');

}

function getSfc(){

if($this->sfc==null){

$this->sfc = new SforcePartnerClient();

}

return $this->sfc;

}

function getSessionId(){

SoapUtils::$sessionId = $this->isSessionExpired();

if(SoapUtils::$sessionId==null){

SoapUtils::$sessionId = $this->login();

}

return SoapUtils::$sessionId;

}

private function createHeader(){

return $sforce\_header = new SoapHeader($this->WS\_NAMESPACE, "SessionHeader",

array("sessionId" => $this->getSessionId()));

}

private function login(){

$mySforceConnection = $this->getSfc();

$mySoapClient = $mySforceConnection->createConnection($this->host.$this->wsdl\_sf);

$this->user = $mySforceConnection->login($this->USERNAME, $this->PASSWORD.$this->SECURITY\_TOKEN);

$this->createSessionServer($this->sessionId\_key, $this->user->sessionId);

return $this->user->sessionId;

}

public function createSessionServer($name,$value){

$data = array();

$data[$name]= $value;

$data['expTime'] = (time()+3600\*8); //8h

$sessionfile = fopen(self::$session\_file, "w");

fputs($sessionfile, json\_encode($data));

fclose($sessionfile);

return true;

}

public function isSessionExpired(){

if(is\_file(self::$session\_file)){

$sessionfile = fopen(self::$session\_file, "r");

$data = json\_decode(fgets($sessionfile, 4096) );

$expiredTime = $data->expTime;

fclose($sessionfile);

if($expiredTime<time()){

return $this->login();

}

else{

return $data->sessionId;

}

}

return null;

}

public function isTimeExpired(){

if(is\_file(self::$session\_file)){

$sessionfile = fopen(self::$session\_file, "r");

$data = json\_decode(fgets($sessionfile, 4096) );

$expiredTime = $data->expTime;

fclose($sessionfile);

if($expiredTime<time()){

return true;

}

else{

return false;

}

}

return false;

}

public function serializeData($file,$value){

$fichier = fopen($file, "w");

fputs($fichier, serialize($value));

fclose($fichier);

return true;

}

public function deserializeData($file){

if (file\_exists($file)){

$fichier = fopen($file,'r');

$data = unserialize(fgets($fichier));

return $data;

}

return null;

}

function getSoapClient()

{

if($this->client==null){

$mySforceConnection = $this->getSfc();

$client = $mySforceConnection->createConnection($this->host.$this->wsdl);

$client->\_\_setSoapHeaders(array($this->createHeader()));

$this->client = $client;

}

return $this->client;

}

function call($name,$params)

{

$client = $this->getSoapClient();

return $client->\_\_call($name,$params);

}

function printObject($obj)

{

ob\_start();

print\_r($obj);

$res = ob\_end\_clean();

return $res;

Cette classe permet de faire connecter l’application avec le WS de Salesforce. Tous les appels web service doivent donc passer par cette classe.

## CollaborateurRepository.php

<?php

namespace Ivalis\CollaborateurBundle\Entity;

use Ivalis\CollaborateurBundle\Manager\Repository;

use Ivalis\CollaborateurBundle\Manager\CollaborateurManager;

use Ivalis\CollaborateurBundle\Entity\Reflection;

use Assetic\Exception\Exception;

/\*\*

\* CollaborateurRepository

\*

\*/

class CollaborateurRepository extends Repository

{

const NEW\_PASSWORD\_SENT = "New password sent";

const MAIL\_INEXISTAN = "Email address does not exist";

/\*\*

\* Gestion de la connexion / Connexion

\*

\* param String login

\* param String password

\* return collaborateur

\*/

public function authenticateUser($login, $password){

$client = $this->soap->getSoapClient();

$collaborateur = $client->AuthenticateUser(array("Login"=>$login,"Password"=>$password));

if(is\_object($collaborateur)){

$collaborateur = $collaborateur->result;

$reflection = new Reflection();

if($collaborateur instanceof \stdClass)

return $reflection->convertStdObject($collaborateur,new Collaborateur());

}

return null;

}

/\*\*

\* Gestion de la connexion / Connexion

\*

\* param String login

\* param String password

\* param String from

\* return collaborateur

\*/

public function authenticateUserFrom($login, $password, $from){

$client = $this->soap->getSoapClient();

$request = $this->context->get("request");

$language = $request->getLocale();

if(strtoupper($language)=='BEEN'){

$language = "en";

}

if(strtoupper($language)=='BEEN'){

$language = "fr";

}

//var\_dump($client);die();

$collaborateur = $client->AuthenticateUserFrom(array("Login"=>$login,"Password"=>$password, "QueryFrom" => $from,"Language"=>$language));

/\*$res = $collaborateur->result;

var\_dump($res);

die("ok");\*/

if(is\_object($collaborateur)){

$collaborateur = $collaborateur->result;

$reflection = new Reflection();

if($collaborateur instanceof \stdClass)

return $reflection->convertStdObject($collaborateur,new Collaborateur());

}

return null;

}

/\*\*

Mot de passe oublié

\* webservice

\* param object Inventorist

\* return

\*/

public function generateUserPassword($email){

$client = $this->soap->getSoapClient();

$collaborateur = $client->GenerateUserPassword(

array("Person"=>array("email"=>$email,"LastName"=>$email))

);

$collaborateur = $collaborateur->result;

$session = $this->context->get('session');

if($collaborateur instanceof \stdClass){

if(isset($collaborateur->PasswordGenerationStatus) && $collaborateur->PasswordGenerationStatus==self::NEW\_PASSWORD\_SENT){

$session->set("msg",$collaborateur->PasswordGenerationStatus);

return true;

}else{

$session->set("msg",$collaborateur->PasswordGenerationStatus);

return false;

}

}

$session->set("msg",null);

return false;

}

/\*\*

Mis à jour des infos du collaborateur

\*

\* param String login

\* param String password

\* return collaborateur

\*/

public function updateCollaborateur($pCollaborateur){

$documentsSave = $pCollaborateur->getDocuments("2");

$client = $this->soap->getSoapClient();

$reflection = new Reflection();

$request = $this->context->get("request");

$language = $request->getLocale();

try{

$collaborateur = $client->UpdateUser(array("Person"=>$reflection->convertObjectToArra($pCollaborateur),"Language"=>$language));

}catch(\Exception $e){

$error = $this->getError($e);

$session = $this->context->get('session');

$session->set("collaborateurTmp",serialize($pCollaborateur));

$session->set("error",$error["code"]);

$session->set("errorMsg",$error["message"]);

return null;

}

$collaborateur = $collaborateur->result;

//var\_dump($collaborateur);die;

if($collaborateur instanceof \stdClass){

//remetre les doc sauvegarder a supprmer quand le ws sur la mise a jour des doc est ok

$collaborateur->Documents = $documentsSave;

return $reflection->convertStdObject($collaborateur,new Collaborateur());

}

else

return null;

}

/\*\*

\* Creation d'un compte

webservice

param object Inventorist

return object EJ\_Person

\* @param $inventorist

\* @return NULL|\stdClass

\*/

private function getError($error){

try{

$matches = array();

preg\_match('#Exception(\((.\*?)\))#', $error->getMessage(), $matches);

$w = $matches[count($matches)-1];

$w = explode(",",$w);

return array("code"=>$w[0],"message"=>str\_replace("\"", "",$w[1]));

}catch(\Exception $e){

die($e->getMessage());

return $error;

}

}

public function inventoryList($inventoristId,$potentialMission,$history,$current,$status = "")

{

$client = $this->soap->getSoapClient();

$reflection = new Reflection();

$inventory = $client->InventoryList(

array(

"PersonId"=>$inventoristId,

"ShowPotentialMissions"=>$potentialMission,

"ShowHistory"=>$history,

"ShowCurrent"=>$current,

"Status"=>$status

)

);

$res = array();

$reflection = new Reflection();

if(isset($inventory->result) && is\_array($inventory->result)){

foreach($inventory->result as $item){

$res[] = $reflection->convertStdObject($item,new Inventaire());

}

}

return $res;

}

/\*\*

\* WS BadgeRequest

\* @param string nbadge => $id de l'utilisateur

\* @return EJ\_Person

\*/

public function BadgeRequest($idPerson){

// connexion WS

$request = $this->context->get("request");

$language = $request->getLocale();

$client = $this->soap->getSoapClient();

$collaborateur = $client->BadgeRequest(array("PersonId"=>$idPerson,'Language'=>$language));

$collaborateur = $collaborateur->result;

if($collaborateur instanceof \stdClass)

return true;

else

return false;

}

/\*\*

\* Valeur possible pour listName

\* $NATIONALITY = "Nationality";

\* $SITUATION = "Situation";

\* $DIPLOME = "Diploma";

\* $ORIGIN = "Origin";

\* @param String $listName

\*/

public function GetSelectListLocale($listName){

$request = $this->context->get("request");

$language = $request->getLocale();

/\*if($listName == 'Country' || $listName == 'Nationality'){

$language = 'en';//$request->getLocale();

}else{

$language = $request->getLocale();

}\*/

//var\_dump($language);

switch(strtoupper($language)){

case "BENL":

case "BEEN":

$language = "en";

break;

case "BEFR":

$language = "fr";

break;

//default: $language = $laguage; break;

}

$client = $this->soap->getSoapClient();

$country = CollaborateurManager::getCountry($language);

//$select = $client->GetSelectListLocale(array("listName"=>$listName,"Language"=>$language));

$select = null;

$filename = $listName.$language.'.txt';

if ($this->soap->isTimeExpired()|| !file\_exists($filename)){

$params = array("listName"=>$listName,"Language"=>$language);

$select = $client->GetSelectList($params);

$this->soap->serializeData($filename,$select);

}

else {

$select = $this->soap->deserializeData($filename);

}

/\*$params = array("listName"=>$listName,"Language"=>$language);

$select = $client->GetSelectList($params);\*/

$res = array();

if(isset($select->result) && is\_array($select->result)){

foreach($select->result as $item){

$tmp = new SelectItem();

$tmp->setId($item);

$tmp->setValue($item);

$res[] = $tmp;

}

}

return $res;

}

/\*\*

\* function to save in WSDL

\* new leaves

\*

\* @param type $\_idPerson

\* @param type $\_type

\* @param type $\_fromDate

\* @param type $\_toDate

\* @param type $\_fromAmPm

\* @param type $\_toAmPm

\* @return boolean

\*/

public function saveNewLeaves($\_idPerson, $\_type = "", $\_fromDate = null, $\_toDate = null, $\_fromAmPm = '', $\_toAmPm = '') {

$request = $this->context->get("request");

$language = $request->getLocale();

$client = $this->soap->getSoapClient();

$params = array(

"PersonId" => $\_idPerson,

"Type" => $\_type,

"FromDate" => $\_fromDate,

"ToDate" => $\_toDate,

"FromAMPM" => $\_fromAmPm,

"ToAMPM" => $\_toAmPm,

"Language" => $language

);

$leaveRequestResponse = $client->LeaveRequest($params);

$collaborateur = $leaveRequestResponse->result;

$reflection = new Reflection();

if($collaborateur instanceof \stdClass)

return $reflection->convertStdObject($collaborateur,new Collaborateur());

else

return null;

}

}

Tous les appels des méthodes WS se font donc au niveau des classes Repository. Chaque appel correspond donc à une méthode WS.

## CollaborateurManager.php

class CollaborateurManager

{

protected $container;

protected $collaborateurRepository;

protected $collaborateur;

const ERROR = "error";

const ERROR\_WS = "error\_ws";

const OK = "ok";

const FROM = "ic";

const ERROR\_DEPARTEMENT\_MODIFICATION = "0003";

public function \_\_construct($collaborateur,$container){

$this->container = $container;

$this->collaborateurRepository = new CollaborateurRepository($container);

$this->collaborateur = $collaborateur;

}

/\*

\* Gestion de la connexion / Connexion

\*/

public function connexion($login, $password){

$collaborateurRepository = $this->collaborateurRepository;

$retConnexionWs = $collaborateurRepository->authenticateUserFrom($login, $password, self::FROM);

if($retConnexionWs instanceof Collaborateur && is\_array($retConnexionWs->getContracts())){

/\*print\_r($retConnexionWs->getContracts());

echo "<br/>----------<br/>";

print\_r($this->sortContract($retConnexionWs->getContracts()));

die;\*/

$retConnexionWs->setContracts($this->sortContract($retConnexionWs->getContracts()));

}

return $retConnexionWs;

}

public function generateUserPassword($email){

return $this->collaborateurRepository->generateUserPassword($email);

}

/\*

\* Mise à jour des infos collaborateur

\*/

public function updateCollaborateur(\Ivalis\CollaborateurBundle\Entity\Collaborateur $collaborateur, Request $request){

// $collaborateur->setFirstName($request->request->get('c\_nom'));

$collaborateurRepository = $this->collaborateurRepository;

$retConnexionWs = $collaborateurRepository->updateCollaborateur($collaborateur);

return $retConnexionWs;

}

public function getInventories(){

if($this->collaborateur!=null){

return $this->collaborateur->getMissions(true);

}

}

public function getAllCallendarCell($month = "",$isExcel=false){

$inv = $this->getInventories();

$res = array();

$invent = array();

if(is\_array($inv)){

$it = 0;

foreach($inv as $item){

//var\_dump($item->inventoryDate);die;

//iic-85 cas de test

// $date1 = new \DateTime(str\_replace('/', '-', $item->inventoryDate));

// $date2 = new \DateTime("now");

if($item->getStatus()==Inventaire::$CONFIRME){ //&& ($date1 > $date2)){

$res[] = $item->getCallendarCell($isExcel);

}

$it++;

}

}

$leaves = $this->collaborateur->getLeaves("2");

if(is\_array($leaves)){

foreach($leaves as $item){

$res[] = $item->getCallendarCell($isExcel,"Y-m-d");

}

}

return $res;

}

public function getInventory($idInventory){

$inventories = $this->getInventories();

if(is\_array($inventories)){

foreach($inventories as $item){

if($item->getIdSF() == $idInventory){

return $item;

}

}

}

return null;

}

public function getLeave($idLeave){

$leaves = $this->collaborateur->getLeaves("2");

if(is\_array($leaves)){

foreach($leaves as $item){

if($item->getIdSF() == $idLeave)

return $item;

}

}

return null;

}

public function requestBadge(){

return $this->collaborateurRepository->BadgeRequest($this->collaborateur->getIdSF());

}

/\*\*

\* function to get next inventory

\*

\* @return type

\*/

public function getNextInventory($status = "") {

$aInventories = $this->getCurrents($status);

$oDateNow = new \DateTime('now');

$oNextInvetory = null;

$aInventories = $this->sort($aInventories);

if(!empty($aInventories)){

foreach($aInventories as $oInventories){

$inventoryDate = str\_replace('/', '-', $oInventories->inventoryDate) ;

$oDateInventory = new \DateTime($inventoryDate);

$diffTimestamp = $oDateInventory->getTimestamp() - $oDateNow->getTimestamp() ;

if($diffTimestamp>=0){

$oNextInvetory = $oInventories;

break;

}

}

}

return $oNextInvetory;

}

public function getSelect($name){

return $this->collaborateurRepository->GetSelectListLocale($name);

}

public static function testMail($email){

$regexp = "/^[A-Za-z\\'0-9-\_!#$%&\*+\\/=?^`{|}~]+([.][A-Za-z\\'0-9-\_!#$%&\*+\\/=?^`{|}~]+)\*[@]([A-Za-z0-9-]+([.][A-Za-z0-9-]+)+){1}$/";

$mailIsValid = preg\_match($regexp, trim($email));

return $mailIsValid;

}

public static function isPhone($phone){

$regex = "/^(\d[\s-]?)?[\(\[\s-]{0,2}?\d{3}[\)\]\s-]{0,2}?\d{3}[\s-]?\d{4}$/i";

return preg\_match( $regex, $phone );

}

/\*\*

\* function to save new leaves

\*

\* @param type $request

\* @return type

\*/

public function saveNewLeaves($request) {

$idPerson = $this->collaborateur->getIdSf();

$absence\_type = $request->get('absence\_type');

$absence\_dateDebut = $request->get('absence\_dateDebut');

$absence\_dateDebutTemps = $request->get('absence\_dateDebutTemps');

$absence\_dateFin = $request->get('absence\_dateFin');

$absence\_dateFinTemps = $request->get('absence\_dateFinTemps');

$absence\_validate = $request->get('absence\_validate');

$absence\_nbrJours = $request->get('absence\_nbrJours');

$oFromDate = new \DateTime($absence\_dateDebut);

$oToDate = new \DateTime($absence\_dateFin);

$xsdFromDate = $oFromDate->format('Y-m-d');

$xsdToDate = $oToDate->format('Y-m-d');

$collaboratteurRepo = $this->collaborateurRepository;

$responseLeaveRequest = $collaboratteurRepo->saveNewLeaves($idPerson, $absence\_type, $xsdFromDate, $xsdToDate, $absence\_dateDebutTemps, $absence\_dateFinTemps);

return $responseLeaveRequest;

}

/\*\*

\*

\* @return array

\*/

public function getLeaves(){

$aNewLeaves = array();

if($this->collaborateur!=null){

$aLeaves = $this->collaborateur->getLeaves();

if(!empty($aLeaves)){

foreach ($aLeaves as $oLeave) {

$oNewLeave = $this->calculateNbJours($oLeave);

array\_push($aNewLeaves, $oNewLeave);

}

}

}else{

return null;

}

return $aNewLeaves;

}

# PARTIE IV : DISCUSSION

# CHAPITRE 7 : ANALYSE DU SYSTEME D’INFORMATION

## Etude générale du système d’information

### Diagramme de cas d’utilisation

Les rôles des diagrammes de cas d’utilisation sont de recueillir, d’analyser et d’organiser les besoins, ainsi que de recenser les grandes fonctionnalités d’un système. Il s’agit donc de la première étape UML pour la conception d’un système.

Il ne faut jamais négliger cette première étape pour produire un site web conforme aux attentes des utilisateurs ciblés. Pour élaborer les cas d’utilisation, il faut se fonder sur des entretiens avec les utilisateurs.

#### Les acteurs

En analysant les besoins exprimés dans le cahier des charges, nous avons déduit les acteurs ainsi que les cas d'utilisations tels que résumés dans le tableau suivant :

Tableau : Acteur et ses cas d'utilisation

|  |  |
| --- | --- |
| **Acteur** | **Cas d'utilisation** |
| **Employé** | Voir statistique et performance |
| Modifier profil |
|  | Voir calendrier de mission |
|  | Voir calendrier d'absence |
|  | Voir sa prochaine mission |
|  | Exporter fichier xls du calendrier |
|  | Editer absence |

#### Diagramme de cas d’utilisation



Figure : Diagramme de cas d'utilisation de l'application

### Diagramme de séquence

#### Diagramme de séquence « Authenfication »

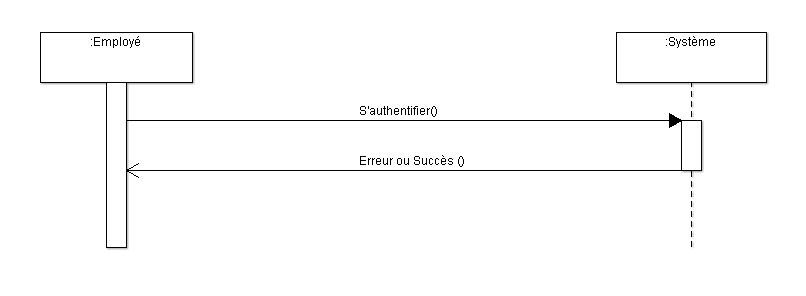


Figure : Diagramme de séquence "authentification"

L’employé fait entrer son mot de passe et son identifiant pour s’authentifier. L’application retourne une erreur quand le mot de passe ou l’identifiant n’est pas reconnu.

#### Diagramme de séquence « Voir statistique et performance »

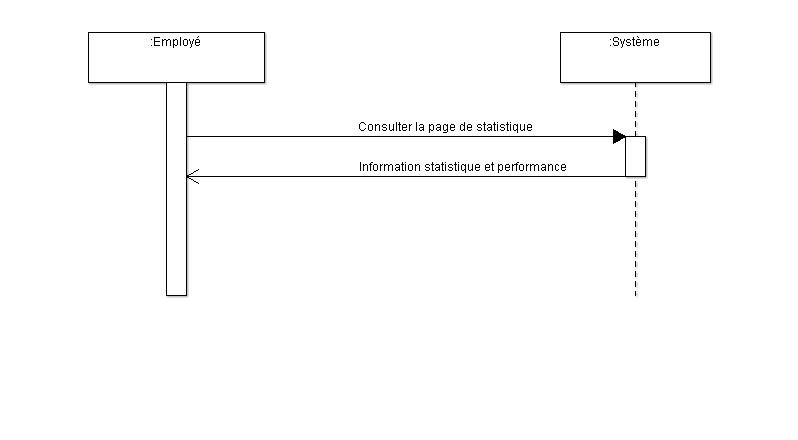


Figure : Diagramme de séquence "Voir statistique et performance"

L’employé rentre dans la page statistique et peut donc consulter les statistiques de ses missions (ses heures de travails,…).

#### Diagramme de séquence « Modifier profil »

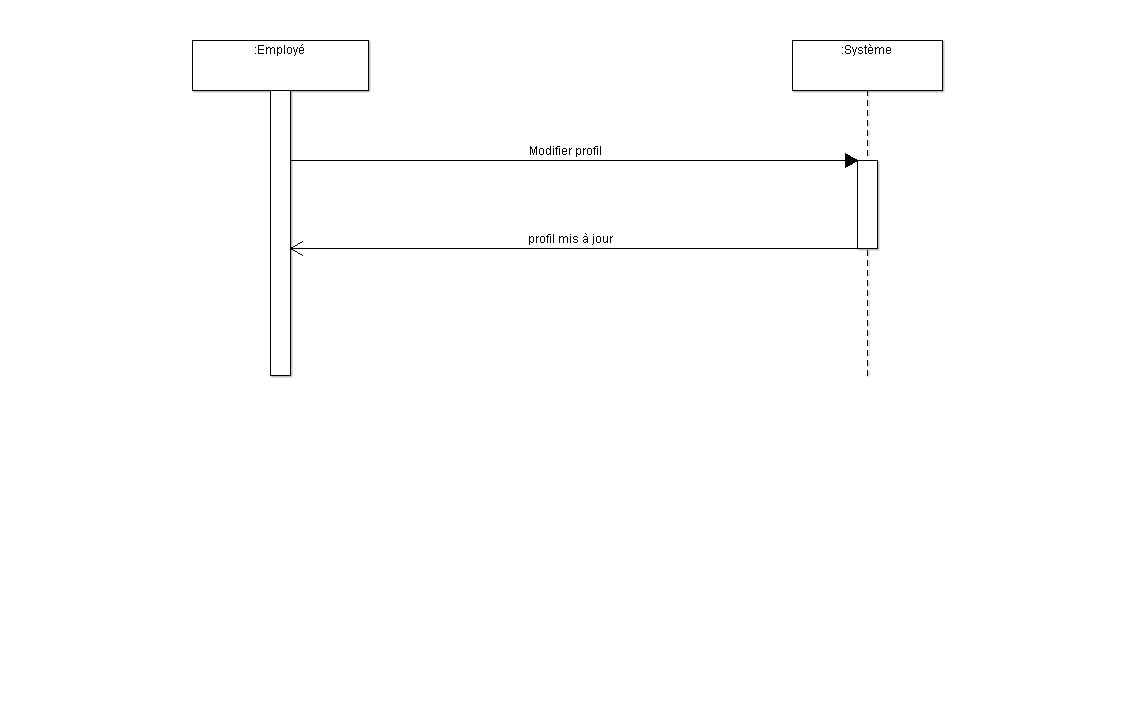


Figure :Diagramme de séquence "Modifier profil"

#### Diagramme de séquence « Voir calendrier de mission »

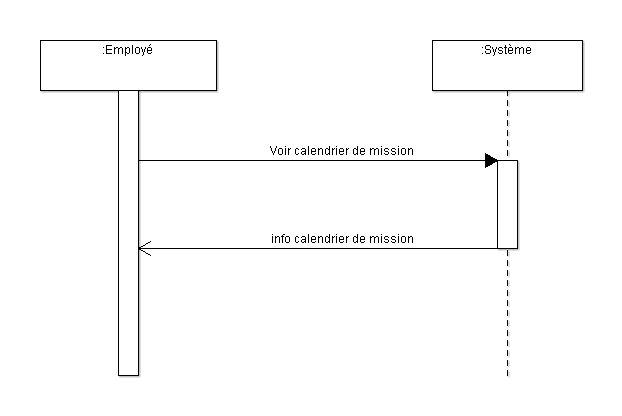


Figure : Diagramme de séquence "Voir calendrier de mission"

Voici donc le diagramme de séquence d’un cas d’utilisation : Voir calendrier de mission

#### Diagramme de séquence « Voir calendrier d’absence »

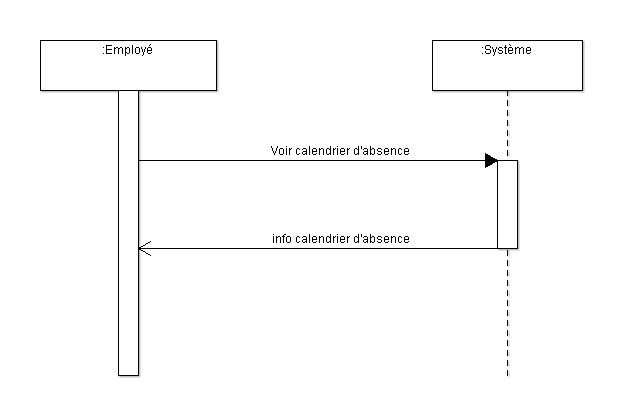


Figure : Diagramme de séquence "Voir calendrier d'absence"

### Diagramme d’activité

C’est un Diagramme associé à un objet particulier ou à un ensemble d'objets, qui illustre les flux entre les activités et les actions. Il permet de représenter graphiquement le déroulement d'un cas d'utilisation.

Le diagramme d’activité se compose des éléments suivants :

* **Une activité** représente une exécution d'un mécanisme, autrement dit, un déroulement d'étapes séquentielles ;
* **Une transition** qui représente Le passage d'une activité vers une autre ;
* **Les gardes** qui représentent la condition de passage d’une activité à une autre dans les transitions conditionnelles ils sont symbolisés par des losanges ;
* **Les barres de synchronisation** sont des barres représentées par une ligne épaisse, le rôle cette barre est de synchroniser le départ de plusieurs transitions qui arrivent de déférentes activités, aboutissant toutes à une activité commune.

#### Diagramme d’activité « Employé »

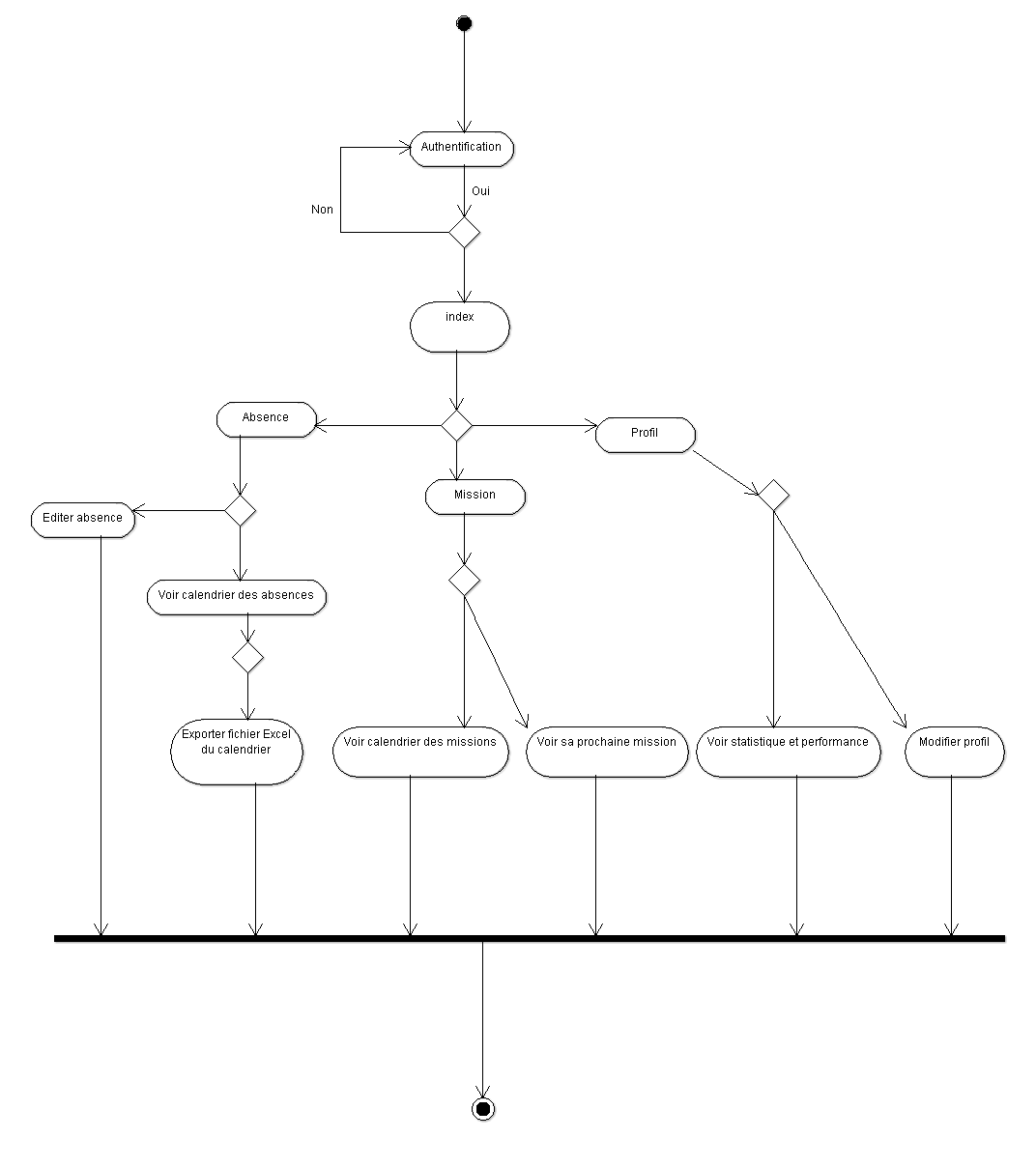


Figure : Diagramme d'activité

Durant tout le processus de création du logiciel, l’analyse constitue la partie la plus difficile et la plus importante. Ce chapitre constitue donc une des parties les plus importantes de ce mémoire. Il décrit toutes les fonctionnalités de l’application et son utilité.

# CONCLUSION

Pour terminer les cinq années d’étude à l’Institut Supérieur Polytechnique de Madagascar, ce mémoire représente donc le fruit de la recherche de ces années d’étude. Il contribue d’ailleurs aussi pour la présentation en vue de l’obtention du diplôme de master II en informatique.

Il traite sur un projet qui réalisé en Symfony2. Techniquement, notre projet se base aussi les web services car les données ne sont pas stockées localement mais cotés Salesforce.

Le projet e-Asako est surtout destiné pour des travailleurs à mi-temps, car les missions peuvent être à n’importe quelle heure. Pendant les périodes de vacance, la plateforme est d’une très grande utilité pour les étudiants, et lycéens. Elle propose donc des travails de vacance.

Nous gardons de ce stage un excellent souvenir, et qui constitue une expérience professionnelle valorisante et encourageante pour nos avenirs.

Nous pensons que cette expérience dans chez eTech nous a offert une bonne préparation à nos insertions professionnelles, car elle fut pour nous une expérience enrichissante et complète qui conforte nos désirs d’exercer nos futurs métiers dans le domaine de développement en informatique.

Ensuite, on peut conclure que les études durant ses cinq dernières années durant lesquelles nous avons étudié au sein de l’Institut Supérieur Polytechnique de Madagascar nous ont permis de s’enrichir nos savoirs et connaissances.

Enfin, nous tenons à exprimer nos satisfactions d’avoir travaillé dans un environnement agréable.

Que la collaboration entre l’ISPM et l’entreprise eTech se développe encore et toujours.