

RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du

Diplôme National d'Ingénieur en Sciences Appliquées et Technologiques

Spécialité : Génie Logiciel et Systèmes d'Information

Par

Alaeddine JELASSI

Mise en place d'une application d'automatisation des processus métier

Encadrant professionnel : Mourad ANEMICHE

Manager Consulting et
Consultant expert

Encadrant académique : Rim KAABI

Maître Assistante

Réalisé au sein de Sopra HR SOFTWARE



RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du

Diplôme National d'Ingénieur en Sciences Appliquées et Technologiques

Spécialité : Génie Logiciel et Systèmes d'Information

Par

Alaeddine JELASSI

Mise en place d'une application d'automatisation des processus métier

Encadrant professionnel : Mourad ANEMICHE

Manager Consulting et
Consultant expert

Encadrant académique : Rim KAABI

Maître Assistante

Réalisé au sein de Sopra HR SOFTWARE



J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant professionnel, **Monsieur Mourad ANEMICHE**

Signature et cachet

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant académique,

Signature

Dédicaces

Je dédie ce travail à :

Mon cher père Mohamed

Qui n'a jamais cessé de me soutenir, m'assister et m'encourager, à celui qui a sacrifié ses plus belles années pour embellir les miennes, je lui dois ma réussite. Aucune dédicace, ne saurait exprimer à sa juste valeur le profond amour que je lui porte...

Ma chère mère Mounira

Aucune dédicace ne puisse exprimer ce que je lui dois, pour sa bienveillance depuis ma plus jeune enfance, son action et son soutien. Elle a comblé ma vie de tendresse, d'action et de compréhension.

Ma adorable soeur Ameni

Pour la merveilleuse enfance que nous avons vécue ensemble, pour leur encouragements incessants et pour leur bienveillance permanente...

Alaeddine

Remerciements

Mes remerciements les plus vifs à dieu le tout puissant qui ne m'a jamais privé de son soutien et son soin divin. J'adresse mes remerciements à tous mes enseignants de l'Institut Supérieur d'informatique pour la qualité de leur enseignement. Je tiens à exprimer ma reconnaissance à mon encadrante académique Madame Rim KAABI pour son encouragement qui m'a aidée à aller jusqu'au bout dans l'élaboration de mon projet de fin d'étude, ainsi que son soutien, sa disponibilité et sa patience. Je tiens à remercier mes encadrants à Sopra HR Software, Monsieur Mourad ANEMICHE, Madame Donia SOULI et Madame hajer BEN AYED pour avoir accepté la délicate charge de m'encadrer et pour les recommandations et les conseils qu'ils m'ont accordés tout au long de ce stage.

Je remercie sincèrement les collaborateurs OSD du Sopra HR ainsi que les membre du départements RD pour leurs conseils et leur aides précieuses. Je remercie également les membres du Jury d'avoir accepté de juger mon travail.

Table des matières

Introduction générale	1
1 Contexte général du projet	3
1.1 Présentation de l'organisme d'accueil	4
1.1.1 Présentation de Sopra HR	4
1.1.2 Présentation de l'équipe « Outsourcing Service Delivery »	6
1.2 Contexte du projet	6
1.3 Étude et critique de l'existant	7
1.4 Solution proposée	8
1.5 Méthodologie de travail	10
2 Étude Préliminaire	11
Introduction	12
2.1 Le concept métier	12
2.2 Architecture des produits de Sopra HR	12
2.3 Description de la base de données HR ACCESS	13
2.3.1 Les structures de données	15
2.3.2 Les informations	16
2.3.3 Organisation dans la base de données	16
2.3.4 Modèle physique de données	17
Conclusion	19
3 Analyse et spécification des besoins	20
Introduction	21
3.1 Les rôles Scrum	21
3.2 Identification des acteurs	21
3.3 Identification des besoins	22
3.3.1 Besoins fonctionnels	22
3.3.2 Besoins non fonctionnels	22
3.4 Backlog de produit	23

3.5	Modélisation des besoins fonctionnels	25
3.5.1	Diagramme de cas d'utilisation général	25
3.5.2	Raffinement et description textuelle des cas d'utilisation prioritaires	26
3.6	Planification des sprints	33
	Conclusion	33
4	Initialisation du projet	35
	Introduction	36
4.1	Initialisation	36
4.2	Architecture de la solution	36
4.2.1	Architecture logique	36
4.2.2	Architecture logicielle	38
4.3	Conception détaillée	41
4.3.1	Le diagramme de déploiement	41
4.3.2	Diagramme de composants	42
4.4	Environnement technique	43
4.4.1	Environnement matériel	43
4.4.2	Environnement logiciel	44
4.4.3	Technologies et langages utilisés	45
4.5	Environnement de travail	46
5	Implémentation et test du Sprint 1 et Sprint 2	48
	Introduction	49
5.1	Mise en place du Sprint 1	49
5.1.1	Baklog Sprint 1	49
5.1.2	Conception	52
5.1.3	Interfaces Sprint 1	54
5.2	Mise en place du Sprint 2	56
5.2.1	Backlog Sprint 2	56
5.2.2	Conception	61
5.2.3	Interfaces Sprint 2	63
5.3	Problèmes rencontrées	64

6 Implémentation et test du Sprint 3 et Sprint 4	66
Introduction	67
6.1 Mise en place du Sprint 3	67
6.1.1 Baklog Sprint 3	67
6.1.2 Conception	72
6.1.3 Interfaces Sprint 3	74
6.2 Mise en place du Sprint 4	76
6.2.1 Baklog Sprint 4	77
6.2.2 Conception	79
6.2.3 Interfaces Sprint 4	81
6.3 Problèmes rencontrées	82
Conclusion générale	84
Bibliographie	85

Table des figures

1.1	Informations relatives à Sopra HR SOFTWARE	4
1.2	Sopra Steria et les solutions RH	5
1.3	Échantillon du canevas Excel de la création d'une société «Partie identification d'une société»	8
1.4	Schéma global de la solution	9
2.1	Schéma de l'architecture HR Access	13
2.2	Organisation des données	14
2.3	Correspondance entre l'organisation de la base données de HR ACCESS et l'organisation de la base données relationnelle	15
2.4	Définition de la structure de données	16
2.5	Organisations physiques de la base de données HR ACCESS	17
2.6	Modèle physique de données de l'application métier	19
3.1	Diagramme de cas d'utilisation général de l'application	26
3.2	Raffinement du cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier»	28
3.3	Raffinement du cas d'utilisation «Gérer les demandes des clients»	30
3.4	Raffinement du cas d'utilisation «Instruire le dossier de la demande»	32
4.1	Architecture physique de HR Access	37
4.2	Architecture physique de notre solution	37
4.3	Architecture logicielle de l'AAPM	39
4.4	Architecture de Drools	40
4.5	Exemple concret de règle métier	41
4.6	Diagramme de déploiement de l'application	42
4.7	Diagramme de composants de l'application	43
4.8	Environnements de travail	46
5.1	Diagramme de classe du premier sprint	53
5.2	Diagramme de séquence «S'authentifier»	54

5.3	Interface authentification des utilisateurs de l'application	55
5.4	Interface d'ajout de demande de la part du client	55
5.5	Interface de la création d'une société «Identification d'une société»	56
5.6	Diagramme de classes du deuxième sprint	62
5.7	diagramme de séquence «Ajouter Dossier Établissement»	63
5.8	identification juridique de l'établissement	64
6.1	Diagramme de classe du troisième sprint	73
6.2	diagramme de séquence «Générer automatiquement le paramétrage des données d'un dossier»	74
6.3	Interface de notifications reçues dans la session du consultant	75
6.4	Interface de la liste des demandes reçues	75
6.5	Interface de la fiche technique générée	76
6.6	Interface de mise à jour de la fiche technique	76
6.7	Diagramme de classe du quatrième sprint	80
6.8	Diagramme de séquence «Modifier une version du dossier»	81
6.9	Interface de consultation d'un client ses demandes effectuées	82
6.10	Interface chargement de version du dossier	82

Liste des tableaux

2.1	Description des tables de l'application AAPM	17
3.1	Backlog du produit	23
3.2	Description textuelle de cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier»	29
3.3	Description textuelle de cas d'utilisation «Gérer les demandes des clients»	31
3.4	Description textuelle de cas d'utilisation «Instruire le dossier de la demande»	32
3.5	Planification des sprints	33
4.1	Caractéristiques de l'ordinateur	43
5.1	Backlog du sprint 1	49
5.2	Backlog du sprint 2	56
5.3	Obstacles rencontrés dans la réalisation des sprints 1 et 2	65
6.1	Backlog du sprint 3	67
6.2	Backlog du sprint 4	77
6.3	Obstacles rencontrés dans la réalisation des sprints 3 et 4	83

Liste des abréviations

- **AAPM** = **A**pplication d'**A**utomatisation des **P**rocessus métier
- **BPO** = **B**usiness **P**rocess **O**peration
- **CSM** = **S**crum **M**aster **C**ertification
- **CSPO** = **C**ertified **S**crum **P**roduct **O**wner
- **GLSI** = **G**énie **L**ogiciel et **S**ystèmes d'**I**nformation
- **MDP** = **M**odèle **P**hysique de **D**onnées
- **OSD** = **O**utsourcing **S**Service **D**elivery
- **PO** = **P**roduct **O**wner
- **SIRH** = **S**ystème d'**I**nformation des **R**essources **H**umaines
- **TMA** = **T**eam **M**aintenance **A**ctivity

Introduction générale

La performance des entreprises dépend aujourd'hui de leur capacité à pouvoir délivrer des services informatiques de façon rapide, fiable et la moins coûteuse. Pour une entreprise qui fournit des services de ressources humaines (RH), ces derniers sont généralement sous la forme de ce que nous appelons des processus métiers.

De ce fait, tous les consultants RH doivent fournir un service de qualité qui répond à l'ensemble des besoins du client. Pour ce faire, il est impératif de travailler sur trois axes : le coût, le temps et la qualité car les risques, en cas d'échec, peuvent être dommageables. En fait, les tâches routinières qui demeurent manuelles, risquent de réduire le taux de productivité et de minimiser le rendement du personnel, mais en plus augmentent le taux d'erreurs et de blocage.

Actuellement, la thématique d'automatisation des processus métiers au sein des entreprises est en plein essor dans le domaine des ressources humaines. De plus les entreprises accordent une importance particulière aux solutions d'automatisations paramétrables. Ceci justifie la forte présence de la notion de moteur de règles qui est une solution idéale pour des applications contenant une certaine logique métier.

C'est dans ce contexte, que Sopra HR Software a décidé de mettre en place une application d'automatisation de processus métier (AAPM) de son système d'information de gestion des ressources humaines (SIRH) HR ACCESS et de faciliter ainsi la tâche des consultants. A cet effet, notre travail consiste à développer une application qui va être l'intermédiaire entre le client et le SIRH de façon que les demandes venants d'un client et qui suivent des processus métier bien déterminés doivent être traitées dans notre application pour qu'elles soient conforme à la structure spécifique et complexe de HR ACCESS afin d'être injectées dans la base de données spécifique de l'SIRH dont nous allons mettre l'accent dessus dans ce qui suit. Les processus métiers que nous allons travailler dessus dans notre projet s'articulent principalement autour de deux axes qui sont la création d'une société et la

création d'un établissement. L'objet de ce présent travail consiste à la conception et la réalisation de cette solution.

Le présent rapport présentera les différentes étapes de la réalisation de ce projet et s'étalera sur six chapitres :

Le premier chapitre « Présentation Générale du Projet » est un chapitre introductif dans lequel nous effectuons une brève description de l'entreprise. Ensuite, nous exposons le cadre général du projet et la solution proposée.

Le second chapitre « Étude Préliminaire » fait l'objet d'une brève description de la méthodologie adoptée, ainsi qu'une description de l'architecture des produits de Sopra HR et la base de données, puis une définition des concepts de base pour la compréhension de notre projet.

Le troisième chapitre nommé « Analyse et Spécification des Besoins » définit la réalisation de ce travail avec la méthodologie Scrum. Nous dégageons également les besoins fonctionnels dans un Backlog de produit et nous définissons le planning des sprints. Par la suite, nous spécifions les besoins fonctionnels illustrés par des diagrammes de cas d'utilisation avec quelques scénarios suivis d'une spécification des besoins non fonctionnels de notre application.

Le quatrième chapitre intitulé « Initialisation du projet » est dédié à l'initialisation du projet et la mise en place de l'environnement de développement, l'architecture ainsi que l'environnement matériel et logiciel.

Les deux chapitres suivants seront dédiés à l'implémentation de notre application où nous nous intéressons à la réalisation des sprints répartis chacun en trois modules, analyse, conception et réalisation.

Nous clôturons, finalement, ce rapport par une conclusion générale dans laquelle nous évaluerons les résultats atteints et nous exposerons les perspectives éventuelles du présent projet.

CONTEXTE GÉNÉRAL DU PROJET

Plan

1	Présentation de l'organisme d'accueil	4
2	Contexte du projet	6
3	Étude et critique de l'existant	7
4	Solution proposée	8
5	Méthodologie de travail	10

Introduction

Le premier chapitre met le travail dans son contexte général. Il se compose de trois parties. La première a pour but de présenter l'organisme d'accueil Sopra HR Software, la deuxième se focalisera sur la présentation de la problématique et la solution proposée et la troisième a pour objectif d'expliquer le choix méthodologique.

1.1 Présentation de l'organisme d'accueil

Nous avons effectué notre stage de fin d'étude au sein de l'unité Outsourcing Service Delivery (OSD) du département service de l'entreprise SOPRA HR Software.

1.1.1 Présentation de Sopra HR

Fondée en 1986, Sopra Steria est l'entreprise mère de Sopra HR Software qui a réalisé un chiffre d'affaire de 3,7 milliards d'euros en 2016, de plus elle est présente dans plus de 20 pays dans le monde et offre un emploi pour 40 000 collaborateurs [1]. La figure 1.1 montre les solutions qu'elle offre dans le domaine de Ressource Humaine (RH).

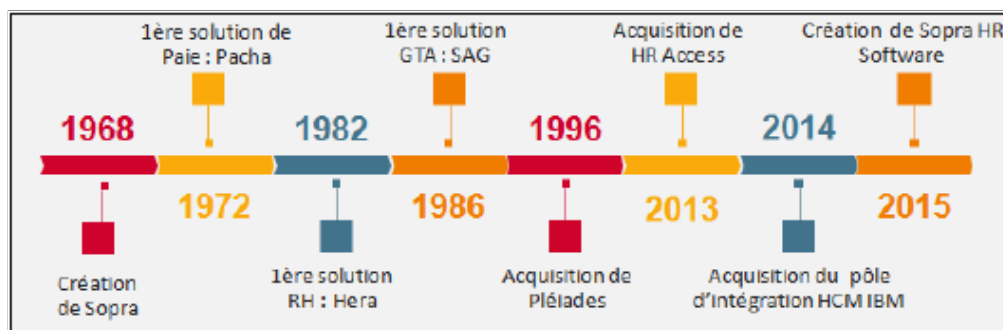


Figure 1.1: Informations relatives à Sopra HR SOFTWARE

Sopra HR Software propose des solutions de RH pour répondre aux besoins des Directions des Ressources Humaines (DRH) et des grandes et moyennes entreprises. La figure 1.2 met en évidence les différentes informations de base concernant Sopra HR



Figure 1.2: Sopra Steria et les solutions RH

Les solutions de Sopra HR sont servies à plus de 850 clients qui les déploient dans plus de 54 pays. Ces solutions satisfont les besoins des entreprises privées et publiques, au sein de tous les secteurs d'activité. Sopra HR répond aux enjeux de croissance, de transformation et de performance de ses clients par une organisation dédiée. En offrant deux lignes d'activités :

- services d'externalisation orientés métiers,
- éditeur de progiciels, HR Suite, complétée par des programmes de formation et des services d'expertise délivrés par des consultants technico-fonctionnels.

Le produit Sopra HR est spécialisé dans la gestion des ressources humaines, caractérisé par son moteur de paie performant qui offre la possibilité de modifier sa configuration au souhait du client. Sopra HR offre deux solutions Pléiades et HR Access pour satisfaire les besoins suivants [4] :

- pilotage et performance (CSP, reporting, analytics, contrôle de gestion sociale, simulation),
- core HR international (référentiel groupe, Benchmark social, politique RH),
- gestion administrative et paie (organisation, processus, gestion des risques professionnels, numérique, déclarations),
- espace collaboratif (Engagement, communication, individualisation, collaboratif),
- gestion des talents (développement RH, formation, Recrutement, mobilité, évaluation),
- gestion des activités et des temps (Planification, adéquation des besoins et des ressources).

1.1.2 Présentation de l'équipe « Outsourcing Service Delivery »

La mission de l'équipe Outsourcing Service Delivery (OSD) est d'accompagner ses clients dans la réussite de leurs projets en :

- faisant le suivi de la facturation,
- animant le comité de suivi, de pilotage et de direction
- assurant le suivi de la paie,
- participant à la correction des anomalies et au développement des évolutions,
- testant et livrant le produit,
- participant aux négociations de renouvellement des contrats.

1.2 Contexte du projet

Sopra HR Software offre plusieurs services métiers à leurs clients. Ces derniers s'effectuent par l'intervention des collaborateurs des deux équipes Team Maintenance Activity (TMA) et Business process operation (BPO) de l'unité organisationnelle OSD, à l'aide des deux lignes de produits HR Access et Pléiades. Le rôle des consultants est, alors, de comprendre et d'analyser les besoins des clients en terme du métier des ressources humaines et de les traduire techniquement via la plateforme HR Access.

L'enjeu principal de la société est, par conséquent, l'amélioration de la qualité de ses services en réduisant tant que possible l'intervention des équipes TMA et BPO dans le cycle de développement et la réalisation des besoins métier de leurs clients.

Suite à ce constat, notre sujet est né d'une réflexion générale dans le contexte de digitalisation et l'automatisation de deux processus métiers existants dans Sopra HR Software qui sont les suivants :

- Le processus de création d'une société,
- Le processus de création d'un établissement.

Ces deux processus métier sont dépendants et contiennent des sous processus métier comme les processus de la création des unités, la création des postes et la création des emplois. On appelle un processus métier un besoin fonctionnel du client qui sera transformé sous forme de paramétrage et de règles métiers et injecté, par la suite, dans HR Access.

1.3 Étude et critique de l'existant

Afin de comprendre l'intérêt de développer une telle application, il est nécessaire réaliser une étude de l'existant au sein de Sopra hr.

Actuellement, à Sopra HR Software, la phase d'analyse des besoins clients, ainsi que leurs mise en production consomment beaucoup de ressources et se fait d'une façon manuelle par les collaborateurs TMA et BPO. En effet, ils passent beaucoup de temps à interagir avec le client afin de comprendre ses besoins spécifiques. Cette phase d'analyse des besoins a comme résultat un canvas Excel bien rempli avec les données du client et vérifié en terme de cohérence avec celles qui existent dans la base HR Access. Une fois ces données sont cohérentes et compatibles avec la logique du métier, on passe à la deuxième étape qui est la mise en production des services analysées précédemment. Ces informations seront validées par les collaborateurs afin d'être introduites techniquement dans le SIRH. Le volume des données pour les deux processus métier que nous allons travailler dessus est assez grand, aussi que l'injection de ces données dans HR ACCESS est une tâche vraiment fastidieuse car ce n'est pas juste une simple saisie de formulaires mais en effet, chaque données nécessite une certaine configuration spécifique au niveau de HR ACCESS avant de l'insérer. Ce processus de mise en production de ces besoins est fastidieux et coûteux en terme de :

- temps
- ressources
- connaissances techniques

La figure 1.3 présente un échantillon du canevas Excel de la création d'une société, plus précisément c'est la phase de l'identification d'une société qui constitue juste 10% en terme de volume de données d'un dossier d'une société.

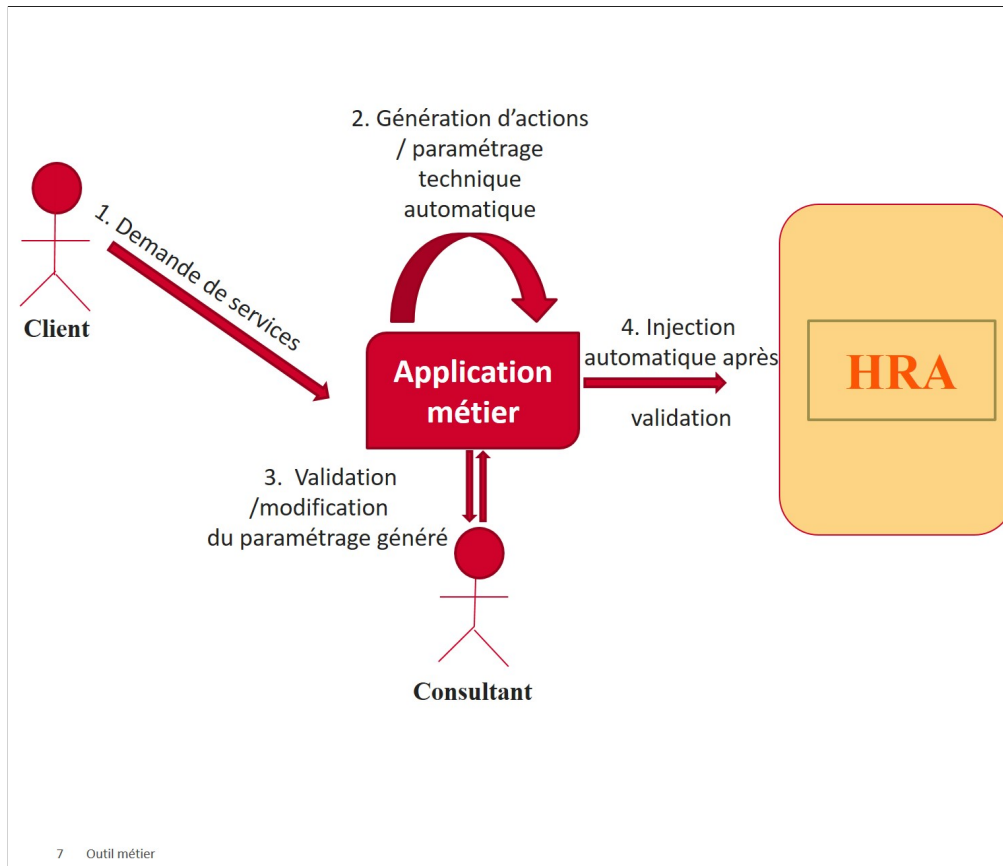


Figure 1.4: Schéma global de la solution

Le déroulement du cycle de la nouvelle solution passe par quatre étapes :

- **Etape 1** : le client instruit son dossier en précisant toutes les informations nécessaire à la bonne prise en compte dans le produit.
- **Etape 2** : prise en compte par l'application des éléments saisis et proposition des actions à faire dans le SIRH de manière automatique. Le lancement des actions est le résultat des traitements effectués par notre moteur de règle Drools, que nous l'avons alimenté par les règles de gestion adéquates aux informations saisies.
- **Etape 3** : validation d'un consultant de l'équipe TMA le paramétrage généré automatiquement. En effet c'est ce paramétrage qui va assurer la cohérence et le mappage entre la base de données classique et la structure spécifique de la base de données de l'SIRH.
- **Etape 4** : Injection des données saisies dans la base de HR Access.

1.5 Méthodologie de travail

Nous avons choisi d'utiliser la méthode de SCRUM [1] afin de gérer notre projet vue qu'elle répond parfaitement aux exigences demandées. Cette méthode va nous permettre, en cas de besoin, d'ajouter ou de modifier des fonctionnalités au niveau d'un sprint désiré sans affecter les autres. En plus, cette méthodologie nous permet de mieux respecter les dates planifiées pour clôturer les différents sprints et de ne pas les dépasser. La méthodologie Scrum appartenant à la méthodologie agile [2] est destinée aux projets de moyenne et haute complexité, ayant comme principe de travail la répartition du projet en des périodes de temps nommés « **Sprints** » qui peuvent durer d'une semaine à un mois. Chaque jour toute l'équipe se réunit pour « **la mêlée quotidienne** » qui dure au maximum 15 minutes afin de mettre au point les tâches en cours de réalisation. A la fin de chaque sprint, l'équipe de développement se réunit pour effectuer « **la revue du sprint** » qui se traduit par la réalisation du bilan des sprint, suivi d'une « rétrospective » pendant laquelle l'équipe révisé le rendu du sprint pour des petites rectifications ainsi que les problèmes rencontrés pour les résoudre par la suite avant de classer la tâche comme « **valide** ». En suivant le même enchaînement pour tous les sprints, on aura enfin la version « **release** » du travail qui est la version finale qui peut être livrée au client.

Par ailleurs, il faut noter que Sopra hr adopte Scrum depuis un certain nombre d'années et la plupart de ses consultants sont certifiés CSM [3] ou/et CSPO [4]. Par conséquent, la culture Agile fait partie de leur quotidien. Mise à part, Sopra hr a investi dans un outil phare qui est JIRA [5] pour que toute la gestion des projets soit outillée et conforme aux pratiques Scrum. Pour toutes ces raisons, on a tranché pour l'adoption de cette méthode afin de bénéficier de la maturité de Sopra hr, son savoir faire et son retour d'expérience.

Conclusion

Ce chapitre est un point de départ pour la présentation de notre projet, puisqu'il décrit son cadre général en présentant l'organisme d'accueil Sopra HR SOFTWARE, la problématique, l'étude et la critique de l'existant, les objectifs à atteindre ainsi qu'une description de la méthodologie de développement SCRUM. Dans le chapitre suivant, nous allons exposer la première étape de notre projet à savoir l'analyse et la spécification de différentes exigences des clients.

ÉTUDE PRÉLIMINAIRE

Plan

Introduction	12
1 Le concept métier	12
2 Architecture des produits de Sopra HR	12
3 Description de la base de données HR ACCESS	13
Conclusion	19

Introduction

Ce chapitre présente une étude préliminaire, allons définir le concept métier pour mieux comprendre notre projet , ensuite nous allons présenter l'architecture des produits de Sopra HR, enfin nous allons décrire la base de données HR ACCESS.

2.1 Le concept métier

Le concept d'une application métier c'est le fait d'arriver à fluidifier les demandes clients afin qu'elles soient prises en compte d'une façon véloce. Les demandes sont purement fonctionnelles qui reflètent le besoin métier. Elles sont formulées par le client en langage non technique, d'où la nécessite d'une application qui joue le rôle d'un intermédiaire entre le client et le SIRH, qui génère le paramétrage nécessaire, afin que cette demande soit pris en considération dans le SIRH. À la date d'aujourd'hui la prise en compte d'une demande de création de société d'une façon manuelle (par exemple) prend entre 4 et 10 semaines, cette période comprend :

- La prise en compte de la demande client,
- Une phase d'analyse incluant les échanges avec le client pour clarifier le besoin,
- Un atelier afin de clarifier certains points compliqués si besoin,
- Une validation du besoin par le client via un document d'analyse détaillé,
- Un paramétrage et un développement de la solution.

2.2 Architecture des produits de Sopra HR

La figure 2.1 présente le schéma de l'architecture HR Access.

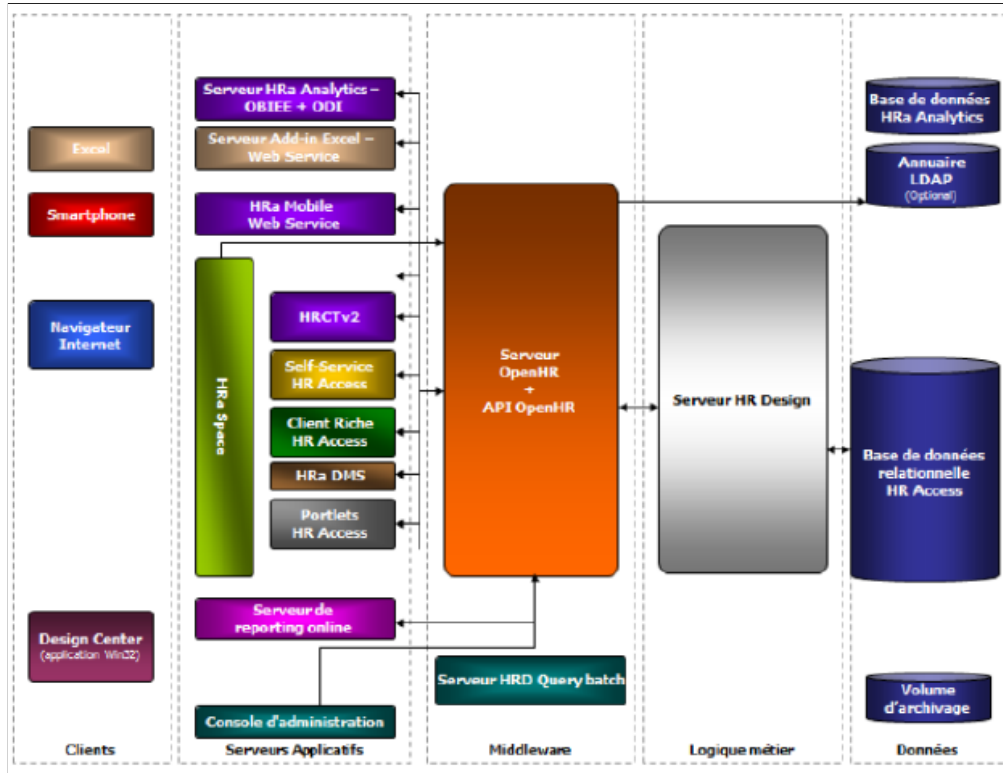


Figure 2.1: Schéma de l'architecture HR Access

L'architecture de HR Access est une architecture 5-tiers avec les différentes couches :

- **clients** : présentent les interfaces qui vont interagir avec l'utilisateur.
- **serveurs applicatifs** : contient l'ensemble des Web services et des API permettant la communication entre la couche présentation et la couche middleware.
- **middleware** : l'API OpenHR présente un élément essentiel de cette architecture sur laquelle sont bâties toutes les applications Java pour accéder aux services et aux données du serveur HR Access.
- **logique métier** : contenant l'ensemble des règles de gestion du logique métier exécuté dans un serveur HR Design.
- **données** : on trouve la base de données relationnelle de HR Access ainsi que l'annuaire LDAP (Lightweight Directory Access Protocol).

2.3 Description de la base de données HR ACCESS

L'architecture de la base de données HR ACCESS adopte une logique conceptuelle différente par rapport aux autres organisations classiques des bases de données. Elle est structurée en rubriques

qui sont définies pour l'entreprise et constitue le dictionnaire.

L'organisation des données est illustrée par la figure 2.2

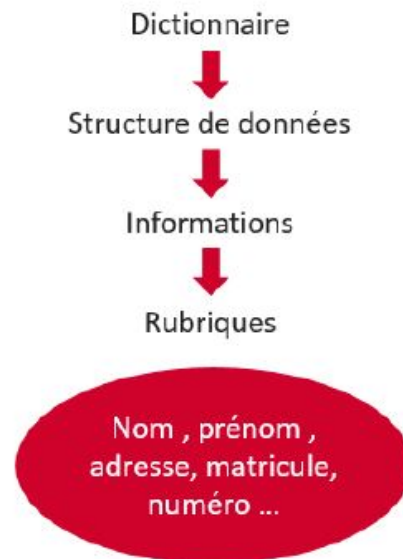


Figure 2.2: Organisation des données

Pour mieux comprendre ces nouvelles notions nous allons essayer de faire une sorte de correspondance entre certains concepts du dictionnaire avec les concepts de la base de données relationnelle.

- La notion de **dictionnaire** n'a pas de correspondance dans la base de données relationnelle.
- La notion de **structure de données** n'a pas de correspondance exacte dans la base de données relationnelle. Cependant, comme une structure de données regroupe un ensemble d'informations, elle peut être représentée par un ensemble de tables relationnelles présentant une affinité fonctionnelle.
- Le concept d'**information** correspond à la notion de table dans la base de données relationnelle : c'est d'ailleurs la définition de l'information (et de ses rubriques) qui sert à générer la DDL pour créer la table.
- La notion de **type de dossier** n'a pas de correspondance exacte dans la base de données relationnelle. Cependant, comme un type de dossier regroupe un sous-ensemble d'informations

d'une structure de données, il peut être représenté par un sous-ensemble des tables relationnelles représentant la structure de données.

- Le concept de **rubrique** correspond à la notion de colonne dans la base de données relationnelle : c'est d'ailleurs la définition de la rubrique qui sert à générer le fragment de DDL pour créer la colonne.

La figure 2.3 présente une correspondance entre la base de données HR ACCESS et la base de données relationnelle.

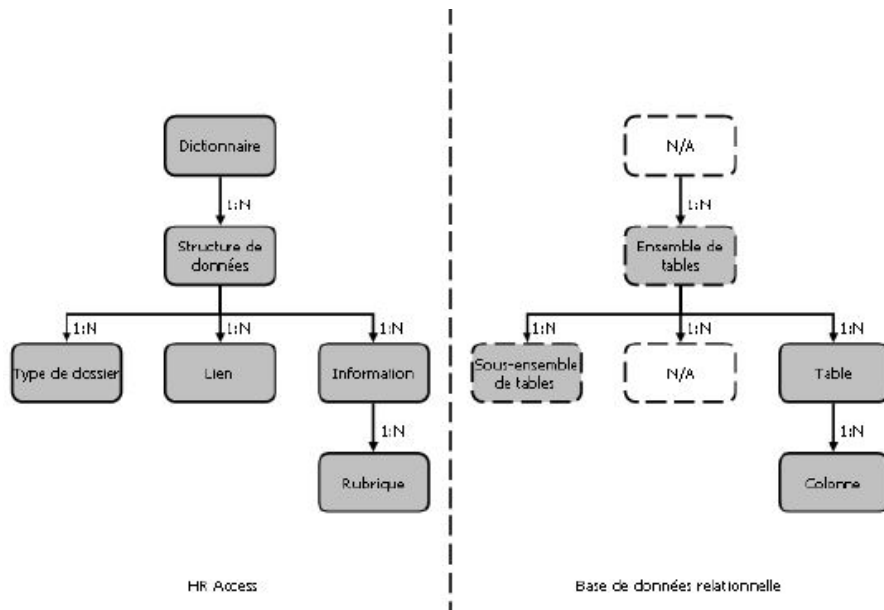


Figure 2.3: Correspondance entre l'organisation de la base données de HR ACCESS et l'organisation de la base données relationnelle

2.3.1 Les structures de données

C'est la racine qui rassemble un ensemble de tables.

Toutes les rubriques sont décrites dans le dictionnaire au sein de la structure de données. Ces rubriques sont soit utilisées par des dossiers ou contenues dans les dossiers.

La structure de données sert à décrire la structure d'un dossier, plus précisément l'ensemble des rubriques qu'il contient. Par convention, le nom de la structure de données est composé de 2 caractères alphanumériques. La définition de la structure de données est présentée dans la figure 2.4.

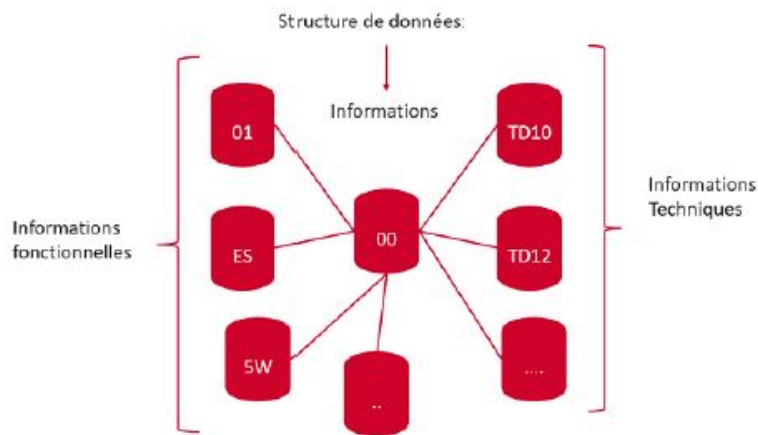


Figure 2.4: Définition de la structure de données

2.3.2 Les informations

Dans les structures de données les informations constituent les tables qui regroupent les rubriques.

Une information regroupe un ensemble de rubriques qui ont un lien logique commun. Le nom de l'information est composé de 2 caractère alphanumérique.

Les tables qui constituent la base de données HR ACCESS se répartissent dans les groupes suivants :

- **table d'informations techniques** : ce sont les tables qui contiennent les données techniques qui ont un lien avec l'exécution des applications et des programmes COBOL.
 - **table d'informations fonctionnelles** : Ces tables contiennent les rubriques réelles de données. Les lignes de ces tables sont alimentées par les données introduites par les utilisateurs.
- Par convention, la première colonne contient l'identifiant unique interne du dossier (NUDOSS).

2.3.3 Organisation dans la base de données

La notion de dossier est allégorique, elle correspond à une description logique. Les données concrètement sont stockées par information.

Une information réelle est contenu dans une table relationnelle. Toutes les occurrences de l'information venant de tous les dossiers sont contenues dans cette table. L'identifiant de la table relationnelle est crée d'une façon automatique, il est composé de :

- Nom de la structure de données à laquelle l'information appartient (2 caractères).
- Nom du segment d'accueil contient la valeur de segment d'accueil qui est souvent le nom de

l'information

L'organisation de la base de données HR ACCESS est présentée dans la figure 2.5.

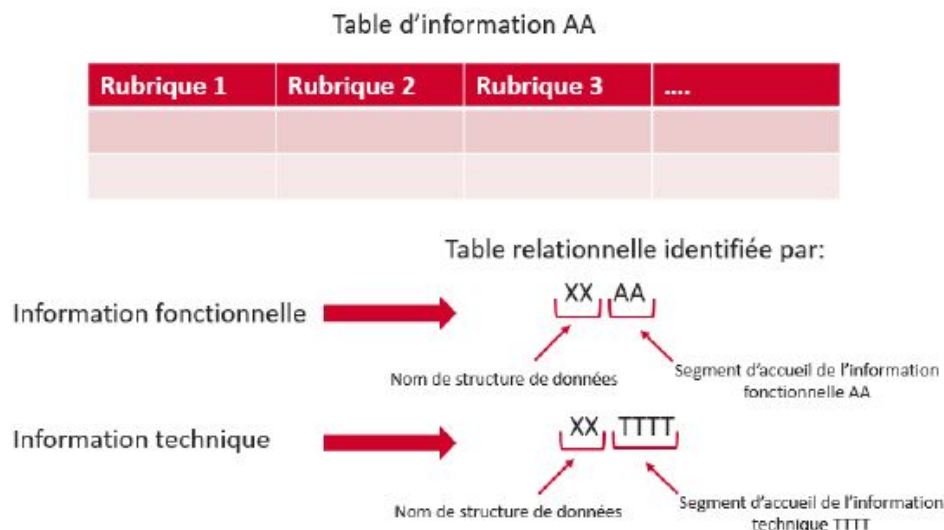


Figure 2.5: Organisations physiques de la base de données HR ACCESS

2.3.4 Modèle physique de données

Le modèle physique de données (MDP) permet de représenter graphiquement la structure de données qui va permettre d'identifier les différentes tables ainsi que les relations entre elles. Il permet de donner une vue globale sur la base de données existante de la ligne de produit HR ACCESS. Notre application est destinée pour tous les clients de Sopra HR qui disposent de produit HR ACCESS. Il est important de noter que chaque client a son propre système spécifique à lui, mais de point de vue base de données, ils ont la même structure pratiquement qui correspond à un modèle de données en étoile. C'est pour cela, nous allons utiliser la base de données de l'environnement de développement nommée DV11.

Le MDP de HR ACCESS contient environ 5000 tables. Pour notre application, nous avons utilisé que 18 tables détaillés dans le tableau 2.1

Tableau 2.1: Description des tables de l'application AAPM

Tables	Description des tables
ZD00	Cette table contient l'identification des dossiers réglementaires
ZD01	Cette table contient la description des dossiers réglementaires

ZD0F	Cette table contient l'adresse complète d'une société, d'un établissement, d'un organisme social.... Et ceci en fonction des caractéristiques propres à chaque pays
ZDG4	Cette table contient les durées théoriques des dossiers réglementaires
ZD0H	cette table contient les données concernant le ou les numéros de téléphone, E-mail, Fax..
ZDAT	Cette table contient les informations de réglementations
ZD39	Cette table contient les types de réglementations
ZDUU	Cette table contient les codes de réglementations
ZDB0	Cette table contient les anciens horaires de référence
ZDAQ	Cette table contient les code du journal pour l'interface comptable au niveau de l'établissement
ZDW3	Cette table contient les établissements comptables
ZD48	Cette table contient la grille spécifique pour établissement
ZDF2	Cette table contient les données médicales
ZDAG	Cette table contient renseignements pour édition TDS-DUCS
ZDBK	Cette table contient les renseignements pour DADS et horaire
ZD4J	c'est une table virtuelle qui contient les paramètres de groupe réglementaire
ZD0A	Cette table contient la législation en vigueur sur chaque société juridique
ZDOI	Cette table contient les renseignements bancaires

La figure 2.6 présente le modèle physique de la base de données existante de l'application métier

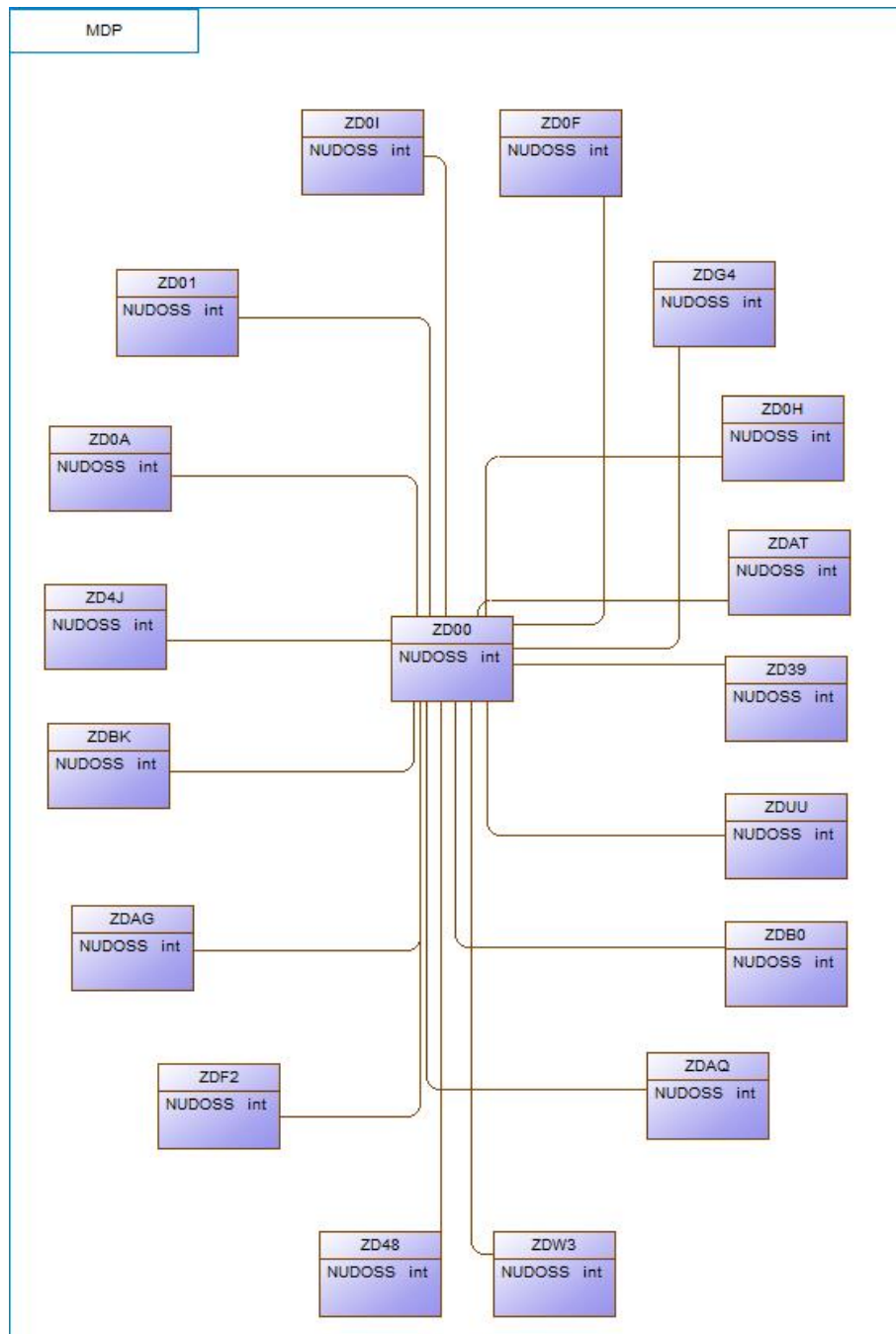


Figure 2.6: Modèle physique de données de l'application métier

Conclusion

Dans ce chapitre, nous avons présenté des nouvelles notions en générales qui sont en relation avec notre projet, d'où nous avons mis l'accent sur l'architecture de HR ACCESS ainsi qu'une partie de son modèle de données qu'on va utiliser. Dans le chapitre suivant, nous allons illustrer la première étape de notre projet à savoir l'analyse et la spécification des besoins.

ANALYSE ET SPÉCIFICATION DES BESOINS

Plan

Introduction	21
1 Les rôles Scrum	21
2 Identification des acteurs	21
3 Identification des besoins	22
4 Backlog de produit	23
5 Modélisation des besoins fonctionnels	25
6 Planification des sprints	33
Conclusion	33

Introduction

Ce chapitre a pour but d'analyser et de spécifier les besoins qui doivent être supportés par l'application demandée. Ceci constitue une étape primordiale pour déterminer une bonne compréhension des besoins afin de faciliter l'élaboration des fonctionnalités demandées. Pour ce faire, nous allons éliciter les besoins fonctionnels et non fonctionnels en passant par l'identification des acteurs. Ensuite, nous allons présenter l'équipe Scrum et le backlog de produit. Pour conclure, nous allons entamer la modélisation des besoins de ce projet suivie de la planification de la release.

3.1 Les rôles Scrum

Scrum est la méthodologie suivie par la société Sopra HR Software pour la gestion de ses projets. Les équipes Scrum sont auto-organisées et pluridisciplinaires. La mission de l'équipe consiste à délivrer un incrément du produit à la fin de chaque sprint. Nous allons tout d'abord présenter notre équipe scrum :

- **Product Owner (PO) :** Mourad ANEMICHE.

Son rôle est de spécifier les caractéristiques fonctionnelles du produit à développer, ainsi que la définition des tâches qui permettront de répondre aux besoins du projet.

- **Scrum master (SM) :** Donia SOULI.

Le Scrum Master assure globalement le bon déroulement des programmes et protège l'équipe de tout problème extérieur. Il assure également l'organisation des réunions et la bonne application de la méthode AGILE de par ce biais.

- **Équipe de développement :** Alaeddine JELASSI.

Elle ne comporte pas d'éléments particuliers si ce n'est son autogestion. La communication et le partage entre les différents développeurs sont fréquents et l'équipe s'adresse directement au Product Owner pour la compréhension des objectifs.

3.2 Identification des acteurs

Un acteur est une entité externe au système. On parle d'une personne ou d'un autre système d'information qui va se servir des services offerts par notre application. Dans le cas de notre application, nous avons identifié :

- **Consultant SIRH :** c'est un collaborateur de l'entreprise qui fait partie de l'équipe TMA

disposant d'un compte dans l'application HR ACCESS avec le rôle « collaborateur ». Il va valider les demandes clients et lancer des traitements automatiques.

- **Client** : c'est un employé de l'entreprise cliente disposant du produit HR ACCESS. Il va passer ses demandes à l'équipe TMA via son espace client.

3.3 Identification des besoins

L'objectif de la phase de spécification des besoins consiste à définir avec détail l'ensemble des fonctionnalités offertes par le système. Pour se faire, nous allons définir les besoins fonctionnels et non fonctionnels. Par la suite, nous allons préparer le backlog de produit comme le dicte le cadre SCRUM. Les besoins fonctionnels vont être modélisés par la suite.

3.3.1 Besoins fonctionnels

Les besoins fonctionnels sont les fonctionnalités qui doivent être présentes dans le système. Les besoins fonctionnels du Consultant SIRH sont :

- Générer les codes de paramétrages à partir du dossier client.
- Consulter l'état d'avancement de création du dossier.
- Modifier le paramétrage généré automatiquement dans la fiche technique.
- Injecter automatiquement les données et les codes générés dans la base HR Access.
- Gérer les comptes clients.
- Gérer les demandes des clients.
- Modification des constantes de veille légale pour tous les clients à la fois.

Les besoins fonctionnels du client sont :

- Introduire les données de création (société/établissement).
- Sauvegarder les versions du dossier lors de la création.
- Envoyer la version finale du dossier de la demande au Consultant SIRH.

3.3.2 Besoins non fonctionnels

Par conjonction aux besoins fonctionnels, notre système doit répondre aux besoins non fonctionnels pour améliorer la qualité de ses services tels que :

- **Ergonomie** : Étant donné que notre application va être utilisée par un client non informaticien donc les interfaces doivent être réalisées d'une façon simple pour que l'utilisateur s'y trouve facilement.
- **Extensibilité** : le système doit être faiblement couplé et conforme à une architecture qui permet tout renouvellement ou ajout des nouveaux services.
- **Sécurité** : : la confidentialité des données utilisateurs doit être garantie. Chaque utilisateur selon son rôle ne doit visualiser que l'espace qui lui est réservé.

3.4 Backlog de produit

Afin de garantir la conformité du produit final avec les exigences du client, les fonctionnalités sont collectées dans le BACKLOG de produit. Elles sont raffinées progressivement durant les sprints. Le BACKLOG de produit est évolutif et peut changer tout au long du projet. Les fonctionnalités sont regroupées en des « user stories ». Le Backlog de produit comprend les champs suivants :

- **ID** : c'est un identifiant unique pour chaque histoire utilisateurs.
- **User story** : c'est une description brève de la fonctionnalité désirée par le client.
- **Story point** : c'est une estimation de l'effort nécessaire à la réalisation d'une histoire.

Dans le tableau 3.1, nous présentons la liste des histoires utilisateurs ainsi que leurs estimations.

Tableau 3.1: Backlog du produit

ID	User Story	Priorité	Story Point
AM1	En tant que client ou consultant, je veux m'authentifier	1	2
AM2	En tant que client, je veux créer un dossier de création de société	1	4
AM3	En tant que Client, je veux préciser l'identification juridique de la société	3	2
AM4	En tant que Client, je veux préciser les liens avec les OPS	3	2
AM5	En tant que Client, je veux créer les dossiers établissements	1	4

AM6	En tant que Client, je veux créer les unités organisationnelles	2	2
AM7	En tant que Client, je veux créer les postes	2	2
AM8	En tant que Client, je veux créer les emplois	2	2
AM9	En tant que Client, je veux indiquer les nouvelles règles de gestion (CP, les nouvelles règles en lien avec les Absences SS, les nouveaux éléments de rémunération à paramétrer, etc)	3	2
AM10	En tant que client, je veux modifier mon dossier	3	3
AM11	En tant que client, je veux enregistrer les versions de mon dossier	3	1
AM12	En tant que client, je veux annuler mon dossier	3	1
AM13	En tant que client, je veux valider mon dossier ce qui génère un ticket EasyVista (CRM)	3	3
AM14	En tant que consultant, je gérer mes demandes reçues	2	3
AM15	En tant que consultant, je veux valider le dossier afin de générer les codes et le paramétrage nécessaire	1	10
AM16	En tant que consultant, je veux vérifier la cohérence de ce paramétrage par rapport au celui présent au SIRH	1	5

AM17	En tant que consultant, je veux modifier le paramétrage afin de bien répondre au besoin du client	1	3
AM18	En tant que consultant, je veux injecter les données de la fiche de paramétrage dans la base HR	1	13

3.5 Modélisation des besoins fonctionnels

Pour donner une vue sur l'ensemble des fonctionnalités offertes par notre application, nous allons les modéliser dans un diagramme de cas d'utilisation global, ensuite nous allons détailler les cas d'utilisation les plus prioritaires.

3.5.1 Diagramme de cas d'utilisation général

La figure 3.1 illustre le diagramme de cas d'utilisation général.

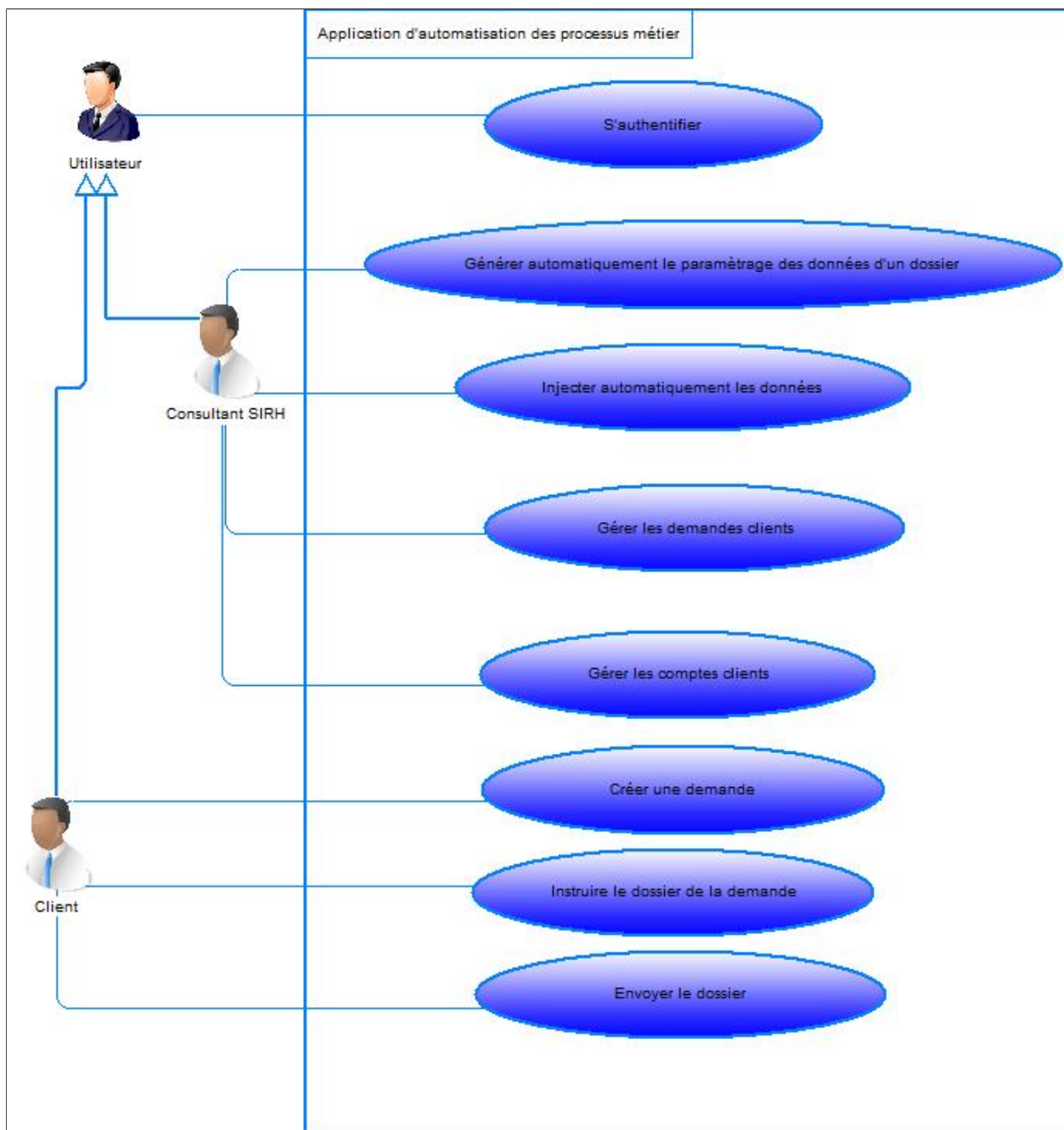


Figure 3.1: Diagramme de cas d'utilisation général de l'application

3.5.2 Raffinement et description textuelle des cas d'utilisation prioritaires

Nous avons choisi quelques cas d'utilisation en fonction de leurs priorités afin de les détailler par des raffinements en précisant les descriptions textuelles. Ces cas d'utilisation sont les suivants :

- «générer automatiquement le paramétrage des données d'un dossier» : En effet la notion de dossier est une notion orientée métier, quand nous disons par exemple création d'un établissement, c'est à dire nous introduisons juste les informations d'identification de l'établissement. Par

contre, pour la création d'un dossier établissement, nous introduisons toutes les informations relatives à tous les éléments qui composent un établissement (unités organisationnelles, poste, etc). En ce qui concerne la génération de paramétrage automatique, en fait chaque information que le client saisi va être stockée dans une base local indépendante du SIRH HR ACCESS. Par la suite, pour que ces informations seront injectées dans la base HR ACCESS, ça nécessite un certain paramétrage de mapping spécifique à chaque information et la où le moteur de règle intervient pour nous générer d'une façon automatique ce paramétrage.

- «Gérer les demande des clients» : ça permet au consultant de visualiser les demandes envoyées par le client ainsi que de gérer l'état de ces demandes.
- «Instruire le dossier de la demande» : ça permet le client de créer le dossier de la demande, ainsi que importer ou exporter son dossier sous forme fichier Excel.

3.5.2.1 Cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier»

La figure 3.2 illustre le raffinement du cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier»

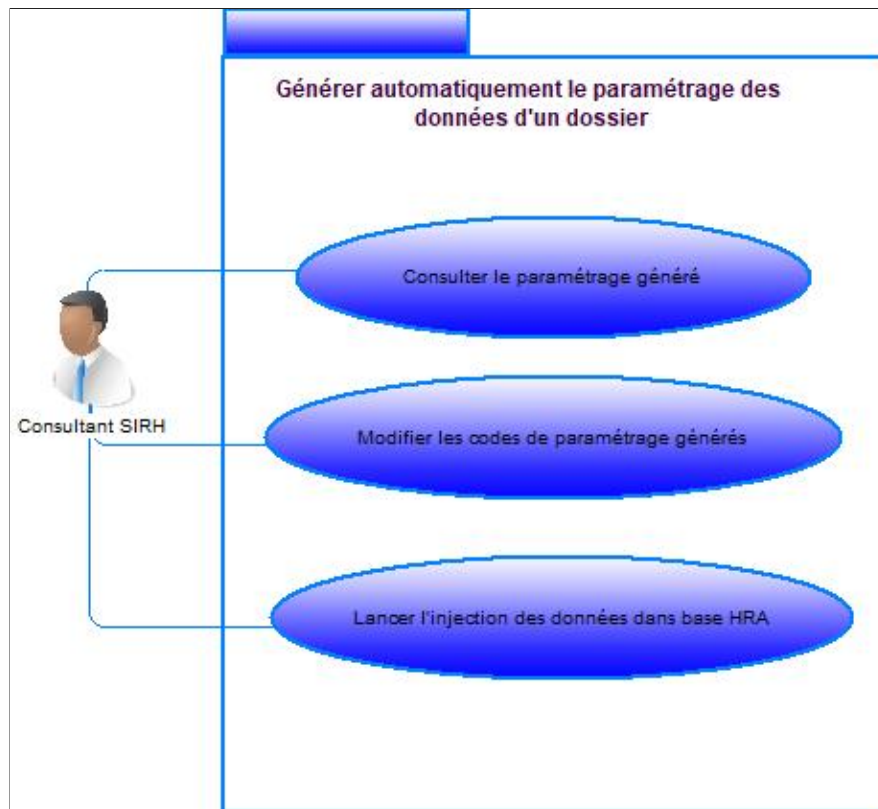


Figure 3.2: Raffinement du cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier»

Le tableau 3.2 illustre la description textuelle du cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier»

Tableau 3.2: Description textuelle de cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier»

Cas d'utilisation	Générer automatiquement le paramétrage des données d'un dossier
Résumé	Générer automatiquement le paramétrage adéquat à la demande du client
Acteur	Consultant
Pré-condition	<ol style="list-style-type: none"> 1. Le consultant doit être authentifié à l'application et au SIRH. 2. Le client doit avoir un dossier à traiter dans la boîte des demandes reçues. 3. Les informations introduites dans le dossier doivent être correctes et cohérentes.
Scénario Nominal	<ol style="list-style-type: none"> 1. Le consultant sélectionne l'onglet des demandes reçus. 2. Le système affiche la liste de toutes les demandes des clients ainsi que leurs états. 3. Le consultant choisit une demande non traitée. 4. Le système va charger les informations relatives à cette demande dans le moteur de règles. 5. Le moteur de règle va générer le paramétrage automatiquement. 6. Le système va afficher la liste des paramétrages générés. 7. Le consultant vérifie le paramétrage généré. 8. Le consultant modifie si besoin certain paramétrage. 9. Le consultant valide le paramétrage généré. 10. Le système va injecter selon ce paramétrage les informations dans la base de données HR ACCESS.

Scénario d'exception	<p>3.1 si le client choisit une demande traitée, le système affiche que la demande est déjà traitée et validée.</p> <p>6.1 si le système détecte une erreur lors de la génération du paramétrage, il va afficher un message d'erreur qui explique la nature de l'anomalie</p> <p>10.1 si le système détecte une incohérence de données lors de l'injection des informations, il va afficher un message indiquant les points d'incohérences</p>
Post condition	L'opération choisie a été effectuée avec succès.

3.5.2.2 Cas d'utilisation «Gérer les demandes des clients»

La figure 3.3 illustre le raffinement du cas d'utilisation «Gérer les demandes des clients»

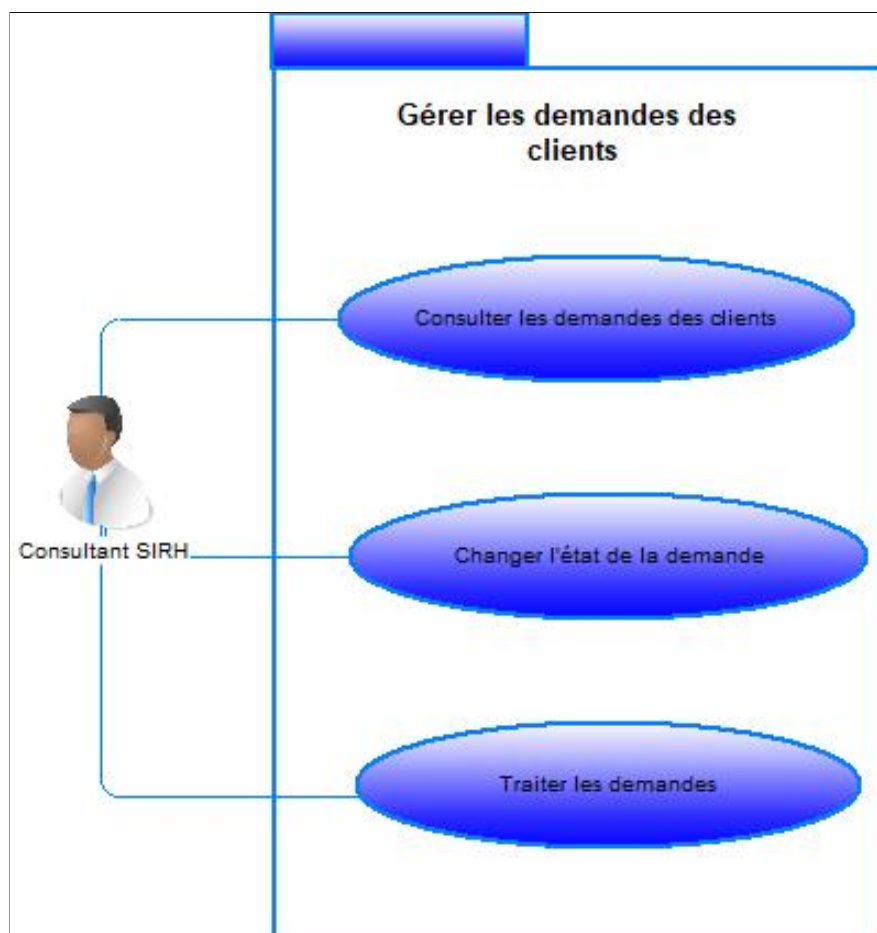


Figure 3.3: Raffinement du cas d'utilisation «Gérer les demandes des clients»

Le tableau 3.3 illustre la description textuelle du cas d'utilisation «Gérer les demandes des clients»

Tableau 3.3: Description textuelle de cas d'utilisation «Gérer les demandes des clients»

Cas d'utilisation	Gérer les demandes des clients
Résumé	Gérer les demandes des clients en consultation ou en modification
Acteur	Consultant
pré-condition	1. Le consultant doit être authentifié à l'application 2. Le consultant doit avoir des demandes dans sa boîte de réception de demandes.
Scénario Nominal	1. le consultant doit accéder à sa boîte de réception. 2. Le système affiche la liste des demandes reçues en temps réel. 3. Le consultant peut changer l'état d'une ou de plusieurs demandes de la liste. 4. Le collaborateur valide le changement d'état de la demande. 5. Le système va changer l'état des demandes dans la base de données. 6. Le client va être informé par le changement de l'état de sa demande. 7. Le collaborateur choisit une demande pour la traiter.
Scénario d'exception	3.1 Si le consultant fait un changement d'état de traitée à non traitée alors le système lui affiche une alerte pour l'empêcher. 7.1 Si le consultant choisit une demande alors le système affiche que la demande est déjà traitée.
Post condition	L'opération choisie a été effectuée avec succès

3.5.2.3 Cas d'utilisation «Instruire le dossier de la demande client»

La figure 3.4 illustre le raffinement du cas d'utilisation «Instruire le dossier de la demande client»

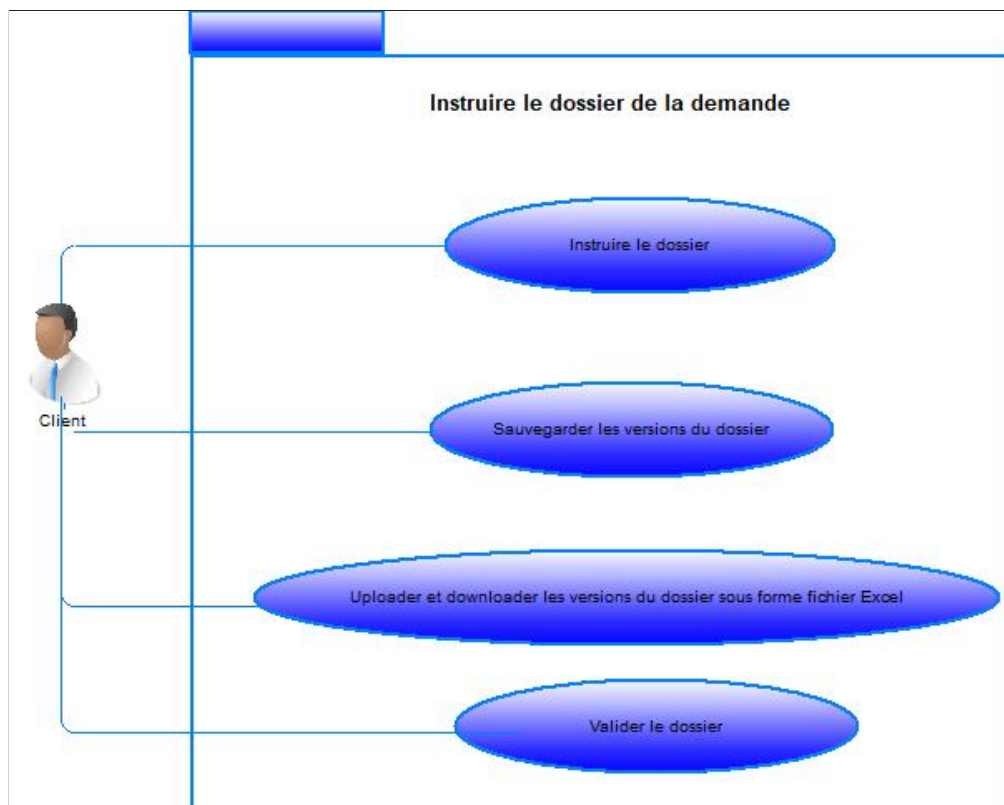


Figure 3.4: Raffinement du cas d'utilisation «Instruire le dossier de la demande»

Le tableau 3.4 illustre la description textuelle du cas d'utilisation «Instruire le dossier de la demande»

Tableau 3.4: Description textuelle de cas d'utilisation «Instruire le dossier de la demande»

Cas d'utilisation	Instruire le dossier de la demande client
Résumé	Instruire le dossier de la demande d'un service métier avec les informations nécessaires à la bonne prise en compte dans le SIRH
Acteur	Client
pré-condition	1. Le client doit être authentifié à l'application

Scénario Nominal	<ol style="list-style-type: none">1. Le client choisit le type de demande qu'il souhaite créer2. Le client remplit les formulaires avec ses informations spécifiques à la demande.3. Le client sauvegarde la version de sa demande4. Le client revient aux versions précédentes pour compléter l'instruction de son dossier.5. Le client upload et download les versions de son dossier via le fichier excel.7. Le client valide sa demande.
Post condition	L'opération choisie a été effectuée avec succès

3.6 Planification des sprints

Une fois que le Backlog du produit est suffisamment complet et ordonnancé, nous avons établi, lors de la réunion de planification des sprints, les durées prévisionnelles du travail à effectuer durant chaque sprint. Après avoir évalué le travail estimé et dû au fait que notre équipe est considérée comme une petite équipe, les sprints que nous avons définis durent trois semaines.

Le tableau 3.5 présente la planification que nous avons suivi.

Tableau 3.5: Planification des sprints

ID Sprint	Nom du Sprint	ID User Story	Durée
Sprint 1	Authentification et création de société	AM1, AM2, AM3, AM4	Du 02-04 au 20-04
Sprint 2	Création dossier établissement	AM5, AM6, AM7, AM8, AM9	Du 23-04 au 09-05
Sprint 3	Génération de paramétrage et injection dans la base HR ACCESS	AM14, AM15, AM16, AM17, AM18	Du 10-05 au 31-05
Sprint 4	Gestion des dossiers	AM10, AM11, AM12, AM13	Du 01-06 au 22-06

Conclusion

Ce chapitre nous a permis de bien délimiter le projet et d'avoir une vision plus claire du sujet. Nous avons décrit les besoins fonctionnels, non fonctionnels, les acteurs et le Backlog produit. Par la suite, il nous a permis de planifier et organiser le temps consacré à la réalisation du projet en identifiant les sprints. Puis, nous avons décrit les cas d'utilisation qui sont nécessaires ainsi leurs descriptions textuelles. Dans le chapitre suivant, nous allons entamer la phase d'initialisation du projet.

INITIALISATION DU PROJET

Plan

Introduction	36
1 Initialisation	36
2 Architecture de la solution	36
3 Conception détaillée	41
4 Environnement technique	43
5 Environnement de travail	46

Introduction

Nous consacrons ce chapitre à la préparation des conditions favorables avant le déclenchement des sprints. Pour ce faire, nous allons débiter par l'initialisation du projet où la conception générale et détaillée va être explicitée. Par la suite, nous allons présenter les architectures adoptées et justifier nos choix technologiques.

4.1 Initialisation

Dans ce projet, vu que nous sommes face à des outils spécifiques à Sopra HR Software, nous avons consacré la première phase du projet pour la documentation. Ensuite, nous avons assisté à des formations de familiarisation avec les nouvelles notions suivantes :

- métier RH,
- le produit HR Access,
- base de données de HR Access,
- outil de développement spécifique à la société : Design Center.

Par la suite, nous avons effectué une auto-formation grâce aux guides des développeurs fournis par nos encadrants pour pouvoir manipuler l'API Java spécifique à OpenHR.

4.2 Architecture de la solution

Dans cette partie, nous allons présenter l'architecture logique et logicielle de notre application en tenant compte l'intégration avec l'architecture existante du produit HR ACCESS.

4.2.1 Architecture logique

Dans cette partie, nous définissons en détails l'architecture physique qu'on a adoptée pour la réalisation de notre application en tenant compte des spécifications précédentes.

L'architecture physique de notre application est basée sur la manipulation de l'API Open HR dont le but est de consommer des services qui nous permettent d'exploiter les données existantes dans la base de données de HR Access.

La figure 4.1 détaille la répartition des matériels qui constitue le système. De plus, Elle expose les composants qu'on a utilisés pour le déploiement de l'application AAPM selon une architecture 4-tiers avec les différentes couches :

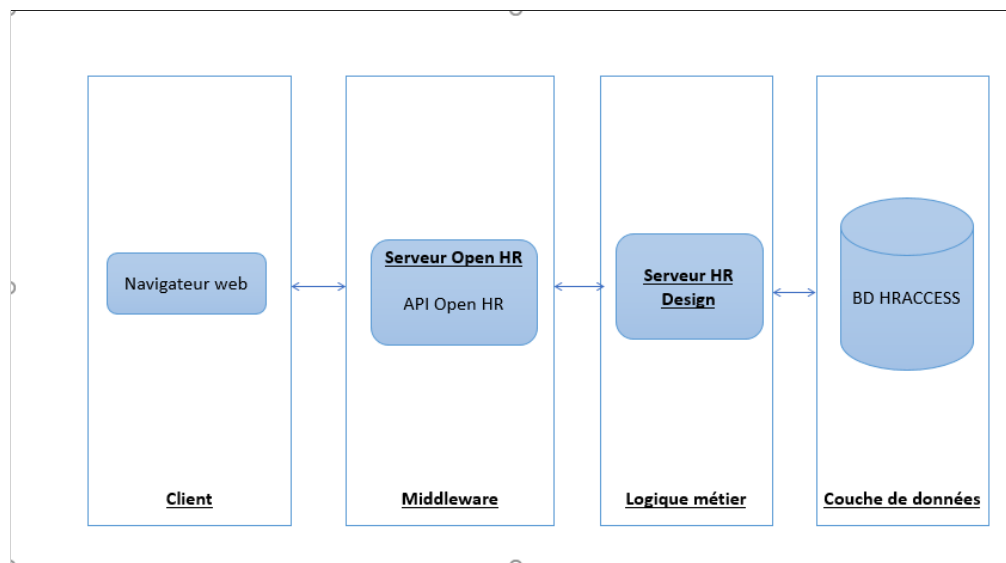


Figure 4.1: Architecture physique de HR Access

Dans cette approche, les couches communiquent entre elles à travers un modèle d'échange dont chacune offre un ensemble de services rendus. Les services d'une couche sont mis en profit de la couche supérieure. Le déploiement de chaque couche est effectué sur un serveur différent. Les ressources matérielles sont représentées sous forme de noeuds qui sont connectées entre eux à l'aide d'un support de communication.

La figure 4.2 représente l'architecture physique de notre application après l'intégration avec l'architecture existante de HR ACCESS Via l'API OpenHR de la couche middleware.

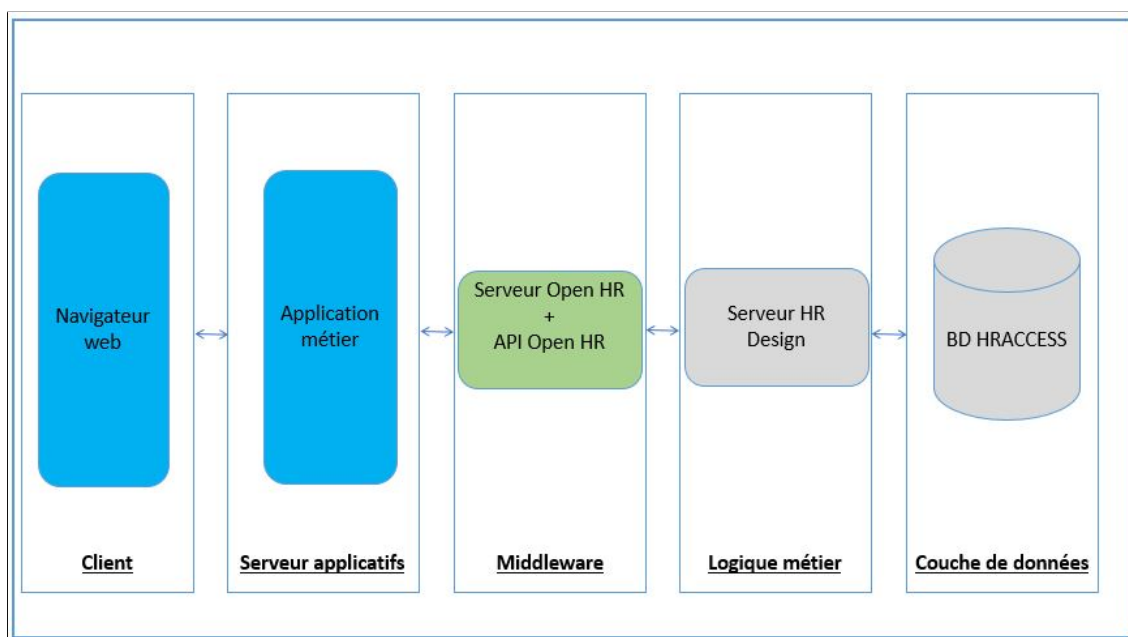


Figure 4.2: Architecture physique de notre solution

Nous avons manipulé les composants colorés en vert et bleu. En effet, nous avons utilisé les services offerts par l'API OpenHR afin d'accéder en lecture ou écriture en temps réel aux données gérées par le serveur HR ACCESS tout en réutilisant la logique fonctionnelle existante de contrôle des données (implémentée sous la forme de programmes COBOL). En ce qui concerne les couches en bleu, ce sont celles, avec un niveau d'abstraction élevé de notre application web. Le reste des composants en gris forment les parties existantes chez SOPRA HR qu'on n'a pas manipulées directement. La mise en place des composants suivants est nécessaire pour l'élaboration de notre application à savoir :

- **Serveur OpenHR** : permet l'accès en lecture/écriture et en temps réel aux données gérées par le serveur HR Access tout en réutilisant la logique fonctionnelle existante de contrôle des données.
- **Serveur HR Design** : représente la partie fonctionnelle de L'application, celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.
- **Serveur de base de données** : permet la consultation des données de la base HR Access et leurs mise à jour tout en s'assurant de la confidentialité.

4.2.2 Architecture logicielle

l'application métier va être hébergée dans un serveur TOMCAT qui communique avec les serveurs Open HR, HR Design et la base de données HR Access. L'architecture physique est représentée par la figure 4.3.

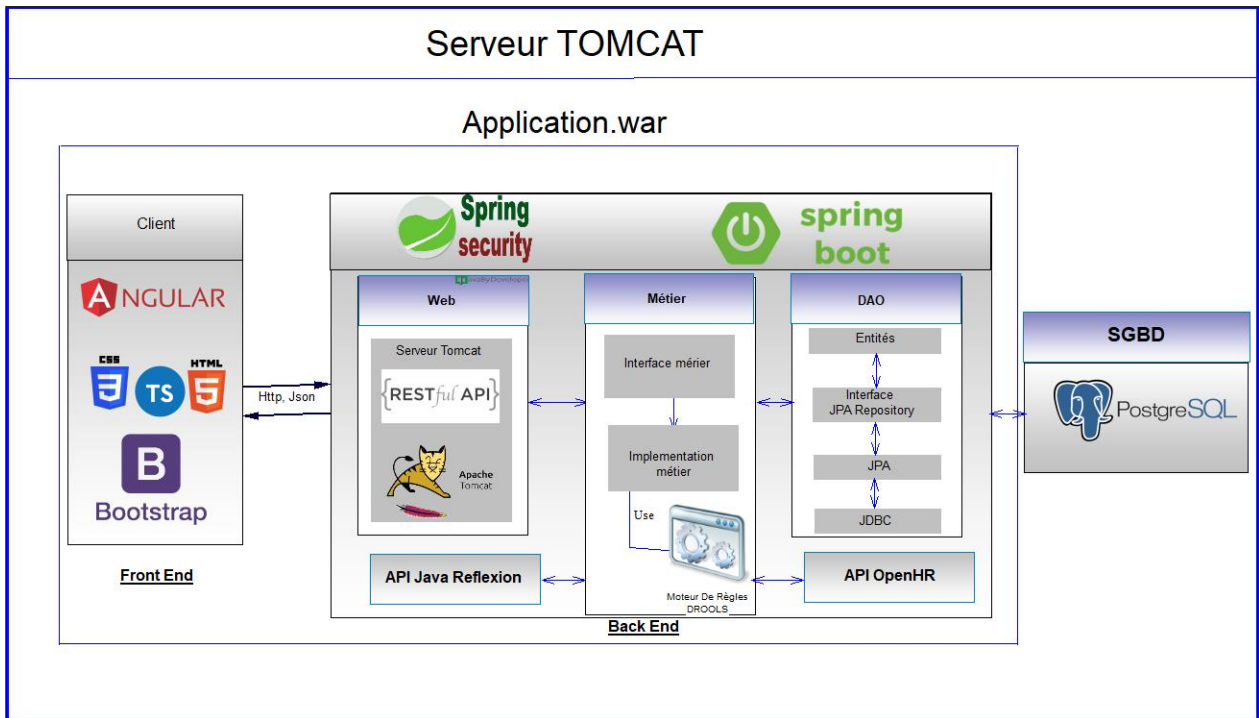


Figure 4.3: Architecture logicielle de l'AAPM

Le back end de l'AAPM est basé sur Spring Boot qui consiste à la séparation des différentes couches d'une application web, à savoir : données, traitements et présentation. Ainsi, cette partie se retrouve segmentée en trois composants essentiels :

- **Couche DAO :** permettant la persistance des données par JPA et Hibernate dont le but est de faire la correspondance entre les objets Java et les tables correspondantes dans la base de données.

- **Couche Métier :**

C'est dans cette couche qu'on doit retrouver toutes les traitements métiers de l'application. Nous avons implémenté dans cette partie, notre moteur de règle Drools que nous l'avons alimenté par les règles de gestion adéquat pour générer des actions métier automatiquement dans le SIRH à travers l'API Open HR. Ces traitements seront implémentés sous forme de services Spring.

Spring est un conteneur léger permettant de simplifier l'intégration des différentes couches via l'utilisation des interfaces JAVA, du design pattern Ioc de la programmation par aspect (AOP). En plus, Spring est totalement portable sur tous les serveurs d'application et totalement intégrable avec toutes les technologies choisies pour le développement de ce projet.

- **Web :** basé sur des Web Services REST sur le protocole HTTPS en utilisant le format

d'échange de données JSON.

- **Drools** : Drools est une plateforme d'intégration de logique métier écrite en java et se repose sur l'algorithme de Rete qui fournit une exécution plus efficace d'un système expert, pour l'amélioration des performances.

Il s'agit d'un moteur de règles qui permet de définir des règles et de les appliquer à des faits(données). Il va faire un raisonnement. Pour cela, il est composé d'un moteur d'inférence qui va chercher et déterminer les règles applicables aux faits (pattern matching). Cet ensemble de règles applicables constitue ce qu'on appelle l'agenda. Il va ensuite exécuter ces règles, on parle alors de déclenchement de règles. En effet la performance de ce moteur de règles n'est remarquable que lorsque nous disposons d'une base de règles assez volumineuse. C'est pour cette raison que nous avons opté à utiliser DROOLS, car chaque client de Sopra HR dispose de plus que 500 règles de gestion dans les deux processus métier que nous travaillons dessus. De plus, nous parlons de plus que 100 clients, donc nous sommes faces à une explosion de règles et de conditions que nous ne pouvons la gérer qu'à travers un tel système expert puissant. [6].

La figure 4.4 montre l'architecture simplifiée de Drools.

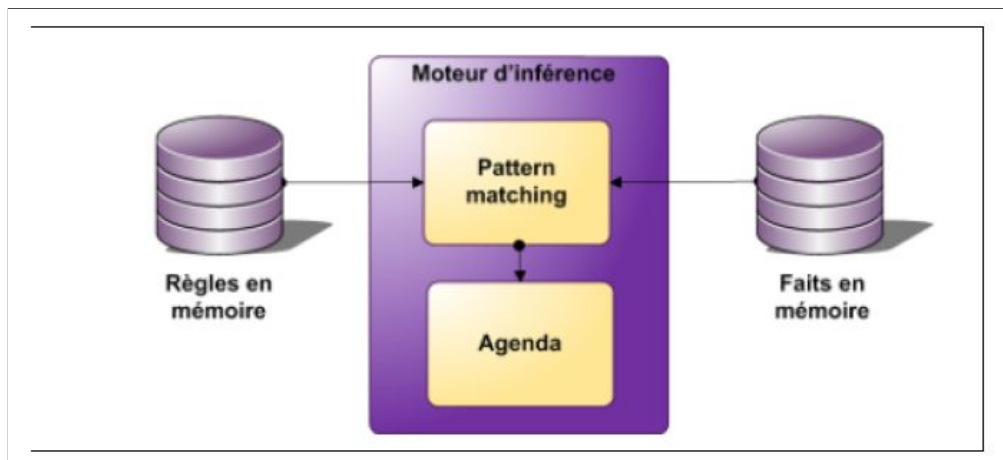


Figure 4.4: Architecture de Drools

Dans notre contexte chaque règle métier, concrètement représente les paramètres de mappage nécessaires d'une information introduite par le client pour qu'elle soit injecter correctement le SIRH. Pour mieux s'expliquer nous allons illustrer un exemple concret de règle métier de notre application exprimé en langage DRL dans la figure 4.5.

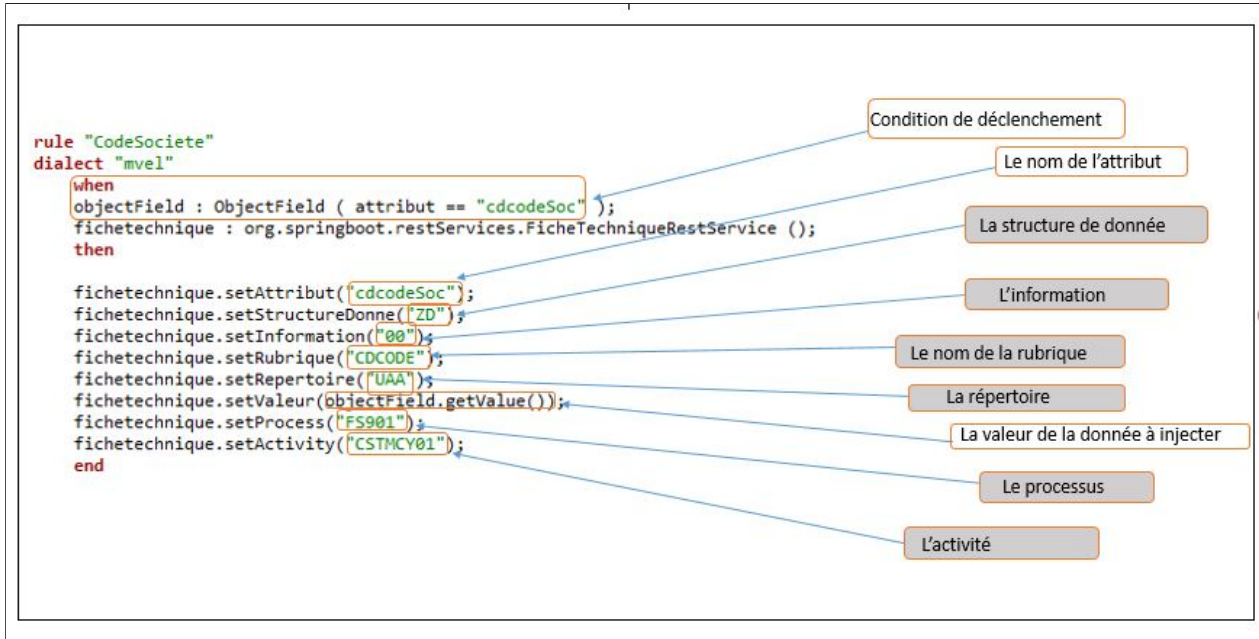


Figure 4.5: Exemple concret de règle métier

Dans cet exemple nous avons pris une règle qui va être déclenchée quand le client introduit le code de la société. Une fois déclenché, le moteur de règle va générer dans la fiche technique la valeur de l'attribut et il va indiquer que ce dernier doit être stocké dans la base de données de HR ACCESS sous la structure de donnée "ZD", qui appartient à l'information "00" et qui représente la rubrique "CDCODE". Ainsi cette information suit le processus « création d'une société », dont il est identifié la base HR ACCESS par le code "FS901". Ce processus fait partie de l'activité nommée "CTMCY01" dans le SIRH.

4.3 Conception détaillée

Dans cette partie nous allons présenter le diagramme de déploiement, ensuite, nous décrirons l'architecture logique de notre application à travers le diagramme de composants.

4.3.1 Le diagramme de déploiement

Les diagrammes de déploiement sont utilisés pour mettre en évidence les utilisateurs de l'application avec les différents serveurs. Ils montrent aussi la distribution des composants logiciels sur la base d'unité d'exécution. Notre application sera hébergée dans un nouveau serveur TOMCAT pour des raisons de sécurité. Les composants en gris constituent les serveurs et la base de données HR ACCESS, nous ne les avons pas manipulés directement, mais la communication avec eux est

assurée par notre serveur d'application.

la figure 4.6 présente le diagramme de déploiement de notre application.

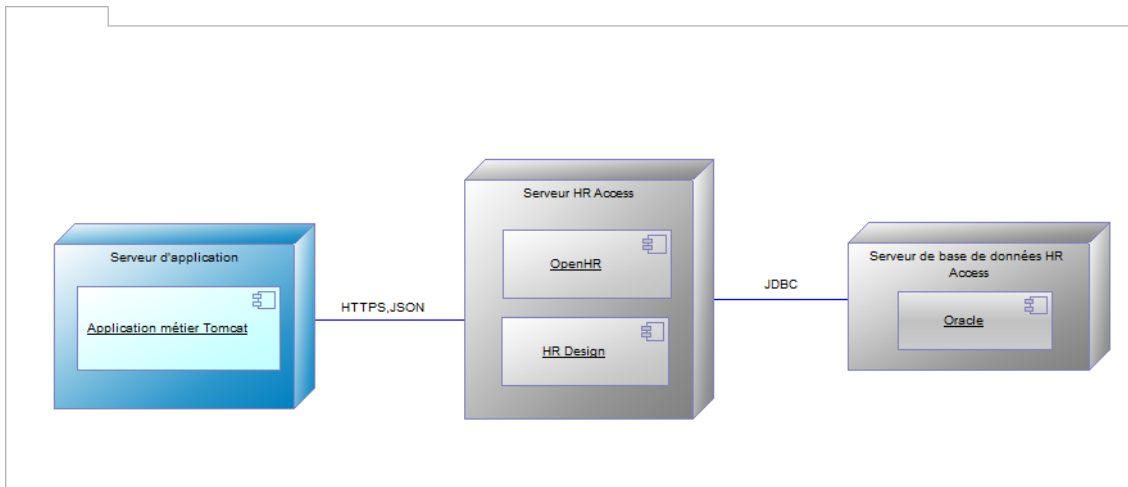


Figure 4.6: Diagramme de déploiement de l'application

On a utilisé dans notre projet les connexions suivantes :

- **HTTPS** : (HyperText Transfer Protocol Secure) c'est un protocole qui permet de récupérer les données dont le contenu est formaté en JSON par le serveur.
- **JDBC** : (Java Database Connectivity) est une interface de programmation qui permet aux applications de se connecter à des serveurs de bases de données indépendamment du SGBD [7] utilisé.

4.3.2 Diagramme de composants

Un diagramme de composants permet de décrire l'architecture physique et statique d'une application. Il permet de mettre en évidence les composants, les interfaces et définir les relations entre eux. En tant que développeurs, nous nous servons de diagramme de composants car il nous offre une vue architecturale de niveau supérieur des systèmes [8].

La figure 4.7 présente le diagramme de composant de notre application.

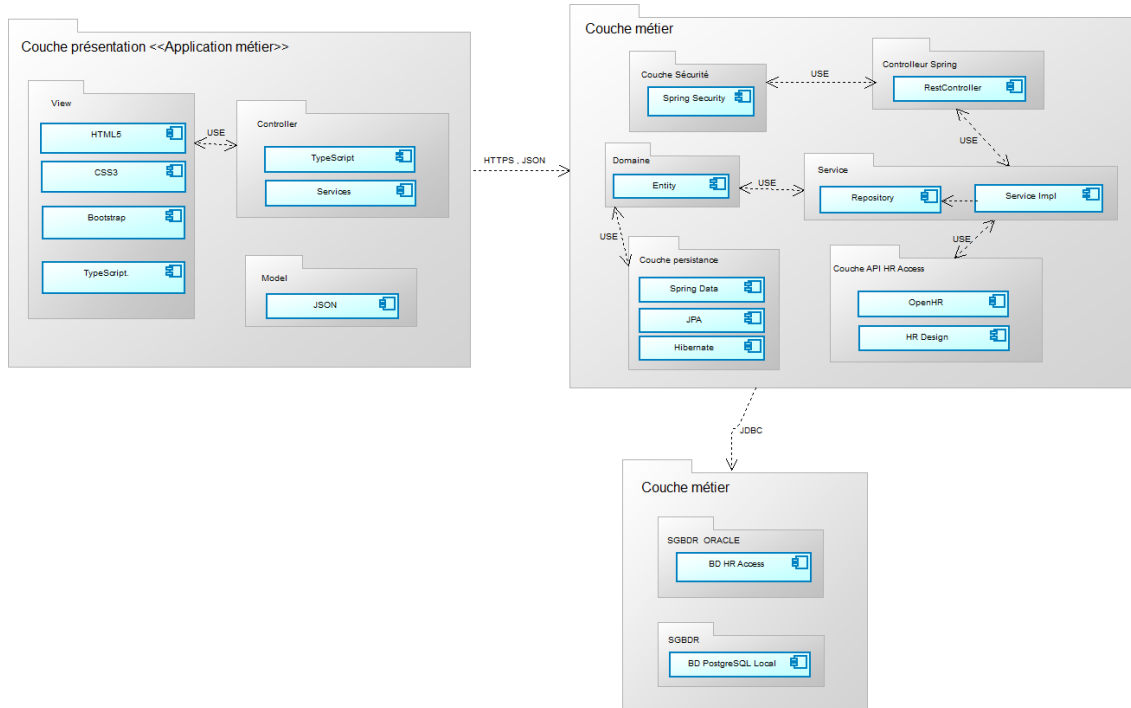


Figure 4.7: Diagramme de composants de l'application

4.4 Environnement technique

Dans cette partie, nous présenterons l'ensemble des équipements mis à disposition afin de réaliser ce travail ainsi que les techniques relatifs à la réalisation de cette partie.

4.4.1 Environnement matériel

Comme station de développement nous avons utilisé un ordinateur portable HP EliteBook embarquant un système d'exploitation Windows 10 ayant les caractéristiques illustré dans le tableau 4.1.

Tableau 4.1: Caractéristiques de l'ordinateur

Caractéristique	Type
Processeur	Intel Core i5 vPro 7th Gen
Mémoire vive (RAM)	8GB
Système	64bits

4.4.2 Environnement logiciel

Pour réaliser ce projet nous avons utilisé les outils et les environnements logiciels suivants :

- **Open HR** : Le serveur openHR démarre et arrête les programmes COBOL du serveur HR ACCESS [9].
- **HR Design** : le serveur HR Design héberge les programmes COBOL. C'est le serveur unique qui communique avec la base de données de HR ACCESS.
- **Design Center** : c'est un IDE de conception et de développement HR ACCESS. il est développé principalement en C++.Il permet d'implémenter les workflows, les traitements, les processus. Nous l'avons utilisé pour comprendre les champs et les tables de HR ACCESS [10].
- **Eclipse Oxygen** : Eclipse est un Environnement de Développement Intégré (EDI) utilisé pour déployer des applications sur le serveur web Tomcat (version 7.0.6) [11].
- **JDK (Java Development Kit)** : pour exécuter les applications Java [12].
- **Putty** : nous avons utilisé Putty pour tester les requêtes SQL qu'on lance vers le serveur de base de données HR ACCESS, ainsi pour visualiser le log du serveur Open HR [13].
- **Tomcat** : c'est l'un des serveurs d'application de Apache Software Fondation. En effet c'est une application multi plate-forme.
- **Oracle** : HR ACCESS repose sur oracle comme serveur de base de données. En effet, Oracle Database est un SGBDR (Système de Gestion de Bases de Données Relationnelle) [14].
- **Maven** : un outil pour la gestion et l'automatisation de production des projets Java et Java EE, créé par Apache Software Foundation [15].
- **Outil de gestion de version SVN** : SVN Subversion est un logiciel de gestion de sources et de contrôle de versions [16].
- **PostgreSQL** : est un système de gestion de base de données relationnelle et objet (SGBDRO) Open Source concurrent aux projets MySQL ou Firebird, eux aussi Open Source, ou Oracle, Sybase ou Microsoft SQL Server, des solutions propriétaires. [17].
- **PowerAMC** : c'est un outil de conception qui nous aide à modéliser les diagrammes UML [18].

4.4.3 Technologies et langages utilisés

Pour la réalisation de cette application nous avons utilisé les technologies et les langages suivants :

- **JavaEE** : est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles [19].
- **Spring** : est un Framework libre pour construire et définir l'infrastructure d'une application java2, dont il facilite le développement et les tests . C'est un conteneur dit "léger" en s'appuyant sur une infrastructure similaire à un serveur d'applications J2EE. Il prend, donc, en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre eux. L'avantage par rapport aux serveurs d'application est qu'avec SPRING, les classes n'ont pas besoin d'être implémentées. [20].
- **Framework Angular 5** : c'est le Framework JavaScript de Google qu'on a utilisé pour le développement de partie front office. Il permet de bien structurer le code en couche et il prend en charge la langage TypeScript.
- **DRL** : c'est le langage dont nous écrivons des règles métier. Nous les appelons règles métier car ils sont spécifique à un domaine bien spécifique et aussi il peuvent être facilement modifier par un expert métier non informaticien [21].
- **Spring Boot** : il nous aide à accélérer le développement des application Spring. ca permet une configuration rapide puisqu'on n'est plus obligé la redéfinir, donc on évite la redondance au niveau du code [22].
- **Spring security** : nous permet d'assurer une authentification sécurisée, aussi le plugin Spring Security nous facilite la gestion des rôles des utilisateurs de l'application [23].
- **JPA (Java Persistence Api)** : c'est un standard qui sert à faire le mapping entre les objets java et les tables physiques dans la base de données [24].
- **JSON** : est une format de échange et de transfert les données [25].
- **HTML5** : est un langage de balisage qui nous permet de créer des pages Web [26].
- **CSS3 (Cascading Style Sheets)** : utilisé dans le contexte du Web pour gérer la disposition des éléments des pages HTML aussi que la description des styles [27].

- **Bootstrap** : c'est un Framework HTML, JS et CSS le plus populaire pour le développement Web responsive [28].
- **Hibernate** : afin de faciliter le développement de la couche persistance de l'application, nous avons opté pour la solution open source « Hibernate » de type ORM [29].
- **TypeScript** : c'est un langage de programmation offert par Microsoft pour mieux sécuriser le code Javascript [30].

4.5 Environnement de travail

Le développement d'une application passe par des différentes phases et environnements avant d'être utilisé par client final.

La figure 4.8 montre les différents types d'environnements :

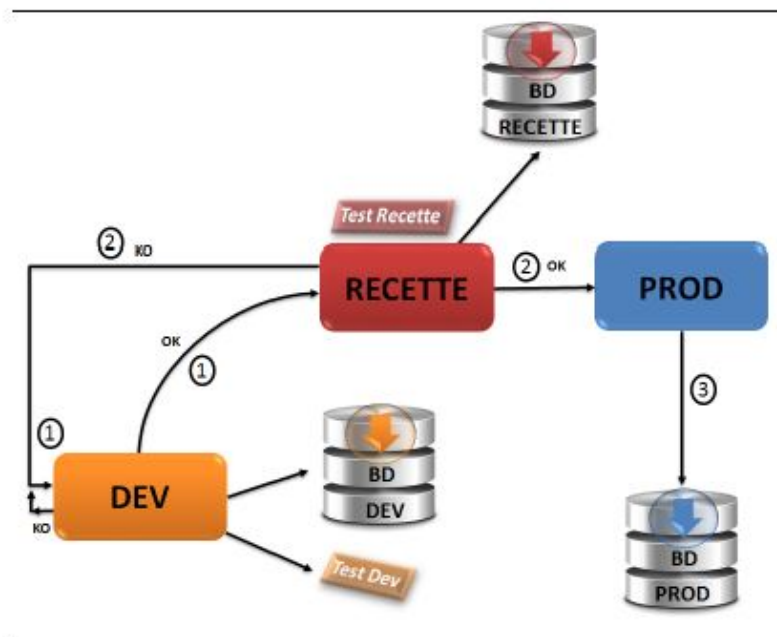


Figure 4.8: Environnements de travail

- **Développement (DEV)** : chaque développeur dispose de son environnement local pour développer des nouvelles fonctionnalités.
- **Recette** : c'est une phase de validation de la production auprès des clients avant la mise en production. Durant cette opération, le client vérifie la conformité du produit livré par le fournisseur par rapport à la commande passé.
- **Production (PROD)** : c'est l'environnement final destiné au clients finaux du produit.

Conclusion

Au cours de ce chapitre, nous avons fait une initialisation du projet. Par la suite nous avons présenté l'architecture de notre application en fonction des technologies utilisées. Ensuite, nous avons décortiqué la logique de l'application à travers une conception détaillée.

Dans le chapitre suivant, nous allons illustrer la réalisation de l'application et nous allons entamer la conception des différents sprints.

IMPLÉMENTATION ET TEST DU SPRINT 1 ET SPRINT 2

Plan

Introduction	49
1 Mise en place du Sprint 1	49
2 Mise en place du Sprint 2	56
3 Problèmes rencontrés	64

Introduction

Ce chapitre fait l'objet d'une présentation du sprint 1 et sprint 2 du projet. L'étude de chaque sprint couvre l'analyse, la conception, la réalisation.

5.1 Mise en place du Sprint 1

Le tableau présente le backlog du sprint 1 qui est composé de l'authentification et l'introduction des différentes informations nécessaires pour la bonne prise en compte dans le SIRH.

5.1.1 Baklog Sprint 1

Le tableau 5.1 présente le backlog du sprint 1.

Tableau 5.1: Backlog du sprint 1

ID	User Story	Tâches	Estimation (h/h)
AM1	En tant que client ou consultant, je veux m'authentifier afin d'accéder à mon espace.	Analyse : Préparation de la maquette qui contient le formulaire de l'authentification des utilisateurs	2
		Conception : définir les classes relatives à l'authentification.	6
		Développement : Développement des parties (DAO- Service - REST service pour OpenHR – Spring Security) pour réaliser l'authentification.	7
AM2	En tant que client, je veux créer un dossier de création de société	Analyse : Préparation de la maquette de création de société	3
		Conception : définir les composants sous forme de classes techniques réutilisables pour le dossier société.	7

		Développement : <ul style="list-style-type: none"> • Préparation des interfaces web, models et web services concernant le dossier société pour la couche front end. • Génération des entités à partir de la base de données. • développement des parties (DAO-Service – REST Web Service – Interface) à consommer par la front office. • Ajouter les règles de gestion concernant la création d'une société dans le moteur de règle Drools. 	9
AM3	En tant que Client, je veux préciser l'identification juridique de la société	Analyse : Préparation des maquettes relatives à l'identification juridique de la société	2
		Conception : Ajouter les informations nécessaires qui répondent à l'identification juridique de la société et définir les règles de gestion adéquates.	6

		Développement : <ul style="list-style-type: none"> • Ajuster les interfaces web, modèles et web services relatifs à l'identification juridique pour la couche front end. • Création des entités en backend afin de générer les tables dans la base de données locale. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. • Spécification des nouvelles règles de gestion à introduire dans la base de règles Drools qui concernent l'identification juridique de la société. 	9
AM4	En tant que Client, je veux préciser les liens avec les OPS	Analyse : Préparation de la maquette qui précise les liens avec les OPS	2
		Conception : Ajouter les informations nécessaires qui répondent à l'identification des liens avec les OPS et définir les règles de gestion adéquates.	5

		Développement : <ul style="list-style-type: none"> • Ajuster les interfaces web, modèles et web services relatifs à l'identification juridique pour la couche front end. • Création des entités en backend afin de générer les tables dans la base de données locale. • Développement des parties (DAO-Service – REST Web Service – Interface) dans la couche métier à consommer par la front office. • Spécification des nouvelles règles de gestion à introduire dans la base de règle Drools qui concernent les liens avec les OPS. 	7
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---

5.1.2 Conception

Cette partie comporte la conception du sprint, qui définit les composants sous forme de classes techniques réutilisables. Elle offre la possibilité de vérifier les prévisions afin de réajuster les histoires utilisateurs en cas d'erreur.

5.1.2.1 Conception Statique

Le diagramme de la figure 5.1 représente un modèle de classes que nous avons conçu lors du premier sprint.

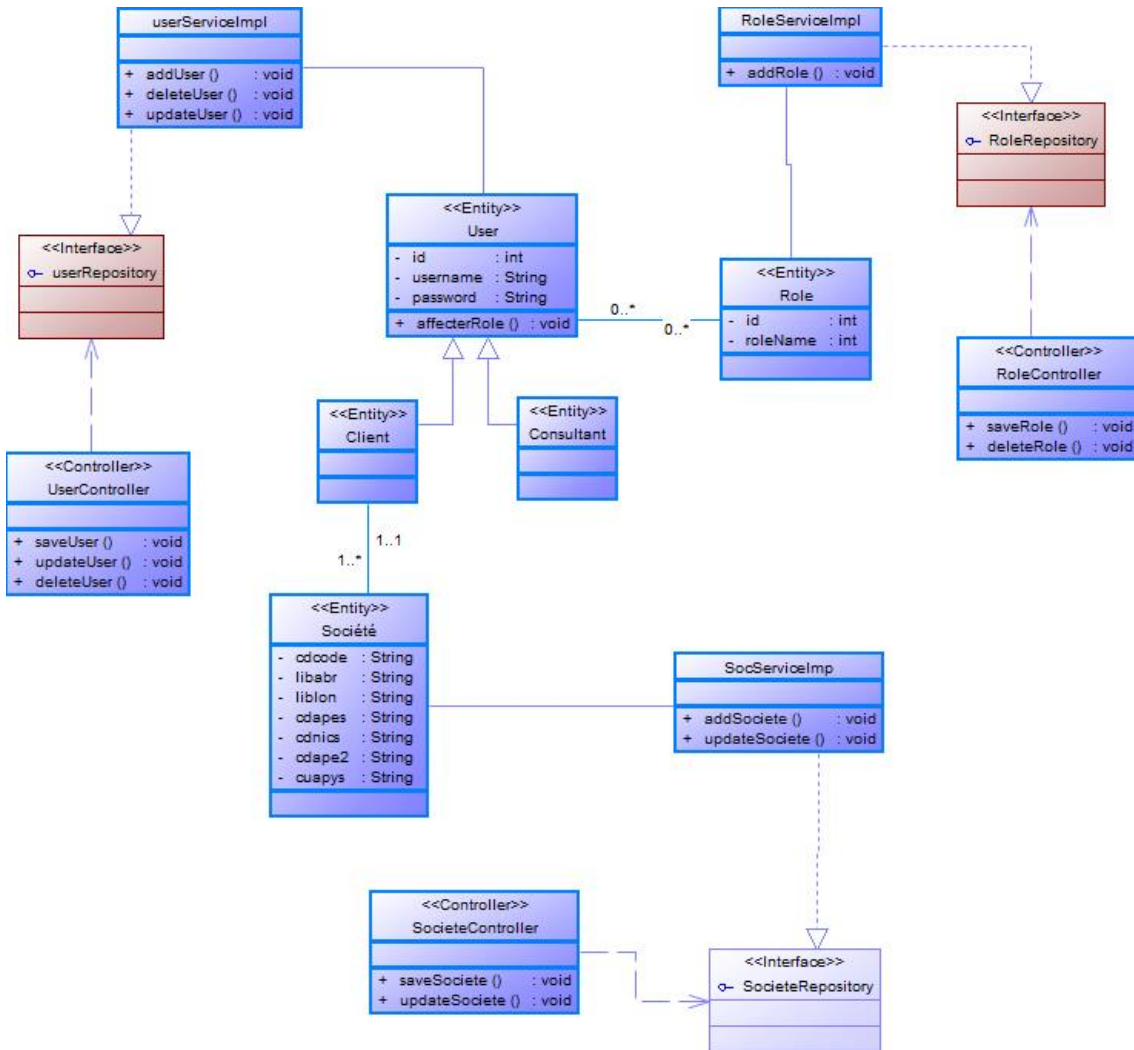


Figure 5.1: Diagramme de classe du premier sprint

Dans notre diagramme de classe, nous trouvons les classes suivantes :

- **Les entités** : chaque entité encapsule toutes les informations relatives à un répertoire donnée à savoir le Client, rôle, société et consultant.
- **Les interfaces** : une interface définit un comportement qui doit être implémenté par une classe métier, sans implémenter ce comportement.
- **Le contrôleur** : reçoit des demandes utilisateurs et invoque des services pour les traitements métier.
- **Les classes métiers** : contiennent les implémentations métier relatives pour la création de société et les utilisateurs de notre application.

5.1.2.2 Conception Dynamique

La figure 5.2 présente le diagramme de séquence relatif au cas d'utilisation «S'authentifier».

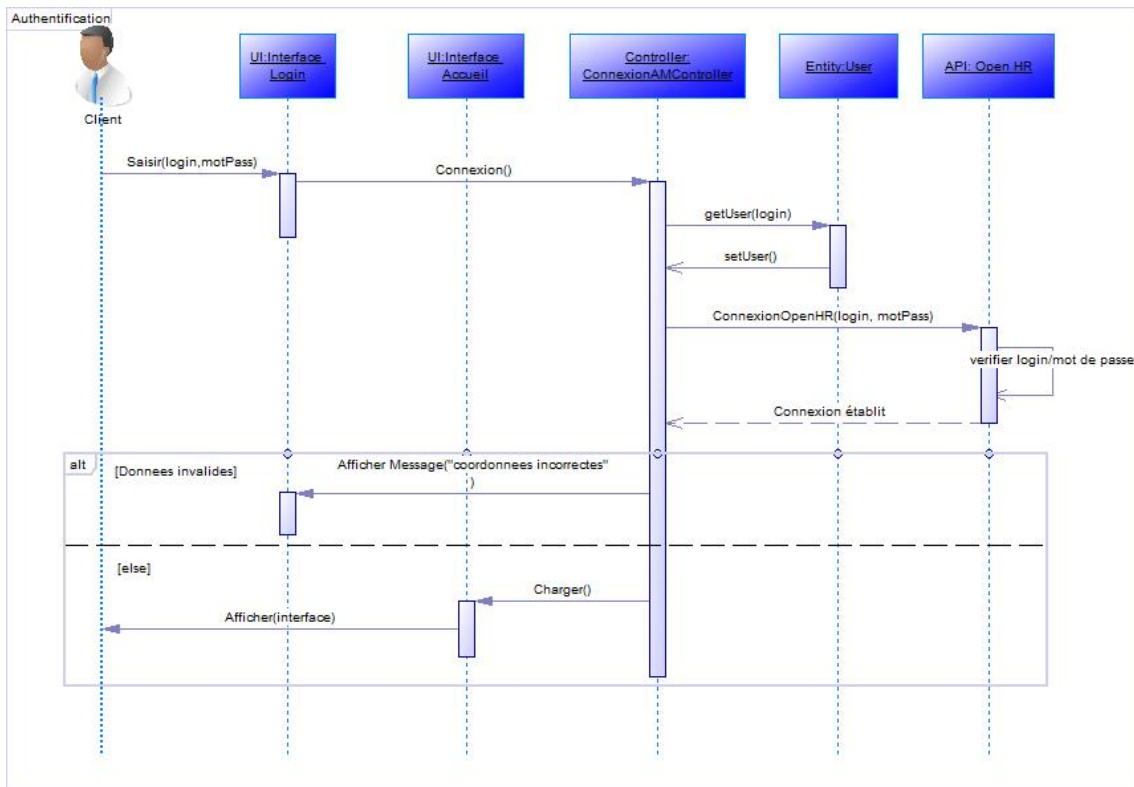


Figure 5.2: Diagramme de séquence «S'authentifier»

5.1.3 Interfaces Sprint 1

Dans ce qui suit, nous présentons quelques scénarios d'exécution pour montrer les fonctionnalités offertes par notre application durant le sprint 1.

5.1.3.1 Vue de l'authentification

Pour les utilisateurs de l'application, le consultant et le client, accèdent à l'application, ils doivent s'authentifier et cela nécessite qu'ils y saisissent leurs identifiants (login et mot de passe) comme présenté dans la figure 5.3.

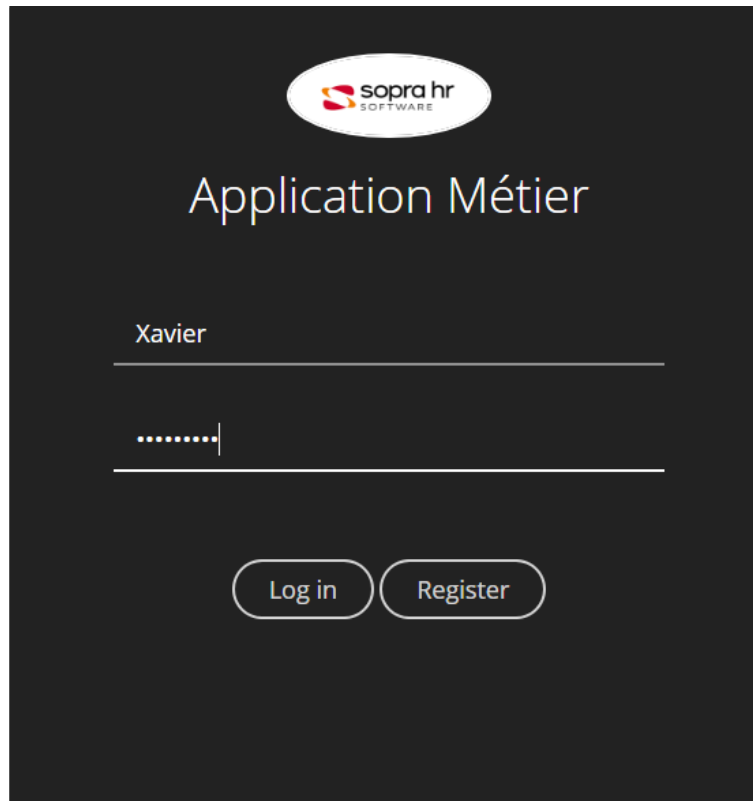


Figure 5.3: Interface authentification des utilisateurs de l'application

5.1.3.2 Vue Ajout demande

Une fois authentifié, le client sera redirigé vers la page d'ajout demande. Dans la figure 5.4, nous présentons comment le client passe cette demande.

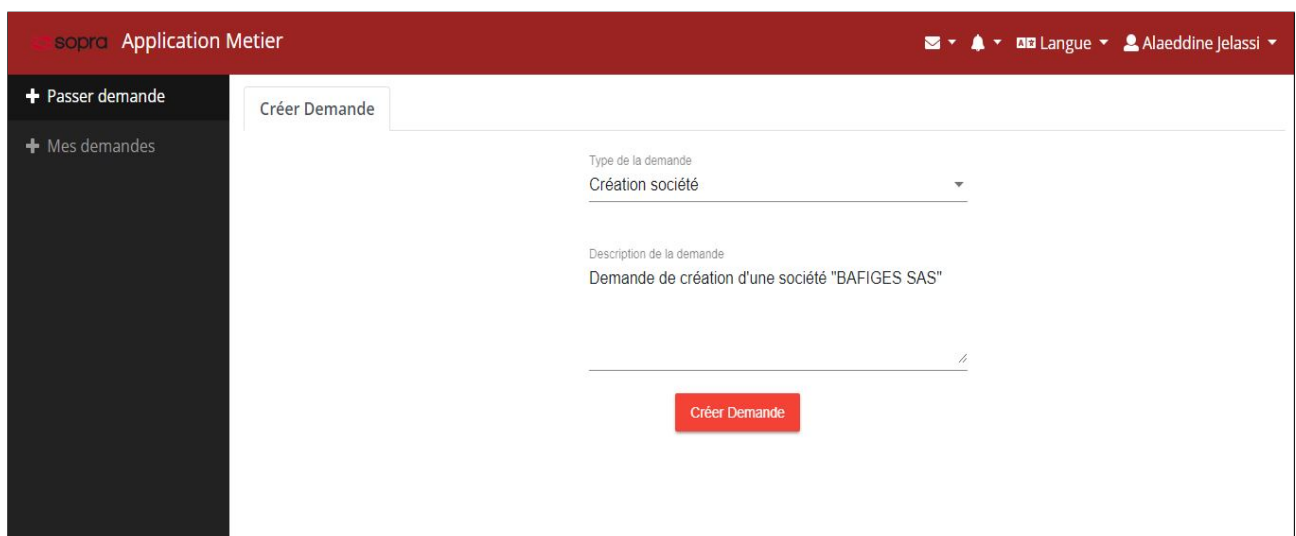


Figure 5.4: Interface d'ajout de demande de la part du client

5.1.3.3 Interface de création de société

Après avoir choisi sa demande, le client commence à saisir les informations nécessaires à la création de société, l'identification juridique, les coordonnées bancaires, les durées théoriques, etc. La figure 5.5 montre les formulaires que le client doit remplir afin de nous fournir les informations nécessaires à la bonne prise en compte par le produit

Figure 5.5: Interface de la création d'une société «Identification d'une société»

5.2 Mise en place du Sprint 2

Pour la mise en place du deuxième sprint, qui a pour objectif la création d'un dossier établissement, nous avons commencé par le backlog du sprint suivi d'une conception, et nous allons clôturer cette partie par la présentation des différentes interfaces réalisées au cours de ce sprint.

5.2.1 Backlog Sprint 2

Le tableau 5.2 présente le backlog du sprint 2 qui contient une liste des tâches fixées par l'équipe Scrum.

Tableau 5.2: Backlog du sprint 2

ID	User Story	Tâches	Estimation (h/h)
----	------------	--------	---------------------

AM5	En tant que Client, je veux créer les dossiers établissement	Analyse : Préparation de la maquette de création d'établissement	2
		Conception : Définir les classes techniques réutilisable qui répondent aux besoins fonctionnels de la création d'établissement et définir les règles de gestion adéquates.	7
		Développement : <ul style="list-style-type: none"> • Création de l'interface sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. • Création des entités en backend afin de générer les tables dans la base de données locale. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. • Alimenter la base de règles Drools avec les règles de gestion qui concerne le dossier d'établissement. 	9
AM6	En tant que Client, je veux créer les unités organisationnelles	Analyse : Préparation de la maquette de créations des unités organisationnelles.	2
		Conception : Définir les classes techniques réutilisables qui répondent aux besoins fonctionnels de la création des unités organisationnelles et définir les règles de gestion adéquates.	

		Développement : <ul style="list-style-type: none"> — Création de l'interface sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. — Création des entités en backend afin de générer les tables dans la base de données locale. — Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. — Alimenter la base de règles Drools avec les règles de gestion qui concerne les unités organisationnelles. 	7
AM7	En tant que Client, je veux créer les postes	Analyse : Préparation de la maquette de créations des postes	2
		Conception : Définir les classes techniques réutilisables qui répondent aux besoins fonctionnels de la création des postes et définir les règles de gestion adéquates.	5

		Développement : <ul style="list-style-type: none"> • Création de l'interfaces sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. • Création des entités en backend afin de générer les tables dans la base de données locale. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. • Alimenter la base de règles Drools avec les règles de gestion qui concerne les postes rattachés à l'établissement. 	8
AM8	En tant que Client, je veux créer les emplois	Analyse : Préparation de la maquette de créations des emplois	2
		Conception : Définir les classes techniques réutilisables qui répondent aux besoins fonctionnels de la création des emplois et définir les règles de gestion adéquates.	5

		Développement : <ul style="list-style-type: none"> • Création de l'interface sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. • Création des entités en backend afin de générer les tables dans la base de données locale. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. • Alimenter la base de règles Drools avec les règles de gestion qui concerne les emplois rattachés aux différents postes 	8
AM9	En tant que Client, je veux indiquer les nouvelles règles métier (Congés payés, les nouvelles règles en lien avec les Absences SS, les nouveaux éléments de rémunération à paramétrer. . . .)	Analyse : Préparation de la maquette qui permet aux clients d'introduire leurs règles métier spécifiques	2
		Conception : Définir les classes techniques réutilisable qui répondent aux besoins fonctionnels de la création des nouvelles règles métier et définir les règles de gestion adéquates.	6

		Développement : <ul style="list-style-type: none"> • Création de l'interfaces sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. • Création des entités en backend afin de générer les tables dans la base de données locale. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. • Alimenter la base de règles Drools avec les règles de gestion qui concerne les règles métier spécifiques. 	8
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---

5.2.2 Conception

Cette partie comporte la conception, qui définit les composants sous forme de classes techniques réutilisables. Elle offre la possibilité de vérifier les prévisions afin de réajuster les histoires utilisateurs en cas d'erreur.

5.2.2.1 Conception Statique

Dans cette partie, nous allons illustrer le diagramme de classe afin de préciser la structure statique de ce sprint.

La figure 5.6 représente un modèle de classes que nous avons conçu lors du deuxième Sprint.

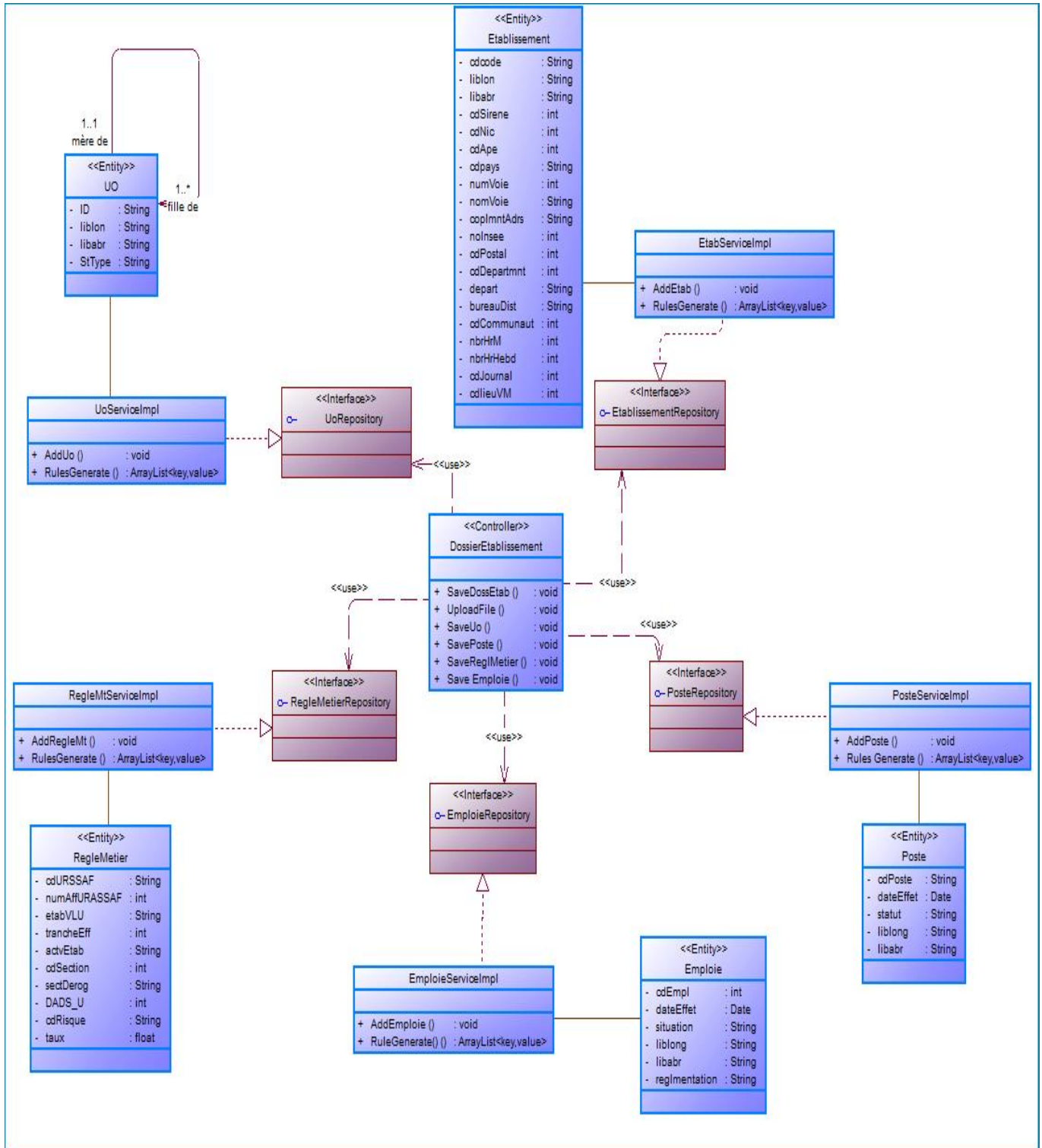


Figure 5.6: Diagramme de classes du deuxième sprint

5.2.2.2 Conception Dynamique

Nous allons mieux décrire cette interaction en présentant le diagramme de séquence de la figure 5.7 relatif au cas d'utilisation «Ajouter Dossier Établissement».

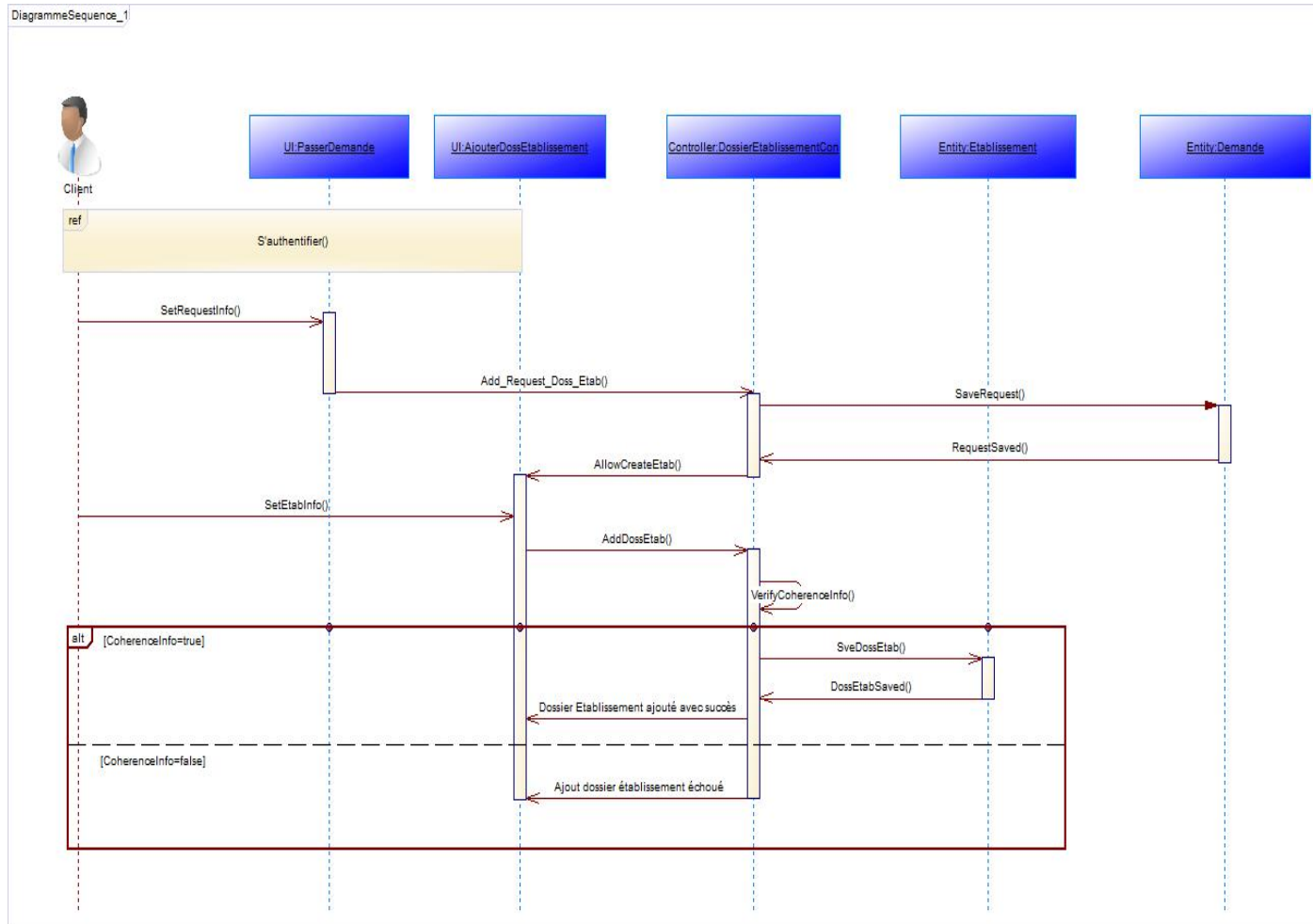


Figure 5.7: diagramme de séquence «Ajouter Dossier Établissement»

5.2.3 Interfaces Sprint 2

La figure 5.8 représente la page de création d'établissement «Identification de l'établissement»

The screenshot shows the 'Application Metier' interface. The top navigation bar is dark red with the 'sopra' logo and the title 'Application Metier'. On the right, there are icons for email, notifications, language, and a user profile 'Alaeddine Jelassi'. A left sidebar contains a menu with 'Création Société' expanded, showing sub-items: 'Définition de la société', 'Définition établissement', 'Définition des unités organisationnelles', 'Définition des emplois', and 'Définition des postes'. Below these are 'Mes demandes' and 'Fiche Technique'. The main content area has tabs: 'Définition', 'Déclaration légale' (active), 'Absences', and 'Paie'. Under 'Déclaration légale', there are sub-tabs: 'Identification', 'Adresse', 'Horaires théoriques', 'Interface', 'Renseignements médicaux', and 'Anciens horaires de'. The 'Identification' sub-tab is active, showing a form with fields: 'Code établissement *', 'Libellé court *', 'Assurance Maladie / Alsace-Moselle *', 'Code NIC *', 'libelle *', 'Code société *', 'Code SIRENE *', and 'Code APE (NAF1) *'. At the top right of the form are buttons for 'Export', 'Import', and 'Save'.

Figure 5.8: identification juridique de l'établissement

5.3 Problèmes rencontrés

Lors de la réalisation du sprint 1 et 2, nous avons rencontré des obstacles puis nous avons proposé des solutions qui sont détaillées dans le tableau 5.3.

Tableau 5.3: Obstacles rencontrés dans la réalisation des sprints 1 et 2

ID Sprint	Obstacles	Solutions proposées
Sprint 1	L'authentification du consultant SIRH doit être effectuée avec le même login et mot de passe de son espace HR Access. La gestion de la sécurité avec Spring Security nécessite la sauvegarde des logins et des mots de passe. Nous avons rencontré un problème lors de la récupération des mots de passe. Par mesure de sécurité, l'équipe PRODOPS au sein de Sopra HR, nous a empêché de récupérer les mots de passe.	Pour assurer une authentification sécurisée, nous avons utilisé l'API OpenHR qui permet la gestion des rôles et des accès. Ensuite, nous avons effectué des tests sur les résultats retournés par l'API OpenHR. Si la connexion est établie et le rôle de l'utilisateur correspond à un consultant SIRH, nous récupérons les paramètres d'authentification pour les passer, par la suite, dans Spring Security d'une façon dynamique, ce qui assure la sécurité de notre application.
Sprint 2	La demande création de dossier société ou établissement est très spécifique et nécessite une compréhension du fonctionnel lors de la saisie des formulaires.	Nous avons demandé l'aide des experts.

IMPLÉMENTATION ET TEST DU SPRINT 3 ET SPRINT 4

Plan

Introduction	67
1 Mise en place du Sprint 3	67
2 Mise en place du Sprint 4	76
3 Problèmes rencontrés	82

Introduction

Ce chapitre fait l'objet d'une présentation du troisième et quatrième sprints. L'étude de chaque sprint couvre l'analyse, la conception, la réalisation.

6.1 Mise en place du Sprint 3

Ce sprint a pour but de développer le module de génération de paramétrage automatique ainsi, la vérification de la cohérence avec la base HR ACCESS pour assurer l'intégrité des données.

6.1.1 Baklog Sprint 3

Le tableau 6.1 présente le bakclog du sprint 3.

Tableau 6.1: Backlog du sprint 3

ID	User Story	Tâches	Estimation (h/h)
AM14	En tant que consultant, je gérer mes demandes reçues	Analyse : Préparation de la maquette de la boîte de réception	3
		Conception : Définir les classes techniques réutilisable qui traduit les besoins fonctionnels de la gestion des demandes .	7

		Développement : <ul style="list-style-type: none"> • Création de l'interfaces sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. • création d'un système de notification qui assure le transfert des demandes en temps réel. • Création des entités en backend afin de générer les tables dans la base de données locale. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. 	9
AM15	En tant que consultant, je veux valider le dossier afin de générer les codes et le paramétrage nécessaire	Analyse : Préparation de la maquette de visualisation du dossier de la demande client ainsi que la maquette de la fiche technique	5
		Conception : Définir les classes techniques réutilisables qui répondent aux besoins fonctionnels de la génération de la fiche technique.	8

		Développement : <ul style="list-style-type: none"> • Création de l'interface sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. • Création des entités en backend afin de générer les tables dans la base de données locale. • Création et intégration du moteur de règles DROOLS dans la partie backend. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. 	12
AM16	En tant que consultant, je veux vérifier la cohérence de ce paramétrage par rapport au celui présent au SIRH	Analyse : Utiliser la même maquette de la génération de la fiche technique.	2
		Conception : Définir les classes techniques réutilisables qui répondent aux besoins fonctionnels de la vérification de cohérence	7

		Développement : <ul style="list-style-type: none"> • Création de la classe de connexion avec la base HR ACCESS via l'API OpenHR. • Création de l'interfaces sous forme des composants Angular 5 aussi que le model et les services pour consommer les web services dans la partie front. • Création des services qui permettent de communiquer avec la base HR ACCESS pour assurer l'intégrité des données contenues dans la fiche technique avec les données existantes déjà dans la base HR ACCESS. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. 	9
AM17	En tant que consultant, je veux modifier le paramétrage afin de bien répondre au besoin du client	Analyse : Préparation de la maquette de modification de fiche technique.	8
		Conception : Définir les classes techniques réutilisables qui répondent aux besoins fonctionnels de la modification du paramétrage.	15

		Développement : <ul style="list-style-type: none"> • Création de l'interface sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front. • Création d'un service qui permet de changer le comportement de moteur de règles une fois, une modification est apportée sur la fiche technique générée. • Création des contrôleurs de type REST pour assurer la communication avec la partie frontale. 	15
AM18	En tant que consultant, je veux injecter les données de la fiche de paramétrage dans la base HR	Analyse : Préparation de la maquette d'injection des données dans la base HR ACCESS.	6
		Conception : Définir les classes techniques réutilisables qui assure l'injection des données de notre base local vers la base HR ACCESS.	16

		Développement : <ul style="list-style-type: none">• Création de l'interface sous forme des composants Angular 5 aussi que le modèle et les services pour consommer les web services dans la partie front.• Création des contrôleurs de type REST pour assurer la communication avec la partie frontale.• Création des services qui assurent le mapping et l'injection des données dans la base HR ACCESS via l'API OpenHR.	20
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

6.1.2 Conception

Après avoir étudié les différentes tâches du sprint 3, nous avons eu recours au diagramme de classe ainsi que quelques diagrammes de séquences pour réaliser la conception.

6.1.2.1 Conception Statique

Pour mieux décortiquer les composantes techniques et mettre en évidence la structure statique de ce sprint nous avons réalisé le diagramme classes de ce sprint, présenté à la figure 6.1.

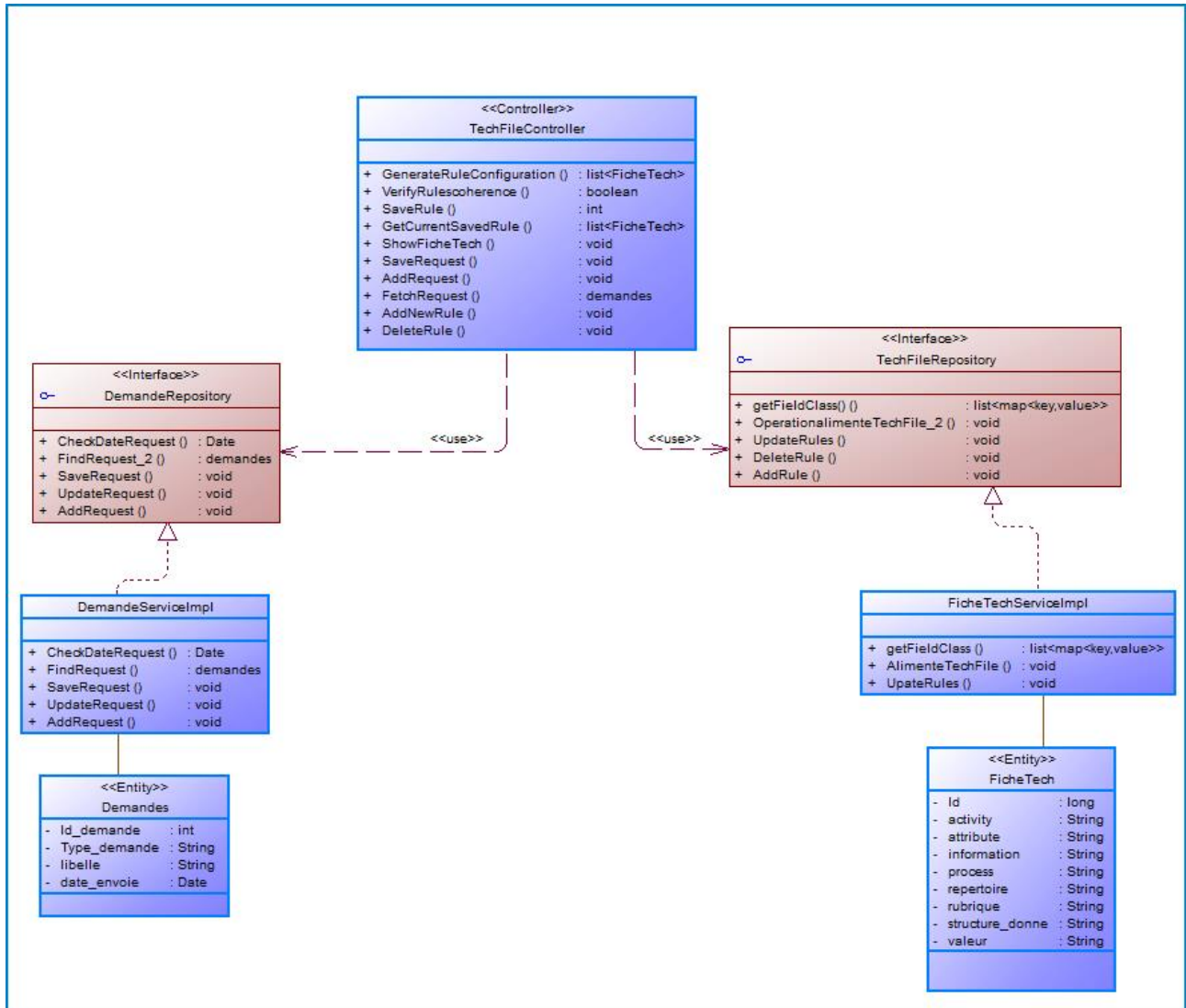


Figure 6.1: Diagramme de classe du troisième sprint

6.1.2.2 Conception Dynamique

Pour schématiser la vue dynamique et les interactions entre l'acteur et le système intervenant dans le sprint 3, nous faisons recours au diagramme de séquence UML. Dans ce contexte nous allons présenter le diagramme de séquence relatif au cas d'utilisation «Générer automatiquement le paramétrage des données d'un dossier» présenté à la figure 6.2.

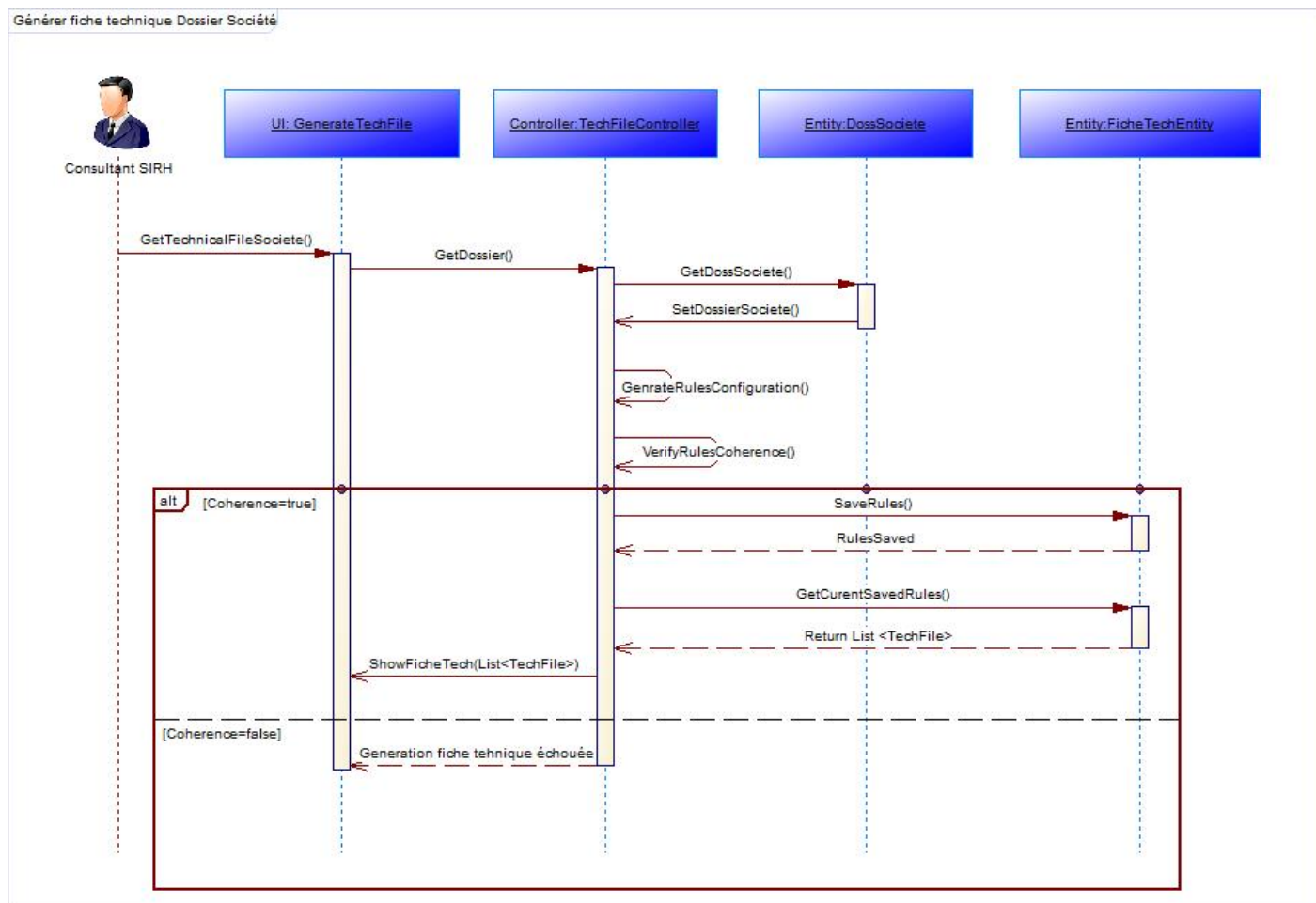


Figure 6.2: diagramme de séquence «Générer automatiquement le paramétrage des données d'un dossier»

6.1.3 Interfaces Sprint 3

La figure 6.3 représente le système de notification dans la session du consultant, suite à l'envoi d'une demande de la part du client.

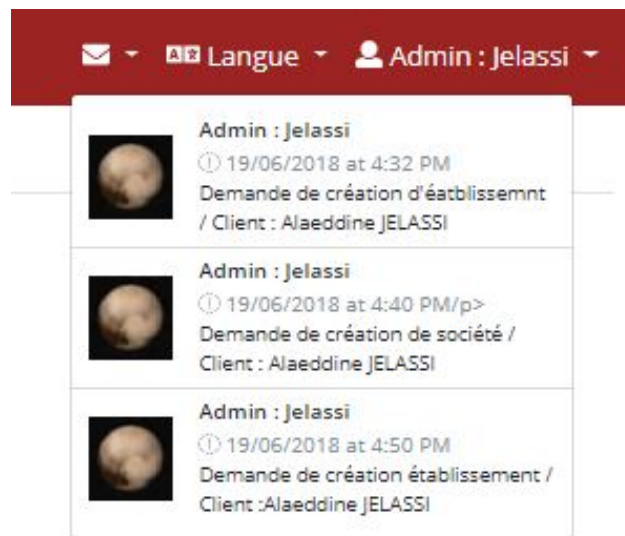


Figure 6.3: Interface de notifications reçues dans la session du consultant

La figure 6.4 représente la liste des demandes reçues, c'est là, où le consultant peut vérifier l'état de chaque demande, ainsi il peut accéder à la demande en consultation et il là il va passer à la même interface de introduction du dossier.



Figure 6.4: Interface de la liste des demandes reçues

La figure 6.5 représente le paramétrage généré automatiquement grâce au moteur de règles, cet échantillon de paramétrage est adéquat à la création d'un établissement.

Activity	Attribut	Information	Process	Repertoire	Rubrique	Structure de donnée	Valeur	Edit
CSTM CY01	curcy	0A	FS901	UAA	CURCY	ZD	EUR	edit
CSTM CY01	compt1	0I	FS901	UAA	COMPT1	ZD	0003726399995	edit
CSTM CY01	idlo00	0A	FS901	UAA	IDLO00	ZD	FR	edit
CSTM CY01	cdstco	0A	FS901	UAA	CDSTCO	ZD	UAA	edit
CSTM CY01	bangfr	0I	FS901	UAA	BANQUE	ZD	30003	edit
CSTM CY01	culan	0I	FS901	UAA	CDLANG	ZD	FR	edit
CSTM CY01	cdcodeSoc	00	FS901	UAA	CDCODE	ZD	17	edit
CSTM CY01	cpadfr	0F	FS901	UAA	CPADFR	ZD	ADRESSE Complémentaire	edit

Figure 6.5: Interface de la fiche technique générée

La figure 6.6 correspond à l'interface qui permet au consultant de modifier le paramétrage générée dans la fiche technique.

Mettre à jour Paramétrage

Activity * CSTMCY01

Attribut * curcy

Information * 0A

Process * FS901

Repertoire * UAA

Rubrique *

Structure * ZD

Confirmation

Êtes vous sûr de modifier le paramétrage courant!!

[Cancel](#) [Confirm](#)

[Confirmer](#)

Figure 6.6: Interface de mise à jour de la fiche technique

6.2 Mise en place du Sprint 4

Pour la mise en place du deuxième sprint, qui a pour objectif la création d'un dossier société, nous avons commencé par le backlog du sprint suivi d'une conception, et nous allons clôturer cette partie par la présentation des différentes interfaces réalisées au cours de ce sprint.

6.2.1 Baklog Sprint 4

Le tableau 6.2 présente le backlog du sprint 4 qui contient une liste des tâches fixées par l'équipe Scrum.

Tableau 6.2: Backlog du sprint 4

ID	User Story	Tâches	Estimation (h/h)
AM14	En tant que client, je veux modifier mon dossier	Analyse : Préparation des maquettes relatives à la modification du dossier	2
		Conception : Préparation de la maquette de modification du dossier client	4
		Développement : <ul style="list-style-type: none">• Développement de l'interface web de la modification de la demande• Création des web services de modification dans la partie front.• développement des parties (DAO-Service – REST Web Service – Interface) de la modification à consommer par la front office.	7
AM15	En tant que client, je veux enregistrer les versions de mon dossier	Analyse : Ajuster la maquette relative à la sauvegarde des versions du dossier	2
		Conception : Définir les composants sous forme de classes techniques réutilisables en relation avec le sauvegarde des versions des dossiers	5

		Développement : <ul style="list-style-type: none"> • Préparation de l'interface web, modèle et web service concernant la version du dossier pour la partie front end. • Création des entités en backend afin de générer les tables dans la base de données locale. • Développement des parties (DAO-Service – REST Web Service – Interface) dans la couche métier à consommer par la front office. 	7
AM16	En tant que client, je veux annuler mon dossier	Analyse : Préparation de la maquette d'annulation du dossier.	2
		Conception : Définition des fonctionnalités permettant l'annulation du dossier	4
		Développement : <ul style="list-style-type: none"> • Préparation de l'interface web, modèle et web service concernant l'annulation du dossier pour la partie front end. • Création d'une fonctionnalité en backend permettant l'affichage des dossiers. • Création d'une fonctionnalité en backend permettant l'annulation du dossier. 	7

AM17	En tant que client, je veux valider mon dossier ce qui génère un ticket	Analyse :	2
		Conception : Spécifier les fonctionnalités permettant la validation du dossier.	2
		Développement : <ul style="list-style-type: none"> • ajuster les interfaces web de validation du dossier. • création d'un service web qui permet la consommation de l'API Easy vista qui génère un ticket avec les informations de la demande. 	7

6.2.2 Conception

Nous allons présenter pour ce sprint le diagramme de classe et le diagramme de séquence.

6.2.2.1 Conception Statique

La figure 6.7 présente les classes contenues dans ce module.

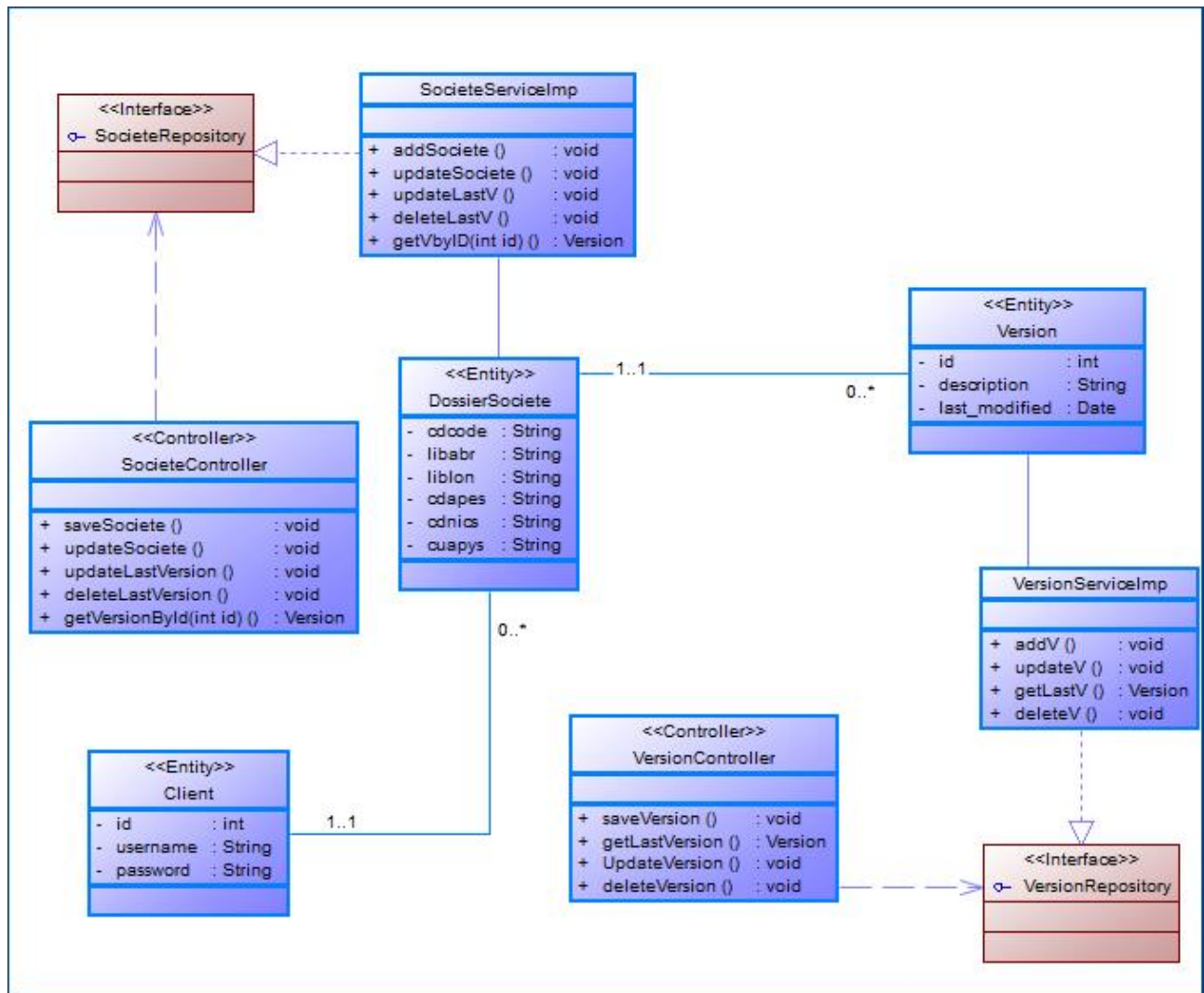


Figure 6.7: Diagramme de classe du quatrième sprint

6.2.2.2 Conception Dynamique

Nous allons mieux décrire ces interactions en présentant le diagramme de séquence relatif au cas d'utilisation "Modifier une version du dossier". La figure 6.8 présente le diagramme de séquence correspondant.

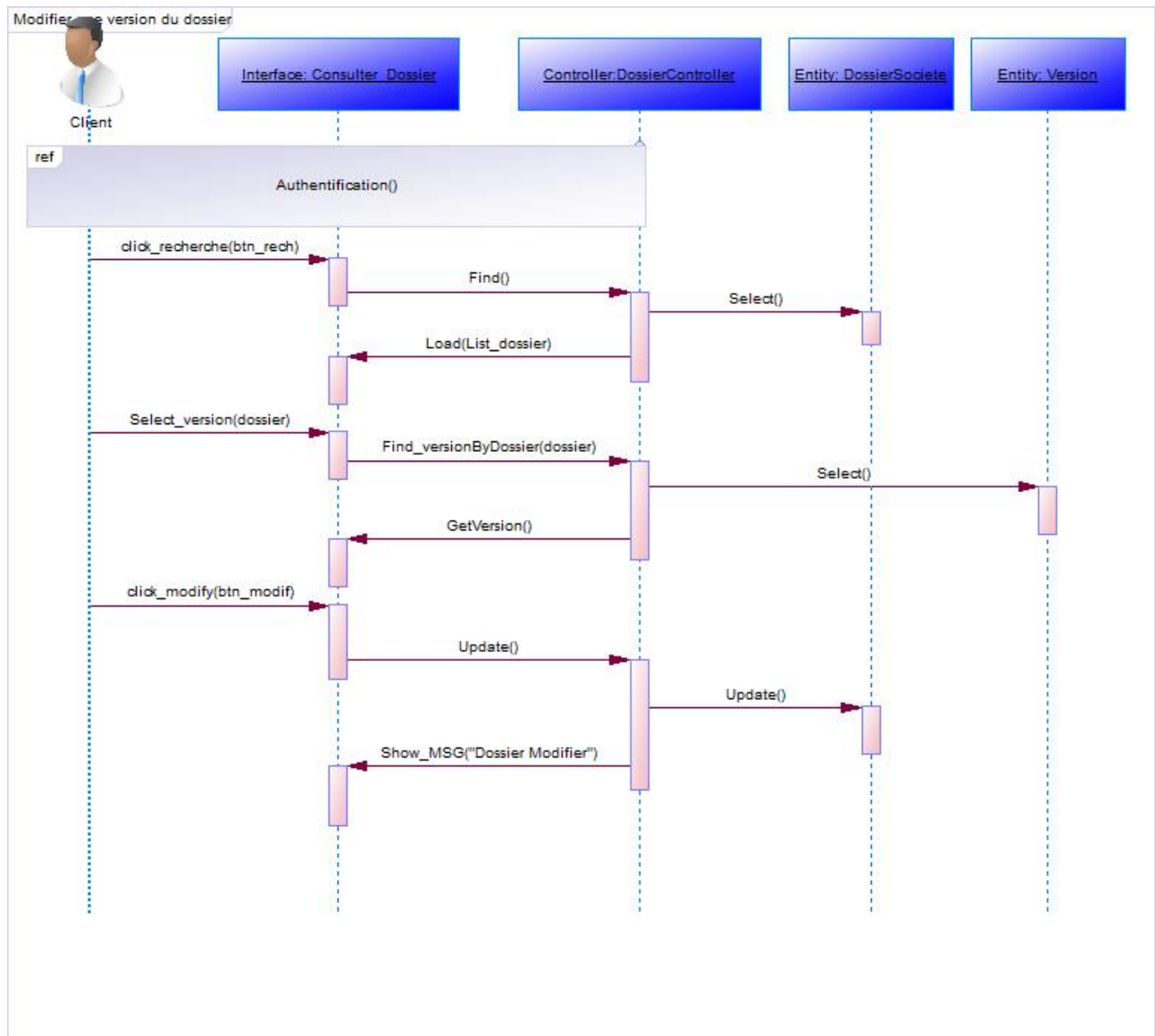


Figure 6.8: Diagramme de séquence «Modifier une version du dossier»

6.2.3 Interfaces Sprint 4

La figure 6.9 représente la page de consultation des demandes effectuées par le client.

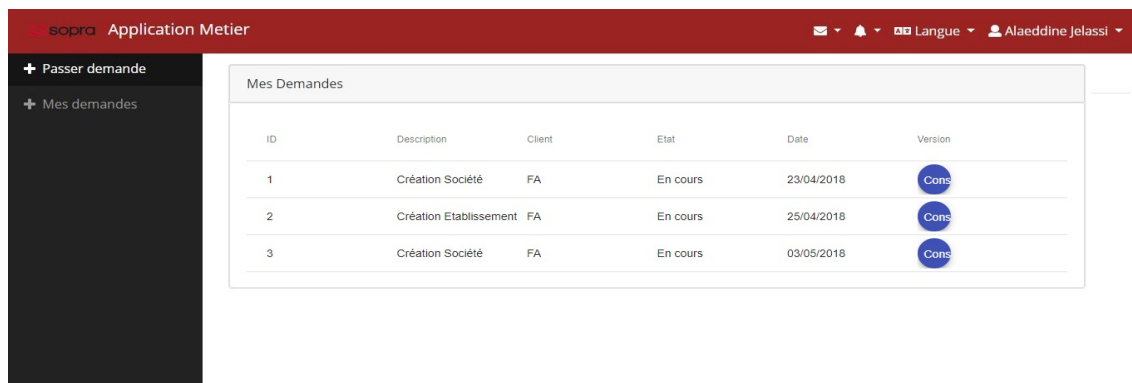


Figure 6.9: Interface de consultation d'un client ses demandes effectuées

La figure 6.10 représente la page de chargement de version du dossier par le client afin de compléter l'introduction du dossier de la demande.

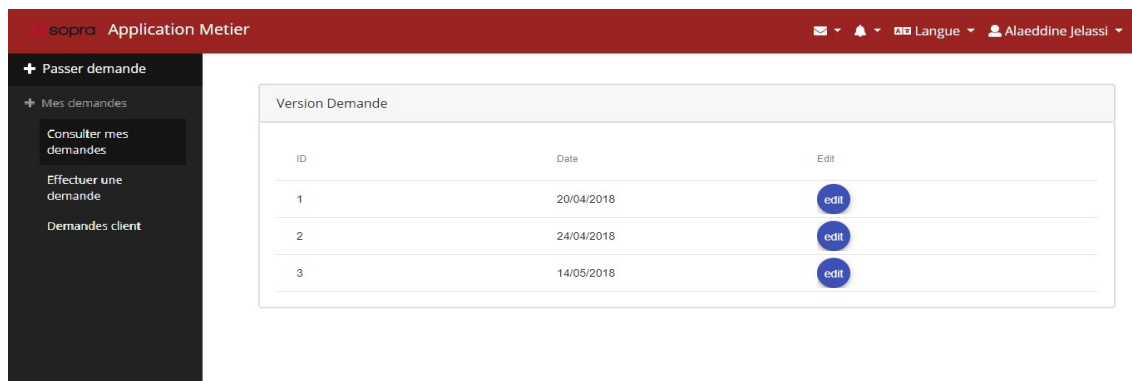


Figure 6.10: Interface chargement de version du dossier

6.3 Problèmes rencontrés

Le tableau 6.3 présente les solutions proposées face aux obstacles rencontrés lors de la réalisation du sprint 3 et 4.

Tableau 6.3: Obstacles rencontrés dans la réalisation des sprints 3 et 4

ID Sprint	Obstacles	Solutions proposées
Sprint 3	Lors de l'injection des données dans la base HR ACCESS, nous avons trouvé que il y a une notion de table virtuelle paramétrées et donc nous ne pouvons pas injecter les données dans le dossier d'injection que nous créons à travers l'API OpenHR.	Pour remédier a ce problème nous avons trouvé l'astuce de faire le paramétrage qui va initialiser les tables virtuelles paramétrée au niveau code java avant de créer le dossier d'injection et alimenter le reste des tables.

Conclusion générale

Au cours de ce projet, nous avons présenté les différentes étapes de la conception et de la réalisation l'Application Métier "AM", durant notre stage de fin d'études au sein de la société Sopra HR Software. Le projet a pour but de réaliser un outil d'optimisation des ressources mis par l'entreprise dans le déroulement des processus métier fourni aux client de sopra HR, tout en générant des règles de gestion et des actions métier dans le SIRH d'une façon automatique.

Nous avons commencé par comprendre le contexte général du projet puis nous avons étudié et analysé les exigences et les besoins. Ensuite, nous avons détaillé la phase d'initialisation du projet. Enfin, nous avons entamé la phase de conception et de réalisation de notre module de création d'une nouvelle organisation.

Le travail dans le cadre de ce PFE, était d'une importance considérable dans la mesure où il nous a servi comme portail vers le monde professionnel et la vie en entreprise.

Sur le plan professionnel, ce stage a été une occasion pour mettre en pratique les connaissances que nous avons acquises durant notre formation et de les enrichir en apprenant les technologies web comme [HTML5, CSS3.0, Angular 5, etc].

Sur le plan humain, ce projet a représenté pour nous une formidable expérience pour nous adapter et nous intégrer dans un environnement professionnel, au contact des collaborateurs de l'Outsourcing, qui nous ont prodigué leurs conseils et nous ont permis de profiter de leurs expérience. Prendre en charge un tel travail au sein d'une entreprise, nous a aidé à développer nos facultés d'analyses, de réflexion et de décision.

Ce travail a accompli ses objectifs. Cependant, il ne prétend pas la perfection. Toutefois, à la lumière des résultats obtenus, nous avons pu dégager des besoins constituant des améliorations au niveau de ce projet. En effet, de point de vue fonctionnel l'application peut être enrichie par le développement d'autres processus métier comme le changement de contrat de veille légale, modification des constantes qui peuvent être automatisées dans le SIRH. De plus, nous comptons rajouter un module de communication avec le client "Chatbot", qui dirige le client vers ses besoins d'une manière intelligente et automatique.

Bibliographie

- [1] F. LOTHON. (). Diagramme de déploiement ; adresse : <https://www.agiliste.fr/guide-de-demarrage-scrum/>.
- [2] L. FLORENT. (2018). Methodologie Agile. [Accès le 5-Juin-2018], adresse : <https://www.agiliste.fr/introduction-methodes-agiles/>.
- [3] D. C. SUPPORT. (2018). Certified.ScrumMaster. [Accès le 5-Juin-2018], adresse : <https://www.scrumalliance.org/get-certified/practitioners/csm-certification>.
- [4] —, (2018). Certified Scrum Product Owner. [Accès le 5-Juin-2018], adresse : <https://www.scrumalliance.org/get-certified/practitioners/cspo-certification>.
- [5] —, (2018). JIRA SOFTWARE. [Accès le 2-Juin-2018], adresse : <https://fr.atlassian.com/software/jira/agile>.
- [6] R. H. COMMUNITY. (2017). DROOLS. [Accès le 4-Juin-2018], adresse : https://docs.jboss.org/drools/release/7.7.0.Final/drools-docs/html_single/index.html#_introduction.
- [7] (). JDBC ; [Consulté le 21-05-2018], adresse : <http://web.maths.unsw.edu.au/~lafaye/CCM/java/javajdbc.htm>.
- [8] (). Diagramme de composant ; [Consulté le 21-05-2018], adresse : <http://uml.free.fr/cours/i-p17.html>.
- [9] H. ACCESS. (2018). Open HR Guide de developpeur OpenHR.Sopra HR Software.
- [10] —, (2018). Open HR Guide de developpeur OpenHR .Sopra HR Software.
- [11] D. C. SUPPORT. (2018). Eclipse EDI. [Accès le 2-Juin-2018], adresse : <http://help.eclipse.org/oxygen/index.jsp?nav=%2F0>.
- [12] ORACLE. (2018). JDK. [Accès le 2-Juin-2018], adresse : <http://www.oracle.com/%20technetwork/java/javase/downloads/index-jsp-138363.html>.
- [13] S. TATHAM. (2018). puTTY. [Accès le 2-Juin-2018], adresse : <http%20:%20/www.putty.org/>.
- [14] ORACLE. (2018). PowerAMC. [Accès le 1-Juin-2018], adresse : <https://www.oracle.%20com/database/index.html>.

- [15] A. COMMUNITY. (2017). Maven. [Accès le 4-Juin-2018], adresse : <https://maven.apache.org/>..
- [16] A. et BSD COMMUNITY. (2017). SVN. [Accès le 4-Juin-2018], adresse : <https://openclassrooms.com/courses/gerez-vos-projets-a-l-aide-du-gestionnaire-de-versions-subversion/un-gestionnaire-de-version..>
- [17] B.CAG. (2018). PostgreSQL. [Accès le 1-Juin-2018], adresse : <https://doc.ubuntu-fr.org/postgresql>.
- [18] I. SYBASE. (2018). PowerAMC. [Accès le 1-Juin-2018], adresse : <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc31025.1610/doc/html/rad1232023292093.html>.
- [19] JMDODOUX. (2018). Java EE. [Accès le 2-Juin-2018], adresse : <http://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>.
- [20] UNIV-PARIS-13. (2018). Spring. [Accès le 2-Juin-2018], adresse : <https://lipn.univ-paris13.fr/~fortier/Wiki/index.php?n=Enseignement.Spring>.
- [21] R. H. COMMUNITY. (2017). DRL. [Accès le 4-Juin-2018], adresse : https://docs.jboss.org/drools/release/7.7.0.Final/drools-docs/html_single/index.html#_drools_drleditor.
- [22] ——. (2018). définition de SpringBoot. [Accès le 5-Juin-2018], adresse : <https://projects.spring.io/spring-framework/#quick-start>.
- [23] ——. (2018). définition de SpringSecurity. [Accès le 5-Juin-2018], adresse : <https://projects.spring.io/spring-framework/#quick-start>.
- [24] M. MUNIER. (2017). JPA. [Accès le 5-Juin-2018], adresse : <https://openclassrooms.com/courses/creez-votre-applicationweb-avec-java-ee/la-persistence-des-donnees-avec-jpa>.
- [25] J. COMMUNITY. (2017). JSON. [Accès le 5-Juin-2018], adresse : <http://www.json.org/jsonfr.html..>
- [26] html COMMUNITY. (2017). HTML. [Accès le 4-Juin-2018], adresse : <https://www.w3schools.com/html/default.asp..>

- [27] C. COMMUNITY. (2017). CSS. [Accès le 4-Juin-2018], adresse : <https://www.w3schools.com/css/default.asp>..
- [28] B. COMMUNITY. (2017). BootStrap. [Accès le 4-Juin-2018], adresse : <http://getbootstrap.com/getting-started/>..
- [29] ORACLE. (2017). Hibernate. [Accès le 5-Juin-2018], adresse : https://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html.
- [30] T. COMMUNITY. (2017). TypeScript. [Accès le 4-Juin-2018], adresse : <https://www.npmjs.com/package/typescript>.

يندرج هذا العمل في إطار مشروع نهاية الدراسات الذي تم داخل الشركة متعددة الجنسيات Sopra HR Software و الهدف من هذا المشروع تنفيذ تطبيق لجعل بعض العمليات التجارية المخصصة لجميع العملاء أوتوماتكية. أهداف هذا التطبيق هي تقليل تدخل المستشارين العاملين في الشركة وتقديم تطبيق أوتوماتيكي ليكون الوسيط ، بين العملاء ونظام معلومات الموارد البشرية (نظام إدارة معلومات الموارد البشرية). سيتمكن هذا التطبيق هاتان العمليتان أوتوماتكيتان و هما: إنشاء شركة وإنشاء مؤسسة. تنفيذ مفهوم جديد يستند إلى Système experts الذي يتمثل في محرك قواعد DROOLS.

كلمات مفاتيح : Systèmes experts, DROOLS

Résumé

Ce travail s'inscrit dans le cadre du Projet de Fin d'Études réalisé au sein de l'entreprise multinationale Sopra HR Software. Ce projet consiste à la mise en place d'une application d'automatisation de certaines processus métier dédié à tous les clients. Les objectifs de cette application sont de diminuer de plus en plus la tâche du consultant et rendre une application d'automatisation, l'intermédiaire, entre le client et le SIRH (Système d'information de gestion des ressources humaines). Cette application va permettre l'automatisation de deux processus métier qui sont : la création d'une société et la création d'un établissement. L'implémentation d'une nouvelle notion qui se base sur les systèmes experts qui est le moteur de règles DROOLS.

Mots clés : SIRH, Processus métier, Automatisation, Moteur de règle, Systèmes experts, DROOLS

Abstract

This work takes part of the Project of the End of Studies realized within the multinational company Sopra HR software. This project consists in the implementation of an automation application for business processes dedicated to all customers. The objectives of this application are to make the task of the consultant more easy and make an automation application as an intermediary between the client and the HRIS (Human Resources Management Information System). This application will enable the automation of two business processes that are: the creation of a company and the creation of an establishment. The implementation of a new notion that is based on the expert systems that is the engine of rules DROOLS.

Keywords : HRIS, business processes, automation, engine of rules, expert systems, DROOLS