



# DOSSIER-PROJET

*Nom de naissance* ▶ LEFRANCOIS

*Nom d'usage* ▶ LEFRANCOIS

*Prénom* ▶ Jean

*Adresse* ▶ 36 rue du Docteur Choquart  
51230 PLEURS

## Titre professionnel visé

Développeur-se logiciel - Niveau III

# Sommaire

---

(à adapter selon le projet)

<b>Remerciements</b>	<b>3</b>
<b>Résumé du projet en anglais</b>	<b>4</b>
<b>Liste des compétences du référentiel</b>	<b>5</b>
<b>Cahier des charges ou expressions des besoins de l'application à développer</b>	<b>6</b>
a)Gestion des utilisateurs	6
b)Gestion du temps	6
c)Gestion des mails et de l'historique	6
d)Base documentaire	6
<b>Spécifications fonctionnelles</b>	<b>7</b>
<b>Spécifications techniques</b>	<b>8</b>
<b>Réalisations</b>	<b>9</b>
<b>Conclusions</b>	<b>10</b>

## REMERCIEMENTS

---

En tout premier lieu, je tiens à remercier mon entreprise La Poste, particulièrement Charles Moureau qui est à l'origine de cette formation. Je remercie aussi l'Espace Mobilité Groupe, notamment le personnel chargé d'organiser et gérer cette formation.

Un remerciement spécial à Mr Laurent Manhes, directeur de la P.D.C. de Provins, qui mis les moyens nécessaires pour que je fasse cette formation.

Je veux aussi remercier Simplon, notamment Laetitia Amourous, Elise Fraisse, Emmanuel Le Pevedic et Nicolas Hovart, qui ont fait que cette formation se déroule le mieux possible et qu'elle puisse aboutir.

Je remercie les autres formateurs qui sont intervenus dans la formation, ainsi qu'Erwann Kezzar pour son soutien et ses conseils tout au long du processus.

Merci à mes collègues Simploniens pour leur sympathie, leur gentillesse et la qualité des échanges qui nous ont permis de progresser.

Un dernier remerciement à Mokhtar Khider et Fred Couriol avec qui j'ai eu plaisir à travailler sur ce projet.

## RÉSUMÉ DU PROJET EN ANGLAIS

For the project, we chose to make an application which manages the absences of the company Simplon.

This application allows an employee to make a request of vacation. An e-mail is sent to his(her) supervisor. The latter confirms or refuses the request. Whatever is the answer, the employee receives an e-mail with justification in case of refusal. If the request is accepted, an e-mail is sent in the service of the human resources, which verifies the validity of the vacation according to the law. If the Rh service refuses the vacation, an e-mail is sent to the employee and to his supervisor with the motive for the refusal.

The supervisor can also seize an absence, if the absence is unexpected or if the employee forgot to make his request of vacation.

The employee also has the possibility of seeing his requests, as well as his periods of absences thanks to a calendar.

The immediate superior has access in calendar of his(her) agents as well as to the calendar of his(her) team to manage better this one.

The service of the human resources has access to the calendar of all the teams as well as to the calendar of the company.

An administrator system is responsible for the management of the database, the creation of a new incomer, a new administrator or a team, by updating or by deleting data.

Pour le projet, nous avons choisi de faire une application qui gère les absences de la société Simplon.

Cette demande permet à un employé de faire une demande de congé. Un e-mail est envoyé à son superviseur. Ce dernier valide ou refuse la demande. Quelle que soit la réponse, l'employé reçoit un courrier électronique avec justification en cas de refus. Si la demande est acceptée, un courrier électronique est envoyé au service des ressources humaines, qui vérifie la validité du congé conformément à la loi. Si le service Rh refuse le congé, un e-mail est envoyé à l'employé et à son superviseur avec le motif du refus.

Le superviseur peut également saisir une absence, si l'absence est inopinée ou si l'employé a oublié de faire sa demande de congé.

L'employé a également la possibilité de voir ses demandes, ainsi que ses périodes d'absences grâce à un calendrier.


Le supérieur hiérarchique a accès au calendrier de ses agents ainsi qu'au calendrier de son équipe afin de mieux gérer celui-ci.

Le service des ressources humaines a accès au calendrier de toutes les équipes ainsi qu'au calendrier de l'entreprise.

Un administrateur système est responsable de la gestion de la base de données, de la création d'un nouvel entrant, d'un nouveau gestionnaire ou d'une équipe, en mettant à jour ou en supprimant des données.


## LISTE DES COMPÉTENCES DU RÉFÉRENTIEL

 *Maquetter une application :*

 *Concevoir une base de données :*

*Notre application utilise une base de donnée. Pour la concevoir, nous avons utilisé MySql Workbench,*

 *Mettre en place une base de données*

 *Développer une interface utilisateur*

 *Développer des composants d'accès aux données*

 *Développer des pages web en lien avec une base de données*

 *Développer une application simple de mobilité numérique*

 *Utiliser l'anglais dans son activité professionnelle en informatique*

## CAHIER DES CHARGES OU EXPRESSIONS DES BESOINS DE L'APPLICATION À DÉVELOPPER

### Demande du client

La société Simplon évolue très rapidement. Le nombre de personnes y travaillant devenant de plus en plus important, elle a souhaité optimiser la gestion de ses absences. De ce fait, elle désire utiliser une application pour gérer les absences de son personnel.

Le salarié se connecte avec son adresse Gmail et un mot de passe . Un premier mot de passe lui est attribué, qu'il doit modifier à sa 1° connexion. Sur son profil, il peut visualiser ses reliquats de congés, ses anciennes demandes, modifier son mot de passe, accéder à son calendrier intégrer à l'application et faire une demande de congé.

Pour cela, il remplit un formulaire pour sa demande de congé. Un mail est automatiquement envoyé à son N+1, avec un lien pour valider ou refuser les congés.

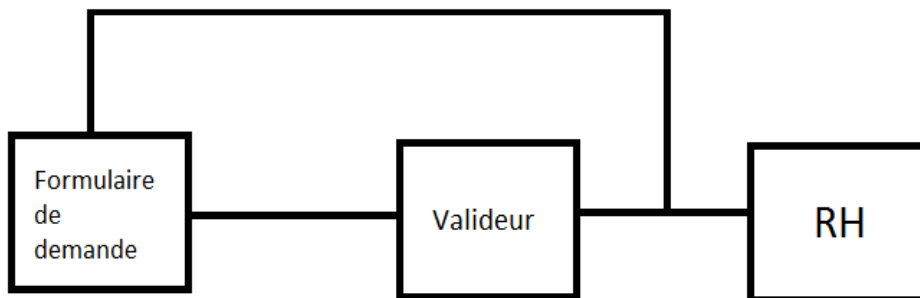
Le salarié est averti par mail de la réponse. Si il y a refus, la demande est enregistré dans la base avec la mention refus et le circuit s'arrête.

Si la demande est acceptée, elle est affichée dans le calendrier de l'agent et de l'équipe. Le N+1 a la possibilité de visualiser les congés de ses subalternes sur une période donnée. Il a aussi la possibilité de saisir des congés maladies ainsi que diverses absences.

Suite à la validation du congé par le N+1, un mail est envoyé au service RH pour information et vérification des règles RH. Le service Rh a la possibilité de visualiser les congés par employé, équipe ou la totalité de l'entreprise.

Un fichier mensuel des absences est généré pour le service paye.

La base de donnée est gérée par un administrateur système, qui pourra y faire toutes les opérations.



Workflow de la demande de congé

## Les différents acteurs

- Employé : La personne qui fait sa demande de congé. Elle visualise ses différentes absences via un calendrier ou un tableau. Elle a le niveau User.
- N+1 : le responsable hiérarchique de l'employé. Il valide les congés des membres de son équipe, visualise les congés de celle-ci grâce à un calendrier. Elle a aussi les mêmes droits de l'employé. Elle a le niveau valideur.
- Service Rh : Le service qui vérifie, d'un point de vue RH, la validité des congés. Elle peut aussi les modifier ou les annuler. Elle a le niveau RH.
- Administrateur de la base : Il saisie les nouveaux arrivants et gère les problèmes de la base. Il a le niveau admin.
- Service paye : C'est un acteur externe qui recevra un état mensuel des congés pour établir la paye.

## Gestion des utilisateurs

---

Les utilisateurs de l'application doivent s'authentifier à l'aide de leur adresse mail ainsi que de leur mot de passe donnant accès aux environnements pour lesquels ils sont accréditer.

Il revient à l'administrateur système de gérer la base des personnes. Il peut ainsi créer, modifier ou supprimer des utilisateurs. Un utilisateur se définit par son nom, son prénom, son matricule, son adresse mail, son ou ses rôles, l'équipe dont il dépend, le service RH le gérant. En cas de création, un mot de passe est généré automatiquement, celui-ci étant à changer dès la première utilisation. A noter que tous les utilisateurs ont à minima le rôle de demandeur.

L'administrateur doit aussi gérer les équipes ainsi que les Responsables, essentiel pour permettre l'usage du calendrier d'équipe.

## Gestion des absences

Différents type d'absences sont gérés par l'application, ainsi, un employé peut formuler une demande d'absence selon le cas de type congé payé, RTT, repos compensateur (récupération suite à heures supplémentaires), congé exceptionnel (maternité, paternité, fête religieuse, événement familial,...).

Un responsable d'équipe peut également déclarer l'absence inopinée d'un employé dont il a la gestion. Il a à sa disposition un formulaire de saisie qui envoie automatiquement un mail au service Rh. Il est aussi possible de déclarer une absence légitime dans le cadre d'un oubli de l'employé ou d'une régularisation.

Les compteurs de congés ou de RTT sont exprimés en jour ce qui n'est pas le cas pour le repos compensateur qui est exprimé en heure.

Le calcul des congés se fait en jour ouvré, car les samedis et dimanches ne sont pas travaillés, moins les jours fériés, si il y en a pendant les congés.

Par défaut, les jours fériés (au nombre de 11) sont considérés comme chômés, mais selon l'organisation de l'entreprise il est possible de définir un ou plusieurs jours



---

fériés comme jours travaillés. De même, les jours fériés tombant un samedi ou un dimanche sont retirés de la liste des jours fériés de l'année.

---

## Circuit de validation des demandes

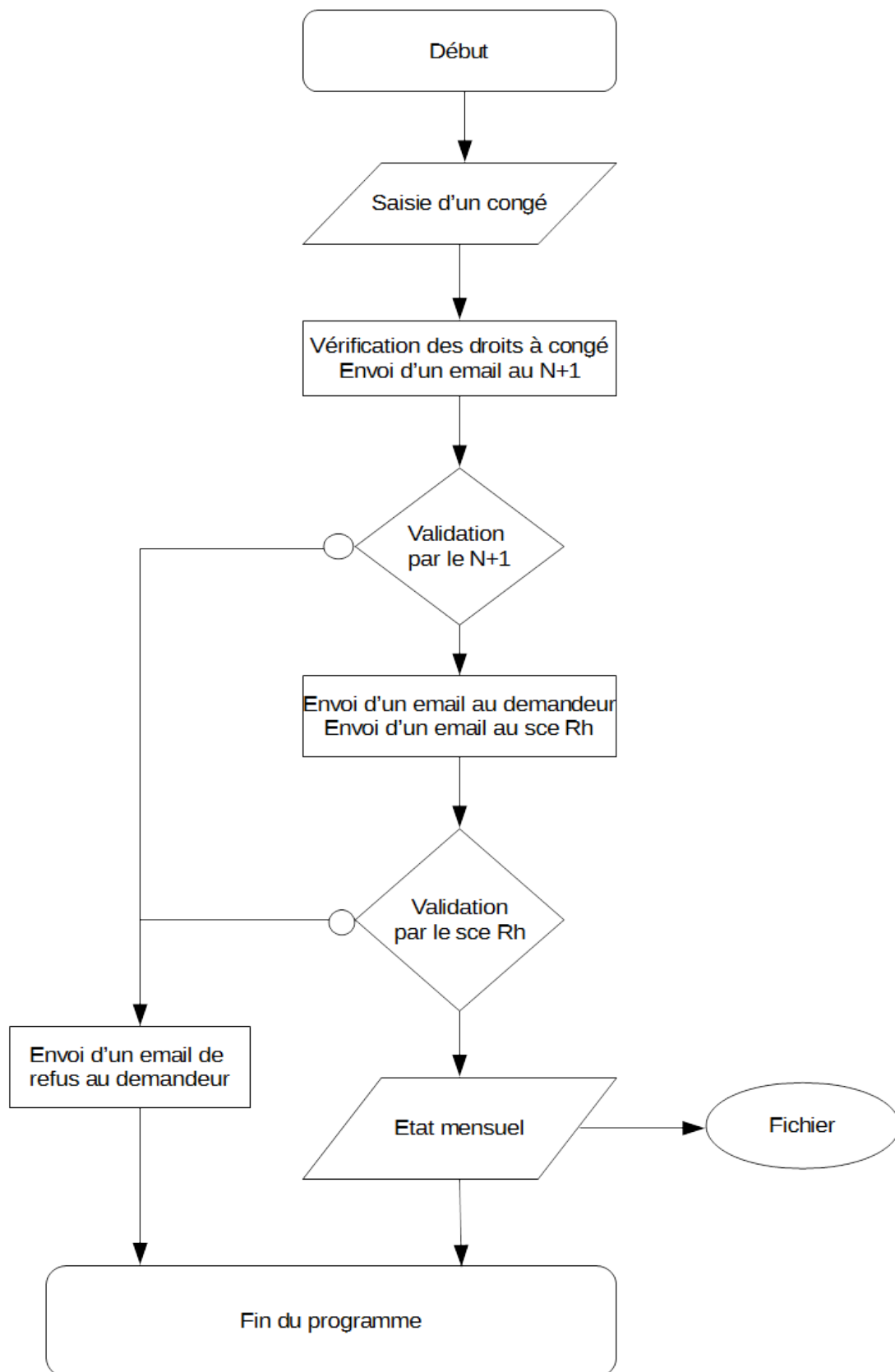
Un employé soumet une demande d'absence par le biais du formulaire prévu à cet effet. Une première vérification est faite au niveau de l'application, notamment en fonction des droits à congé et de la durée des congés (un congé payé ne peut dépasser 4 semaines).

La validation de la demande déclenche automatiquement l'envoi d'un mail destiné au N+1. Ce mail contient un lien permettant de valider ou non la demande.

Si la demande est validée, deux mails sont transmis, le premier au demandeur pour lui rendre compte de la validation et le deuxième destiné au service RH pour vérification en fonction des lois (pour certains types de congés, ex : congé parental, mariage, naissance...).

Dans le cas où le valideur, le service RH oppose un refus, un mail mentionnant le motif du refus est transmis au demandeur. Cela entraîne l'arrêt du workflow.

Le demandeur peut à tout moment suivre le parcours de sa demande. Il aura aussi la possibilité de relancer sa demande en attente de validation voire de l'annuler.

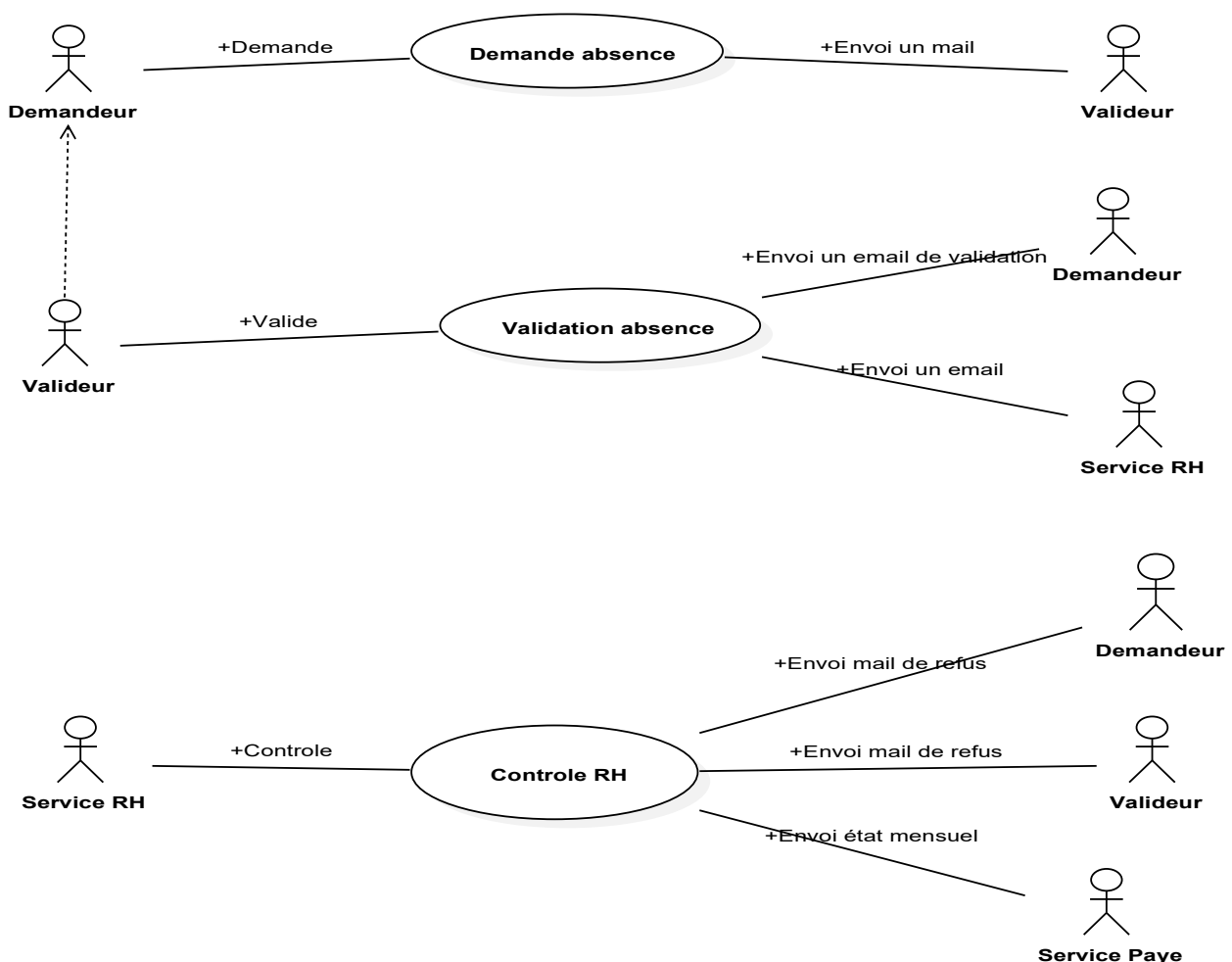
*Algorithme de la demande de congé*

## SPÉCIFICATIONS FONCTIONNELLES

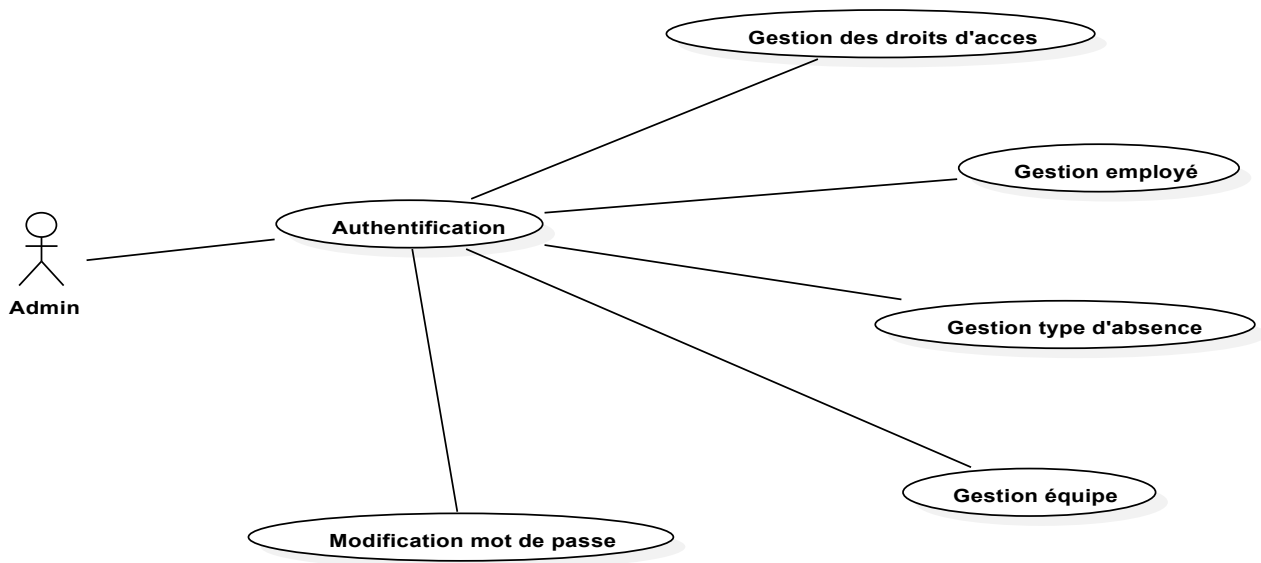
Le maquettage de l'application s'est fait avec des diagrammes UML (Unified Modeling Language). Les diagrammes UML permettent de visualiser la conception d'une application. C'est un langage de modélisation graphique utilisé dans la majorité des entreprises. Nous avons utilisé Star UML pour les diagrammes.

### Use case

Il permet d'identifier les possibilités d'interaction entre l'application et les acteurs (intervenants extérieurs à l'application), c'est à dire toutes les fonctionnalités que doit fournir l'application.



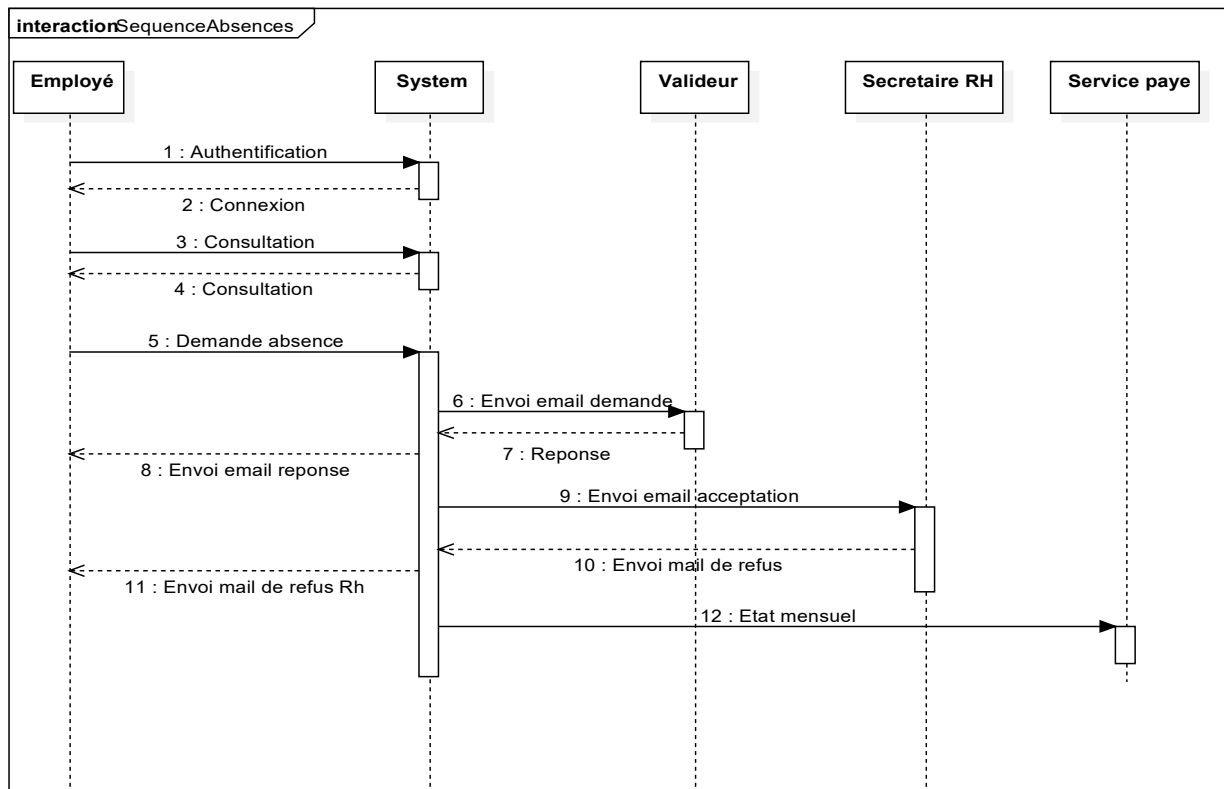
User case des employés, Valideur et Service Rh



User cas de l'administrateur système

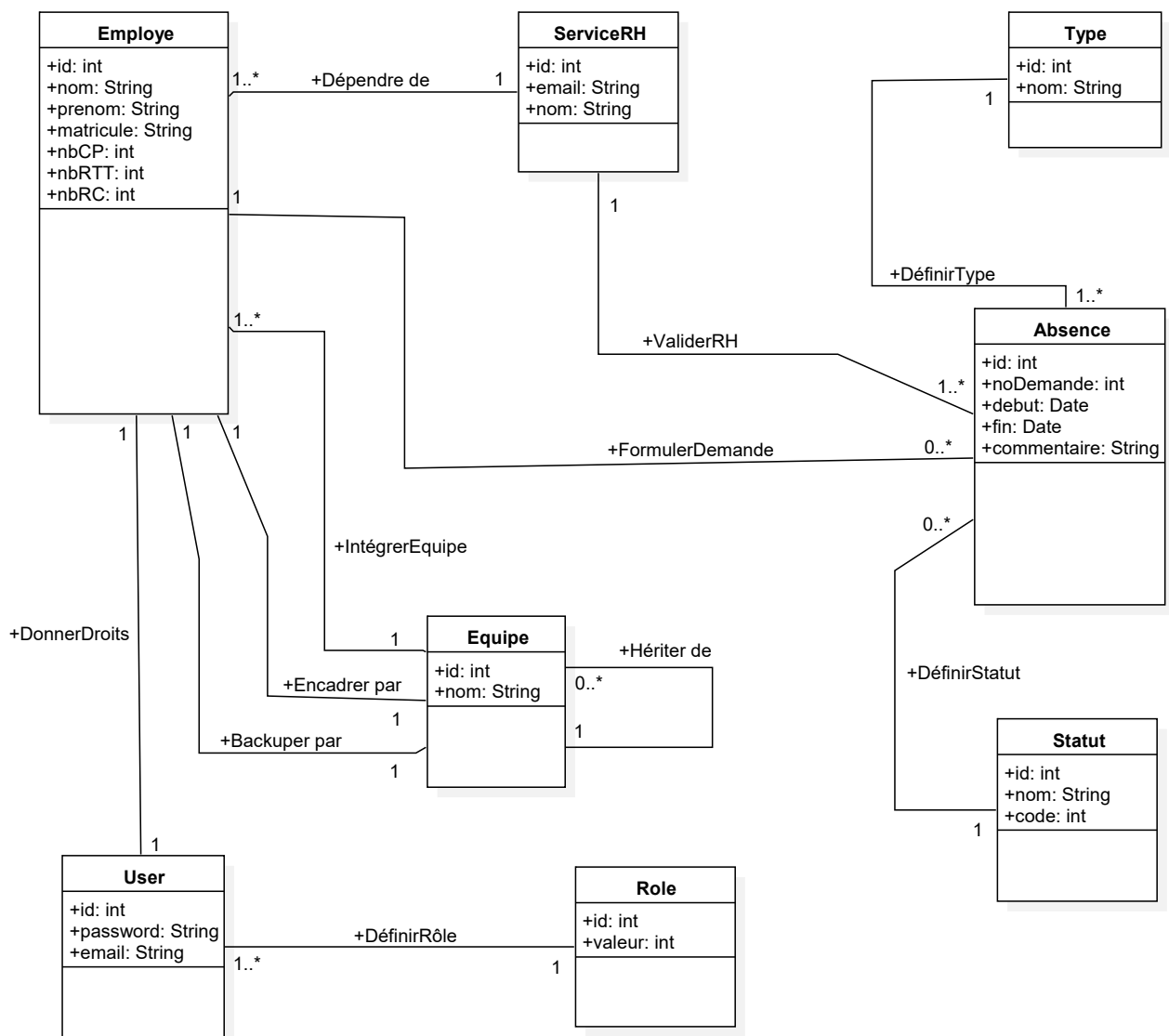
## Diagramme de séquence

Le diagramme de séquence représente la vie de l'application. Chaque action y est représenté de façon chronologique. Chaque intervenant est représenté par une ligne de vie. La communication se fait par des messages, envoi et réponse.



## Diagramme de classes

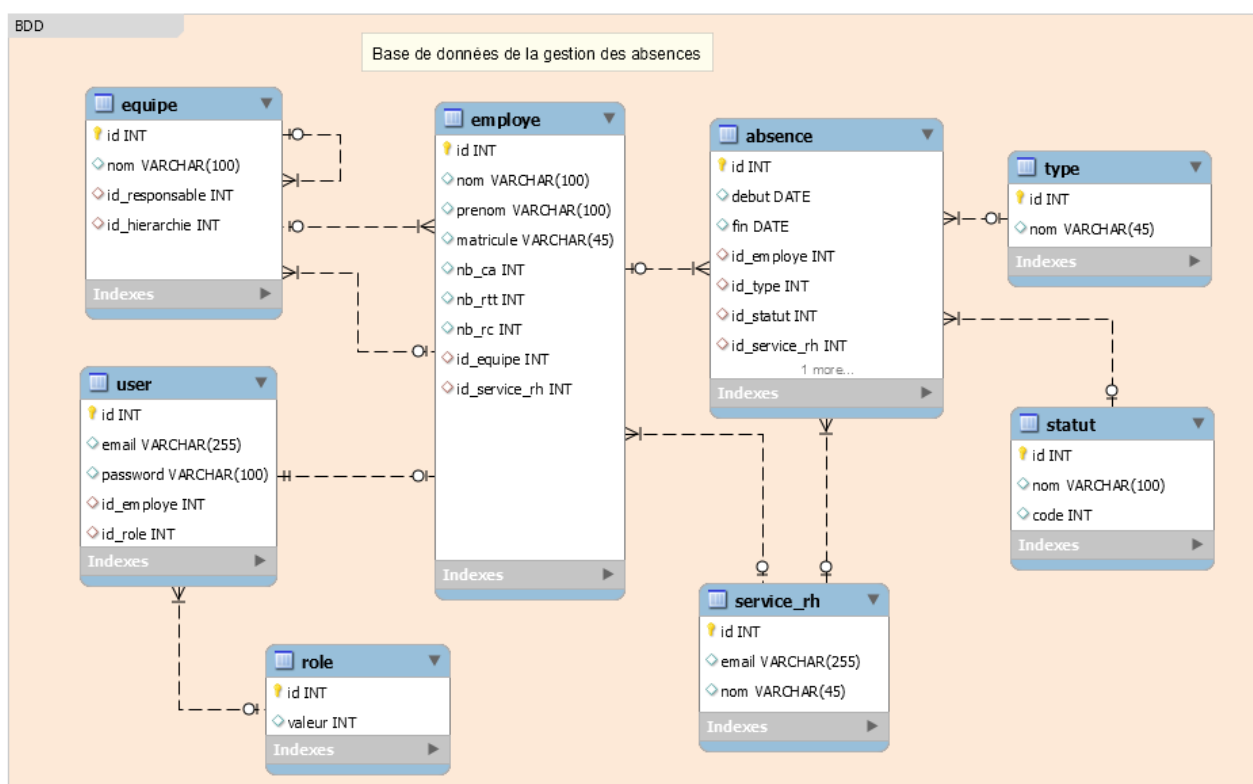
Il représente les classes intervenants dans le système. C'est une définition statique des éléments qui composent un système et leurs relations. Chaque classe représente un objet qui figurera dans application. C'est ce qui va générer le code en langage Java. Par conséquent, nous avons respecté les règles de nommage de Java.



## Modèle conceptuel des Données

Notre application nécessite une base de données, afin de pouvoir stocker les informations.

Le modèle conceptuel de données est une représentation graphique et structurée des informations de la base de données. Il matérialise les différentes entités ainsi que leurs relations entre elles. Cela permet d'avoir une base de données bien structurée, sans informations redondantes ou inutiles. Pour modéliser notre base, nous avons utilisé WorkBench, édité par la communauté MySQL.



## Maquettage de l'application

Le maquettage de l'application (ou Mock-up) est une étape indispensable dans la réalisation d'une application. Grâce à cela, nous pouvons voir le contenu des pages est organisé de façon logique, si la navigation est intuitive et agréable, quels types de données doivent être affichées. Cela permet aussi de savoir si il y a une cohérence graphique entre chaque page. Pour réaliser ces Mock-Up, nous avons utilisé Balsamiq.

The image shows a wireframe of a web application's authentication page. The browser's address bar displays 'http://gestion-absences/authentification'. The page has a header with a 'Company' logo on the left and navigation links 'Accueil', 'Aide', and 'Déconnexion' on the right. The main content area is titled 'Authentification' and contains a login form with two input fields: 'Adresse mail' and 'Mot de passe'. Below these fields are two buttons: 'Se connecter' and 'Modifier votre mot de passe'. The footer includes a copyright notice: '©Copyright juillet 2017 - Par Fred, Jean et Mokhtar de La Poste@Simplon' and a 'Powered' logo on the right.

Écran d'authentification



GESTION DES ABSENCES

http://gestion-absences/nouvelledemande

Company NOM Prénom - Matricule [Nouvelle demande](#) | [Mon calendrier](#) | [Calendrier Equipe](#) | [Accueil](#) | [Aide](#) | [Déconnexion](#)

### Nouvelle demande d'absence

Type  Congés payés   
 Date de début  RTT   
 Date de fin  RC   
 Valideur

©Copyright juillet 2017 - Par Fred, Jean et Mokhtar de La Poste@Simplon

## Écran de demande de congé

GESTION DES ABSENCES

http://gestion-absences/calendrierperso

Company NOM Prénom - Matricule [Nouvelle demande](#) | [Mon calendrier](#) | [Calendrier Equipe](#) | [Accueil](#) | [Aide](#) | [Déconnexion](#)

### Mon calendrier

#### CALENDRIER INDIVIDUEL

Mars 2017							Avril 2017							Mai 2017						
Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di
27	28	1	2	3	4	5	27	28	29	30	31	1	2	1	2	3	4	5	6	7
6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14
13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21
20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28
27	28	29	30	31	1	2	24	25	26	27	28	29	30	29	30	31	1	2	3	4

#### LEGENDE

Type d'absence	Validée	Formulée
Jour non travaillé		
Congé payé		
RTT		
Repos		
Autre absence		

< Avril 2017 >

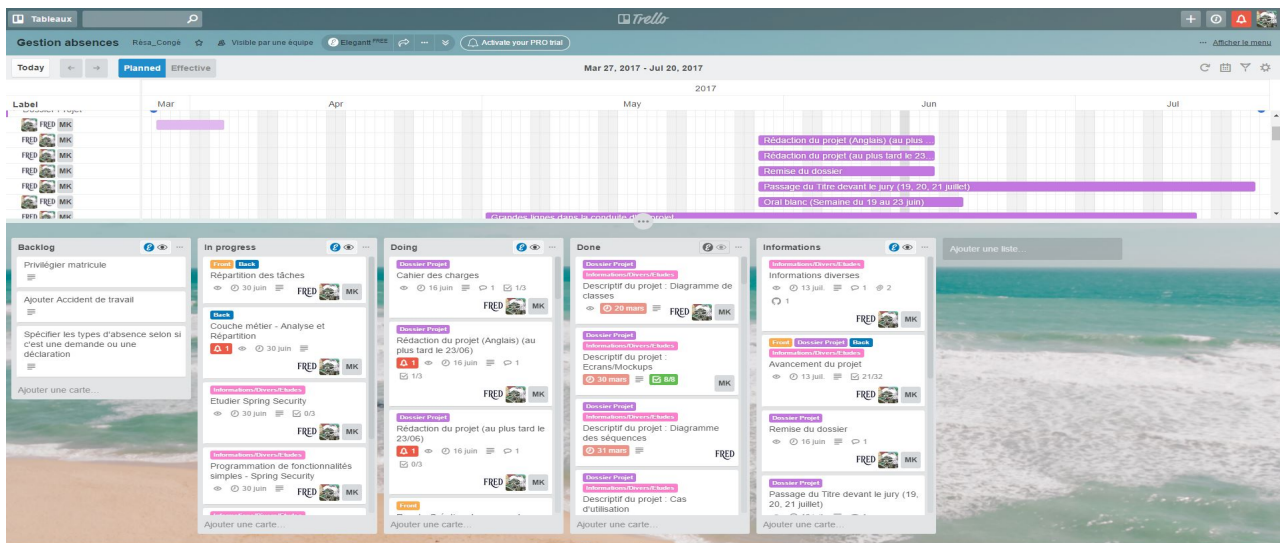
©Copyright juillet 2017 - Par Fred, Jean et Mokhtar de La Poste@Simplon

## Calendrier individuel de l'employé

## SPÉCIFICATIONS TECHNIQUES

## Organisation du projet

Le développement du projet a été effectué en groupe. Nous sommes 3 à participer à ce projet. Pour organiser et repartir le travail, nous avons utilisé un outil de gestion de projet en ligne : Trello. Cela a permis de se repartir le travail de façon efficace.

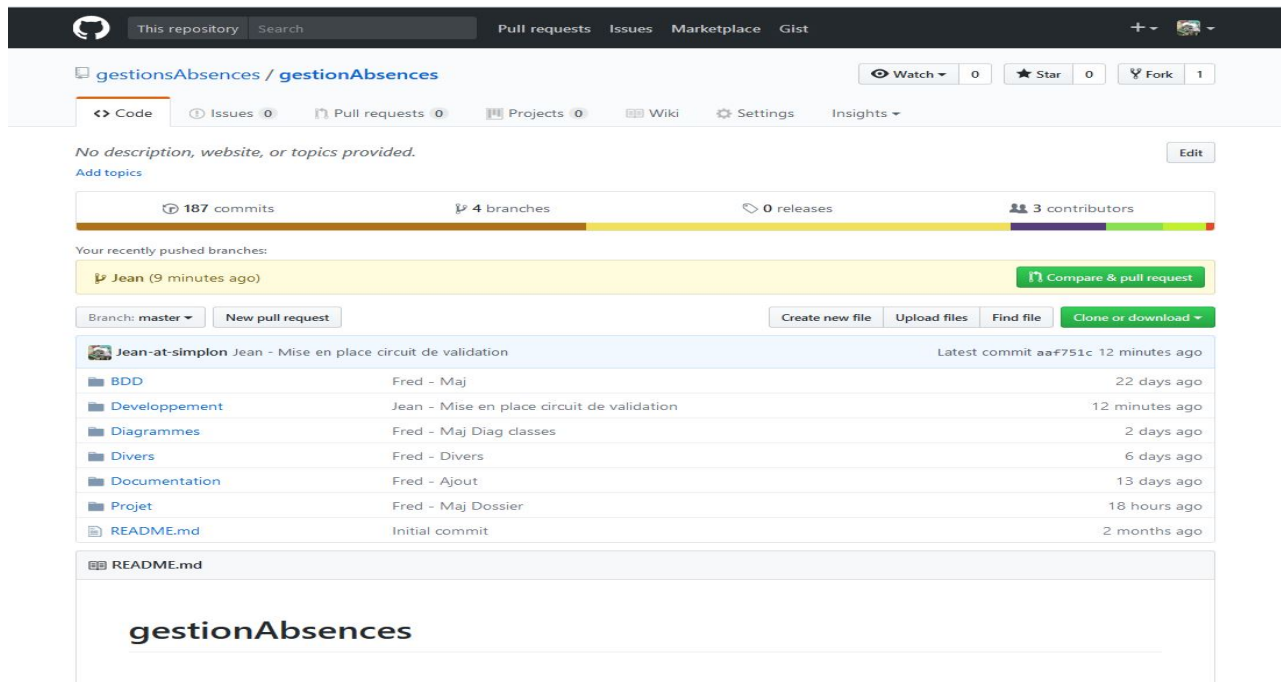


### Trello

Pour la sauvegarde et la vie du projet, nous avons utilisé l'outil de versionnage Git et sa version internet Github, du fait de sa gratuité.

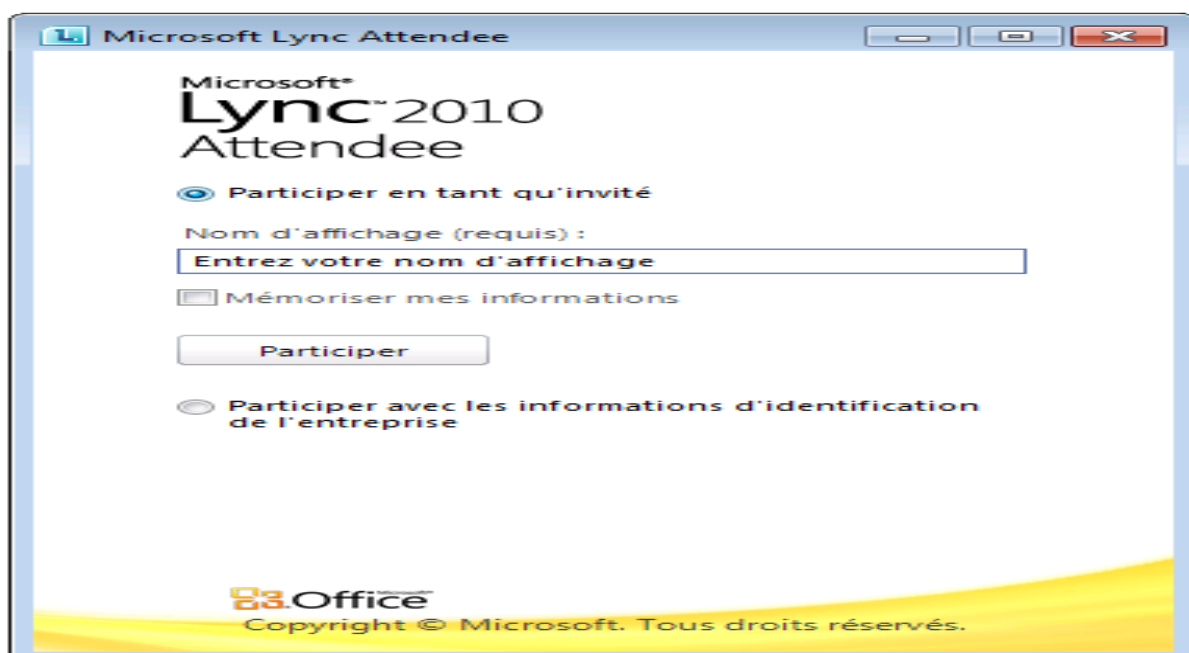
Git est un outil qui permet de stocker des données, tout en conservant la chronologie des modifications.

Il est décentralisé, ce qui permet de travailler à plusieurs sur un même projet, et de récupérer les anciennes versions de ce projet, en cas d'erreur.



## Git et Github

Pour la communication au sein de l'équipe, nous avons utilisé Microsoft Lync Attendee. C'est un logiciel de communication, regroupant messagerie instantanée, chat et vidéo-conférence. En vidéo-conférence, il permet le partage des écrans, de documents pour améliorer la communication.



## Back End

Le choix du langage pour le back-end a été Java EE.

Java Edition Enterprise est une plate forme normalisée, destinée au développement de logiciels pour les entreprises et les serveurs d'application. La plate forme comporte toutes les interfaces de programmation de Java SE (java Standard Edition), ainsi que des interfaces de programmation destinées à créer des applications web, envoyer des courriels, mettre en œuvre et utiliser des services web, mettre en œuvre de la persistance de données et des transactions.

Cela correspond parfaitement à ce que nous souhaitons réaliser, faire une application web qui utilise la persistance de données, les transactions, et gérer l'envoi de mails, en plus de la couche métier, à savoir le contrôle et la validation des demandes d'absences.

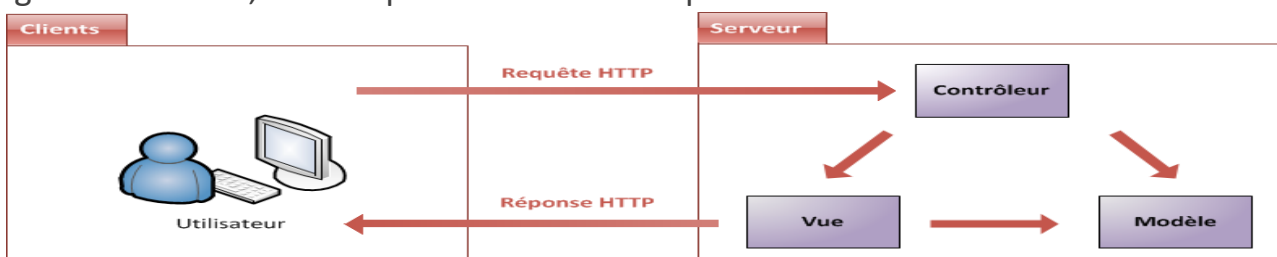
Le choix s'est porté sur le micro-framework Spring Boot pour sa simplicité d'utilisation.

Spring Boot permet d'utiliser le paradigme de P.O.A. (programmation orienté aspect), c'est à dire, l'utilisation de fonctionnalités indépendantes les unes des autres mais réutilisables entre elles. Ce paradigme repose aussi sur l'architecture MVC (modele – vue -contrôleur).

L'architecture MVC repose sur 3 modules :

- La vue : contient la présentation de l'interface graphique.
- Le modèle : contient les données à afficher
- Le contrôleur : contient la logique concernant les actions effectuées par l'utilisateur.

Outre le fait que chaque module puisse être développé individuellement et subir des évolutions de la même façon, l'architecture MVC permet d'offrir une plus grande sécurité, car chaque module est indépendant l'un de l'autre.



---

Le protocole utilisé pour la liaison entre le front et le back est basée sur la technologie REST (Representational State Transfer). C'est une architecture client/serveur qui dissocie le client du serveur, chacun ayant ses propres responsabilités. Ce qui permet de faire évoluer le client ou le serveur de manière totalement indépendante.

La communication entre les deux se fait au moyen d'une URI (Uniform Resource Identifier). Celle-ci est composée d'une adresse (ex: <http://www.simplomco.com>), associé à une entête dans laquelle se trouve un nombre limité de verbes(opérations). Ces verbes permettent d'effectuer des opérations de lecture (GET), écritures (POST), mises à jour (UPDATE) et suppression (DELETE) de données.

Cela implique aussi que le serveur ne garde aucune trace des requêtes du client, celui-ci doit envoyer toute l'information nécessaire pour permettre au serveur de comprendre la requête. Cela libère de nombreuses interactions entre le client et le serveur.

Le serveur, quant à lui, offre la possibilité de mettre en cache cette réponse, donc de décharger les contraintes aux serveurs, et des requêtes inutiles de la part des clients.

Le format des requêtes et des réponses peut être choisi, à savoir Xml, Json ou Yaml. Nous avons choisi Json, qui est le standard le plus utilisé actuellement.

Cela permet d'avoir une interface uniforme et un système hiérarchisé par couche, conforme au MVC.

Nous avons associé Spring Boot à Maven (outils de gestion et d'automatisation de production des projets java). Il permet de configurer facilement des projets à l'aide d'un fichier POM.xml (Project Object Model) et d'ajouter des librairies additionnelles pour étendre les possibilités de l'application. L'autre avantage de Maven est de pouvoir gérer le cycle de vie du développement d'une application, depuis son commencement jusqu'à son déploiement.

Autre avantage de Spring Boot est l'inversion de contrôle (IoC). Spring Boot contrôle la création des objets, leurs dépendance. Il gère le cycle de vie des objets et nous permet de nous concentrer principalement sur la couche métier de notre projet.

Nous avons utilisé des librairies additionnelles pour notre projet :

- Spring Security : sécurise une application par un module d'authentification ainsi que la gestion des accès de l'application (pages autorisées ou non à certaines personnes),
- Joda-Time : facilite la gestion et le calcul des dates tout en étant compatible avec Java (sera implémenté en natif dans Jee 8),
- JavaMail : permet d'utiliser le courrier électronique dans une application en java, nous n'avons utilisé que la partie envoi de mails,
- Thymeleaf : moteur de template en java, il génère du Html. Nous l'avons utilisé pour les templates des mails, afin d'externaliser ceux-ci et leur offrir un cycle de vie plus long.
- Junit : Framework open source pour le développement et l'exécution de tests unitaires. Les tests unitaires permettent de vérifier le bon fonctionnement de certaines méthodes ou certaines classes.
- Cobertura et Mockito : Cobertura permet de savoir ou faire les tests unitaires ainsi que mesurer le taux de couvertures des tests. Mockito simule certaines méthodes ou classes afin de faire des tests unitaires de façon totalement isolés et indépendantes du programme.

Pour la base de données , notre choix s'est portée sur MySQL, Il s'agit d'un système de gestion de base de données relationnels (S.G.B.D.R.) gratuit et universel. Il est utilisé par la majorité des hébergeurs et donc aisément portable d'un hébergeur à l'autre.

La relation entre Java et MySql se faisant à l'aide d'un O.R.M (Object Relational Mapping), nous avons choisi. Hibernate est un framework open source qui gère la persistance des données. Il gère lui-même les méthodes du C.R.U.D. (Creat, read, Update, Delete). La quantité de code ainsi épargnée est très importante. Autre avantage d'hibernate, il masque à la couche métier l'existence d'une base de données et améliore la sécurité de l'application.

Nous l'avons associé à JPA (Java Persistence Api) qui propose un niveau d'abstraction plus élevé. Cette Api permet de transformer des objets vers la base de données et réciproquement. Elle repose sur des entités annotés et sur un gestionnaire de ces entités (EntityManager). Ce gestionnaire propose des fonctionnalités simples pour

---

manipuler les objets et est responsable de l'état des entités et de leur persistance dans la base de données. Ces entités sont de simples POJOs (Plain Old Java Object) , sérialisable, sans constructeur, et accessible par des méthodes getter et setter, dénommé JavaBean.

## RÉALISATIONS



## CONCLUSIONS