
number	cols	text	text	text	text	text
number	rows	number	rows	spt<=-number	rows	spt>=width
						text

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

Sami GHORBEL

Conception et Développement d'une plateforme d'apprentissage automatique

Encadrant professionnel : **Monsieur Aymen BEL ARBI**

Ingénieur R&D

Encadrant académique : **Madame Olfa MOURALI**

Maître Assistante

Réalisé au sein de SOFRECOM



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

Sami GHORBEL

Conception et Développement d'une plateforme d'apprentissage automatique

Encadrant professionnel : **Monsieur Aymen BEL ARBI**

Ingénieur R&D

Encadrant académique : **Madame Olfa MOURALI**

Maître Assistante

Réalisé au sein de SOFRECOM



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

Encadrant professionnel, **Monsieur Aymen BEL ARBI**

Signature et cachet

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

Encadrant académique, **Madame Olfa MOURALI**

Signature

Dédicaces

A ceux qui ont été mon meilleur exemple dans ma vie, ceux qui m'ont beaucoup donné sans rien attendre en retour, mon cher père et ma chère mère. Que ce projet soit le témoignage de ma gratitude et ma profonde reconnaissance pour l'amour, les sacrifices, le dévouement et le courage qu'ils m'ont prodigués,

A mes chers deux frères Wassim et Slim, je vous souhaite un avenir radieux, plein de succès et de bonheur,

A Tous ceux qui me sont chers qu'ils trouvent dans ce travail l'expression de mes sentiments les plus affectueux.

« Nulle dédicace ne peut exprimer ce que je vous dois »

Sami GHORBEL

Remerciements

Au terme de ce travail, j'exprime mes vifs remerciements à toute personne ayant contribué, de près ou de loin, à la réalisation de ce travail.

Ma profonde gratitude s'adresse à la société Sofrecom Tunisie qui m'a bien accueilli et à mon manager Monsieur Khaled KSONTINI pour son engagement permanent, son soutien constant et la confiance totale qu'il m'a accordée.

Je voudrais exprimer, ma profonde reconnaissance à mon encadrant de la société «SOFRECOM» Monsieur Aymen BEL ARBI pour l'effort et le soin accordés afin de réussir ce projet. Je le remercie infiniment pour ses conseils et son suivi tout au long de la durée de mon stage.

J'adresse aussi mes profonds remerciements à Madame Olfa MOURALI, pour son encadrement, pour son assistance, pour tout le temps précieux qu'elle m'a octroyé et pour ses conseils qui m'ont été bien utiles pour la rédaction de ce rapport.

Je tiens, par ailleurs, à exprimer mon immense reconnaissance à :, pour l'honneur qu'elle a bien voulu me faire en acceptant de présider le jury chargé d'évaluer ce travail. Et Monsieur ... , d'avoir accepté d'examiner ce travail.

J'espère que le présent projet soit à la hauteur de vos attentes.

Table des matières

Introduction générale	1
1 Contexte général	2
1.1 Présentation de l'organisme d'accueil : SOFRECOM	3
1.1.1 Groupe Sofrecom	3
1.1.2 Sofrecom Tunisie	3
1.2 Étude et analyse de l'existant	4
1.2.1 Analyse de l'existant	4
1.2.2 Critique de l'existant	4
1.3 Problématique	5
1.4 Solution proposée	5
1.5 Méthodologie de développement à adopter : SCRUM	5
1.5.1 Rôles dans SCRUM	6
1.5.2 Planification d'un projet par SCRUM	6
1.6 Diagramme de GANTT théorique	7
2 Analyse et spécification des besoins	9
2.1 Analyse des besoins	10
2.1.1 Identification des acteurs	10
2.1.2 Besoins fonctionnels	10
2.1.3 Besoins non fonctionnels	12
2.2 Diagramme des cas d'utilisation global	13
2.3 Backlog du produit	14
2.4 Planification des sprints	16
3 Initialisation du projet	18
3.1 Environnement de développement	19
3.1.1 Environnement matériel	19
3.1.2 Environnement logiciel	19
3.2 Architecture de l'application	22

3.2.1	Architecture physique de l'application	22
3.2.2	Architecture logicielle de l'application	22
3.3	Diagramme de déploiement	24
3.4	Architecture des composants Angular	24
4	Release 1 Prédiction	26
4.1	Sprint 1 Création d'un modèle de prédiction	27
4.1.1	Backlog du Sprint	27
4.1.2	Spécification fonctionnelle	29
4.1.3	Apprentissage Automatique	31
4.1.3.1	Apprentissage Automatique supervisé	32
4.1.3.2	Apprentissage Automatique non supervisé	32
4.1.4	Solutions existantes	33
4.1.4.1	Classification	34
4.1.4.2	Régression	40
4.1.5	Solution envisagée	43
4.1.5.1	Réseau de neurones	43
4.1.5.2	Préparation des données	43
4.1.5.3	Architecture de réseau de neurones	45
4.1.5.4	Évaluation du modèle	47
4.1.5.5	Réalisation	49
4.2	Sprint 2 Gestion des prédictions des employés	51
4.2.1	Backlog du Sprint	51
4.2.2	Spécification fonctionnelle	53
4.2.3	Conception	59
4.2.3.1	Diagramme de classes	59
4.2.3.2	Diagrammes de séquences	62
4.2.4	Réalisation	64
4.2.5	Test et validation	71
5	Release 2 Administration et visualisation	73
5.1	Sprint 3 Authentification et gestion des utilisateurs	74

5.1.1	Backlog du Sprint	74
5.1.2	Spécification fonctionnelle	75
5.1.3	Conception	79
5.1.3.1	Diagramme de classes	79
5.1.3.2	Diagrammes de séquences	81
5.1.4	Réalisation	84
5.1.4.1	Test et validation	87
5.2	Sprint 4 Visualisation du tableau de bord	89
5.2.1	Spécification fonctionnelle	89
5.2.1.1	Backlog du Sprint	89
5.2.1.2	Diagrammes de cas d'utilisations détaillés du sprint 4	89
5.2.2	Conception	89
5.2.2.1	Diagrammes de séquences	89
5.2.3	Réalisation	89
5.2.3.1	Test et validation	89
	Conclusion générale et Perspectives	90
	Bibliographie	91

Table des figures

1.1	Le Framework SCRUM	7
1.2	Diagramme de GANTT théorique	7
2.1	Diagramme de cas d'utilisation global	13
2.2	Planification des sprints	17
3.1	Évolution des bibliothèques Deep Learning	21
3.2	Architecture physique de l'application	22
3.3	Architecture logicielle de l'application	23
3.4	Diagramme de déploiement	24
3.5	Architecture des composants Angular	25
4.1	Diagramme de cas d'utilisation « Créer un modèle de prédiction »	29
4.2	Apprentissage Automatique supervisé	32
4.3	Structure de nos données	33
4.4	Matrice de confusion	34
4.5	Matrice de confusion de SVM	35
4.6	Précision du modèle SVM	35
4.7	Matrice de confusion d'arbre de décision	36
4.8	Précision du modèle d'arbre de décision	36
4.9	Matrice de confusion de forêt d'arbres décisionnels	37
4.10	Précision du modèle de forêt d'arbres décisionnels	37
4.11	Matrice de confusion de KNN	38
4.12	Précision du modèle de KNN	38
4.13	Matrice de confusion de Gradient Boosting Classifier	39
4.14	Précision du modèle de Gradient Boosting Classifier	39
4.15	Test sur le modèle de régression linéaire	40
4.16	Test sur le modèle de Gradient Boosting Regressor	41
4.17	Test sur le modèle de forêt d'arbres décisionnels de régression	41
4.18	Création du modèle GBM avec H2O Flow	42

Table des figures

4.19 Évaluation du GBM et importance de chaque variable explicative	42
4.20 Script de conversion des données	43
4.21 Exemple de conversion des données	44
4.22 données normalisées	44
4.23 données divisées	45
4.24 Architecture de notre réseau de neurones	46
4.25 Fonction d'activation « Relu »	47
4.26 Évaluation de notre modèle de réseau de neurones	47
4.27 Visualisation de la fonction d'erreur avec « Tensorboard »	48
4.28 Visualisation des informations réelles des employés	48
4.29 Visualisation des sorties prédites des employés	49
4.30 Interface d'import des données	50
4.31 Appel de service web de division des données	50
4.32 Résultat retourné de service web de prédiction	50
4.33 Diagramme de cas d'utilisation « Gérer les prédictions »	53
4.34 Diagramme de cas d'utilisation « Gérer les employés »	54
4.35 Diagramme de séquences système « Effectuer une prédiction »	57
4.36 Diagramme de séquences système « Modifier un employé »	58
4.37 Diagramme de séquences système « Supprimer un employé »	59
4.38 Diagramme de classes	61
4.39 Diagramme de séquences objet « Effectuer une prédiction »	62
4.40 Diagramme de séquences objet « Modifier un employé »	63
4.41 Diagramme de séquences objet « Supprimer un employé »	64
4.42 Interface de gestion des employés	65
4.43 Interface d'ajout d'un nouvel employé	65
4.44 Interface de recherche d'un employé	66
4.45 Interface de consultation des informations d'un employé	66
4.46 Interface de modification des informations d'un employé	67
4.47 Interface de confirmation de prédiction d'un employé	67
4.48 Interface de chargement de prédiction d'un employé	68
4.49 Interface de résultat de prédiction d'un employé	68

4.50 Interface de suppression d'un employé	69
4.51 Interface de succès de suppression d'un employé	69
4.52 Interface de consultation des employés prédits	70
4.53 Interface de consultation de l'historique des prédictions d'un employé	70
4.54 Interface d'export de l'historique des informations d'un employé sous format PDF	71
5.1 Diagramme de cas d'utilisation « Gérer les utilisateurs »	76
5.2 Diagramme de classes	80
5.3 Diagramme de séquences objet « s'authentifier »	81
5.4 Diagramme de séquences objet « Ajouter un compte utilisateur »	82
5.5 Diagramme de séquences objet « Modifier un compte utilisateur »	83
5.6 Diagramme de séquences objet « Supprimer un compte utilisateur »	84
5.7 Interface de l'authentification	85
5.8 Interface de gestion des comptes utilisateurs	85
5.9 Interface d'ajout d'un compte utilisateur	86
5.10 Interface de modification des informations d'un compte utilisateur	86
5.11 Interface de suppression d'un compte utilisateur	87
5.12 Interface de succès de suppression d'un compte utilisateur	87

Liste des tableaux

2.1 Backlog du produit	14
4.1 Backlog du Sprint 1	27
4.2 Description du diagramme de cas d'utilisation «Importer les données des employés» .	30
4.3 Description du diagramme de cas d'utilisation «Construire un modèle de prédition»	30
4.4 Description du diagramme de cas d'utilisation «Évaluer le modèle de prédition» . .	31
4.5 Description du diagramme de cas d'utilisation « Effectuer des prédictions » . . .	31
4.6 Backlog du Sprint 2	51
4.7 Description du diagramme de cas d'utilisation «Effectuer une prédition»	54
4.8 Description du diagramme de cas d'utilisation «Exporter les informations d'un employé»	55
4.9 Description du diagramme de cas d'utilisation «Ajouter un employé»	56
4.10 Description du diagramme de cas d'utilisation «Supprimer un employé»	56
4.11 Tests fonctionnels du sprint 2	71
5.1 Backlog du Sprint 3	74
5.2 Description du diagramme de cas d'utilisation «s'authentifier»	76
5.3 Description du diagramme de cas d'utilisation «Consulter la liste des utilisateurs» .	77
5.4 Description du diagramme de cas d'utilisation «Ajouter un compte utilisateur» . .	77
5.5 Description du diagramme de cas d'utilisation «Supprimer un compte utilisateur» .	78
5.6 Tests fonctionnels du sprint 3	88

Liste des abréviations

- **API** = Application Programming Interface
- **DSI** = Direction des Systèmes d'Information
- **OLS** = Orange Labs Services
- **REST** = Representational state transfer

Introduction générale

CONTEXTE GÉNÉRAL

Plan

1	Présentation de l'organisme d'accueil : SOFRECOM	3
2	Étude et analyse de l'existant	4
3	Problématique	5
4	Solution proposée	5
5	Méthodologie de développement à adopter : SCRUM	5
6	Diagramme de GANTT théorique	7

Introduction

Ce chapitre constitue le point de départ du rapport de stage de fin d'études. Nous allons commencer tout d'abord par présenter l'organisme d'accueil SOFRECOM et sa filiale SOFRECOM Tunisie. Nous allons énoncer par la suite la problématique et les objectifs de notre projet et enfin nous allons spécifier la méthodologie et la planification de travail pour sa réalisation.

1.1 Présentation de l'organisme d'accueil : SOFRECOM

Nous allons présenter dans cette section notre organisme d'accueil.

1.1.1 Groupe Sofrecom

Sofrecom, filiale d'Orange, développe depuis 50 ans un savoir-faire unique dans les métiers de l'opérateur, ce qui en fait un leader mondial du conseil et de l'ingénierie télécom. Ces dernières années, plus de 200 acteurs majeurs, dans plus de 100 pays, ont confié à Sofrecom la conduite de leurs projets stratégiques et opérationnels. Le réseau de savoir-faire de Sofrecom (Know-How Network) garantit également le transfert de connaissances et de compétences pour une évolution continue basée sur des méthodologies certifiées internationalement [1].

1.1.2 Sofrecom Tunisie

Sofrecom Tunisie lancée en Octobre 2012, est la plus jeune et la plus importante filiale du groupe Sofrecom en zone Afrique et Moyen Orient. En 5 ans, elle a pu se positionner en tant qu'un leader d'ingénierie en télécommunications. Sofrecom Tunisie compte aujourd'hui 500 experts, et deux clients majeurs qui font partie du groupe Orange : Orange Labs Services (OLS) et Direction des Systèmes d'Information (DSI) France. Sofrecom Tunisie propose à ses clients une gamme riche de prestations organisées autour de huit métiers [2] :

- Ingénierie ;
- Architecture ;
- Support et maintenance ;
- Sécurité informatique.
- Expertise technique.
- Développement ;

- Innovation ;
- Consulting.

Notre projet concerne le métier du développement.

1.2 Étude et analyse de l'existant

Dans cette section, nous allons effectuer l'analyse et le critique de l'existant.

1.2.1 Analyse de l'existant

La direction resources humaine de Sofrecom utilise une application web développée en 2017 pour l'aider à prendre de la décision concernant la démission de ses employés dans le futur. Cette application est basée sur un système d'information open source H2o.ai permettant la création des modèles d'apprentissage automatique.

Pour construire un modèle de prédiction de "Turnover" de Sofrecom, l'utilisateur de ressource humaine collecte d'abord des données dans un fichier d'extension ".csv" qui représente les caractéristiques des employés (nom, age, matricule, expérience, école, manager, département...). Après cette phase, l'utilisateur importe ce fichier dans le système d'information H2o.ai. Enfin, il effectue plusieurs tâches sur ce fichier telles que : analyse de fichier, choix des arguments de modèle à construire, choix de variable à prédire (qui est la démission dans notre cas) et sélection d'algorithme pour la construction de modèle afin d'effectuer des prédictions.

1.2.2 Critique de l'existant

L'application développée contient plusieurs contraintes :

- Téléchargement et activation d'un module dans le navigateur permettant à l'application d'accéder à les services offertes par le système H2o.ai ;
- Lenteur dans la gestion des prédictions de la démission des employés ;
- Utilisation d'une solution générique de prédiction ;
- Lors de la fermeture de système d'information H2o.ai, il n'y aura pas de trace sur les modèles créés ainsi que l'historique de prédiction ;
- Modèle de prédiction n'est pas performant.

Cette manière de procéder a un impact négatif sur le traitement de l'information des employés pouvant engendrer des pertes de temps considérables et de grands risques de perte des résultats de prédiction d'où la nécessité d'une application qui a pour but d'optimiser la gestion, le traitement et l'organisation de ces informations.

1.3 Problématique

Sofrecom utilise une application basée sur la plateforme de l'apprentissage automatique H2o.ai. Elle utilise cette application pour traiter leurs propres informations afin de faire des prédictions pour la démission de ses employés à l'avenir.

Le modèle de prédiction développé par H2o.ai ne permet pas de répondre aux nouvelles données mise à jour de Sofrecom et même pour accéder aux services de H2o.ai, il faudra installer un module dans le navigateur pour chaque utilisateur.

Face aux difficultés rencontrées, une décision a été prise pour le développement d'une solution propre et spécifique à la société Sofrecom permettant de remédier à ces problèmes afin d'optimiser ces différentes tâches pour faciliter le traitement de prédiction et construire un modèle de prédiction de turnover performant.

1.4 Solution proposée

La solution proposée consiste à développer et implémenter une application propre et spécifique à la société Sofrecom basée sur un modèle d'apprentissage automatique de TURNOVER performant. L'application propose une visualisation des différentes informations des employés de Sofrecom et permet d'effectuer des prédictions en une seule action sans avoir recours aux différentes autres activités (import de fichier, analyse, choix de variable à prédire...), elle permet de garder l'historique de cette dernière et d'exposer un tableau de bord dynamique.

1.5 Méthodologie de développement à adopter : SCRUM

Dans tous les secteurs d'activités, la seule manière possible pour arriver à atteindre les objectifs du travail à réaliser est de suivre une démarche bien précise sur laquelle vont reposer toutes les étapes du projet en cours.

Afin d'atteindre nos objectifs et de développer notre application informatique le plus rapidement possible, notre choix a été fixé sur la méthodologie SCRUM. En effet, il s'agit de la meilleure

méthodologie itérative qui peut être adaptée dans le cadre de notre projet vue qu'elle a été conçue pour améliorer la productivité dans les équipes auparavant paralysées par des méthodologies classiques.

1.5.1 Rôles dans SCRUM

Le "Product owner" est la personne qui porte la vision du produit à réaliser, responsable de la gestion du "product backlog" et travaille en interaction avec l'équipe de développement. Il s'agit généralement d'un expert du domaine métier du projet. M. Khaled Ksontini est le "Product owner". Le "Scrum master" est la personne qui doit maîtriser SCRUM et s'assurer que ce dernier est bien compris et appliqué. M. Aymen BEL ARBI est le "Scrum master". L'équipe transforme les besoins en fonctionnalités pour aboutir à un incrément utilisable et livrable à la fin de chaque itération. L'équipe comporte un étudiant en Génie Logiciel et Systèmes d'Information à l'ISI, qui est le responsable de développement de l'application.

Avantages du SCRUM :

- Flexibilité ;
- Créativité ;
- Productivité.

1.5.2 Planification d'un projet par SCRUM

Pour appliquer correctement SCRUM, il faut comprendre le cycle de vie d'un sprint pendant un processus SCRUM. Le processus, illustré dans la figure 1.1, est décrit ci-dessous :

- le Product owner crée le "product backlog" en déterminant et priorisant les user Story ;
- Pendant la planification du sprint, l'équipe choisit un ensemble de "user story" les plus prioritaires à partir du "product backlog" pour construire le sprint Backlog ;
- L'équipe implémente les "user story" pendant une période qui dure de 2 à 4 semaines ;
- Durant le sprint, l'équipe se réunit chaque jour, "Daily Scrum", pour synchroniser les tâches..
- A la fin du sprint, le travail doit être achevé pour faire une démonstration au client ;
- Le sprint est clôturé par un "sprint review" pour discuter les prochaines étapes du projet et par un "sprint retrospective" pour parler des manières à appliquer pour rendre l'équipe plus productive [3].

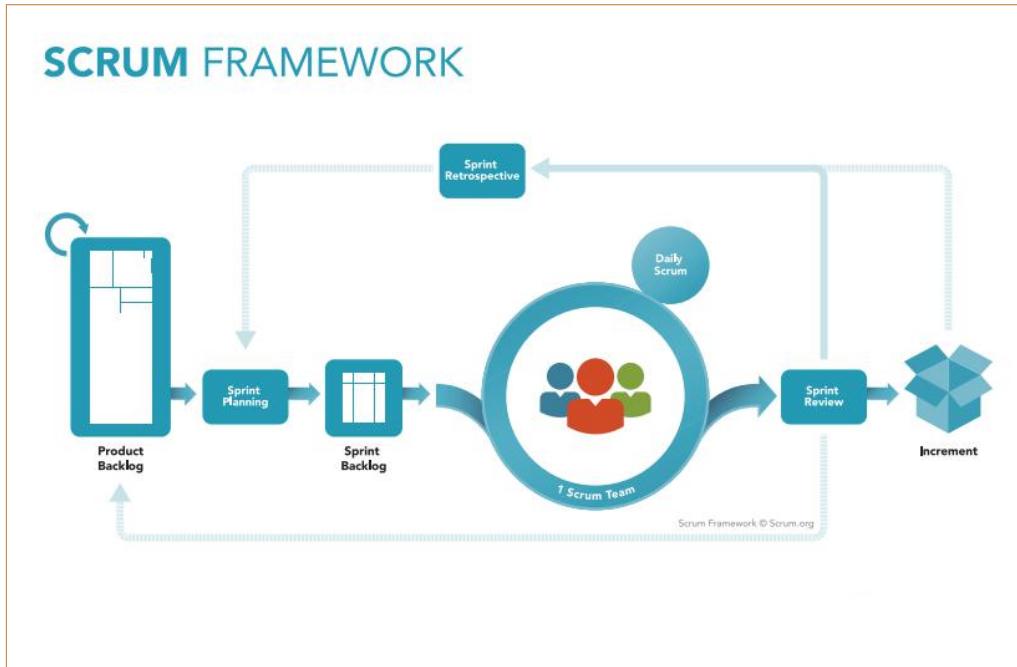


Figure 1.1: Le Framework SCRUM

1.6 Diagramme de GANTT théorique

Le diagramme de la figure ci-après décrit le déroulement chronologique des étapes du stage de fin d'études. Pendant le premier mois, les tâches préliminaires comme la documentation et l'étude théorique du projet ont été effectuées. Ces travaux ont été suivis par l'étude des besoins, qui a abouti à la spécification des besoins et à l'élaboration des diagrammes de cas d'utilisation. Ensuite, la conception et la réalisation de l'application a été mise en oeuvre et ce en parallèle avec son développement pour chaque release. Enfin, le travail réalisé (produit développé) a été testé dans chaque sprint .

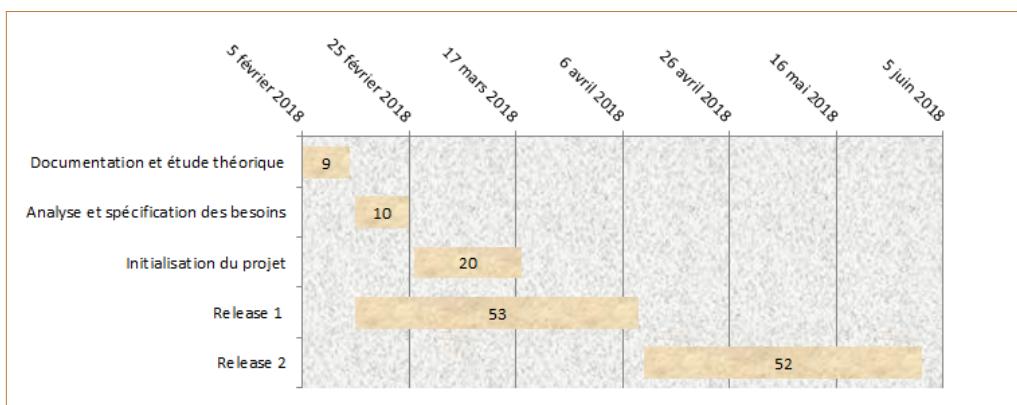


Figure 1.2: Diagramme de GANTT théorique

Conclusion

Dans ce chapitre, nous avons introduit l'organisme d'accueil Sofrecom. De plus, Nous avons fait l'étude de l'existant et la solution proposée ainsi que la méthodologie adoptée. Le chapitre suivant sera consacré pour l'analyse et la spécification des besoins.

ANALYSE ET SPÉCIFICATION DES BESOINS

Plan

1	Analyse des besoins	10
2	Diagramme des cas d'utilisation global	13
3	Backlog du produit	14
4	Planification des sprints	16

Introduction

Dans ce chapitre, nous allons identifier les besoins fonctionnels et les besoins non fonctionnels de notre système ainsi que nous allons présenter le diagramme de cas d'utilisation global en utilisant UML. Nous allons établir ensuite le Backlog Produit et la planification des sprints.

2.1 Analyse des besoins

Dans cette section, nous allons identifier les acteurs de notre système, exposer les besoins fonctionnels et les besoins non fonctionnels de notre application.

2.1.1 Identification des acteurs

Un acteur est une entité externe au système. Il représente une personne ou un autre système informatique qui attend un ou plusieurs services offerts par une interface d'accès. Il interagit avec le système par envoi ou réception des messages [4].

Les acteurs qui interagissent avec l'application sont de deux types :

- Administrateur : c'est la personne possédant le privilège de plus haut niveau et il a l'accès à toutes les fonctionnalités du notre application. Il permet aussi de gérer les différents utilisateurs.
- Utilisateur de ressource humaine : chaque utilisateur de ressource humaine a le droit de gérer les informations des employés de Sofrecom, d'effectuer des prédictions et de consulter le tableau de bord.

2.1.2 Besoins fonctionnels

Il s'agit principalement des fonctionnalités offertes par l'application aux utilisateurs. Ces besoins expriment une action devant être effectuée par l'application en réponse à une demande. Cette application offre les fonctionnalités suivantes :

- Créer un modèle de prédition ;
- S'authentifier ;
- Gérer les employés ;
- Importer les données des employés ;
- Gérer les prédictions ;

- Consulter l'historique des prédictions effectuées ;
- Consulter le tableau de bord ;

Nous allons dans ce stade détailler le contenu des fonctionnalités importantes de notre système.

- Créer un modèle de prédiction :

Le processus de création d'un modèle de prédiction implique de fournir un algorithme d'apprentissage qui permet de trouver des dépendances et des patterns entre des données nommées données d'entraînement.

Les données d'entraînement doivent contenir la bonne réponse, connue sous le nom de cible ou d'attribut cible. L'algorithme d'apprentissage trouve des règles dans les données d'entraînement qui mettent en correspondance les attributs des données d'entrée avec la cible (la variable de prédiction) et il produit un modèle qui capture ces patterns.

Nos données concernent les informations des employés de Sofrecom (matricule, civilité, âge, expérience de travail avant sofrecom, expérience de travail dans sofrecom, expérience de travail totale, poste, pôle, manager, situation familiale, seniorité) et la cible concerne la démission des employés (« 1 » : l'employé va démissionner, « 0 » : l'employé va rester dans Sofrecom).

Nous allons créer un modèle pour obtenir des prédictions sur de nouvelles données pour lesquelles nous ne connaissons pas la cible. Notre modèle devra être capable d'effectuer des prédictions sur la sortie des employés.

- Gérer les prédictions :

Après la création du modèle, l'utilisateur peut effectuer une prédiction sur la démission d'un employé existant ou un nouveau employé. Il peut aussi consulter et supprimer des prédictions effectuées.

- Gérer les employés :

L'utilisateur peut gérer les informations concernant les employés (civilité, âge, expérience de travail dans sofrecom, poste, pôle, manager...). L'utilisateur peut ajouter un nouvel employé, modifier les informations concernant un employé, supprimer ou rechercher un employé et consulter la liste des employés.

— Consulter le tableau de bord :

Un tableau de bord est un échantillon réduit d'indicateurs permettant à l'utilisateur de suivre l'évolution des résultats et les écarts par rapport à des valeurs de référence. Il permet, aussi, d'évaluer la performance, de réaliser un diagnostic de la situation et de progresser de façon continue.

Le tableau de bord de notre application offre plusieurs indicateurs au niveau de suivi des employés :

- Classés selon (civilité, manager, situation familiale...) ;
- Nombre de prédition par date ;
- Estimation de nombre de démissions par date.

2.1.3 Besoins non fonctionnels

Ces besoins correspondent aux caractéristiques de l'application, ils permettent d'identifier les contraintes internes et externes du domaine étudié. L'application a les exigences non fonctionnelles suivantes :

a) Sécurité

La sécurité permet de définir les niveaux d'accès possibles à l'application. Un utilisateur non authentifié ne possède pas le droit d'accéder à l'application. Seuls les utilisateurs qui ont été créés par l'administrateur ont les droits d'accéder.

b) Ergonomie

L'application doit offrir une interface conviviale et facile à utiliser.

c) Portabilité et compatibilité

L'application doit être portable sur tous les environnements logiciels (Windows, Mac OS, Linux...) et compatible avec tous les navigateurs.

d) Maintenabilité

L'architecture doit permettre l'évolution et assurer l'extensibilité de l'application.

d) Performance

Les performances d'exécution de l'application sont, généralement, décrits en termes de (temps de réponse, espace mémoire nécessaire, nombre d'utilisateurs...). Notre application doit avoir un temps minimal pour rendre une vue dans le navigateur.

2.2 Diagramme des cas d'utilisation global

Le cas d'utilisation décrit l'action offerte par l'application en réponse à la demande d'un utilisateur. Le diagramme de cas d'utilisation représente les fonctionnalités définies et les interactions entre les utilisateurs et l'application [5].

Le diagramme de cas d'utilisation ci-après expose les différents besoins généraux, qui seront détaillés par la suite.

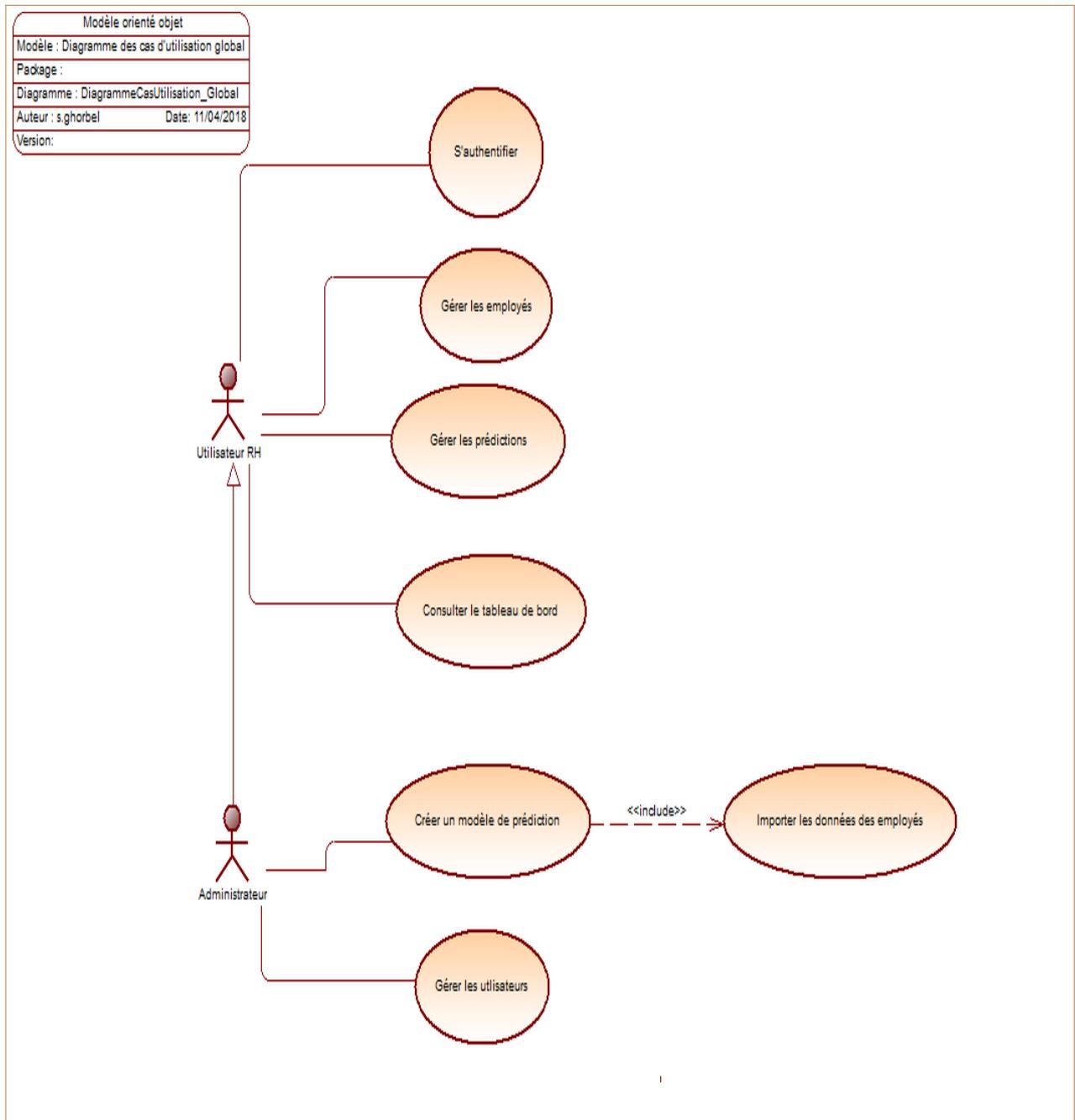


Figure 2.1: Diagramme de cas d'utilisation global

2.3 Backlog du produit

Le tableau 2.1 représente le backlog du produit. Ce tableau est une liste ordonnée qui fournit les éléments suivants , qui sont traduits dans celui-ci par des "user story" dont :

- ID représente l'identifiant du user story ;
- Scénario ou User Story : décrit un comportement du produit visible pour les utilisateurs et qui leur apporte de la valeur ou de l'utilité ;
- Priorité de user story est affectée selon la valeur métier et l'ordre de réalisation, où le chiffre « 1 » désigne la priorité la plus élevée, « 2 » désigne la priorité élevée, « 3 » désigne la priorité normale ;
- Pour estimer la complexité de nos user story, nous les avons notées par A pour normal, B pour difficile, C pour dur et D pour très dur.

Tableau 2.1: Backlog du produit

ID	Fonctionnalité	Scénario ou User Story	Priorité	Complexité
1	Création du modèle de prédiction.	En tant qu'administrateur, je veux importer un fichier contenant toutes les informations des employés.	1	C
2	Création du modèle de prédiction.	En tant qu'administrateur, je veux préparer mes données.	1	D
3	Création du modèle de prédiction.	En tant qu'administrateur, je veux construire un modèle de prédiction.	1	D
4	Création du modèle de prédiction.	En tant qu'administrateur, je souhaite évaluer mon modèle de prédiction	1	D
5	Création du modèle de prédiction.	En tant qu'utilisateur, je veux effectuer des prédition sur la démission des employés.	1	D
6	Gestion des prédictions.	En tant qu'utilisateur, je veux effectuer une prédition sur la démission d'un employé.	1	D

7	Gestion des prédictions.	En tant qu'utilisateur, je souhaite consulter la liste des employés dont j'ai effectué des prédictions sur eux.	1	B
8	Gestion des prédictions.	En tant qu'utilisateur, je souhaite consulter l'historique de prédictions pour chaque employé et exporter toutes ses informations sous format PDF.	2	C
9	Gestion des prédictions.	En tant qu'utilisateur, je souhaite rechercher la liste des employés dont j'ai effectué des prédictions sur eux.	2	B
10	Gestion des employés.	En tant qu'utilisateur, je veux ajouter un employé.	2	B
11	Gestion des employés.	En tant qu'utilisateur, je veux modifier un employé.	2	B
12	Gestion des employés.	En tant qu'utilisateur, je veux supprimer un employé.	2	A
13	Gestion des employés.	En tant qu'utilisateur, je souhaite consulter et effectuer une recherche sur la liste des employés.	2	C
14	Gestion des employés.	En tant qu'utilisateur, je souhaite exporter toutes les informations d'un employé sous format PDF.	2	C
15	Authentification.	En tant qu'utilisateur, je dois m'authentifier pour accéder à l'application	3	D
16	Gestion des utilisateurs.	En tant qu'administrateur, je veux ajouter un utilisateur.	3	B
17	Gestion des utilisateurs.	En tant qu'administrateur, je veux modifier un utilisateur	3	B

18	Gestion des utilisateurs.	En tant qu'administrateur, je veux supprimer un utilisateur.	3	A
19	Gestion des utilisateurs.	En tant qu'administrateur, je souhaite consulter et effectuer une recherche sur la liste des utilisateurs.	3	C
20	Gestion des utilisateurs.	En tant qu'administrateur, je souhaite attribuer un rôle à un utilisateur.	3	C
21	Visualisation du tableau de bord.	En tant qu'utilisateur, je souhaite consulter le nombre des employés classé par manager, seniorité, civilité et situation familiale.	3	B
22	Visualisation du tableau de bord.	En tant qu'utilisateur, je souhaite consulter le nombre des prédictions effectuées par date.	3	D
23	Visualisation du tableau de bord.	En tant qu'utilisateur, je souhaite consulter des statistiques sur les prédictions d'un employé.	3	D

2.4 Planification des sprints

Une fois que le backlog de produit est terminé, nous avons établi la réunion de planification. Le but de cette réunion est de préparer le plan de travail et d'identifier le backlog des sprints.

Pour notre projet, nous avons choisi de développer deux releases. Chaque release est composée de deux sprints avec une durée au maximum de deux mois.

La figure 2.2 illustre notre plan de travail.

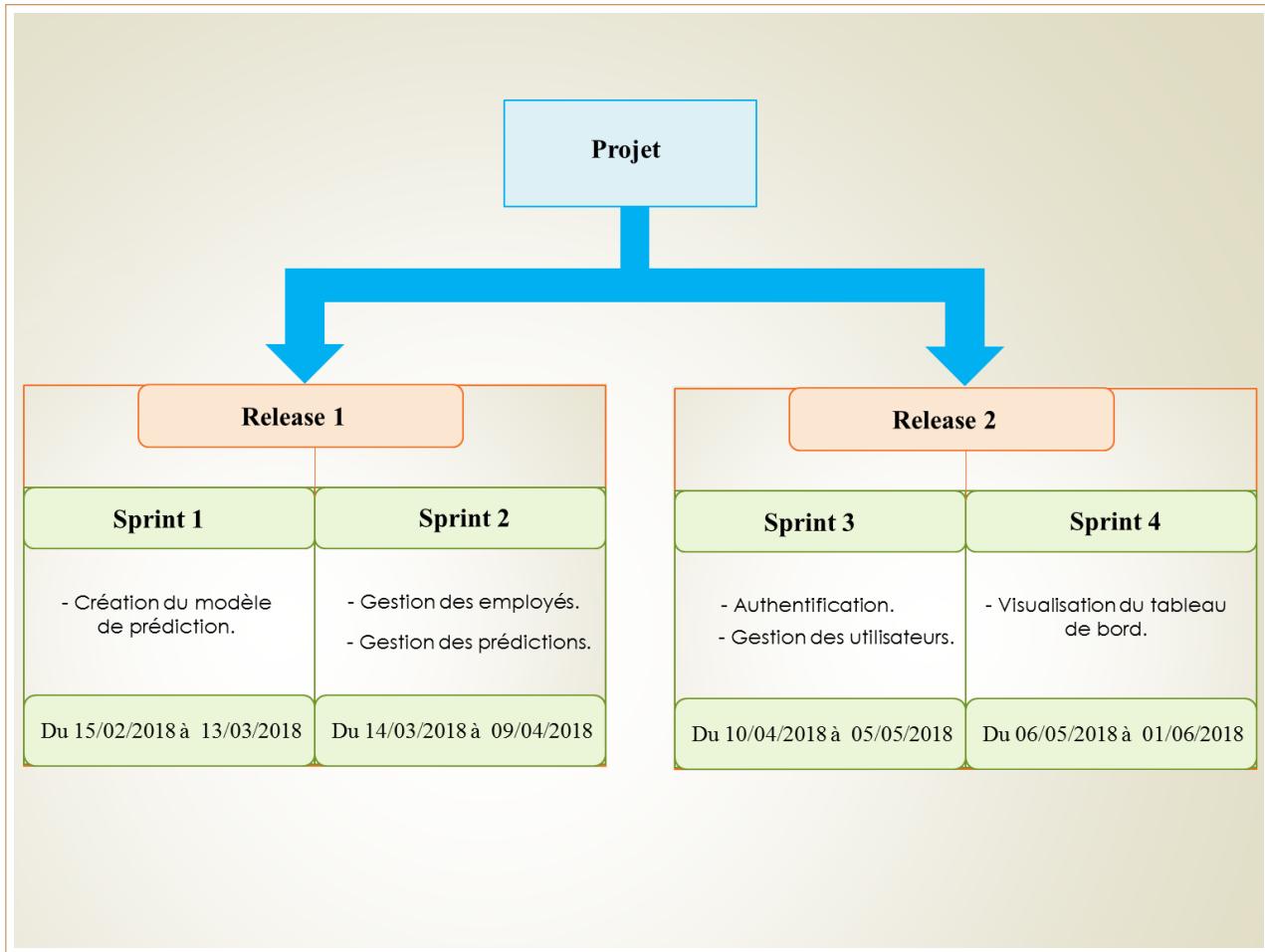


Figure 2.2: Planification des sprints

Conclusion

À travers ce chapitre, les différents besoins fonctionnels et non fonctionnels du projet ont été spécifiés.

Ils étaient, également, modélisés, en élaborant les diagrammes de cas d'utilisation correspondants par recours au langage de modélisation UML.

Le chapitre suivant, intitulé initialisation du projet, sera consacré à la définition de notre environnement de développement et les technologies utilisées.

INITIALISATION DU PROJET

Plan

1	Environnement de développement	19
2	Architecture de l'application	22
3	Diagramme de déploiement	24
4	Architecture des composants Angular	24

Introduction

Dans ce chapitre, la première étape de Scrum qui est l'initialisation du projet sera traitée. C'est en fait à propos de la phase de définition de l'architecture du produit à produire et la préparation de l'environnement de développement. Cette phase ne se termine pas par une livraison.

3.1 Environnement de développement

Dans cette section, nous allons présenter les outils et les technologies utilisés pour la conception et la réalisation de notre application.

3.1.1 Environnement matériel

Pour élaborer ce projet, un micro-ordinateur portable DELL E5540 fourni par la société a été utilisé. Il est exploité sous le système d'exploitation Windows 7 et il possède les caractéristiques suivantes :

- Processeur Intel Core i5-4210 U, 2.40 GHz ;
- Mémoire vive «RAM» d'une capacité de 8 GO ;
- Disque dur d'une capacité de 1 TO ;
- Ecran de 15,6 pouces.

3.1.2 Environnement logiciel

Outil de modélisation UML : PowerAMC

L'outil «PowerAMC» offre plusieurs méthodes de modélisation, accessibles aux informaticiens de tout niveau, parmi elles : Merise, UML, Data Warehouse, et processus métiers [6].

Cet outil est retenu pour la représentation des schémas des différents diagrammes (d'utilisation, de classes et de séquences).

Environnement de développement intégré (IDE) : IntelliJ IDEA

IntelliJ IDEA est un environnement de développement intégré (IDE) largement destiné à des langages comme Java, Scala, Angular... [7].

Nous avons utilisé cet environnement pour le développement de notre application avec Angular et Node.js.

Postman

c'est un outil fourni par Google permet d'appeler et tester une API Web.

Node.js

Node.js est une plate-forme pour la construction des applications serveurs rapides et évolutives en JavaScript [8].

Notre choix a été basé sur Node.js vu que notre application est une application web dynamique qui demande de la performance et de la portabilité.

HTML 5 / CSS 3

HTML est considéré comme un langage de balisage qui a pour but d'organiser et de structurer des pages web. CSS est le complémentaire de HTML, permet de définir des styles de ces pages [9].

Les versions 3 et 5 ont été choisies et utilisées respectivement pour CSS et HTML dans le but de créer des interfaces graphiques pour notre application.

Javascript

JavaScript est le langage de programmation du HTML et du Web. Il est utilisé dans les pages web interactives [10].

Bootstrap

Bootstrap est le framework HTML, CSS et JavaScript le plus populaire pour le développement des sites web réactifs et mobiles [11].

La version 4 de cette technologie a été choisie et qui a été utilisée pour améliorer et fournir une interface conviviale.

Angular 4

Angular 4 est un framework javascript qui facilite la création d'applications sur le web. Il combine les modèles déclaratifs, l'injection de dépendance et les meilleures pratiques intégrées pour résoudre les problèmes de développement [12].

MongoDb

MongoDB stocke des données dans des documents flexibles de type JSON, ce qui signifie que les champs peuvent varier d'un document à un autre et que la structure des données peut être modifiée au fil du temps [13].

La version 3 de Mongodb a été choisie dans l'objectif de gérer de grandes quantités de données.

Python

Python est un langage de programmation interprété, interactif, orienté objet et de haut niveau.

Nous avons réalisé notre modèle de prédiction avec Python et précisément avec Tensorflow [14].

TensorFlow

TensorFlow est un framework de programmation pour le calcul numérique qui a été rendu open source par Google en Novembre 2015. Depuis sa sortie, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des frameworks les plus utilisés pour le Machine learning et Deep learning, comme le montrent les dernières comparaisons suivantes, faites par François Chollet (auteur de la librairie Keras).

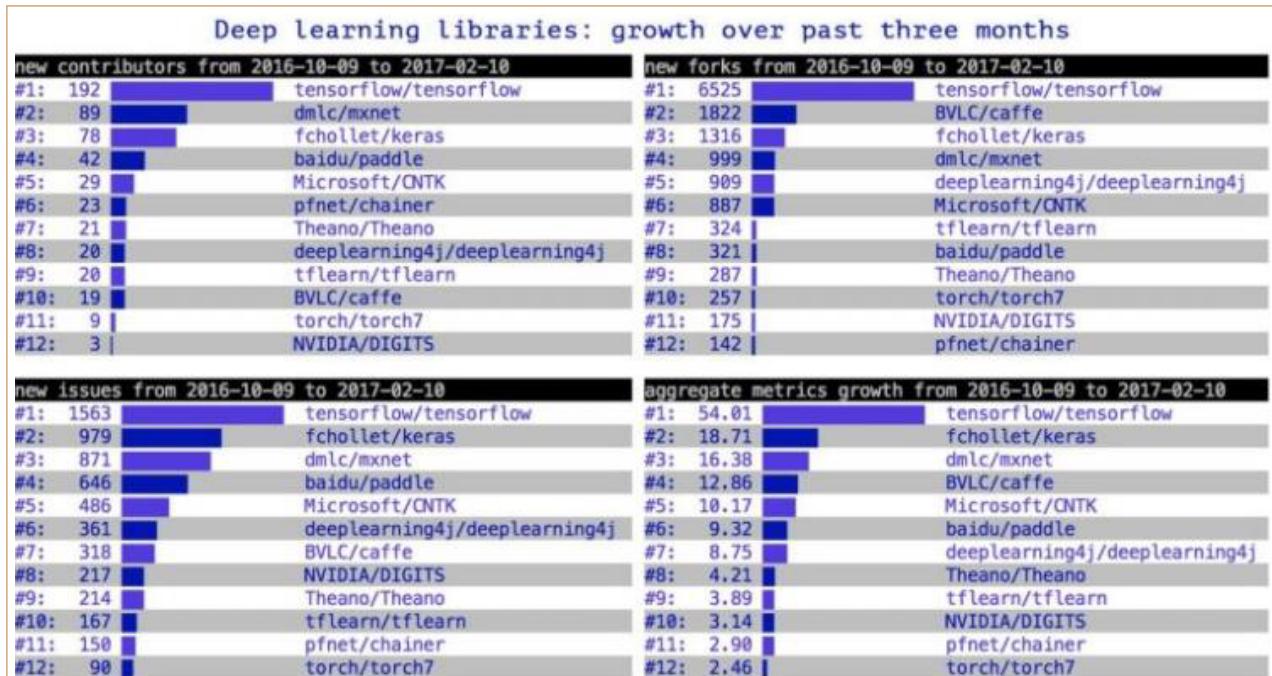


Figure 3.1: Évolution des bibliothèques Deep Learning

Parmi les avantages de Tensorflow :

- Compatibilité avec multi-plateformes (Linux, Mac OS, et même Android et iOS) ;
- Exposition des APIs en Python, C++, Java et Go ;
- Temps de compilation très courts ;
- Prise en charge des calculs sur CPU et GPU ;
- Documentation extrêmement bien fournie avec de nombreux exemples et tutoriels.

Alors qu'il a été développé pour optimiser les calculs numériques complexes, Tensorflow est aujourd'hui particulièrement plus utilisé pour le deep learning (réseaux de neurones) [15].

Flask :

Nous avons utilisé Flask pour exposer les différentes phases (import des données, préparation des données, création du modèle de prédiction....) sous format des services web (API REST) dans le but d'interagir avec Node.js via un format JSON.

3.2 Architecture de l'application

Dans cette section, nous allons présenter l'architecture physique et logicielle de notre application.

3.2.1 Architecture physique de l'application

Pour la réalisation de nos systèmes, nous avons adopté l'architecture 3-tiers. Ce modèle d'architecture est décomposé en 3 parties logiques distinctes, chacune ayant un rôle bien défini :

- Le client qui correspond dans notre cas à un navigateur dans l'ordinateur ou le téléphone d'un utilisateur. Son rôle est d'afficher les données et de permettre à l'utilisateur final d'interagir avec elles ;
- Le serveur web reçoit une demande du client, la traite et fait appel au serveur de base de données, pour servir le client ;
- Le serveur de données, fournissant au serveur web les données dont il a besoin.

La figure 3.2 représente l'architecture physique de l'application.

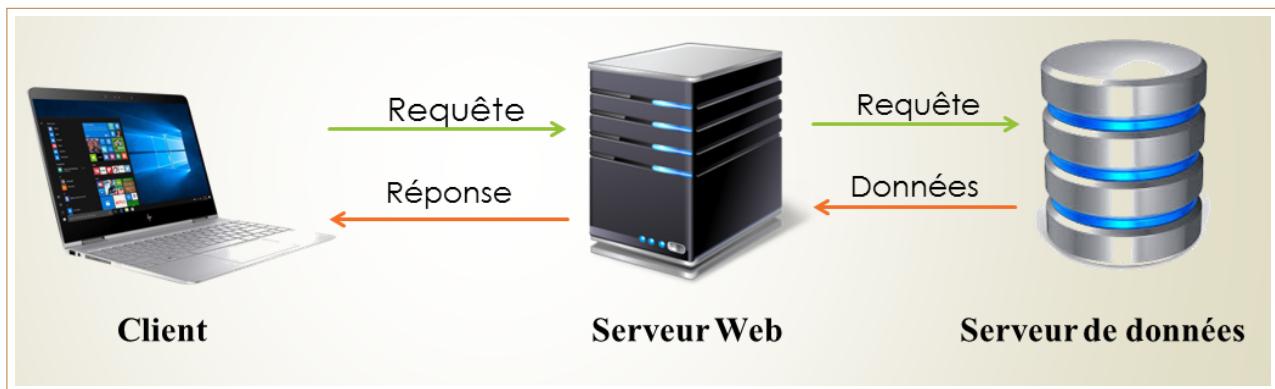


Figure 3.2: Architecture physique de l'application

3.2.2 Architecture logicielle de l'application

La définition de l'architecture de l'application est une phase primordiale dans le processus de conception. Elle dépend d'un nombre de paramètres dont les besoins en matière de la performance, les perspectives, l'évolutivité, la modularité et l'extensibilité.

La figure 3.3 illustre une représentation de l'architecture de notre application.

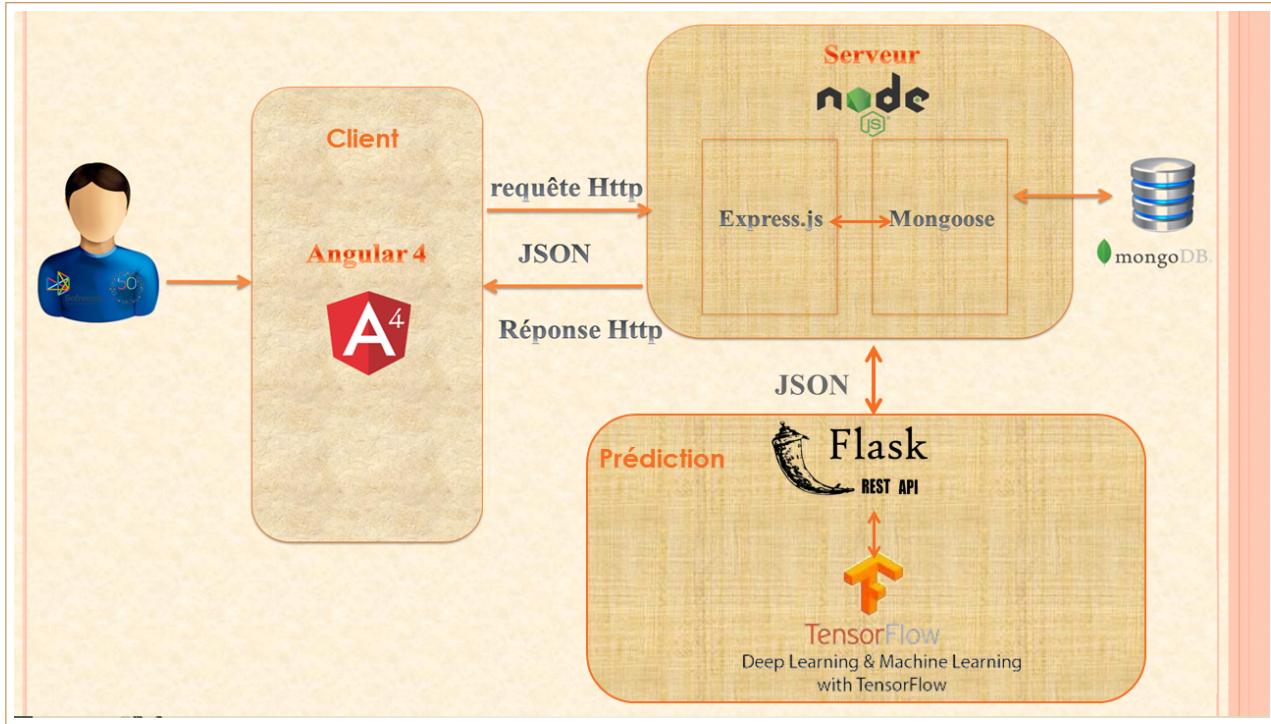


Figure 3.3: Architecture logicielle de l'application

Pour mieux comprendre cette architecture, nous allons procéder à l'explication des différentes parties :

- Au niveau du client, nous avons adopté le framework Angular version 4 qui nous permet de réaliser une application web monopage (Single Page Application). Cette dernière assure la fluidité de l'expérience utilisateur et évite le chargement d'une nouvelle page à chaque action demandée ;
- Sur la partie serveur, nous avons opté pour le framework léger Express.js (basé sur node.js) pour recevoir et interpréter les requêtes entrantes du côté client et interagir avec la base de données Nosql MongoDB à travers le framework de node.js Mongoose. Mongoose est un Mappeur de document d'objet (Object Document Mapper ou ODM) qui permet de définir des objets avec un schéma fortement typé qui est mappé à un document MongoDB ;
- Les différentes phases de préparation des données, de construction du modèle de prédiction et d'évaluation du service de prédiction sont réalisées avec Tensorflow (basé sur python). Elles sont exposées au format des services web (API REST) pour interagir avec le serveur node.js à travers le framework Express.js.

3.3 Diagramme de déploiement

Ce diagramme décrit le déploiement de notre architecture logicielle. En effet, il permet de décrire la disposition physique des ressources matérielles qui constituent notre système et de montrer la distribution des composants sur ces matériels. Le diagramme de déploiement modélise les ressources matérielles par des nœuds, et spécifie les liaisons entre eux [16].

La figure 3.4 présente le diagramme de déploiement de notre application.

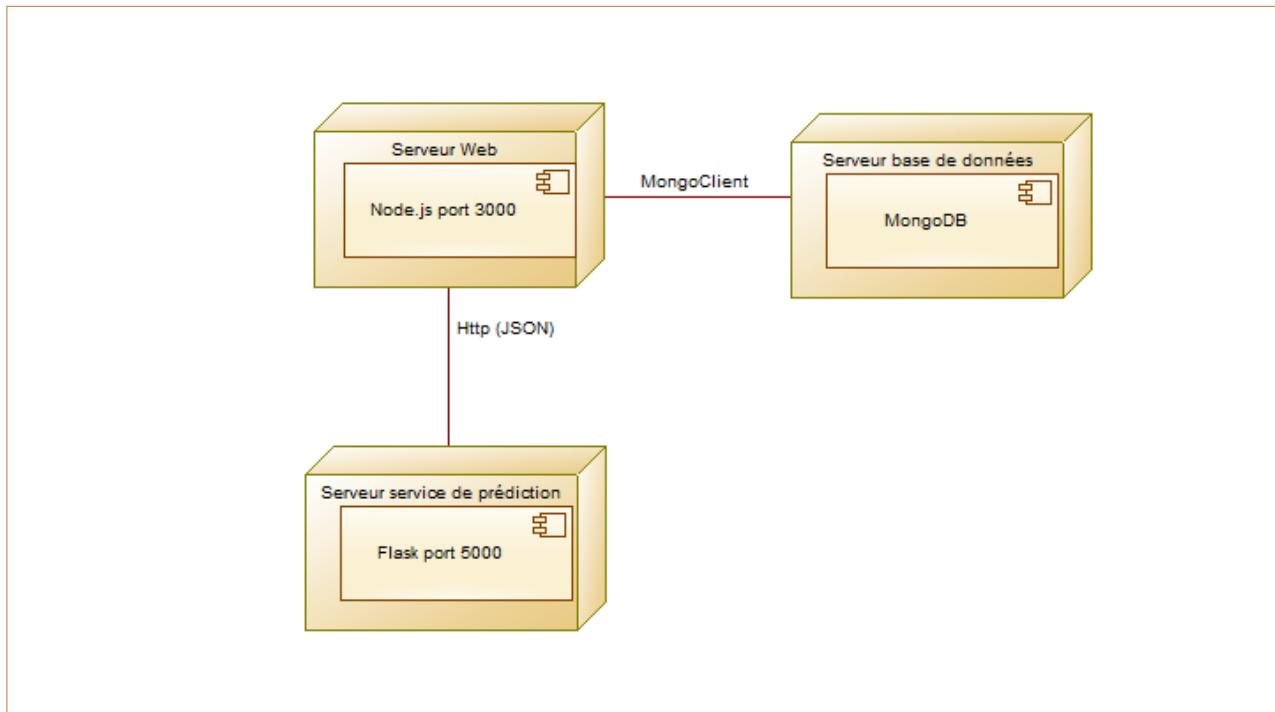


Figure 3.4: Diagramme de déploiement

3.4 Architecture des composants Angular

Nous allons nous intéresser à la partie front de notre application développée avec le framework Angular version 4. En effet, Angular 4 est orienté composant et chaque composant est défini comme étant un élément réutilisable [17].

La figure 3.5 illustre l'architecture des composants de notre partie front où nous distinguons, en rose les fichiers HTML, en orange les fichiers TypeScript, en vert les fichiers CSS et en bleu le paquetage ou composant qui englobe HTML, CSS et TypeScript. Les flèches désignent une relation d'utilisation, par exemple Root Component utilise SideBar Component qui utilise lui-même Utilisateur Component.

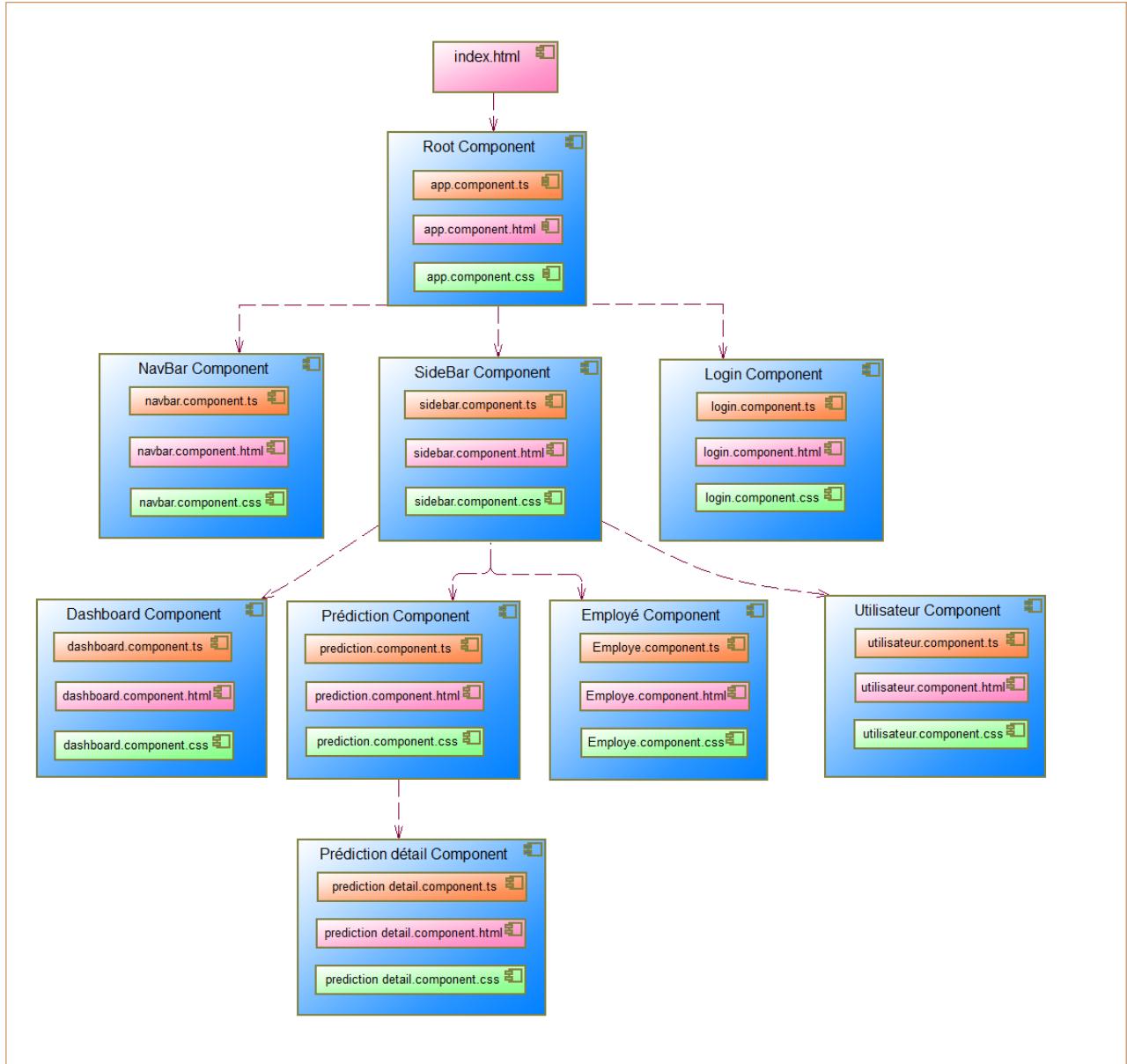


Figure 3.5: Architecture des composants Angular

Un composant fonctionne en associant la logique d'une classe TypeScript avec un template HTML et un style CSS. Pour notre diagramme, le point d'entrée c'est le fichier `index.html` qui est notre unique page web comportant le `Root Component`. Cette page n'est autre qu'un arbre de composants. Le `Root component` fait appel à d'autres composants qui peuvent aussi intégrer plusieurs composants.

Conclusion

À travers ce chapitre, nous avons présenté l'environnement de développement de notre application en justifiant nos choix technologiques. Le chapitre suivant sera consacré à la présentation de notre première release.

RELEASE 1 PRÉDICTION

Plan

1	Sprint 1 Création d'un modèle de prédition	27
2	Sprint 2 Gestion des prédictions des employés	51

Introduction

Une release est constituée d'une suite d'itérations (sprint) qui se terminent quand les incrément de ces derniers construisent un produit d'une valeur suffisante pour les utilisateurs finaux.

La période des releases est définie par le Product Owner en coopération avec son équipe Scrum. Notre première release sera composée de deux sprints, dont chacun, ayant une vitesse de 28 jours. Dans ce chapitre, nous allons traiter les "user story" de nos sprints pour produire un produit potentiellement livrable.

4.1 Sprint 1 Création d'un modèle de prédiction

Le sprint est le cœur de Scrum. Il s'agit d'un bloc de temps pendant lequel un incrément du produit est effectué. Tous les sprints d'une release ont une durée constante et ne se chevauchent jamais, c'est-à-dire qu'un sprint ne peut pas commencer tant que le précédent n'est pas encore terminé. Avant de se lancer dans un sprint, l'équipe Scrum doit obligatoirement définir le but de ce dernier qui est défini en terme métier et non pas en terme technique pour qu'il soit compréhensible par les membres en dehors de l'équipe [18].

4.1.1 Backlog du Sprint

Le Backlog du sprint contient une liste des tâches identifiées par l'équipe Scrum qui devront être réalisées avant la fin de sprint.

Tableau 4.1: Backlog du Sprint 1

ID	User Story	ID Tâche	Tâche	Estimation (jours)
1	En tant qu'administrateur, je veux importer un fichier contenant toutes les informations des employés.	1.1	Installer les différents packages nécessaires de python et Tensorflow.	1
		1.2	Développer et tester un service web d'import.	1
		1.3	Mettre en place l'interface d'import.	1

ID	User Story	ID Tâche	Tâche	Estimation (jours)
2	En tant qu'administrateur, je veux préparer mes données.	2.1	Éliminer les lignes vides du fichier qui a été importé.	2
		2.2	Développer et tester un service web qui permet de convertir toutes les données en données numériques et les mettre à la même échelle.	5
		2.3	Implémenter et tester un service web qui permet de diviser les données en des données d'apprentissage pour créer notre modèle et des données de test pour évaluer notre modèle construit.	1
3	En tant qu'administrateur, je veux construire un modèle de prédiction.	3.1	Évaluer et adapter nos données aux différentes solutions existantes pour créer notre modèle de prédiction.	2
		3.2	Construire un algorithme basé sur le réseau de neurones comme étant notre modèle de prédiction vu sa grande performance et l'exposer sous format de service web.	6
4	En tant qu'administrateur, je souhaite évaluer mon modèle.	4.1	Développer et tester un service web qui permet d'évaluer et tester la performance de notre modèle de prédiction avec les données de test.	6

ID	User Story	ID Tâche	Tâche	Estimation (jours)
5	En tant qu'utilisateur, je veux effectuer des prédictions sur la démission des employés.	5.1	Développer et tester un service web qui permet d'effectuer des prédictions.	1
		5.2	Mettre en place l'interface de prédiction.	1

4.1.2 Spécification fonctionnelle

La figure 4.1 illustre le diagramme de cas d'utilisation détaillé de «Créer un modèle de prédiction».

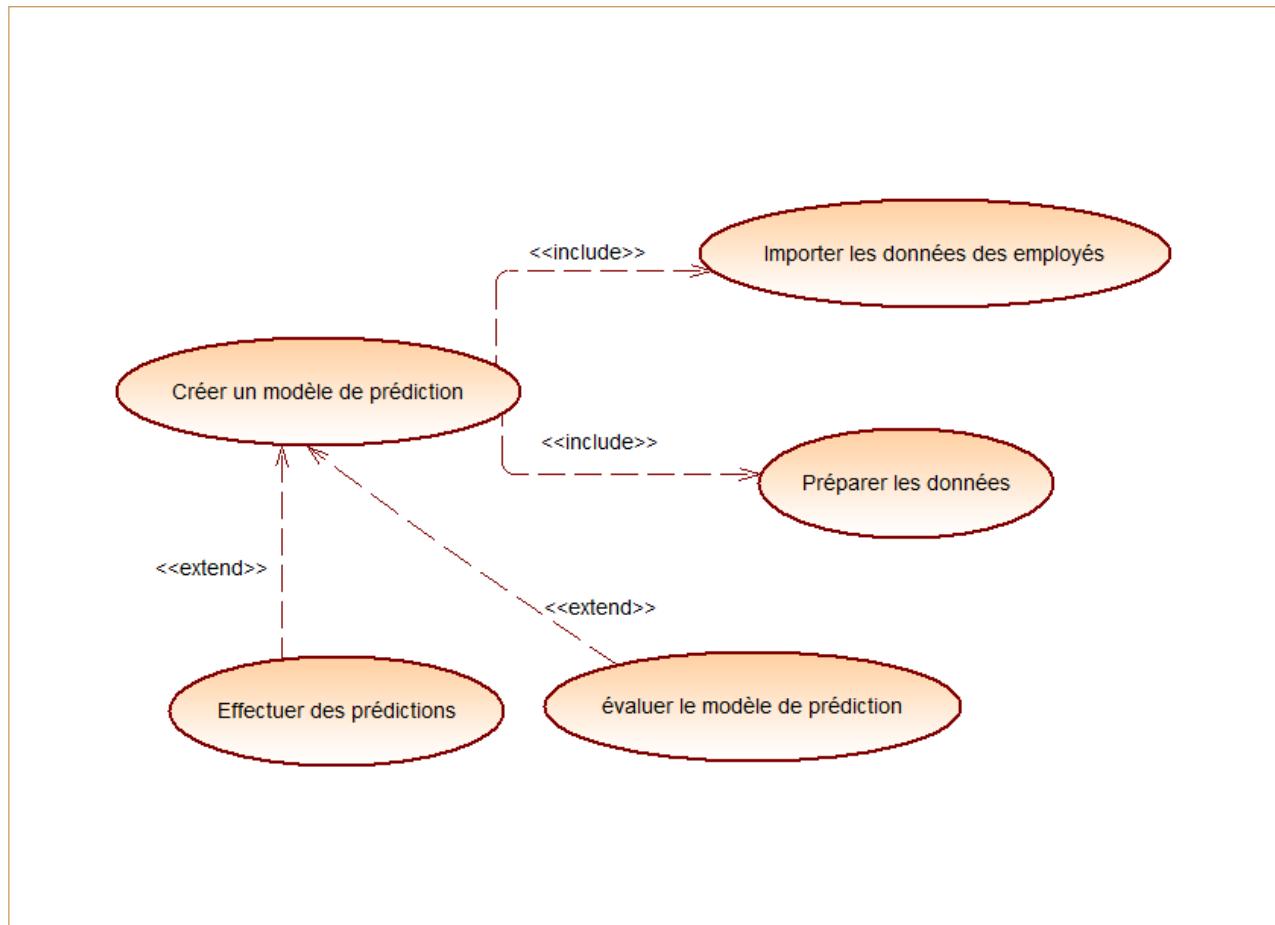


Figure 4.1: Diagramme de cas d'utilisation «Créer un modèle de prédiction»

Le tableau 4.2 présente la description textuelle du cas d'utilisation « Importer les données des employés ».

Tableau 4.2: Description du diagramme de cas d'utilisation «Importer les données des employés»

Cas d'utilisation	Importer les données des employés.
Acteur	Administrateur.
Précondition	Les données doivent être sous format «csv».
Post-condition	Données importées.
Description du scénario	<ul style="list-style-type: none"> — L'administrateur accède à l'interface d'import. — L'administrateur sélectionne un fichier contenant les informations des employés.
Exception	Erreur de format de fichier ou de connexion.

Le tableau 4.3 présente la description textuelle du cas d'utilisation « Construire un modèle de prédiction ».

Tableau 4.3: Description du diagramme de cas d'utilisation «Construire un modèle de prédiction»

Cas d'utilisation	Construire un modèle de prédiction.
Acteur	Administrateur.
Précondition	Les données pour l'apprentissage et pour le test existent.
Post-condition	Modèle construit.
Description du scénario	<ul style="list-style-type: none"> — L'administrateur appelle le service web de création du modèle.
Exception	Erreur de format de données ou de connexion.

Le tableau 4.4 présente la description textuelle du cas d'utilisation «Évaluer le modèle de prédiction».

Tableau 4.4: Description du diagramme de cas d'utilisation «Évaluer le modèle de prédiction»

Cas d'utilisation	Évaluer le modèle de prédiction.
Acteur	Administrateur.
Précondition	Le modèle doit être créé.
Post-condition	Modèle évalué.
Description du scénario	<ul style="list-style-type: none"> — L'administrateur appelle le service web qui permet d'évaluer le modèle de prédiction .
Exception	Erreur de connexion.

Le tableau 4.5 présente la description textuelle du cas d'utilisation « Effectuer des prédictions ».

Tableau 4.5: Description du diagramme de cas d'utilisation « Effectuer des prédictions »

Cas d'utilisation	Effectuer des prédictions.
Acteur	Administrateur.
Précondition	Le modèle doit être créé.
Post-condition	Prédiction effectuée.
Description du scénario	<ul style="list-style-type: none"> — L'administrateur effectue des prédictions sur la démission des employés.
Exception	Erreur de format de données ou de connexion.

4.1.3 Apprentissage Automatique

Dans cette partie, nous allons construire notre modèle de prédiction qui est basé sur l'apprentissage automatique ou Machine learning en anglais. En effet, l'apprentissage automatique est défini comme étant un champ d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés. Pour être plus clair, ce que fait le machine learning, c'est apprendre à résoudre un problème de manière automatique en utilisant les données. Dans le cadre du traitement de la donnée,

il s'agit de construire un modèle obtenu directement à partir d'exemples. L'objectif des algorithmes de machine learning est de minimiser ce qu'on appelle l'erreur, c'est-à-dire de se tromper le moins possible [19].

Nous distinguons différents types d'apprentissage automatique dont les plus connus sont les suivants :

4.1.3.1 Apprentissage Automatique supervisé

Récupération des données annotées à partir de leurs sorties pour entraîner le modèle, c'est-à-dire une classe cible ou une étiquette qui leur a déjà été attribuée et nous voulons que l'algorithme devienne capable de le prédire sur de nouvelles données non annotées une fois entraînées.

La figure 4.2 décrit les différentes étapes de l'apprentissage automatique supervisé.

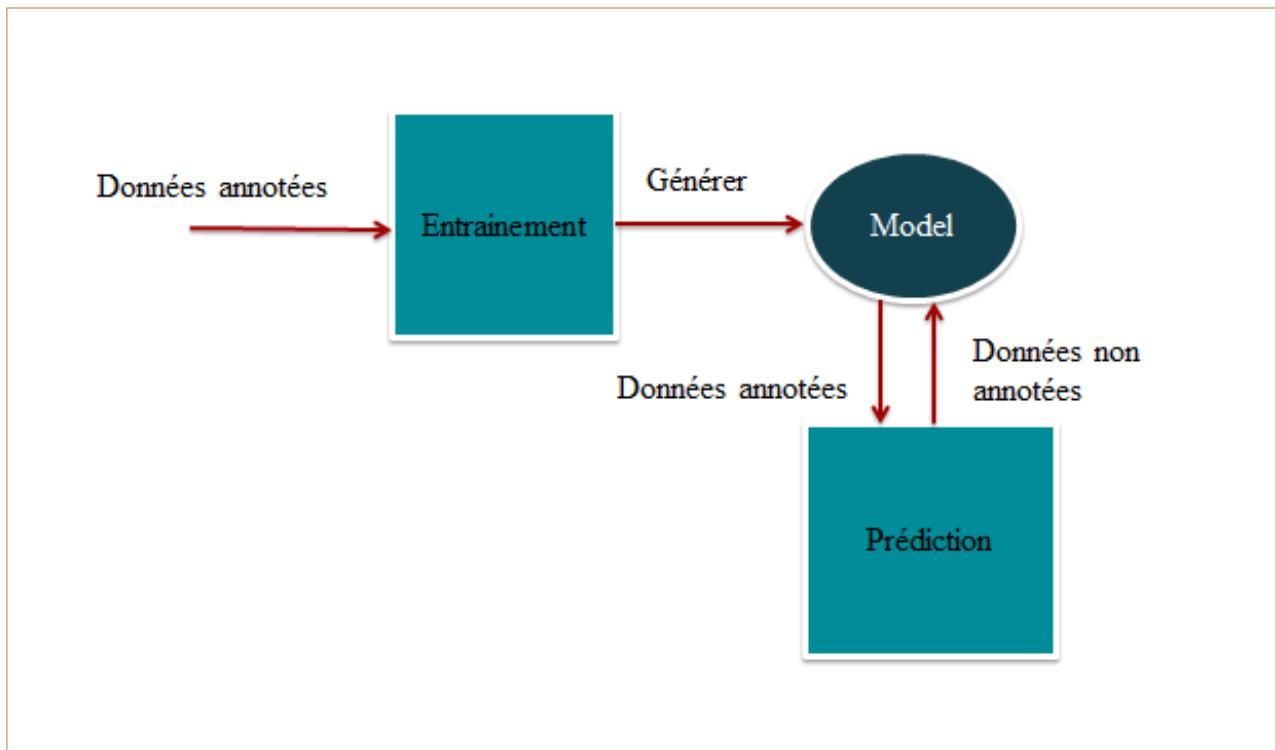


Figure 4.2: Apprentissage Automatique supervisé

4.1.3.2 Apprentissage Automatique non supervisé

Les données d'entrées ne sont pas annotées. L'algorithme d'entraînement s'applique dans ce cas à trouver seul les similarités et distinctions au sein de ces données et à regrouper ensemble celles qui partagent des caractéristiques communes.

La principale différence lorsque nous parlons d'apprentissage non supervisé est que nous n'avons plus d'exemples. Il n'est donc plus possible d'évaluer la performance de l'algorithme en comparant le

résultat avec les valeurs observées. Les problèmes d'apprentissage non supervisé les plus populaires sont les analyses de regroupement [20].

4.1.4 Solutions existantes

La phase la plus importante dans l'apprentissage automatique, c'est d'identifier de quel type d'apprentissage nous allons utiliser pour répondre à notre problème.

La figure 4.3 expose la structure de nos données.



Matricule	Age	Expérience	Manager	Département	Pôle	Ecole	Démission
1	24	0	M1	Développement	P1	ISI	0
2	30	5	M2	RH	P2		1
3	40	10	M2	Développement	P1	INSAT	0
4	35	5	M1	Développement	P1	ENSI	1
5	28	4	M2	Développement	P2	SUPCOM	0
6	55	25	M3	Management	P2	SUPCOM	0
7	32	6	M3	Développement	P2	ISI	0
8	32	8	M2	Développement	P2	ENSI	1
9	25	1	M1	Développement	P1	ISI	1
10	27	3	M2	Développement	P2	SUPCOM	0

Figure 4.3: Structure de nos données

Cette structure est une représentation de nos données contenues dans un fichier excel comportant toutes les informations des employés de la société. Nos données sont annotées et nous avons pour chaque donnée sa sortie (label, étiquette) qui est démission considérée comme étant une variable de prédiction (« 1 » : l'employé va démissionner, « 0 » : l'employé va rester dans Sofrecom).

Notre choix s'est porté sur l'apprentissage supervisé vu que toutes nos données sont étiquetées et nous allons utiliser les différents algorithmes de cet apprentissage pour construire notre modèle de prédiction.

4.1.4.1 Classification

C'est lorsque le type de sortie que nous attendons de notre programme est une valeur discrète (une catégorie). Dans notre cas, nous avons deux classes pour la démission (0 ou 1).

Nous avons appliqué plusieurs algorithmes de classification sur nos données et nous allons évaluer chaque modèle avec la matrice de confusion comme la montre la figure 4.4.

Classes Réelles	p	Vrais Positifs (TP)
	n	Faux Positifs(FP)
	p	Vrais Négatifs (TN)
	n	Faux Négatifs(FN)

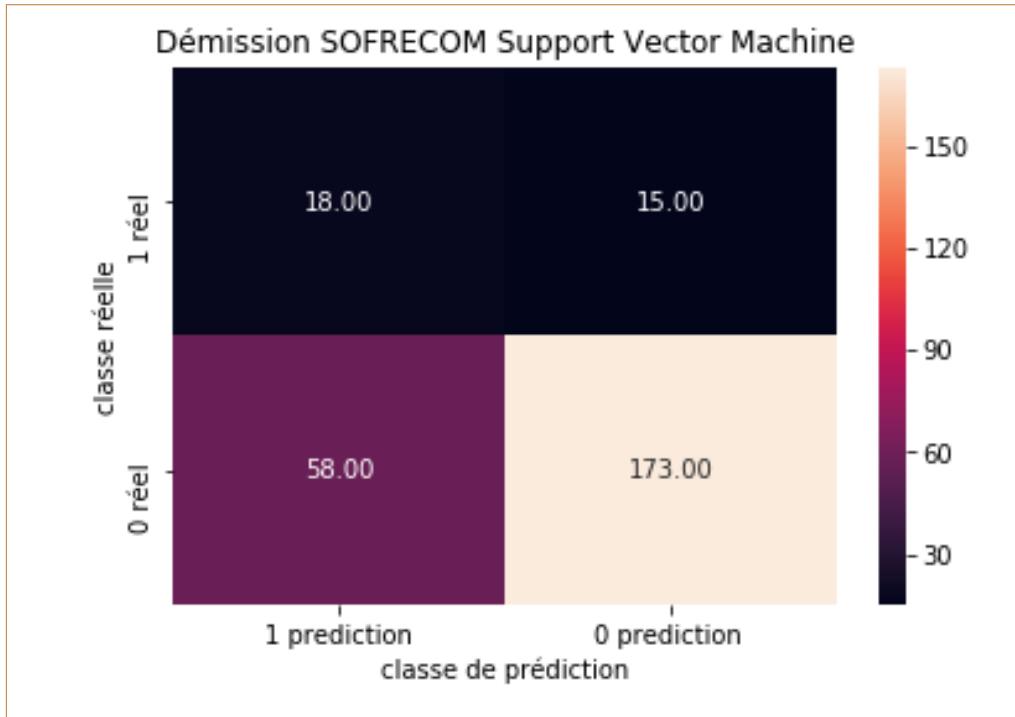
Figure 4.4: Matrice de confusion

Nous allons définir chaque notion présentée dans la figure 4.4 :

- TP (Vrais positifs) : Mesure la proportion de positifs correctement identifiés comme tels ;
- TN (Vrais négatifs) : Mesure la proportion de négatifs qui sont correctement identifiés comme tels ;
- FP (Faux positifs) : Est un résultat qui indique qu'une condition donnée a été remplie, quand ce n'est pas le cas. C'est à dire, par erreur, un effet positif a été supposé ;
- FN (Faux négatifs) : Est lorsqu'un résultat de test indique qu'une condition a échoué, alors qu'elle était réussie.

Machine à Vecteurs de Support (SVM)

C'est un algorithme de classification binaire qui a pour objectif de trouver la séparation entre deux classes d'objets avec l'idée que plus la séparation est large, plus la classification est robuste [21]. Nous avons appliqué le modèle SVM sur nos données et nous avons obtenu comme résultats suivants concernant la performance de ce modèle comme la montre la figure 4.5.

**Figure 4.5:** Matrice de confusion de SVM

Ce modèle contient 73 erreurs sur 264 données et il contient un taux de précision de 0.69 comme la montre la figure 4.6.

	<code>precision</code>	<code>recall</code>	<code>f1-score</code>	<code>support</code>
0	0.75	0.92	0.83	188
1	0.55	0.24	0.33	76
<code>avg / total</code>	<code>0.69</code>	0.72	0.68	264

Figure 4.6: Précision du modèle SVM

NB : La précision est la fraction des instances pertinentes parmi les instances récupérées. Elle est calculée à l'aide de la formule suivante :

$$Precision = \frac{\sum TP}{\sum (TP + FP)}$$

Ce modèle ne permet pas de donner la valeur de la prédiction sous un format de probabilité (pourcentage). De plus, le taux d'erreur est très grand.

Arbre de décision

Les arbres de décision sont l'une des structures de données majeures de l'apprentissage statistique. Leur fonctionnement repose sur des heuristiques qui, tout en satisfaisant l'intuition, donnent des résultats remarquables [22]. Nous avons appliqué le modèle arbre de décision sur nos données et nous avons obtenu comme résultats suivants concernant la performance de ce modèle comme la montre la figure 4.7.

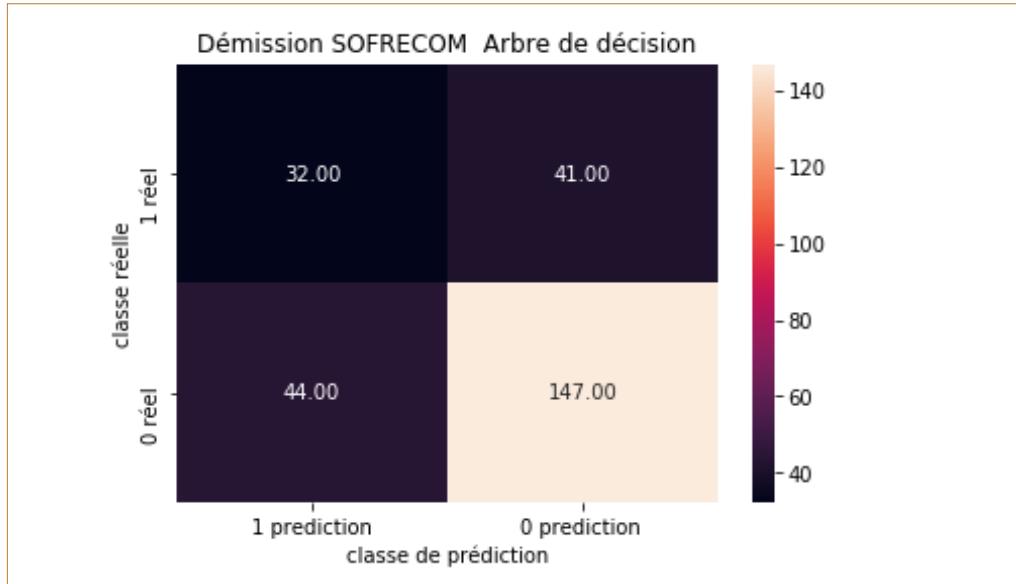


Figure 4.7: Matrice de confusion d'arbre de décision

Ce modèle contient 85 erreurs sur 264 données et il contient un taux de précision de 0.67 comme la montre la figure 4.8.

	precision	recall	f1-score	support
0	0.77	0.78	0.78	188
1	0.44	0.42	0.43	76
avg / total	0.67	0.68	0.68	264

Figure 4.8: Précision du modèle d'arbre de décision

Ce modèle ne permet pas de donner la valeur de la prédiction sous un format de probabilité (pourcentage). De plus, le taux d'erreur est très grand.

Forêt d'arbres décisionnels (Random Forest)

L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents [23].

Nous avons appliqué le modèle forêt d'arbres décisionnels sur nos données et nous avons obtenu comme résultats suivants concernant la performance de ce modèle comme la montre la figure 4.9.

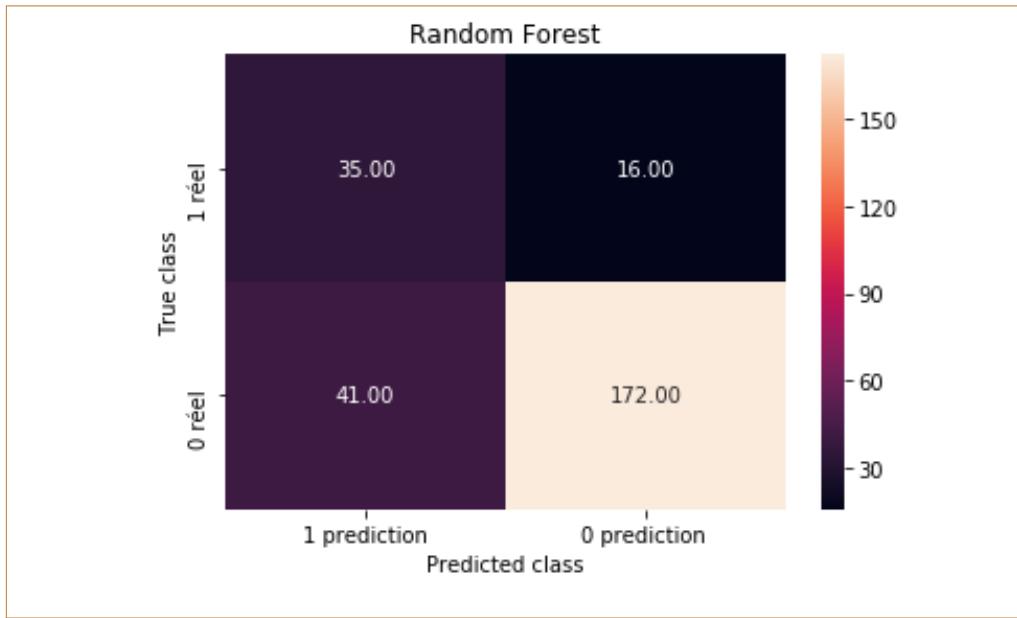


Figure 4.9: Matrice de confusion de forêt d'arbres décisionnels

Ce modèle contient 57 erreurs sur 264 données et il contient un taux de précision de 0.77 comme la montre la figure 4.10.

	precision	recall	f1-score	support
0	0.81	0.91	0.86	188
1	0.69	0.46	0.55	76
avg / total	0.77	0.78	0.77	264

Figure 4.10: Précision du modèle de forêt d'arbres décisionnels

Ce modèle ne permet pas de donner la valeur de la prédiction sous un format de probabilité (pourcentage). De plus, le taux d'erreur est très grand.

K plus proches voisins (K-Nearest Neighbors ou KNN)

C'est une méthode d'apprentissage supervisé. Il existe une base de données d'apprentissage composée de N paires "entrées-sorties". Pour estimer la sortie associée à une nouvelle entrée x , la méthode des k plus proches voisins consiste à prendre en compte (de manière identique) les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x [24].

Nous avons appliqué le modèle KNN sur nos données et nous avons obtenu comme résultats suivants concernant la performance de ce modèle comme la montre la figure 4.11.

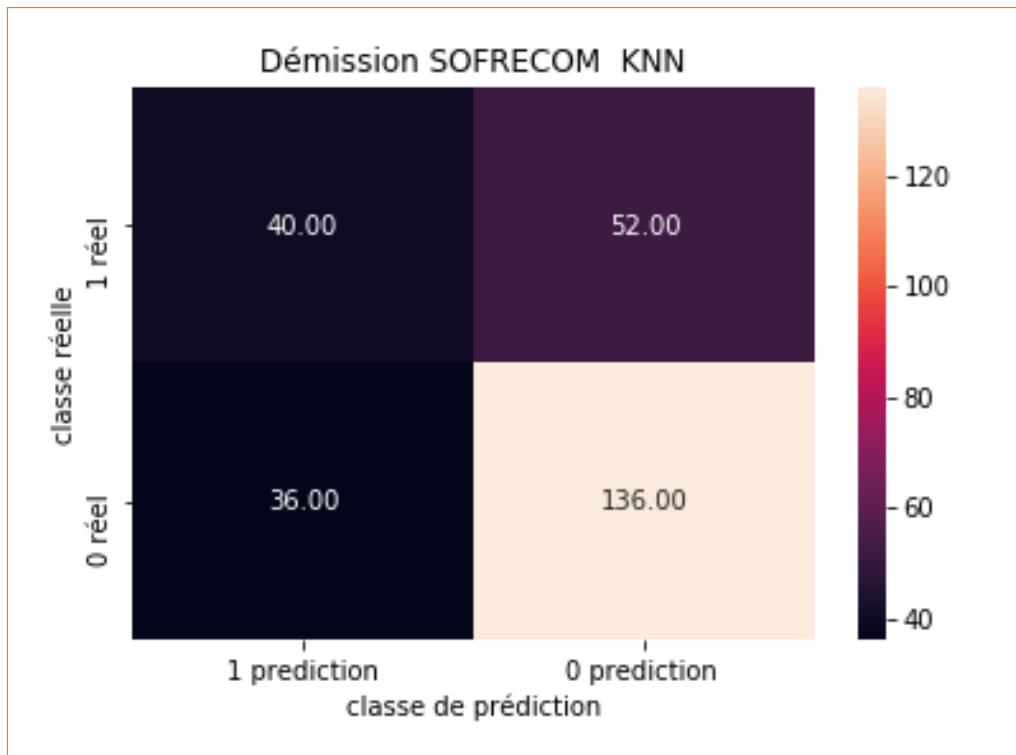


Figure 4.11: Matrice de confusion de KNN

Ce modèle contient 88 erreurs sur 264 données et il contient un taux de précision de 0.69 comme la montre la figure 4.12.

	precision	recall	f1-score	support
0	0.79	0.72	0.76	188
1	0.43	0.53	0.48	76
avg / total	0.69	0.67	0.68	264

Figure 4.12: Précision du modèle de KNN

Ce modèle ne permet pas de donner la valeur de la prédiction sous un format de probabilité (pourcentage). De plus, le taux d'erreur est très grand.

Gradient Boosting Classifier

Le gradient boosting est une technique d'apprentissage machine pour les problèmes de régression et de classification, qui produit un modèle de prédiction sous la forme d'un ensemble de modèles de prédiction faibles, généralement des arbres de décision. Il construit le modèle par étapes, comme les autres méthodes de boosting, et il les généralise en permettant l'optimisation d'une fonction de perte arbitraire différentiable [25].

Nous avons appliqué le modèle KNN sur nos données et nous avons obtenu comme résultats suivants concernant la performance de ce modèle comme la montre la figure 4.13.

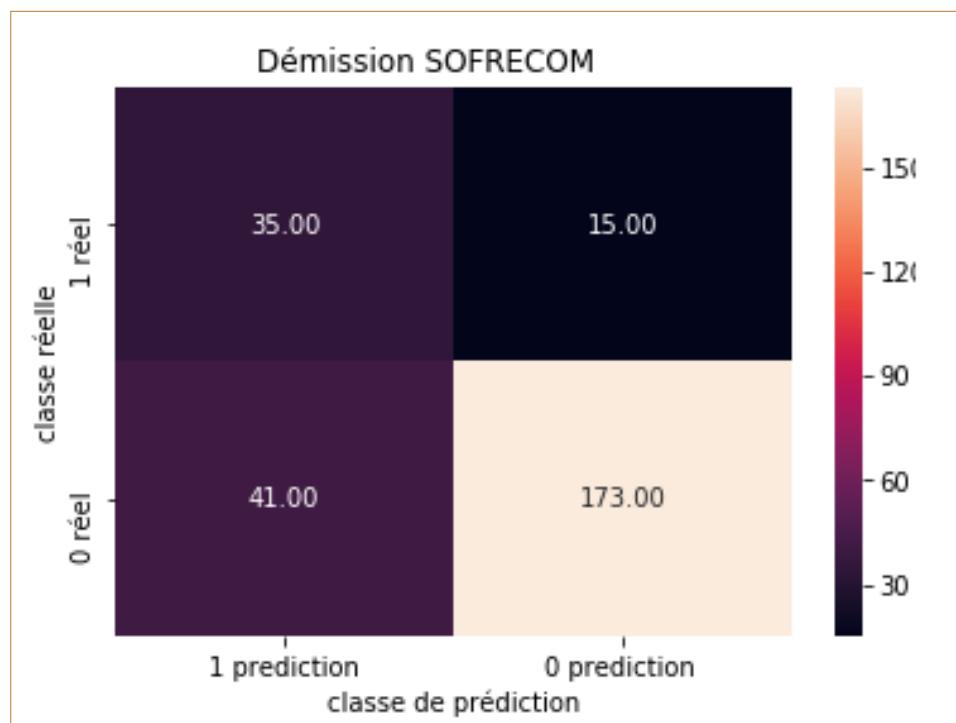


Figure 4.13: Matrice de confusion de Gradient Boosting Classifier

Ce modèle contient 56 erreurs sur 264 données et il contient un taux de précision de 0.78 comme la montre la figure 4.14.

	precision	recall	f1-score	support
0	0.81	0.92	0.86	188
1	0.70	0.46	0.56	76
avg / total	0.78	0.79	0.77	264

Figure 4.14: Précision du modèle de Gradient Boosting Classifier

Ce modèle ne permet pas de donner la valeur de la prédiction sous un format de probabilité (pourcentage). De plus, le taux d'erreur est très grand.

4.1.4.2 Régression

C'est lorsque le type de sortie que nous attendons de notre programme est une valeur continue ou quantitative [26].

Dans cette partie, nous allons aborder les différents algorithmes de régression sur nos données.

Régression linéaire

C'est un modèle qui cherche à établir une relation linéaire entre une variable, dite expliquée, et une ou plusieurs variables, dites explicatives.

$$Y = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + \dots + w_n * x_n + b$$

- Y : variable dépendante (expliquée) ;
- xi : variable indépendante (explicative) ;
- b : ordonnée à l'origine (biais) ;
- wi : poid/pente (variation moyenne de la valeur d'Y par rapport à xi).

Dans notre cas, nous avons la démission comme variable expliquée et comme variables explicatives (civilité, expérience avant sofrecom, expérience sofrecom, expérience totale, sériorité, niveau académique, poste, pôle, manager, âge, situation familiale).

Nous avons appliqué le modèle régression linéaire sur nos données et nous avons testé sur 20 données comme la montre la figure 4.15.

Sortie réelle	Sortie obtenue après prédiction
<pre>print(y_test[:20])</pre> 542 0 164 0 658 0 65 0 152 0 182 0 639 0 20 0 638 0 693 0 464 0 179 1 400 1 370 0 426 0 213 0 37 1 820 0 209 0 15 0 Name: DEM, dtype: int64	<pre>print(y_pred[:20].reshape(-1,1))</pre> [[0.41237431] [0.41123383] [0.13262338] [0.42118522] [0.48036921] [0.04948867] [0.43494222] [0.41345456] [0.01484482] [0.05530289] [0.53919396] [0.30280497] [0.41683623] [0.36018497] [0.00473446] [0.40259839] [0.46727539] [-0.00315743] [0.35772666] [0.20410087]]

Figure 4.15: Test sur le modèle de régression linéaire

Comme la montre la figure 4.15, le modèle construit n'est pas performant et n'est pas adapté à notre problème.

Gradient Boosting Regressor

De même pour l'algorithme de Gradient Boosting Regressor, le modèle appliqué sur nos données n'est pas adapté à notre problème comme la montre la figure 4.16.

Sortie réelle	Sortie obtenue après prédiction
<pre>print(y_test[:20])</pre> <pre>542 0 164 0 658 0 65 0 152 0 182 0 639 0 20 0 638 0 693 0 464 0 179 1 400 1 370 0 426 0 213 0 37 1 820 0 209 0 15 0 Name: DEM, dtype: int64</pre>	<pre>print(y_pred[:20].reshape(-1,1))</pre> <pre>[[0.34680363] [0.5159753] [0.02767915] [0.03208368] [0.69260771] [0.52820397] [0.06774141] [0.53755737] [0.26261826] [0.15771549] [0.22218292] [0.79621954] [0.24419521] [0.34083188] [0.28961228] [0.12679752] [0.32997504] [0.03809112] [0.14399118] [0.67727785]]</pre>

Figure 4.16: Test sur le modèle de Gradient Boosting Regressor

Forêt d'arbres décisionnels de régression (Random Forest Regressor)

De manière similaire pour l'algorithme de forêt d'arbres décisionnels de régression, le modèle appliqué sur nos données n'est pas adapté à notre problème comme la montre la figure 4.17.

Sortie réelle	Sortie obtenue après prédiction
<pre>print(y_test[:20])</pre> <pre>542 0 164 0 658 0 65 0 152 0 182 0 639 0 20 0 638 0 693 0 464 0 179 1 400 1 370 0 426 0 213 0 37 1 820 0 209 0 15 0 Name: DEM, dtype: int64</pre>	<pre>print(predictions[:20].reshape(-1,1))</pre> <pre>[[0.5172963] [0.49278957] [0.03620642] [0.38625266] [0.27510703] [0.37805115] [0.22331378] [0.30992682] [0.29262555] [0.12632246] [0.34466247] [0.43357948] [0.50094603] [0.42035925] [0.24546972] [0.32400389] [0.32097552] [0.1694405] [0.21495822] [0.23259676]]</pre>

Figure 4.17: Test sur le modèle de forêt d'arbres décisionnels de régression

Gradient Boosting Machine (GBM) de H2O Flow

C'est avec cet algorithme que sofrecom construit le modèle de prédiction sur la démission de ses employés. Avec cet algorithme, nous avons créé un nouveau modèle avec nos nouvelles données actualisées comme la montre la figure 4.18.

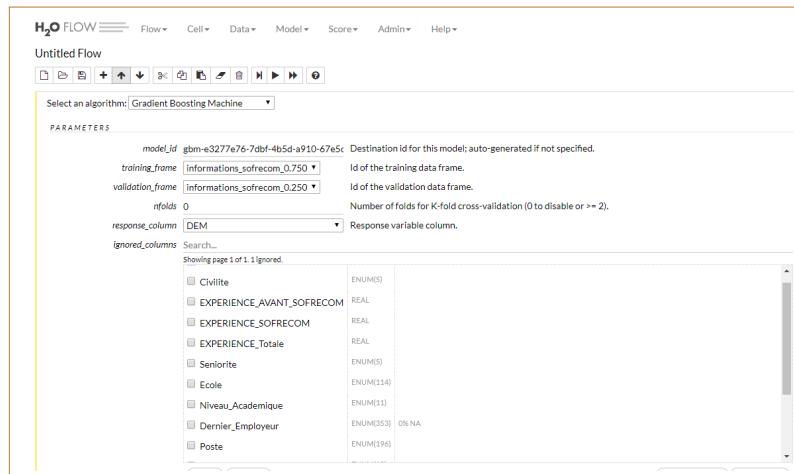


Figure 4.18: Création du modèle GBM avec H2O Flow

Nous avons construit notre modèle avec les données d'apprentissage (75% de totalité nos données) et les autres données qui sont les données de test sont utilisées pour l'évaluation de notre modèle.

La figure 4.19 décrit l'évaluation de notre modèle avec l'importance de chaque variable explicative.

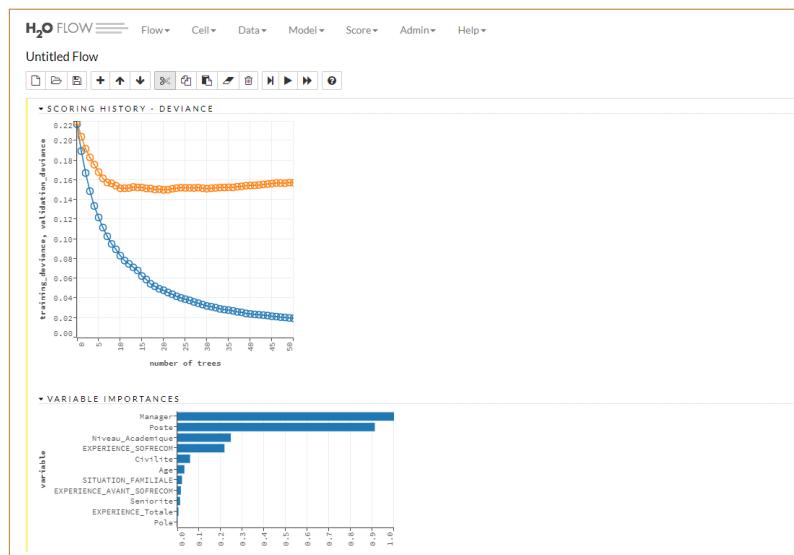


Figure 4.19: Évaluation du GBM et importance de chaque variable explicative

Les deux courbes (données d'apprentissage en bleu et données de test en orange) ne sont pas concordantes, elles sont par contre très éloignées l'une de l'autre ce qui nous indique que ce modèle n'est pas adapté à notre problème.

4.1.5 Solution envisagée

Notre solution consiste à aller en profondeur et construire un réseau de neurones qui permet de regrouper plusieurs algorithmes pour pouvoir traiter nos données et avoir un modèle de prédiction performant. Les algorithmes d'apprentissage en profondeur tentent d'apprendre des fonctionnalités de haut niveau à partir de données. Par conséquent, l'apprentissage en profondeur réduit la tâche de développement d'un nouvel extracteur de caractéristiques pour chaque problème.

La différence la plus importante entre l'apprentissage en profondeur et l'apprentissage automatique traditionnel est sa performance à mesurer quand l'échelle des données augmente.

4.1.5.1 Réseau de neurones

C'est un modèle inspiré du fonctionnement cérébral, qui se compose de couches, dont au moins une est cachée, contenant des unités connectées simples, ou neurones, suivies de non-linéarités. Les neurones reçoivent plusieurs valeurs d'entrée et génèrent une valeur de sortie. Le neurone calcule la valeur de sortie en appliquant une fonction d'activation (transformation non linéaire) à une somme pondérée des valeurs d'entrée [27].

Parmi les frameworks les plus connus pour construire un réseau de neurones, nous trouvons Tensorflow. En effet, TensorFlow propose plusieurs Estimators prédéfinis, notamment le plus connu est « DNNRegressor » que nous allons l'utilisé dans notre cas.

4.1.5.2 Préparation des données

Dans la couche d'entrée de réseau de neurones, les données doivent être des données numériques. Alors, la première phase de préparation des données consiste à convertir les données catégorielles (un ensemble discret de valeurs possibles) en des données numériques et pour ce cas nous allons utiliser la bibliothèque de python « map » comme la montre la figure 4.20.

```
train['Civilite'] = train.Civilite.str.strip()
train['Civilite'] = train.Civilite.str.lower()
Civilite = {label: idx for idx, label in
            enumerate(np.unique(train['Civilite']))}
train['Civilite'] = train['Civilite'].map(Civilite)
```

Figure 4.20: Script de conversion des données

Les données sont converties en indexant chaque donnée catégorielle en donnée numérique comme la montre la figure 4.21 pour la variable explicative « civilité ».

```
Civilité
{'monsieur': 2, 'mademoiselle': 1, 'madame': 0}
```

Figure 4.21: Exemple de conversion des données

Normalisation des données

La deuxième phase de préparation qui est la normalisation des données consiste à convertir une plage réelle de valeurs en une plage standard de valeurs, généralement de 0 à 1. Supposons que la plage naturelle d'une certaine caractéristique s'étende de 800 à 6 000. En effectuant diverses soustractions et divisions, nous pouvons normaliser ces valeurs à une plage s'étendant de 0 à +1.

Dans notre cas, nous allons utiliser la bibliothèque «MinMaxScaler» de la bibliothèque «sklearn.preprocessing» qui est basée sur la formule suivante :

$$X_{norm} = \frac{(X - X_{min})}{(X_{max} - X_{min})}$$

Exemple de nos données traitées après la phase de la normalisation (l'échelle de toutes nos données est devenue entre 0 et 1) comme la montre la figure 4.22.

```
127.0.0.1 - - [22/Apr/2018 14:08:19] "GET /Preprocessing_train HTTP/1.1" 200 -
```

	Civilite	EXPERIENCE_AVANT_SOFRÉCOM	EXPERIENCE_SOFRÉCOM	\
0	0.0	0.014286	1.000000	
1	0.0	0.000000	0.998428	
2	1.0	0.268571	0.998428	
3	1.0	0.285714	0.988994	
4	1.0	0.331429	0.988994	
5	0.5	0.308571	0.985849	
6	1.0	0.388571	0.985849	
7	0.0	0.040000	0.985849	
8	1.0	0.371429	0.976415	
9	1.0	0.400000	0.976415	
10	0.5	0.000000	0.973270	
11	1.0	0.800000	0.963836	
12	0.0	0.285714	0.955975	
13	1.0	0.628571	0.951258	
14	1.0	0.205714	0.933962	
15	0.5	0.000000	0.932390	

Figure 4.22: données normalisées

Division des données

Cette phase consiste à diviser nos données en des données d'apprentissage pour construire et entraîner notre modèle et des données de test pour évaluer la performance de notre modèle. Comme la montre la figure 4.23, nous avons choisi de diviser nos données en 80% pour les données d'apprentissage et les autres 20% pour les données de test.

```
# Train and Test : diviser train en x_test & x_train
x_train, x_test, y_train, y_test = train_test_split(training_set[FEATURES], prediction_set, train_size=0.8, test_size=0.2,
random_state=42)
```

Figure 4.23: données divisées

4.1.5.3 Architecture de réseau de neurones

Nous allons nous intéresser au perceptron multicouche (architecture de réseau de neurones) qui est organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement. Il s'agit donc d'un réseau à propagation directe (feed-forward) qui comporte L couches et chaque neurone d'une couche est totalement connecté aux neurones de la couche suivante. Ce réseau est organisé en couches [28] :

- Une couche : un groupe de neurones uniformes sans connexion les uns avec les autres ;
- Transformation des données :
 - Une couche reçoit des données d'entrée et les transforme en des données de sortie grâce à une fonction d'activation ;
 - Une couche au moins n'est pas linéaire ;
 - Les dimensions d'entrée et de sortie peuvent être différentes.
- Au moins 2 couches (une couche dite cachée et une couche de sortie) ;
- Structuration sans cycle (feed-forward) : une couche ne peut utiliser que les sorties des couches précédentes ;
- La méthode « rétro propagation du gradient » pour faire l'apprentissage ;
- Nombre de neurones cachés : Le choix de la taille d'une couche cachée n'est pas implicite et doit être ajusté ;
- Mesure de l'erreur sur le corpus d'apprentissage.

Nous allons identifier maintenant l'architecture de notre modèle (réseau de neurones) qui est basée sur le perceptron multicouche comme la montre la figure 4.24.

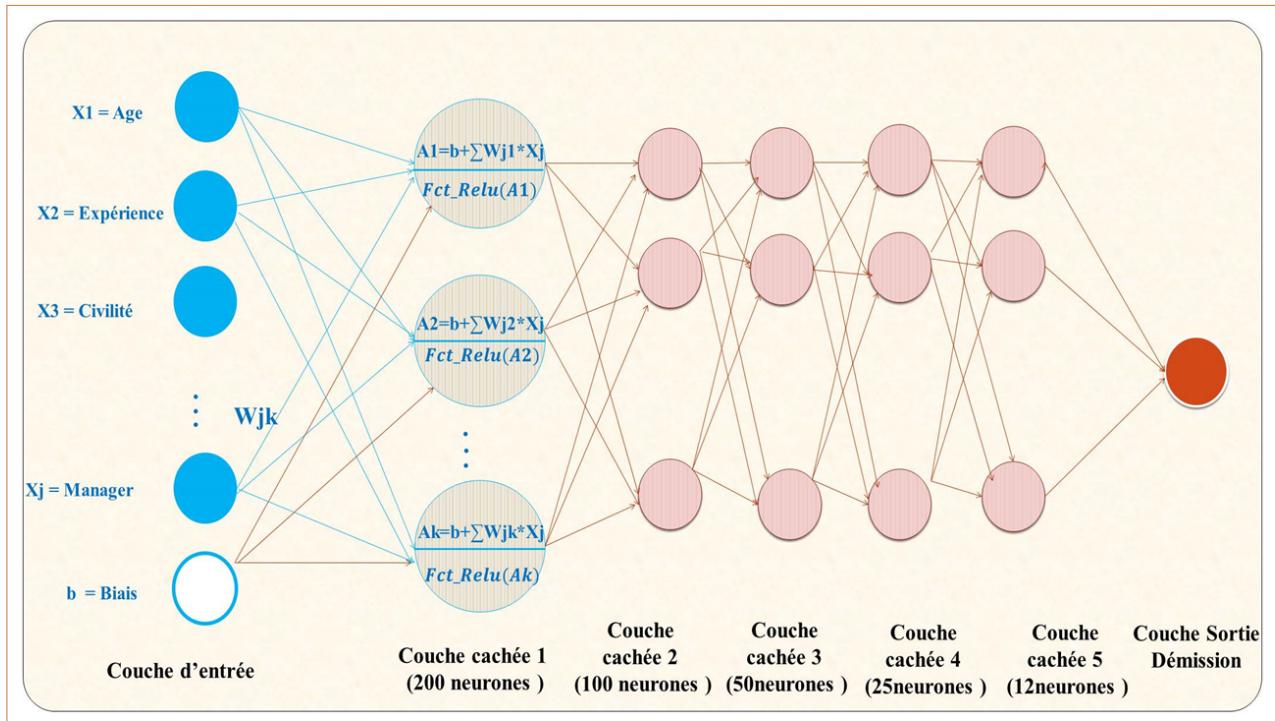


Figure 4.24: Architecture de notre réseau de neurones

Après plusieurs tentatives, nous avons fixé notre architecture de réseau de neurones qui répond à notre besoin (couche d'entrée + 5 couches cachées + couche de sortie). Pour mieux comprendre cette architecture, notre réseau de neurones est composé de :

- une couche d'entrée : cette couche contient les variables explicatives concernant les informations des employés (civilité, expérience avant sofrecom, expérience sofrecom, expérience totale, seniorité, niveau académique, poste, pôle, manager, âge, situation familiale) ;
- b : biais ;
- un système de poids (W_{jk}) liés à ces entrées ;
- cinq couches cachées comportant respectivement 200,100,50,25,12 neurones ;
- couche de sortie : démission (la variable cible ou expliquée) ;
- fonction d'activation « Relu » dans chaque neurone de couche cachée : Relu est une fonction d'activation non linéaire la plus simple et la plus utilisée.

Cette fonction d'activation transforme la sortie en 0 si l'entrée est inférieure à 0 sinon la sortie est égale à l'entrée comme la montre la figure 4.25.

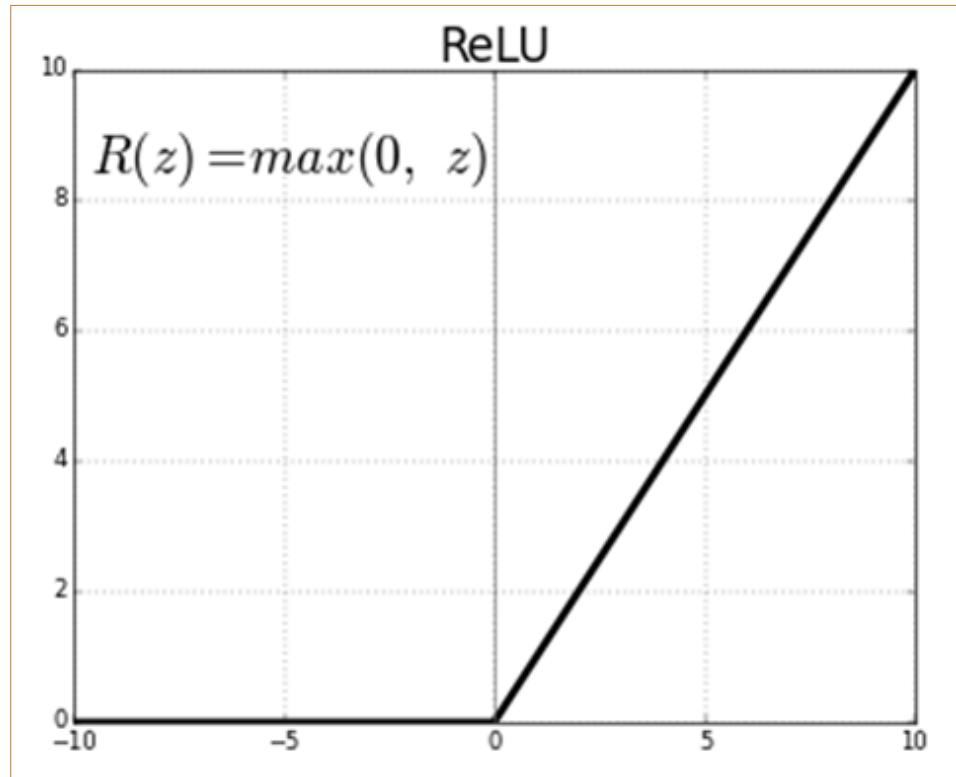


Figure 4.25: Fonction d'activation « Relu »

4.1.5.4 Évaluation du modèle

La phase d'évaluation du modèle est la plus importante pour avoir une bonne visualisation sur la performance de notre modèle. En effet, lors de la création de notre modèle, Tensorflow nous donne des indications sur la valeur de la fonction d'erreur (loss) pendant chaque itération comme la montre la figure 4.26.

The screenshot shows a Jupyter Notebook interface. The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. On the right, there are buttons for Logout, Trusted, and Python 3. The main area displays a series of INFO-level log messages from Tensorflow, showing the loss value and global step for each iteration from 401 to 4901. At the bottom of the log, it says "INFO:tensorflow:Saving checkpoints for final step: 0.058894785". Below the log, a red box highlights an incoming HTTP request: "127.0.0.1 - [25/Mar/2018 21:52:28] "GET /Model HTTP/1.1" 200 -". The bottom left corner shows the "In []:" prompt.

Figure 4.26: Évaluation de notre modèle de réseau de neurones

Notre fonction d'erreur (loss) est définie par défaut dans Tensorflow comme étant l'erreur quadratique moyenne.

En effet, tensorflow offre un outil performant « Tensorboard » qui nous permet de visualiser les différentes valeurs de la fonction d'erreur pendant chaque itération comme la figure 4.27.



Figure 4.27: Visualisation de la fonction d'erreur avec « Tensorboard »

La fonction d'erreur est égale à 0.23 pendant la première itération et elle a diminué à 0.09284 dans l'itération 3000.

La valeur de notre fonction d'erreur finale est égale à 0.05 dans l'itération 5000.

Prédiction des données de test sur notre modèle

Nous allons tester la performance de notre modèle avec les données de test. Comme la montre la figure 4.28, les données réelles des employés ainsi que la valeur de leur démission.

	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI								
1	Matricule	ANNE_SORTI	MOIS_SORTI	Moyenne_pr	Nombre_mo	Nombre_mo	Raison_de_c	Destination	Nationalité	Poste	Pôle	Manager	Date_de_Na_Age	SITUATION_IDEM										
2	2	2015	8					TUN	Ingénieur d/Sofrec	Industrie	01/12/1986	31.31	Célibataire		1									
3	3	2013	10					TUN	Ingénieur D/Sofrec	Industrie	08/12/1986	31.3	Célibataire		1									
4	4	1900	1					TUN	Architecte F/Sofrec	Industrie	16/06/1986	31.78	Marié	0										
5	5	2014	6					TUN	Ingénieur D/Sofrec	Industrie	11/09/1986	31.54	Marié	1										
6	6	2016	7					TUN	Architecte t/Sofrec	Industrie	15/01/1986	32.03	Marié	1										
7	7	1900	1					TUN	Architecte s/Sofrec	Industrie	27/04/1984	33.92	Marié	0										
8	8	2017	9					TUN	Architecte sc/Sofrec	Industrie	22/08/1981	36.6	Célibataire	0										
9	9	2015	8					TUN	Ingénieur d/Sofrec	Industrie	10/01/1986	32.21	Célibataire	1										
10	10	2013	7					TUN	Ingénieur D-Sofrec	Industrie	28/01/1986	32	Célibataire	1										
11	11	2015	10					TUN	Ingénieur d/Sofrec	Industrie	26/09/1986	31.5	Célibataire	1										
12	12	2013	6					TUN	Architecte t/Sofrec	Industrie	18/01/1979	39.19	Marié	1										
13	13	2015	3					Changement France																
14	14	1900	1					TUN	Architecte fc/Sofrec	Industrie	15/03/1983	35.04	Marié	1										
15	15	1900	1					TUN	Responsable Sofrec	Industrie	14/01/1984	34.2	Marié	0										
16	16	2014	11					TUN	Ingénieur D-Sofrec	Industrie	22/10/1981	36.43	Marié	0										
17	17	1900	1					TUN	Responsable Sofrec	Industrie	09/01/1983	35.05	Marié	0										
18	18	1900	1					TUN	Responsable Sofrec	Industrie	04/02/1981	37.14	Marié	0										
19	19	1900	1					TUN	Responsable Sofrec	Industrie	21/04/1986	31.93	Marié	0										
20	20	2015	11					TUN	Responsable Sofrec	Industrie	26/06/1984	33.75	Célibataire	1										
21	21	1900	1					TUN	Directeur Pm/Sofrec	Industrie	31/05/1981	36.82	Marié	0										

Figure 4.28: Visualisation des informations réelles des employés

Nous allons appliquer la prédiction sur notre modèle construit. La figure 4.29 nous montre la variable de démission prédictive pour chaque employé.

	A	B
1	DEM	Matricule
2	0.98203915	2
3	1.0709717	3
4	0.02558884	4
5	1.1329253	5
6	1.1404423	6
7	0.009017587	7
8	0.12673901	8
9	1.0928261	9
10	1.0229594	10
11	1.0528553	11
12	1.0211921	12
13	1.0577695	13
14	0.161409	14
15	0.039829165	15
16	1.1147225	16
17	0.006626606	17
18	0.1619541	18
19	0.001809880	19
20	0.88010454	20
21	0.013666332	21

Figure 4.29: Visualisation des sorties prédictives des employés

Ce modèle répond parfaitement à notre problème avec un taux de précision de 0.95.

4.1.5.5 Réalisation

Dans cette partie, nous allons exposer l'interface d'import du fichier contenant les informations des employés pour la construction de notre modèle et dont nous voulons effectuer des prédictions comme la montre la figure 4.30.

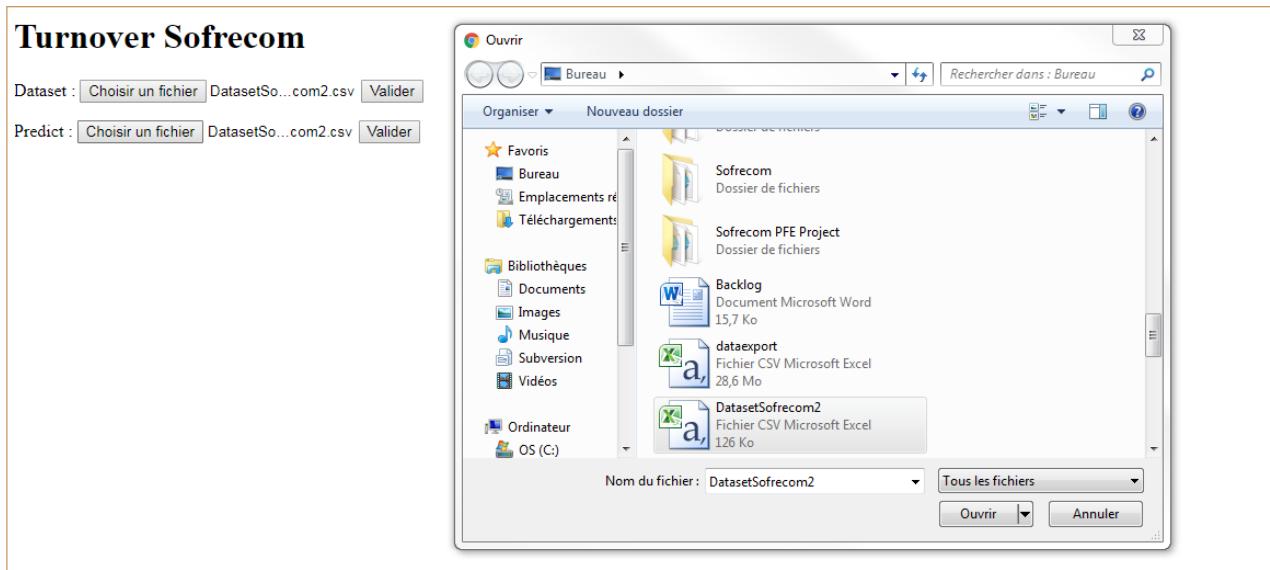


Figure 4.30: Interface d'import des données

Les différentes phases que nous avons abordées précédemment (préparation des données, division des données, création du modèle, prédiction) sont exposées sous format de service web. Exemple d'appel du service web de division des données comme la montre la figure 4.31.

```
127.0.0.1 - - [22/Apr/2018 14:10:33] "GET /split HTTP/1.1" 200 -
```

Figure 4.31: Appel de service web de division des données

Exemple du résultat retourné du service web de prédiction comme la montre la figure 4.32.

```
{"\"DEM\":0.0,\"Matricule\":1023}, {"\"DEM\":0.01,\"Matricule\":1026},  
{"\"DEM\":0.0,\"Matricule\":1029}, {"\"DEM\":0.0,\"Matricule\":1030},  
{"\"DEM\":0.008,\"Matricule\":1035}, {"\"DEM\":0.961,\"Matricule\":2},  
{"\"DEM\":0.979,\"Matricule\":10}, {"\"DEM\":0.964,\"Matricule\":11},  
{"\"DEM\":0.983,\"Matricule\":23}, {"\"DEM\":0.014,\"Matricule\":25},  
{"\"DEM\":0.978,\"Matricule\":33}, {"\"DEM\":0.978,\"Matricule\":39},
```

Figure 4.32: Résultat retourné de service web de prédiction

4.2 Sprint 2 Gestion des prédictions des employés

Ce sprint a pour but de développer les deux modules de notre application qui sont la gestion des prédictions et la gestion des employés.

4.2.1 Backlog du Sprint

Une fois, nous avons défini le but de notre sprint, il est temps de décider quelles sont les user story à inclure dans ce dernier.

Le tableau 4.6 résume le backlog de notre deuxième sprint :

Tableau 4.6: Backlog du Sprint 2

ID	User Story	ID Tâche	Tâche	Estimation (jours)
6	En tant qu'utilisateur, je veux effectuer une prédiction sur la démission d'un employé.	6.1	Création de l'interface de prédiction sur la démission des employés.	1
		6.2	Implémenter les méthodes nécessaires pour l'appel de service web de prédiction et l'affichage de son résultat.	1
		6.3	Tester l'affichage de prédiction.	1
7	En tant qu'utilisateur, je souhaite consulter la liste des employés prédits.	7.1	Création de l'interface graphique de consultation.	1
		7.2	Implémenter les méthodes nécessaires pour la consultation.	1
		7.3	Tester la consultation de la liste des employés prédits.	1
8	En tant qu'utilisateur, je souhaite exporter l'historique des prédictions d'un employé sous format PDF.	8.1	Création de l'interface graphique de consultation de l'historique des prédictions.	1
		8.2	Implémenter les méthodes nécessaires pour la consultation.	1

ID	User Story	ID Tâche	Tâche	Estimation (jours)
		8.3	Tester la consultation de l'historique des prédictions.	1
9	En tant qu'utilisateur, je souhaite rechercher la liste des employés prédits.	9.1	Implémenter les méthodes nécessaires pour la recherche.	1
		9.2	Tester la recherche des prédictions des employés.	1
10	En tant qu'utilisateur, je veux ajouter un employé.	10.1	Création de l'interface graphique d'ajout d'un employé.	1
		10.2	Implémenter les méthodes nécessaires pour l'ajout.	1
		10.3	Tester l'ajout d'un employé.	1
11	En tant qu'utilisateur, je veux modifier un employé.	11.1	Création de l'interface graphique de modification d'un employé.	1
		11.2	Implémenter les méthodes nécessaires pour la modification.	1
		11.3	Tester la modification d'un employé.	1
12	En tant qu'utilisateur, je veux supprimer un employé.	12.1	Création d'un message d'alerte pour la confirmation de suppression d'un employé.	1
		12.2	Implémenter les méthodes nécessaires pour la suppression.	1
		12.3	Tester la suppression d'un employé.	1
13	En tant qu'utilisateur, je souhaite consulter et effectuer une recherche sur la liste des employés.	13.1	Implémenter les méthodes nécessaires pour la recherche d'un employé.	1
		13.2	Tester la recherche des employés.	1

ID	User Story	ID Tâche	Tâche	Estimation (jours)
14	En tant qu'utilisateur, je souhaite exporter toutes les informations d'un employé sous format PDF.	14.1	Implémenter les méthodes nécessaires pour l'exportation des informations d'un employé.	1
		14.2	Tester le format d'exportation.	1

4.2.2 Spécification fonctionnelle

La figure 4.33 présente le diagramme de cas d'utilisation détaillé de «Gérer les prédictions».

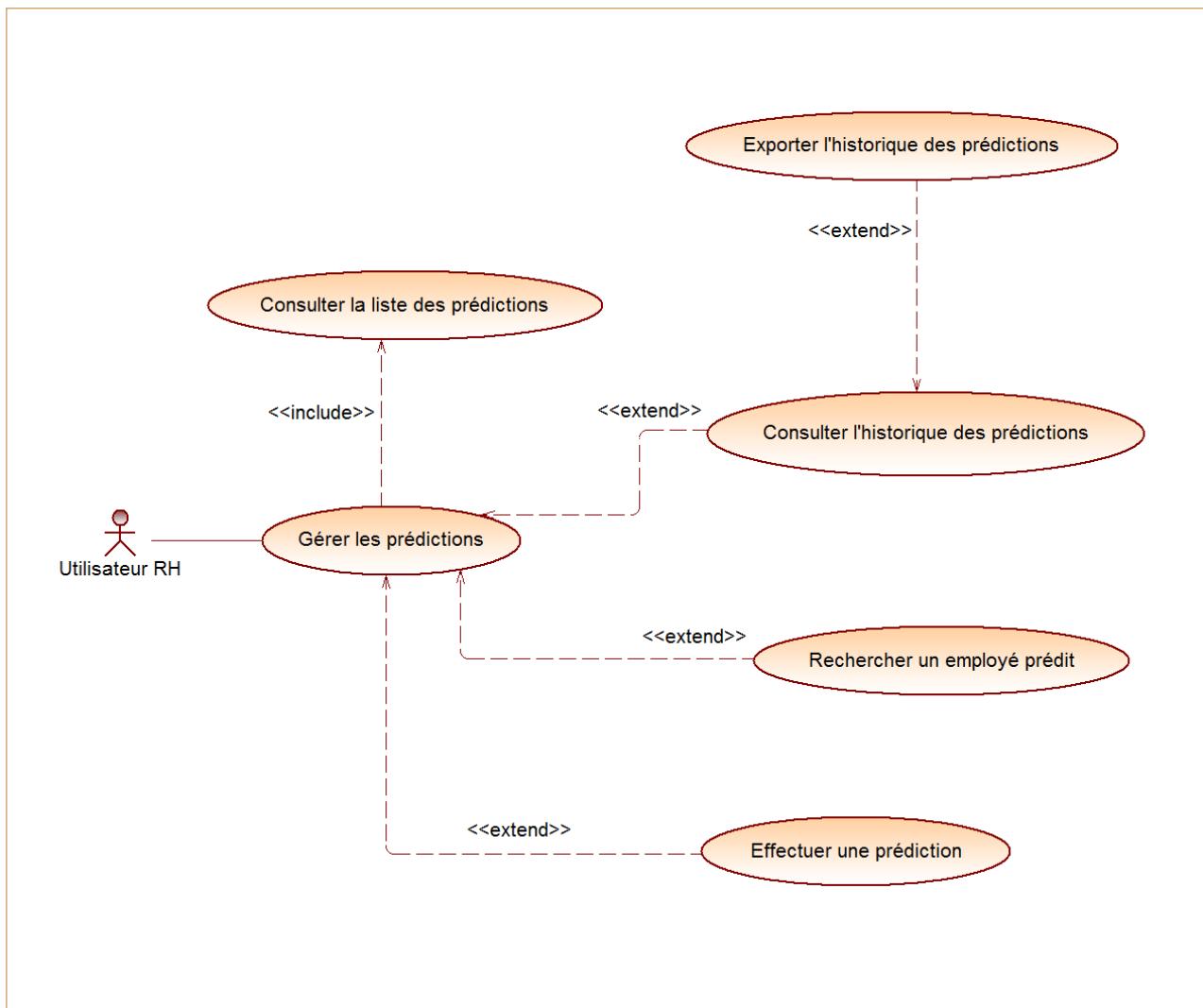


Figure 4.33: Diagramme de cas d'utilisation « Gérer les prédictions »

La figure 4.34 présente le diagramme de cas d'utilisation détaillé de «Gérer les employés».

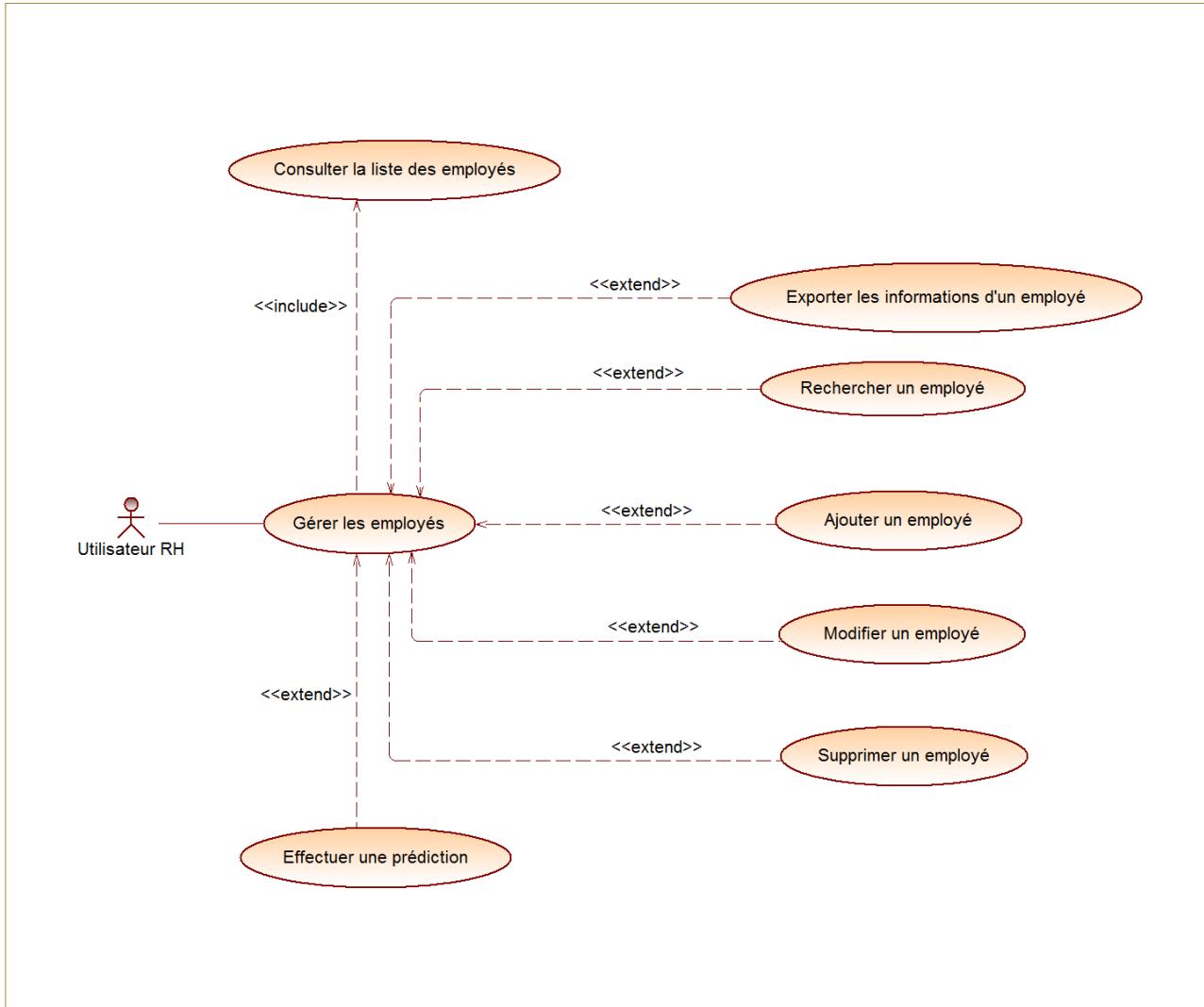


Figure 4.34: Diagramme de cas d'utilisation « Gérer les employés »

Le tableau 4.7 présente la description textuelle du cas d'utilisation « Effectuer une prédition ».

Tableau 4.7: Description du diagramme de cas d'utilisation «Effectuer une prédition»

Cas d'utilisation	Effectuer une prédition.
Acteur	Utilisateur RH ou l'administrateur.
Précondition	L'utilisateur doit accéder à l'interface de gestion des prédictions ou gestion des employés.
Post-condition	Prédiction effectuée avec succès.

Description du scénario	<ul style="list-style-type: none"> — L'utilisateur choisit un employé. — Le système renvoie un message d'alerte pour confirmer la prédiction. — L'utilisateur confirme la prédiction de l'employé choisi. — Le système enregistre le résultat de la prédiction et affiche un message de succès de l'opération.
Exception	Erreur de connexion.

Le tableau 4.8 présente la description textuelle du cas d'utilisation « Exporter les informations d'un employé ».

Tableau 4.8: Description du diagramme de cas d'utilisation « Exporter les informations d'un employé»

Cas d'utilisation	Exporter les informations d'un employé.
Acteur	Utilisateur RH ou l'administrateur.
Précondition	L'utilisateur doit accéder à l'interface de gestion des employés.
Post-condition	Fichier PDF téléchargé.
Description du scénario	<ul style="list-style-type: none"> — L'utilisateur choisit un employé à exporter ses informations. — Le système affiche un message de confirmation. — L'utilisateur confirme l'exportation des informations de l'employé choisi. — Le système retourne un fichier PDF contenant toutes les informations de l'employé choisi.
Exception	Erreur de connexion.

Le tableau 4.9 présente la description textuelle du cas d'utilisation « Ajouter un employé ».

Tableau 4.9: Description du diagramme de cas d'utilisation «Ajouter un employé»

Cas d'utilisation	Ajouter un employé.
Acteur	Utilisateur RH ou l'administrateur.
Précondition	L'utilisateur doit accéder à l'interface de gestion des employés.
Post-condition	Employé ajouté.
Description du scénario	<ul style="list-style-type: none"> — L'utilisateur choisit d'ajouter un nouvel employé. — Le système renvoie le formulaire d'ajout d'un nouvel employé. — L'utilisateur remplit les champs nécessaires et valide la saisie. — Le système enregistre les données et affiche un message de succès.
Exception	Erreur de connexion.

Le tableau 4.10 présente la description textuelle du cas d'utilisation « Supprimer un employé ».

Tableau 4.10: Description du diagramme de cas d'utilisation «Supprimer un employé»

Cas d'utilisation	Supprimer un employé.
Acteur	Utilisateur RH ou l'administrateur.
Précondition	L'utilisateur doit accéder à l'interface de gestion des employés.
Post-condition	Employé et ses prédictions sont supprimés .
Description du scénario	<ul style="list-style-type: none"> — L'utilisateur choisit de supprimer un employé. — Le système affiche un message de confirmation de suppression. — L'utilisateur confirme la suppression de l'employé choisi. — Le système supprime l'employé et ses prédictions et affiche un message de succès.
Exception	Erreur de connexion.

Diagramme de séquences système de l'opération «Effectuer une prédiction»

La figure 4.35 représente le diagramme de séquences système de l'opération «Effectuer une prédiction».

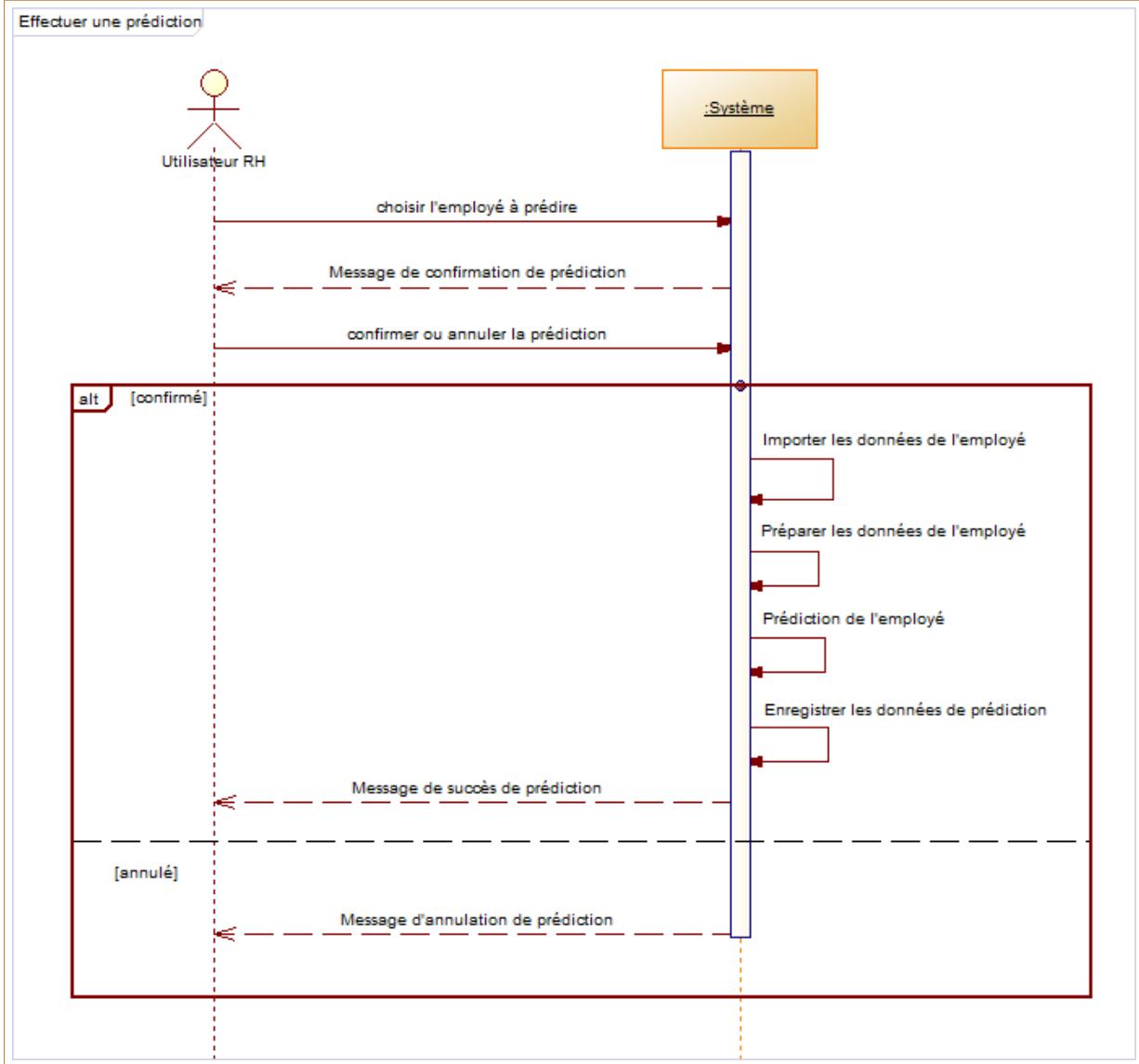


Figure 4.35: Diagramme de séquences système « Effectuer une prédiction »

Diagramme de séquences système de l'opération «Modifier un employé»

La figure 4.36 représente le diagramme de séquences système de l'opération «Modifier un employé».

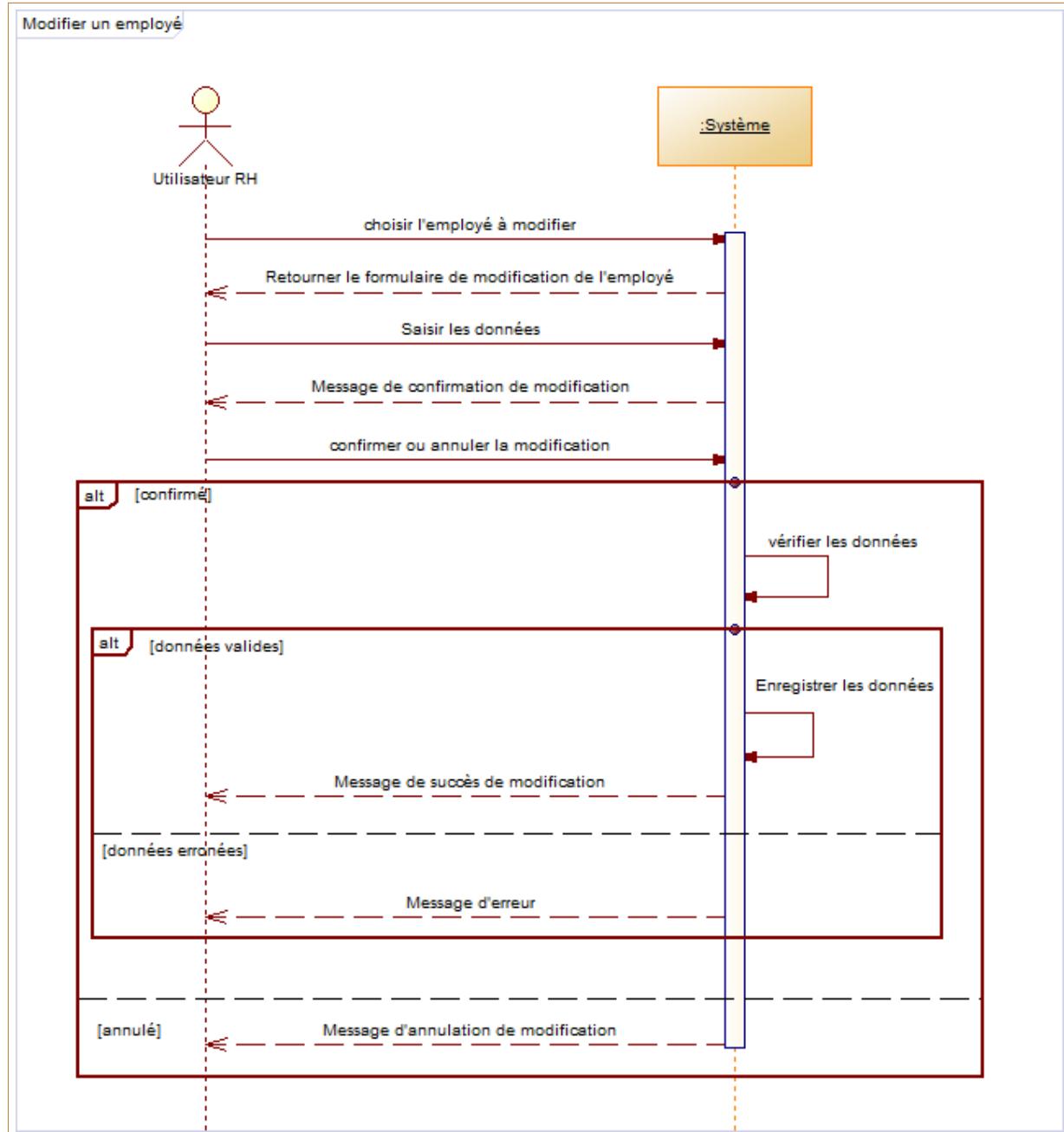


Figure 4.36: Diagramme de séquences système « Modifier un employé »

Diagramme de séquences système de l'opération «Supprimer un employé»

La figure 4.37 représente le diagramme de séquences système de l'opération «Supprimer un employé».

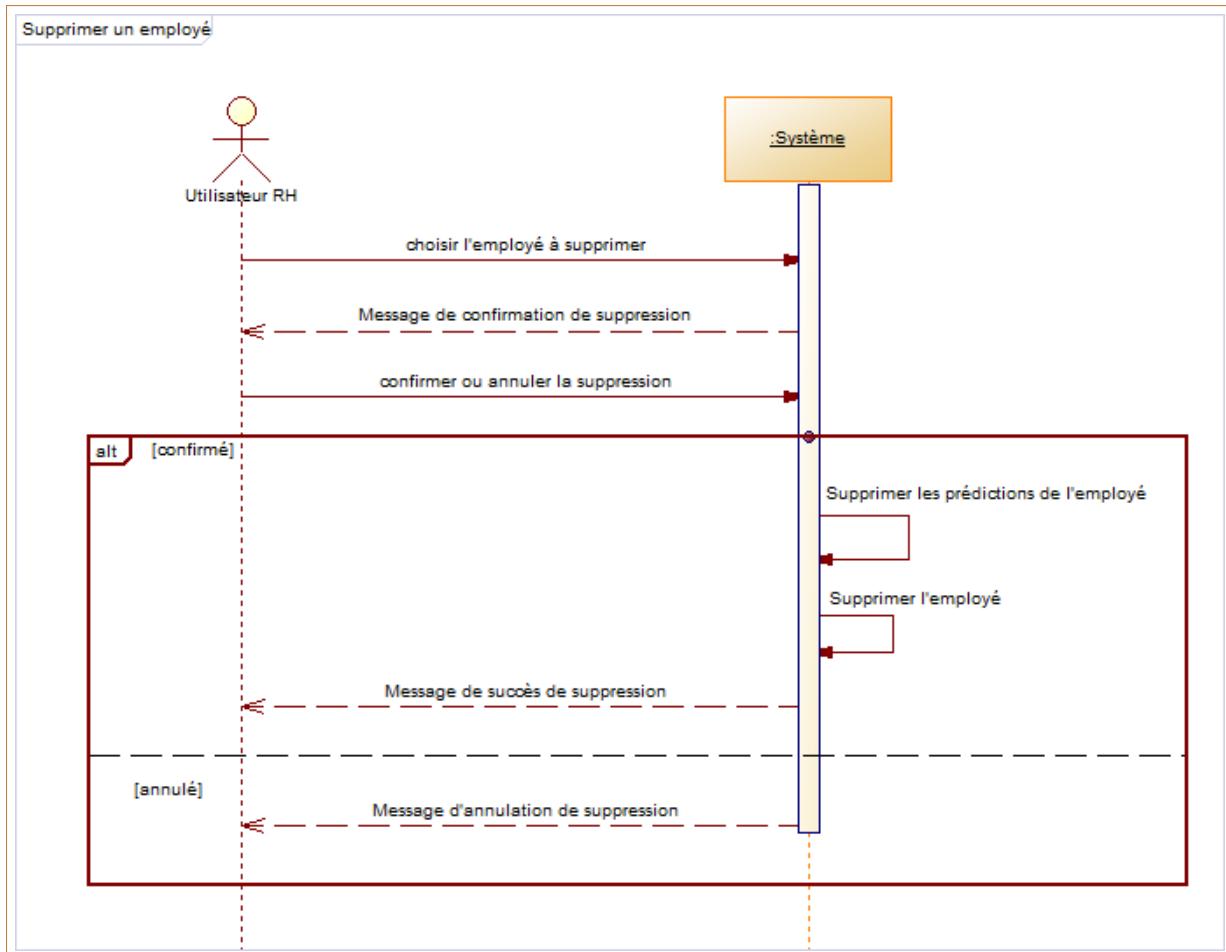


Figure 4.37: Diagramme de séquences système « Supprimer un employé »

4.2.3 Conception

Dans cette section, nous allons présenter le diagramme de classe ainsi que les différents diagrammes de séquences détaillés.

4.2.3.1 Diagramme de classes

La figure 4.38 définit les classes utilisées, pour mettre en place la gestion des prédictions et la gestion des employés.

Pour mieux comprendre ce diagramme, les contrôleurs « EmployeeController » et « PredictionController » jouent le rôle d'intermédiaire entre les entités, le service de prédiction et la vue qui interagit avec l'utilisateur.

Au niveau du modèle, l'entité « EmployeeModel » peut contenir plusieurs prédictions et l'entité « PredictionModel » est associée à un seul employé.

La prédiction des démissions des employés est assurée par le service « ServiceTensorflow ».

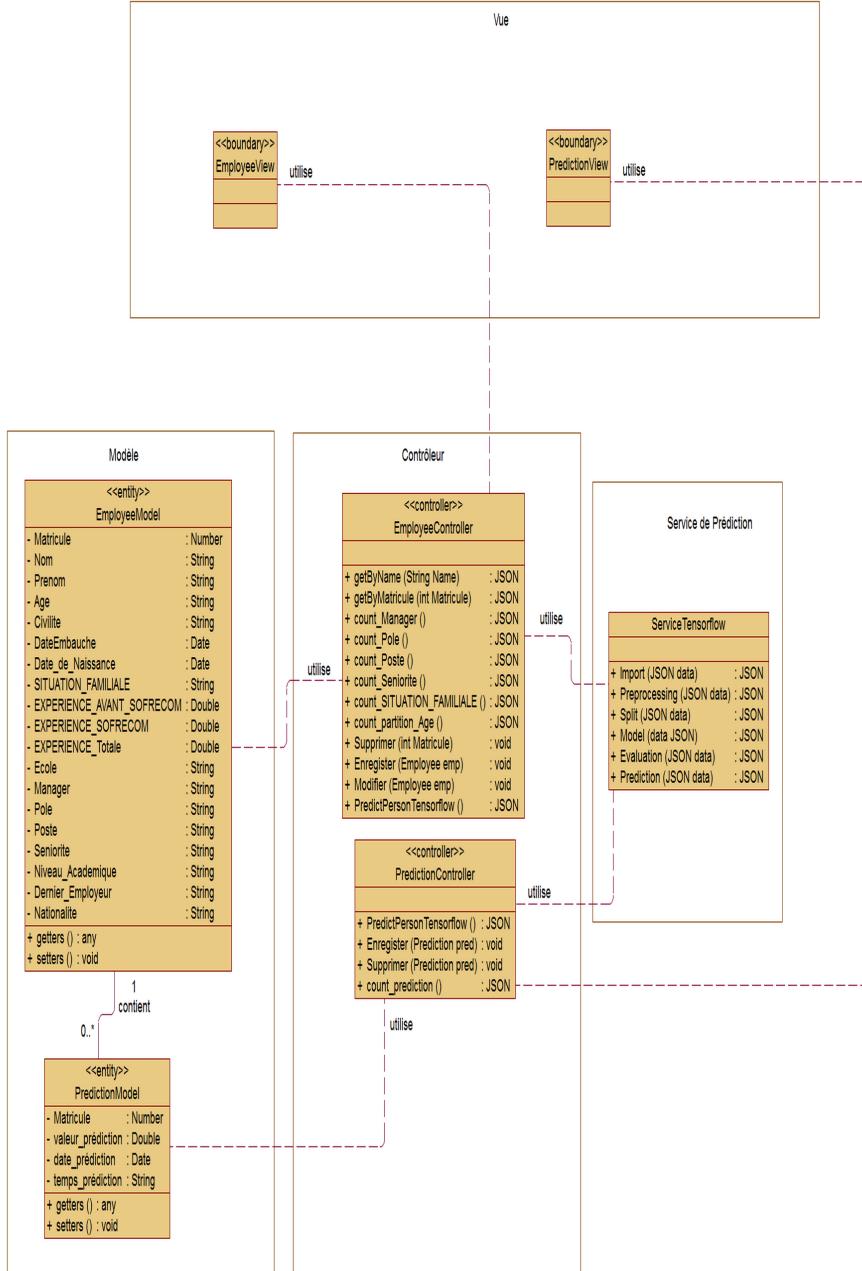


Figure 4.38: Diagramme de classes

4.2.3.2 Diagrammes de séquences

Nous présentons dans ce qui suit les diagrammes de séquences détaillés du deuxième sprint.

Diagramme de séquences objet « Effectuer une prédiction »

La prédiction est la fonctionnalité la plus importante dans notre application.

Pour effectuer ce traitement, l'utilisateur RH choisit un employé à prédire à partir de la vue « PredictionView ». Le contrôleur « PredictionController » récupère les données de l'employé et invoque le service « ServiceTensorflow » pour effectuer la prédiction.

Une fois le traitement est terminé, le contrôleur ordonne d'ajouter le résultat de l'opération dans l'entité « PredictionModel » et le retourne à la vue.

Un message de réussite sera affiché à l'utilisateur en indiquant le résultat de la prédiction.

La figure 4.39 décrit le déroulement du processus de la réalisation de cette tâche.

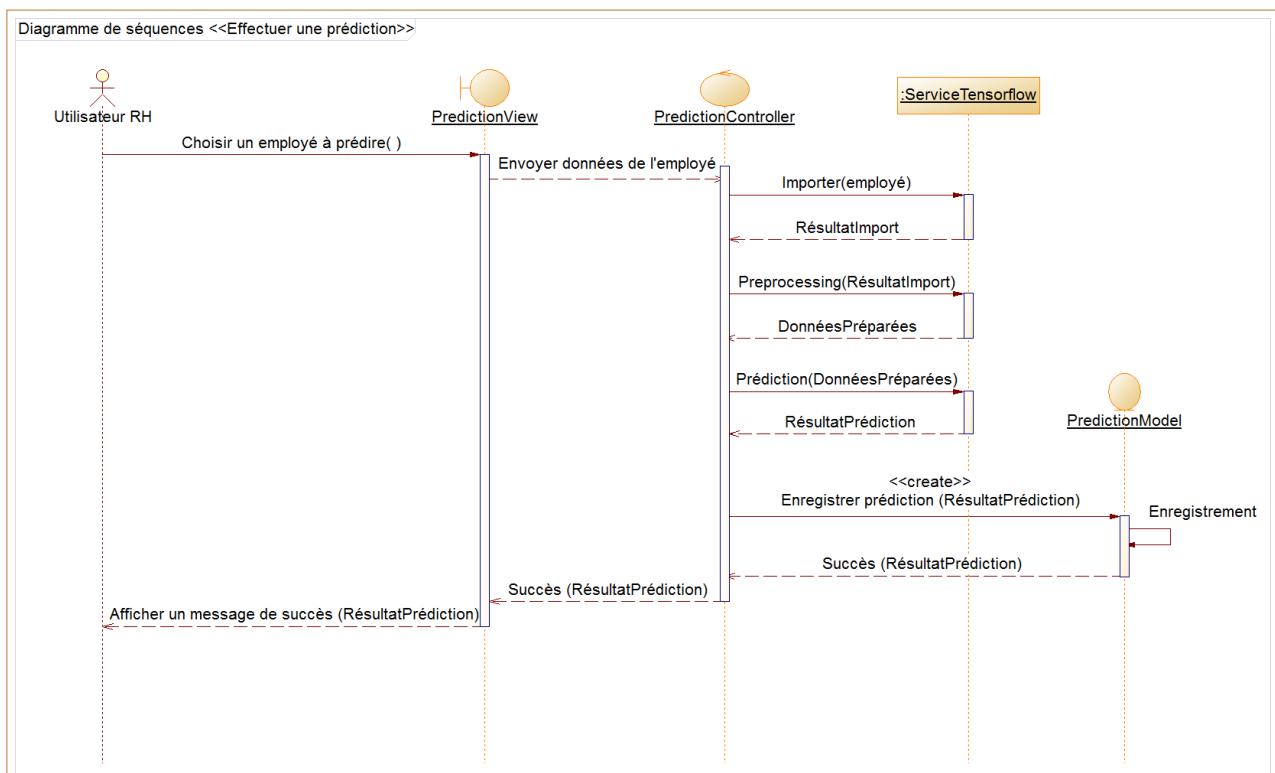


Figure 4.39: Diagramme de séquences objet « Effectuer une prédiction »

Diagramme de séquences objet « Modifier un employé »

L'utilisateur RH peut modifier des données relatives à un employé déjà pris en charge par l'application. Il sélectionne l'employé à modifier à partir de l'interface « EmployeeView » et prend en charge les modifications à apporter.

Le contrôleur « EmployeeController » récupère les informations de l'employé et ordonne la mise à jour de l'employé dans l'entité « EmployeeModel ».

La figure 4.40 expose le déroulement du processus de la réalisation de cette tâche.

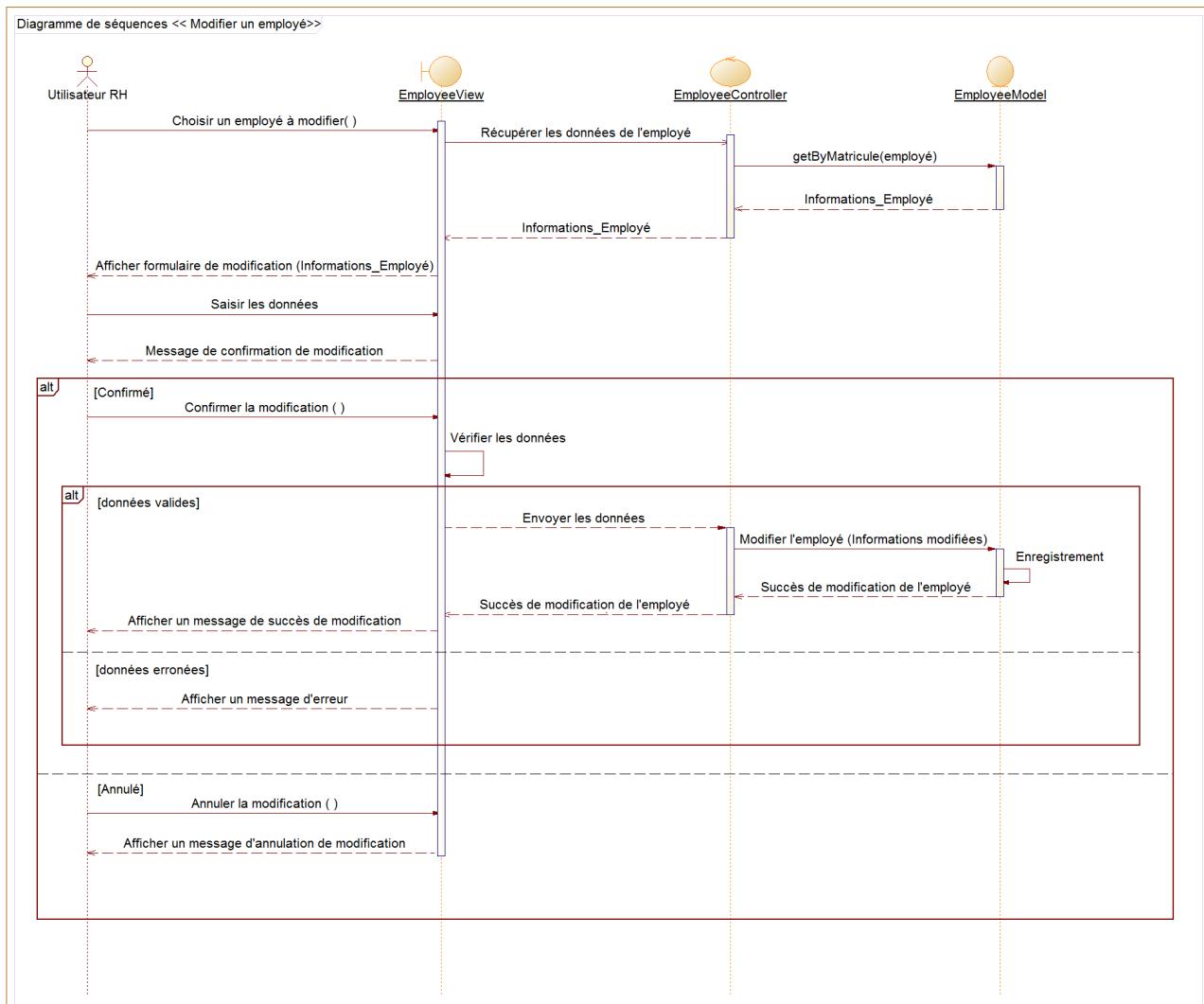


Figure 4.40: Diagramme de séquences objet « Modifier un employé »

Diagramme de séquences objet « Supprimer un employé »

La suppression d'un employé est à la charge de l'utilisateur de l'application. Ce dernier sélectionne l'employé concerné à partir de l'interface « EmployeeView ». Une fois l'employé a été choisi, le contrôleur « EmployeeController » ordonne la suppression des prédictions associées à cet employé afin de respecter les contraintes d'intégrités et puis il ordonne la suppression de l'employé.

La figure 4.41 illustre le déroulement du processus de la réalisation de cette tâche.

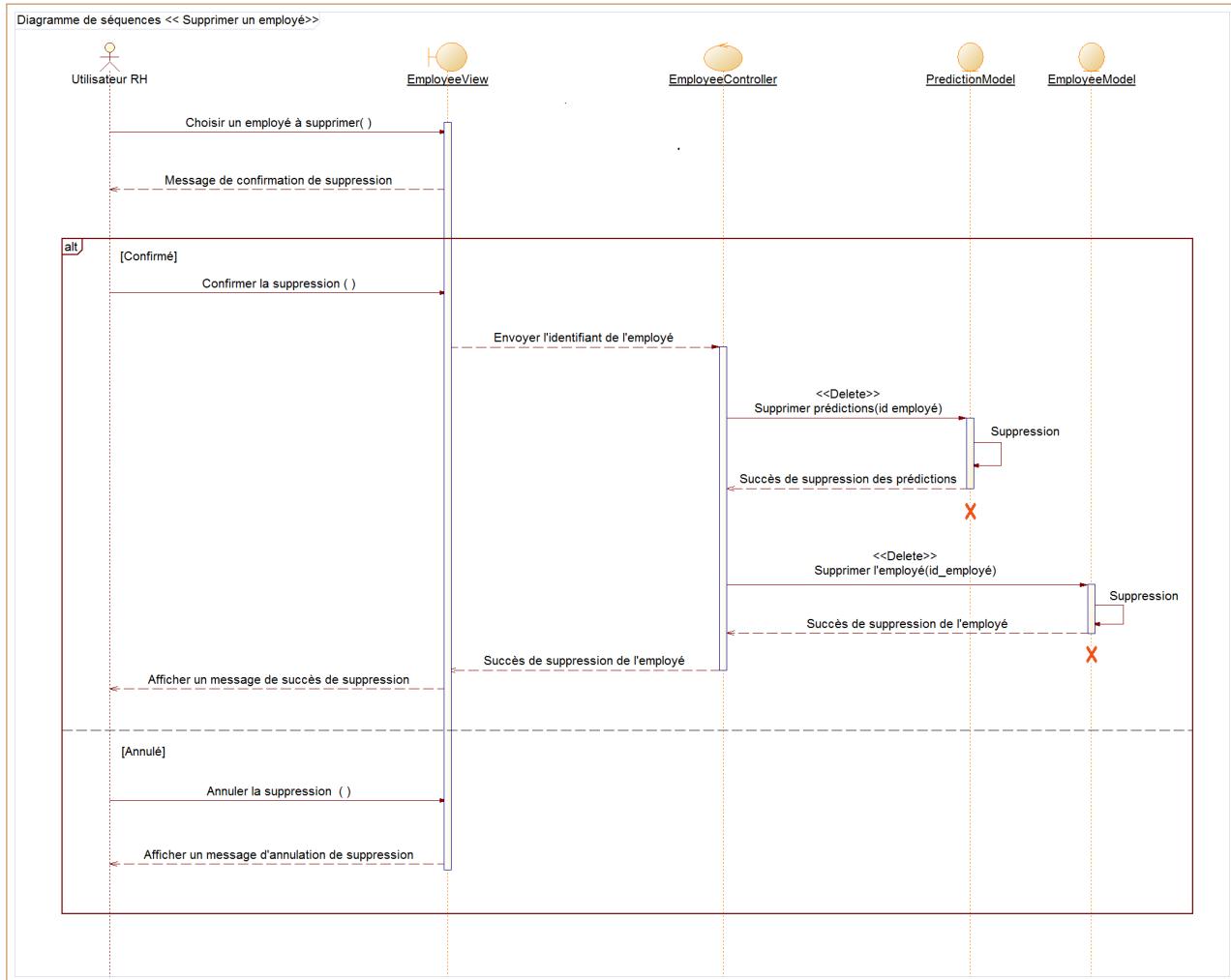


Figure 4.41: Diagramme de séquences objet « Supprimer un employé »

4.2.4 Réalisation

Dans cette partie, nous allons exposer les différentes interfaces de notre application réalisées dans le deuxième sprint.

Interface de gestion des employés

L'utilisateur RH accède à la page de gestion des employés. Il peut consulter, ajouter, prédire, modifier, supprimer un employé ou exporter ses informations comme illustré sur la figure 4.42.

Matricule	NOM	Civilite	SITUATION_FAMILIALE	Date d'Embauche	
2	...	Monsieur	Célibataire	09/11/2011	
3	...	Monsieur	Célibataire	09/11/2011	
4	...	Madame	Marié	09/11/2011	
5	...	Mademoiselle	Marié	09/11/2011	
6	...	Monsieur	Marié	09/11/2011	
7	...	Madame	Marié	14/11/2011	
8	...	Monsieur	Célibataire	14/11/2011	
10	...	Monsieur	Célibataire	21/11/2011	
9	...	Monsieur	Célibataire	14/11/2011	
14	...	Monsieur	Marié	05/12/2011	

Figure 4.42: Interface de gestion des employés

Interface d'ajout d'un nouvel employé

La figure 4.43 représente l'interface permettant à l'utilisateur RH d'ajouter un nouvel employé.

Le formulaire d'ajout contient plusieurs informations (nom, civilité, poste, manager...).

• Ajouter un nouveau Employé

Compléter toutes les informations

Matricole:	4001
NOM:	GHBORBEL
Name:	Sami
Age:	24.11
Civilite:	Choisis Civilité * Monsieur <small>Type : Monsieur</small>
DateEmbauche:	15/06/2018
Date_de_Naissance:	01/06/1993
SITUATION_FAMILIALE:	Choisis SITUATION_FAMILIALE * Célibataire <small>Type : Célibataire</small>

Figure 4.43: Interface d'ajout d'un nouvel employé

Interface de recherche d'un employé

L'utilisateur RH peut effectuer une recherche d'un employé selon la matricule ou le nom de ce dernier comme illustré sur la figure 4.44.

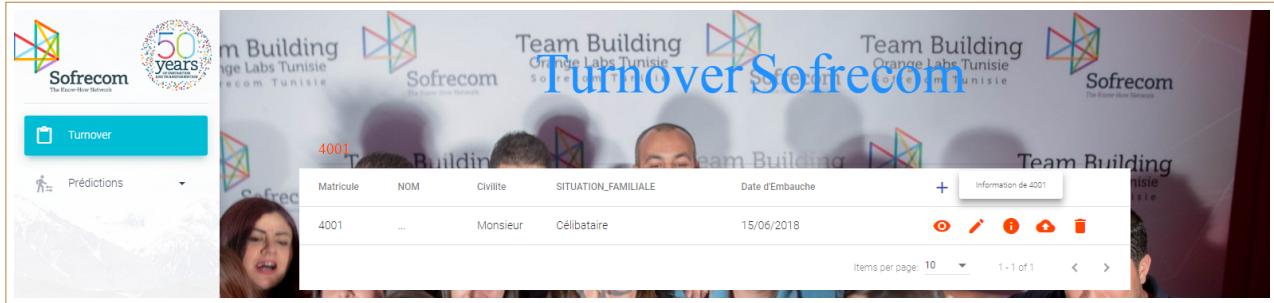


Figure 4.44: Interface de recherche d'un employé

Interface de consultation des informations d'un employé

La figure 4.45 représente l'interface permettant à l'utilisateur RH de consulter les informations d'un employé (nom, prénom, civilité, situation familiale, poste, manager, école...).



Figure 4.45: Interface de consultation des informations d'un employé

Interface de modification des informations d'un employé

Comme illustré par la figure 4.46, l'utilisateur RH peut modifier les informations d'un employé.

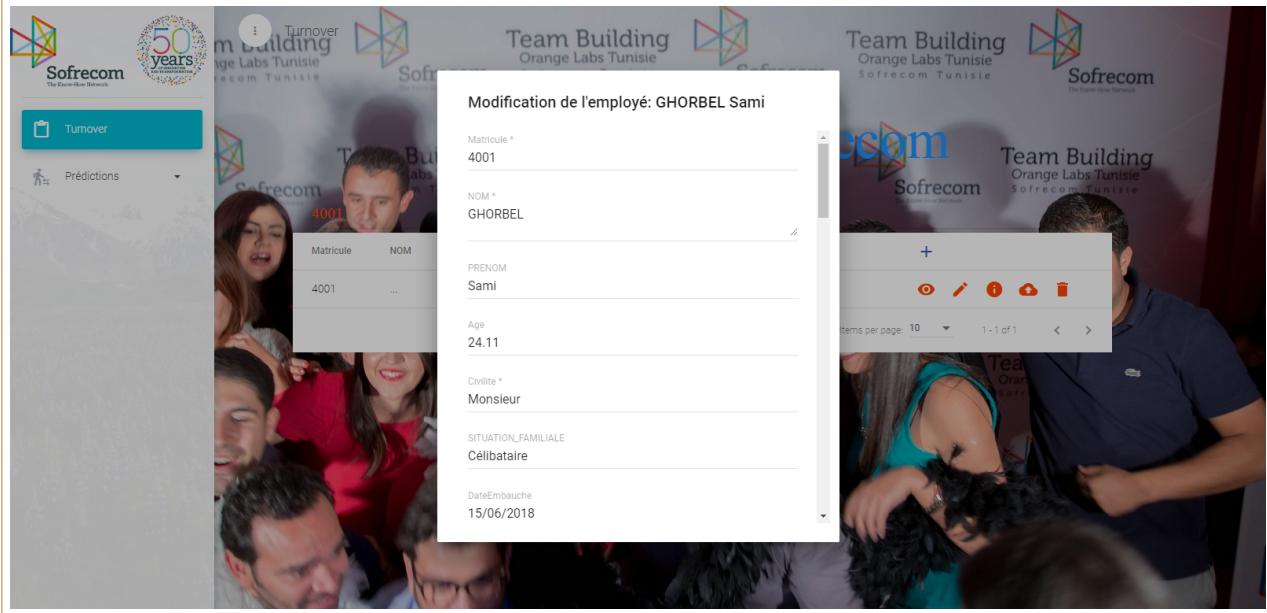


Figure 4.46: Interface de modification des informations d'un employé

Interface de confirmation de prédiction d'un employé

Lors de la prédiction, l'utilisateur RH choisit un employé. Une alerte sera déclenchée pour la confirmation de la prédiction sur l'employé concerné comme le montre la figure 4.47.

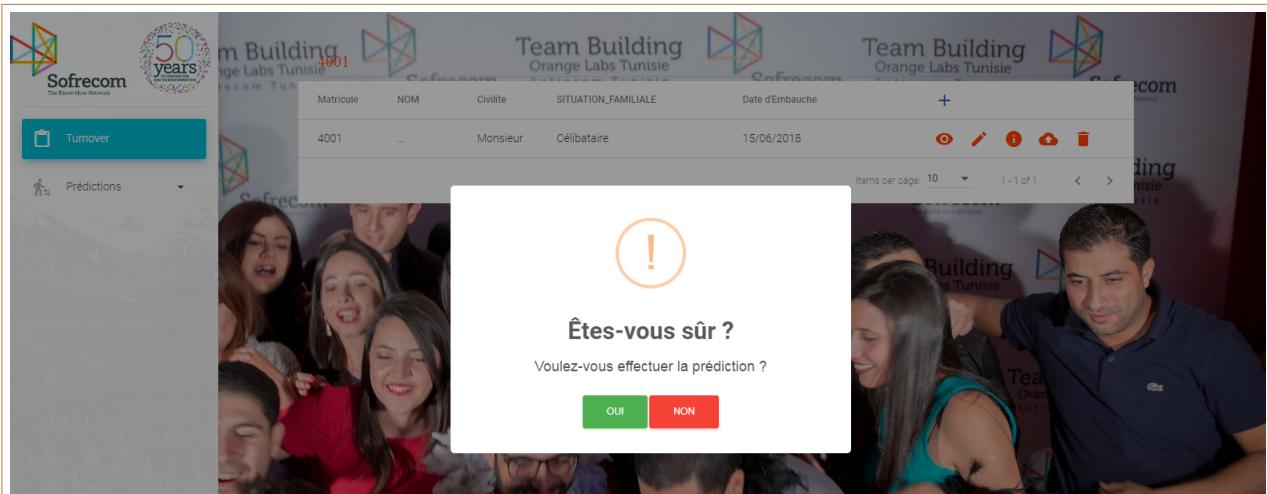


Figure 4.47: Interface de confirmation de prédiction d'un employé

Interface de chargement de prédiction d'un employé

Le traitement de prédiction prend un peu de temps pour retourner un résultat. Une alerte de chargement est affichée à l'utilisateur RH comme le montre la figure 4.48.

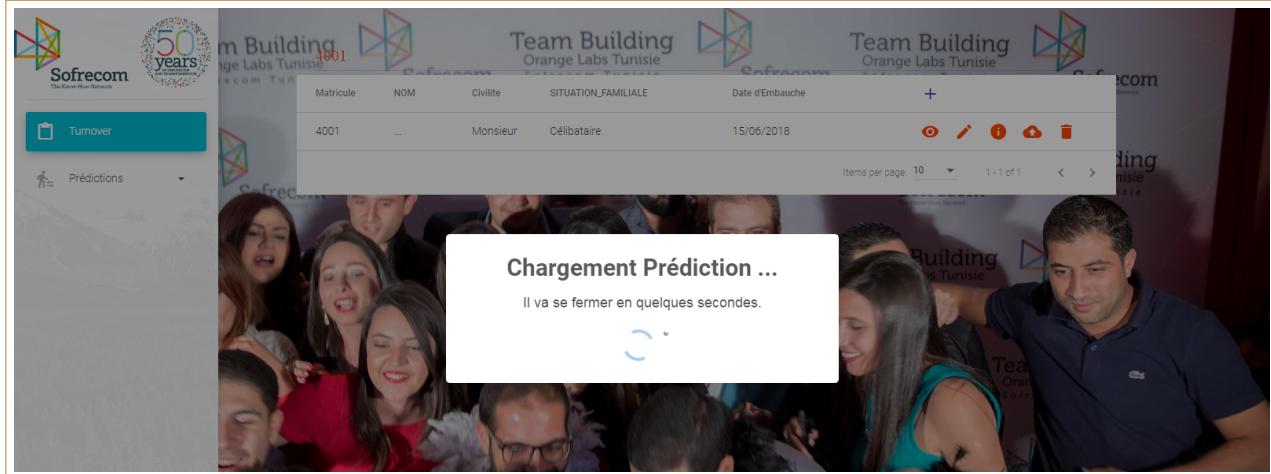


Figure 4.48: Interface de chargement de prédiction d'un employé

Interface de résultat de prédiction d'un employé

Une fois le traitement de prédiction est terminé, une alerte sera déclenchée pour indiquer le résultat retourné comme illustré sur la figure 4.49.

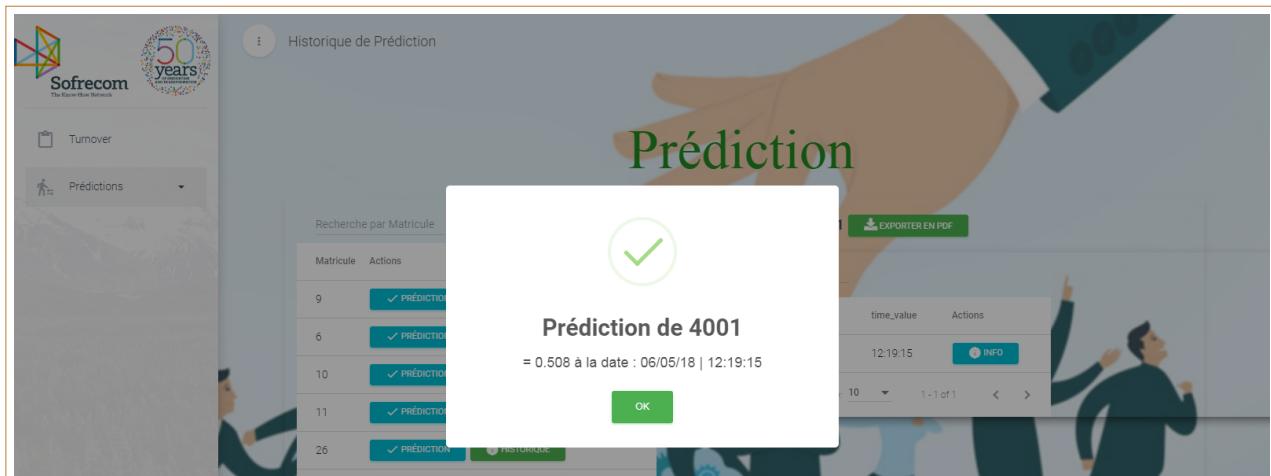


Figure 4.49: Interface de résultat de prédiction d'un employé

Interface de suppression d'un employé

L'utilisateur RH choisit un employé à supprimer. Une alerte de confirmation sera déclenchée pour confirmer la suppression de l'employé concerné comme indiqué sur la figure 4.50.

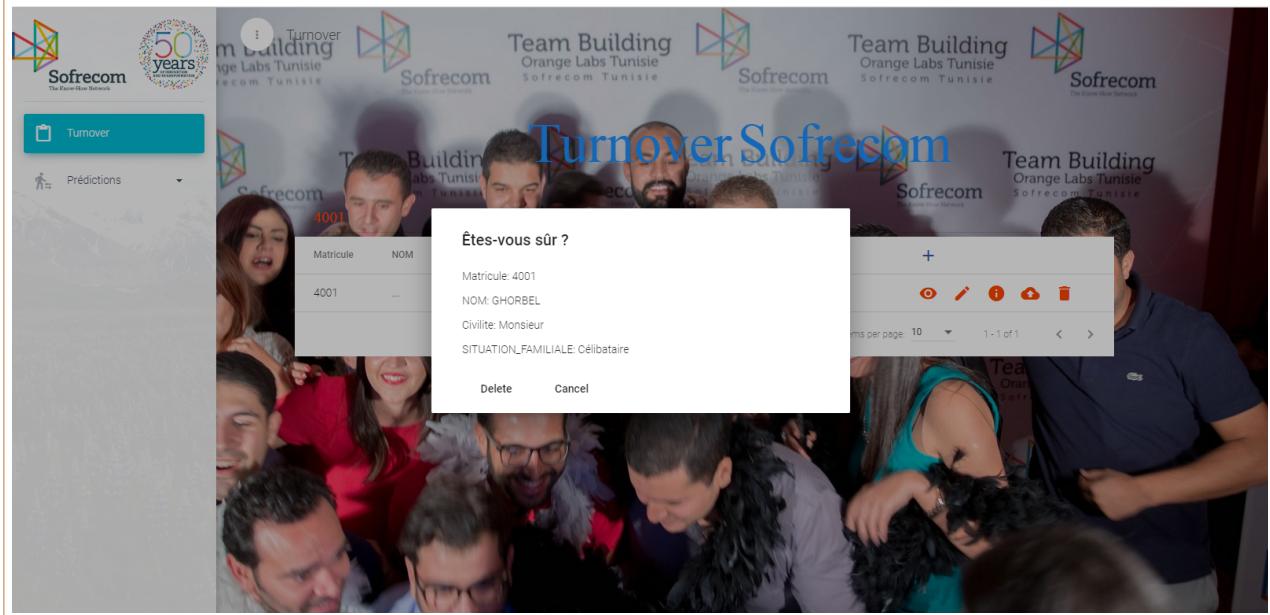


Figure 4.50: Interface de suppression d'un employé

Interface de succès de suppression d'un employé

Dès que l'utilisateur RH confirme la suppression de l'employé, une alerte de succès sera affichée comme le montre la figure 4.51 .

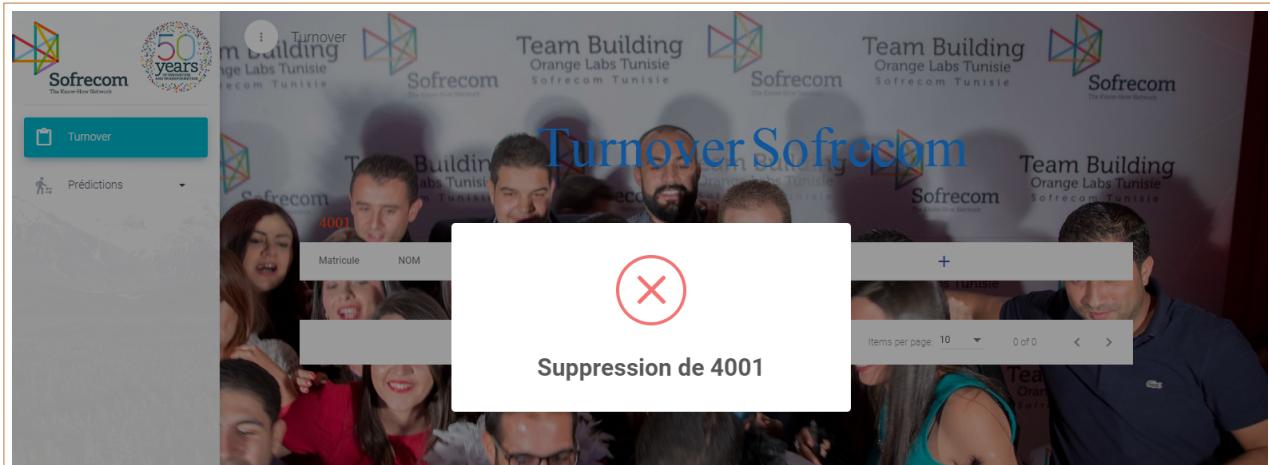


Figure 4.51: Interface de succès de suppression d'un employé

Interface de consultation des employés prédis

Comme le montre la figure 4.52, l'utilisateur RH peut consulter la liste des employés dont les prédictions y figurent.

The screenshot shows the Sofrecom HR application interface. On the left, there is a sidebar with icons for Turnover, Prédictions (selected), Model, and Liste Prédictions. The main area has a large green header 'Prédiction' with a background illustration of several people in business attire. A search bar at the top says 'Recherche par Matricule'. Below it is a table titled 'Matricule Actions' containing 14 rows of employee data. Each row includes a checkbox labeled 'PRÉDICTION', a green button labeled 'HISTORIQUE', and employee details like matricule numbers (e.g., 9, 6, 10, 11, 26, 225, 220, 74, 136, 272). At the bottom of the table are dropdowns for 'Items per page' (set to 10) and a page number indicator '1 - 10 of 67'.

Figure 4.52: Interface de consultation des employés prédis

Interface de consultation de l'historique des prédictions d'un employé

L'utilisateur RH peut consulter l'historique des prédictions de chaque employé comme indiqué sur la figure 4.53.

This screenshot shows the Sofrecom HR application interface focusing on the historical predictions for employee 4001. The sidebar and overall layout are similar to Figure 4.52. The main area features a large green header 'Prédiction' with a background illustration. A sub-header 'Historique de Prédiction' is visible above a table. The table is titled '4001' and has columns 'Matricule' and 'Actions'. It shows one entry for employee 4001 with a 'PRÉDICTION' button and an 'HISTORIQUE' button. To the right, a section titled 'Les informations de 4001' contains a table with columns 'predict_value', 'date_value', 'time_value', and 'Actions'. This table lists two entries with values 0.508, 06/05/18, and 12:19:15, each with an 'INFO' button. Navigation controls like 'Items per page' (10) and a page indicator '1 - 2 of 2' are also present.

Figure 4.53: Interface de consultation de l'historique des prédictions d'un employé

Interface d'export de l'historique des informations d'un employé sous format PDF

La figure 4.54 représente l'interface permettant à l'utilisateur RH de consulter l'historique des prédictions et des informations d'un employé exporté sous format PDF.



Figure 4.54: Interface d'export de l'historique des informations d'un employé sous format PDF

4.2.5 Test et validation

Nous avons rassemblé dans le tableau 4.11 un ensemble de scénarios de cas de tests fonctionnels associés au sprint 2.

Tableau 4.11: Tests fonctionnels du sprint 2

Cas de test	Démarches	Comportement attendu	Résultat
Effectuer une prédiction	Demander la page de gestion des employés. valider la prédiction d'un employé.	Affichage de la page de gestion des employés. Chargement et affichage du résultat de la prédiction.	Conforme

Cas de test	Démarches	Comportement attendu	Résultat
Ajouter un employé	Demander la page de gestion des employés. Cliquer sur ajouter. Remplir et valider le formulaire d'ajout.	Affichage de la page de gestion des employés. Affichage de formulaire d'ajout. Ajout effectué.	Conforme
Modifier un employé	Demander la page de gestion des employés. Cliquer sur modifier. Saisir et valider le formulaire de modification .	Affichage de la page de gestion des employés. Affichage de formulaire de modification. Modification effectuée.	Conforme
Supprimer un employé	Demander la page de gestion des employés. Cliquer sur supprimer. Confirmer la suppression.	Affichage de la page de gestion des employés. Affichage d'un message de confirmation de suppression. Suppression effectuée.	Conforme

Après avoir vérifié le bon fonctionnement du code et des fonctionnalités, le deuxième livrable a été testé et validé par le Product Owner. Il est à noter que plusieurs autres tests ont été effectués et conclus à conforme tels que :

- consulter la liste des employés prédicts ;
- exporter l'historique des prédictions d'un employé ;
- rechercher un employé prédit ;
- exporter toutes les informations d'un employé.

Conclusion

À travers ce chapitre, nous avons réussi à produire un incrément ayant suffisamment de valeur pour le client et pourra être utilisé dans un environnement de production. Le chapitre suivant sera consacré pour produire une nouvelle release couvrant les fonctionnalités d'administration et visualisation.

RELEASE 2 ADMINISTRATION ET VISUALISATION

Plan

1	Sprint 3 Authentification et gestion des utilisateurs	74
2	Sprint 4 Visualisation du tableau de bord	89

Introduction

Ce chapitre fait l'objet d'une description du deuxième release du projet qui est la réalisation de module d'administration (authentification et gestion des utilisateurs) et de visualisation (tableau de bord).

Il est constitué de deux sprints, à savoir le sprint 3 : authentification et gestion des utilisateurs et le sprint 4 : Visualisation de tableau de bord.

L'étude de chaque sprint couvre l'analyse, la conception, la réalisation et les tests fonctionnels.

5.1 Sprint 3 Authentification et gestion des utilisateurs

Ce sprint a pour but de développer les parties d'authentification et de gestion des utilisateurs.

5.1.1 Backlog du Sprint

Dans cette partie, nous allons présenter à travers le tableau 5.1 les différents user story pour le sprint 3.

Tableau 5.1: Backlog du Sprint 3

ID	User Story	ID Tâche	Tâche	Estimation (jours)
15	En tant qu'utilisateur, je dois m'authentifier pour accéder à l'application.	15.1	Implémenter le service AuthService.	3
		15.2	Implémenter les méthodes nécessaires pour l'authentification.	1
		15.3	Préparer l'interface d'authentification.	1
		15.4	Tester l'authentification.	1
16	En tant qu'administrateur, je veux ajouter un utilisateur.	16.1	Implémenter les méthodes nécessaires pour l'ajout.	1
		16.2	Préparer l'interface graphique.	1
		16.3	Tester l'ajout d'un compte utilisateur.	1

ID	User Story	ID Tâche	Tâche	Estimation (jours)
17	En tant qu'administrateur, je veux modifier un utilisateur.	17.1	Implémenter les méthodes nécessaires pour la modification.	1
		17.2	Préparer l'interface graphique.	1
		17.3	Tester la modification d'un compte utilisateur.	1
18	En tant qu'administrateur, je veux supprimer un utilisateur.	18.1	Implémenter les méthodes nécessaires pour la suppression.	1
		18.2	Tester la suppression d'un compte utilisateur.	1
19	En tant qu'administrateur, je souhaite consulter et effectuer une recherche sur la liste des utilisateurs.	19.1	Implémenter les méthodes nécessaires pour la recherche et la consultation.	1
		19.2	Préparer l'interface graphique.	1
20	En tant qu'administrateur, je souhaite attribuer un rôle à un utilisateur.	20.1	Implémenter les méthodes nécessaires pour la gestion des rôles.	2
		20.2	Préparer l'interface graphique.	2

5.1.2 Spécification fonctionnelle

Le diagramme de la figure 5.1 clarifie le fonctionnement de base d'un administrateur en gérant les utilisateurs. Il doit s'authentifier pour pouvoir accéder à la gestion, il peut ajouter, modifier ou supprimer un compte utilisateur.

La figure 5.1 présente le cas d'utilisation détaillé de « Gérer les utilisateurs ».

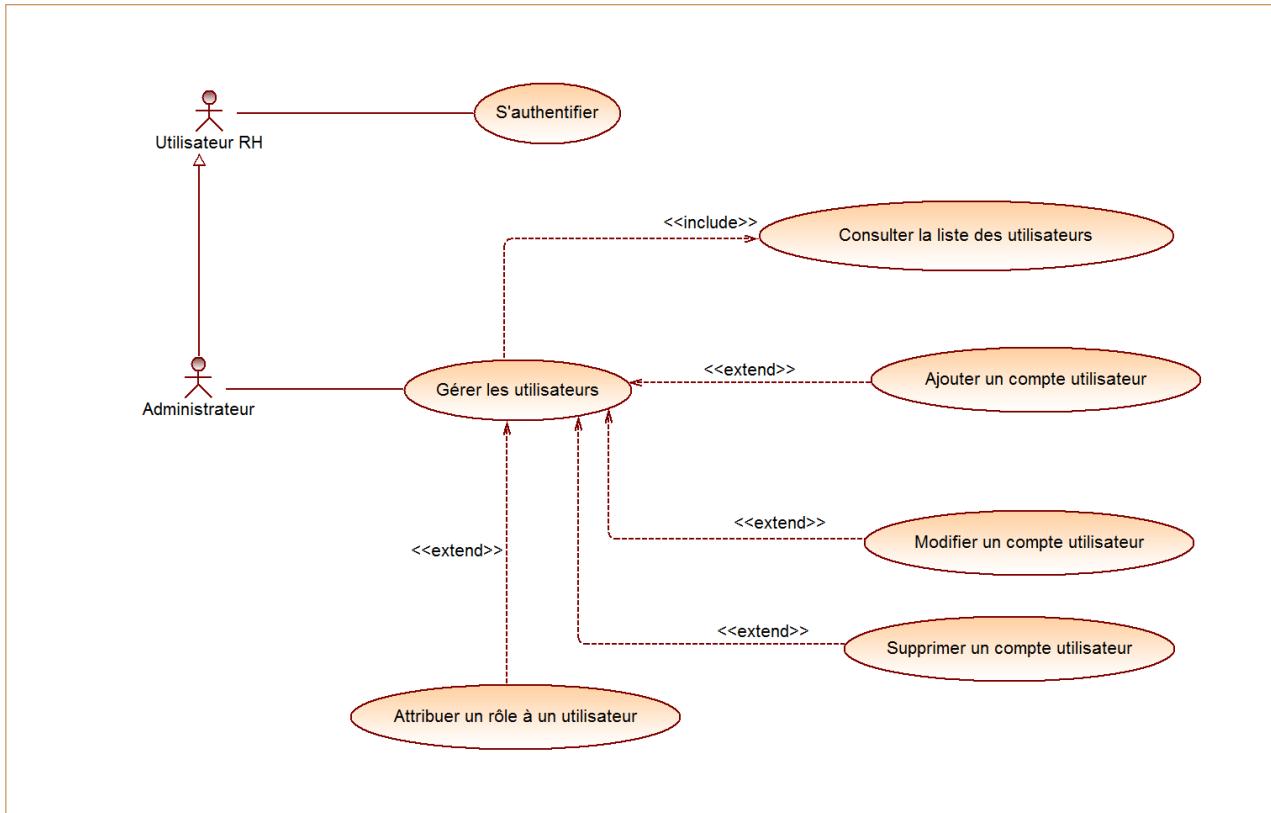


Figure 5.1: Diagramme de cas d'utilisation « Gérer les utilisateurs »

Le tableau 5.2 présente la description textuelle du cas d'utilisation «s'authentifier».

Tableau 5.2: Description du diagramme de cas d'utilisation «s'authentifier»

Cas d'utilisation	s'authentifier.
Acteur	Administrateur, Utilisateur RH.
Précondition	L'utilisateur doit saisir son code (« login») et son mot de passe.
Post-condition	L'utilisateur est authentifié.
Description du scénario	<ul style="list-style-type: none"> — L'utilisateur accède à la page de l'authentification. — L'utilisateur saisit son code (« login») et son mot de passe. — L'application vérifie les coordonnées de l'utilisateur dans la base de données. — L'application donne l'autorisation à l'utilisateur pour accéder à son propre espace de travail.

Cas alternatif	L'application renvoie l'utilisateur à la page de l'authentification suivie d'un message d'erreur.
----------------	---

Le tableau 5.3 présente la description textuelle du cas d'utilisation «Consulter la liste des utilisateurs».

Tableau 5.3: Description du diagramme de cas d'utilisation «Consulter la liste des utilisateurs»

Cas d'utilisation	Consulter la liste des utilisateurs.
Acteur	Administrateur.
Précondition	L'administrateur s'est authentifié.
Post-condition	<ul style="list-style-type: none"> — Liste des utilisateurs affichée. — Liste des utilisateurs filtrée.
Description du scénario	<ul style="list-style-type: none"> — L'administrateur choisit de consulter la liste des utilisateurs. — Le système récupère la liste des utilisateurs. — L'administrateur saisit le nom d'utilisateur pour filtrer la liste. — Liste des utilisateurs filtrée s'affiche.
Exception	Erreur de connexion.

Le tableau 5.4 présente la description textuelle du cas d'utilisation «Ajouter un compte utilisateur».

Tableau 5.4: Description du diagramme de cas d'utilisation «Ajouter un compte utilisateur»

Cas d'utilisation	Ajouter un compte utilisateur.
Acteur	Administrateur.
Précondition	L'administrateur s'est authentifié.
Post-condition	Compte utilisateur ajouté avec succès.

Description du scénario	<ul style="list-style-type: none"> — L'administrateur choisit d'ajouter un nouveau compte utilisateur. — Le système renvoie le formulaire d'ajout d'un compte utilisateur. — L'administrateur remplit les champs nécessaires et valide la saisie. — Le système vérifie les champs. — Le système enregistre les données et affiche la liste des utilisateurs mise à jour.
Exception	Erreur de connexion.

Le tableau 5.5 présente la description textuelle du cas d'utilisation «Supprimer un compte utilisateur».

Tableau 5.5: Description du diagramme de cas d'utilisation «Supprimer un compte utilisateur»

Cas d'utilisation	Supprimer un compte utilisateur.
Acteur	Administrateur.
Précondition	L'administrateur s'est authentifié.
Post-condition	<ul style="list-style-type: none"> — Compte utilisateur sélectionné est supprimé. — Liste des utilisateurs affichée est mise à jour.
Description du scénario	<ul style="list-style-type: none"> — L'administrateur choisit un compte utilisateur à supprimer. — Le système affiche un message de confirmation. — L'administrateur confirme la suppression. — Le système retourne la liste des utilisateurs mise à jour.
Exception	Erreur de connexion.

5.1.3 Conception

5.1.3.1 Diagramme de classes

La figure 5.2 définit les nouvelles classes utilisées en bleu pour le sprint 3.

Le contrôleur « UserController » joue le rôle d'intermédiaire entre l'entité « UserModel » et la vue « UserView » qui interagit avec l'utilisateur. Au niveau du modèle, l'entité « UserModel » peut gérer plusieurs employés et l'entité « EmployeeModel » est gérée par un seul utilisateur.

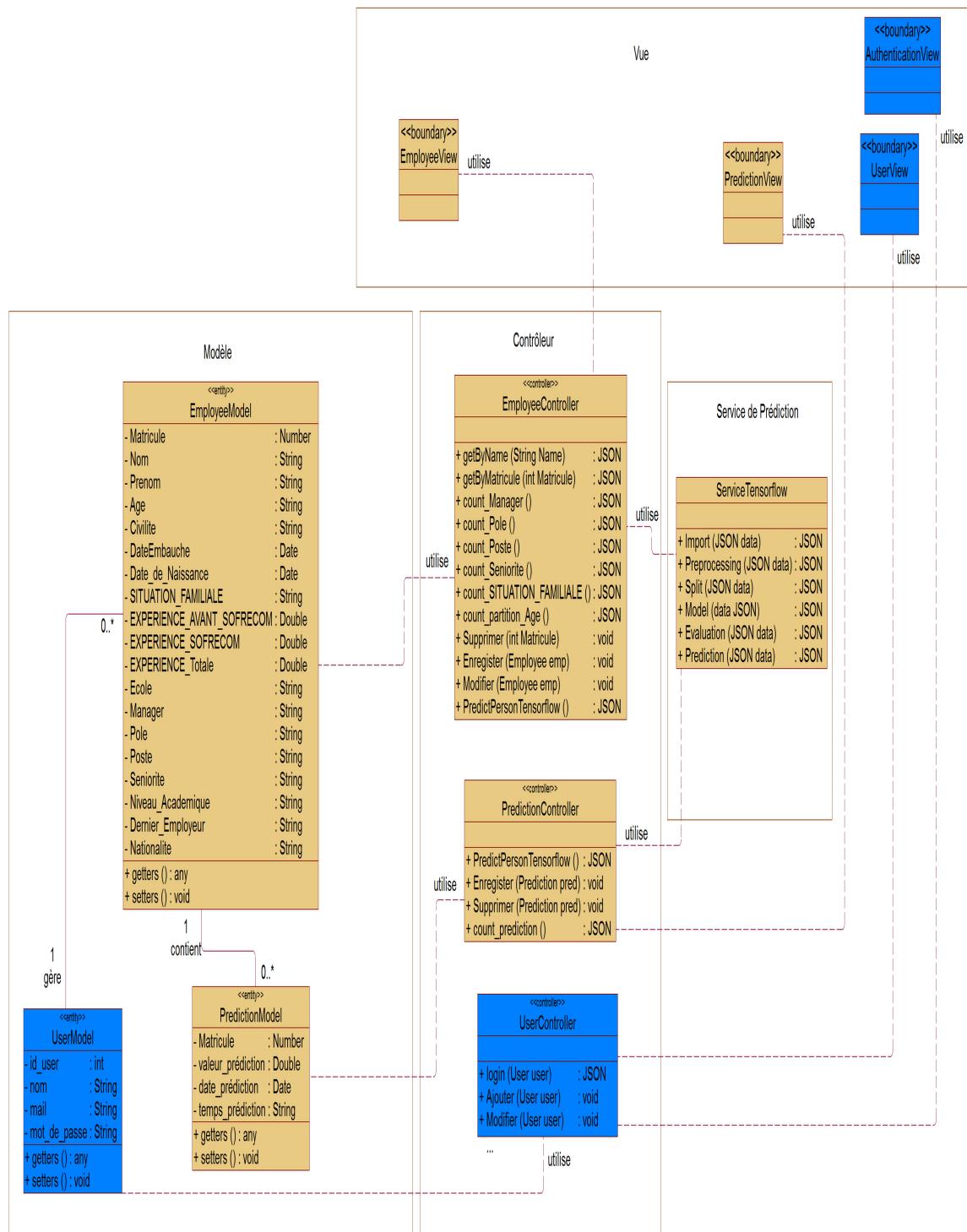


Figure 5.2: Diagramme de classes

5.1.3.2 Diagrammes de séquences

Diagramme de séquences objet « S'authentifier »

L'authentification est indispensable pour gérer l'accès des utilisateurs et la sécurité de l'application.

Les différentes étapes de cette fonctionnalité sont représentées dans la figure 5.3.

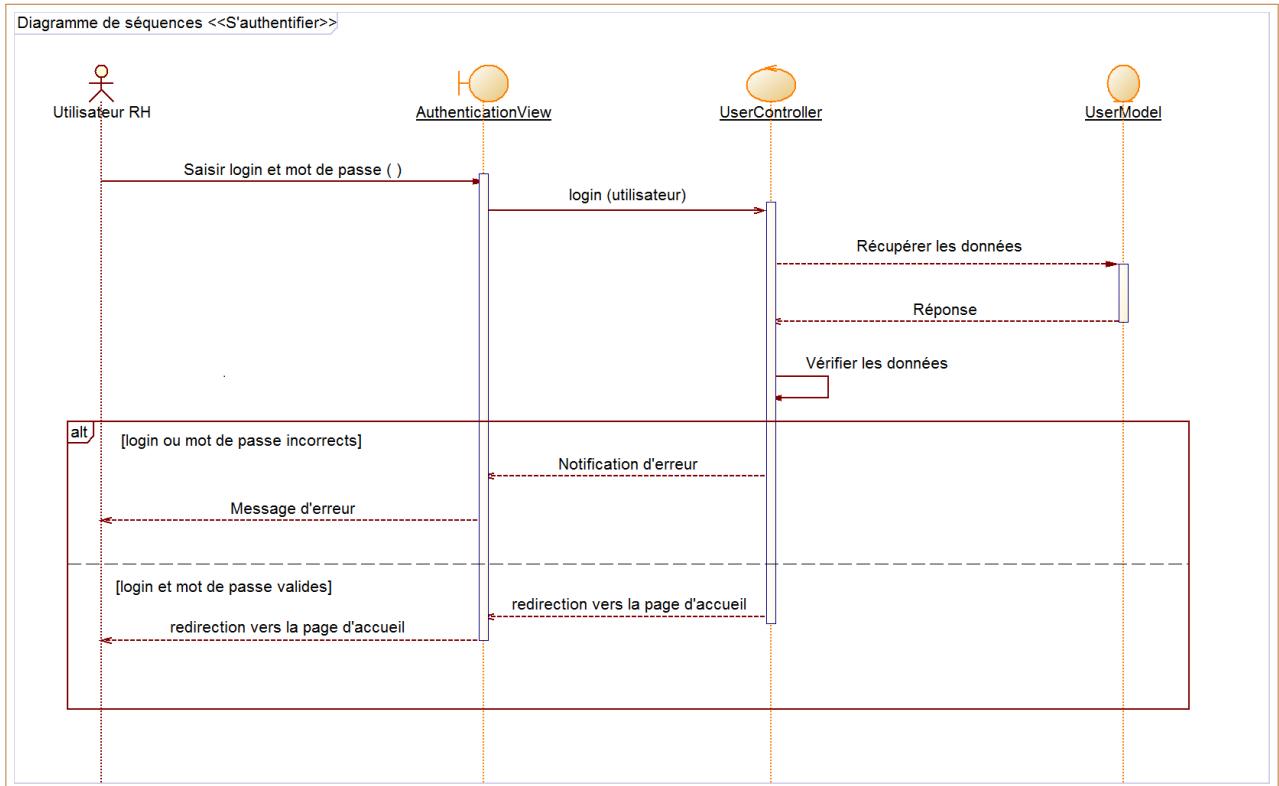


Figure 5.3: Diagramme de séquences objet « s'authentifier »

Diagramme de séquences objet « Ajouter un compte utilisateur »

Chaque acteur intervenant dans l'application dispose d'un compte utilisateur. L'ajout d'un nouvel utilisateur (compte utilisateur) est assuré par l'administrateur de l'application. Ce dernier saisit les données relatives au nouvel utilisateur dans l'interface « UserView ». Une fois saisies, ces données sont prises en charge par le contrôleur « UserController » qui procède à la vérification de leur validité. Une fois ces données validées, le contrôleur ordonne la création d'un nouveau compte utilisateur dans l'entité « UserModel ». Si ce n'est pas le cas, un message d'erreur est affiché.

La figure 5.4 illustre le déroulement du processus de la réalisation de cette tâche.

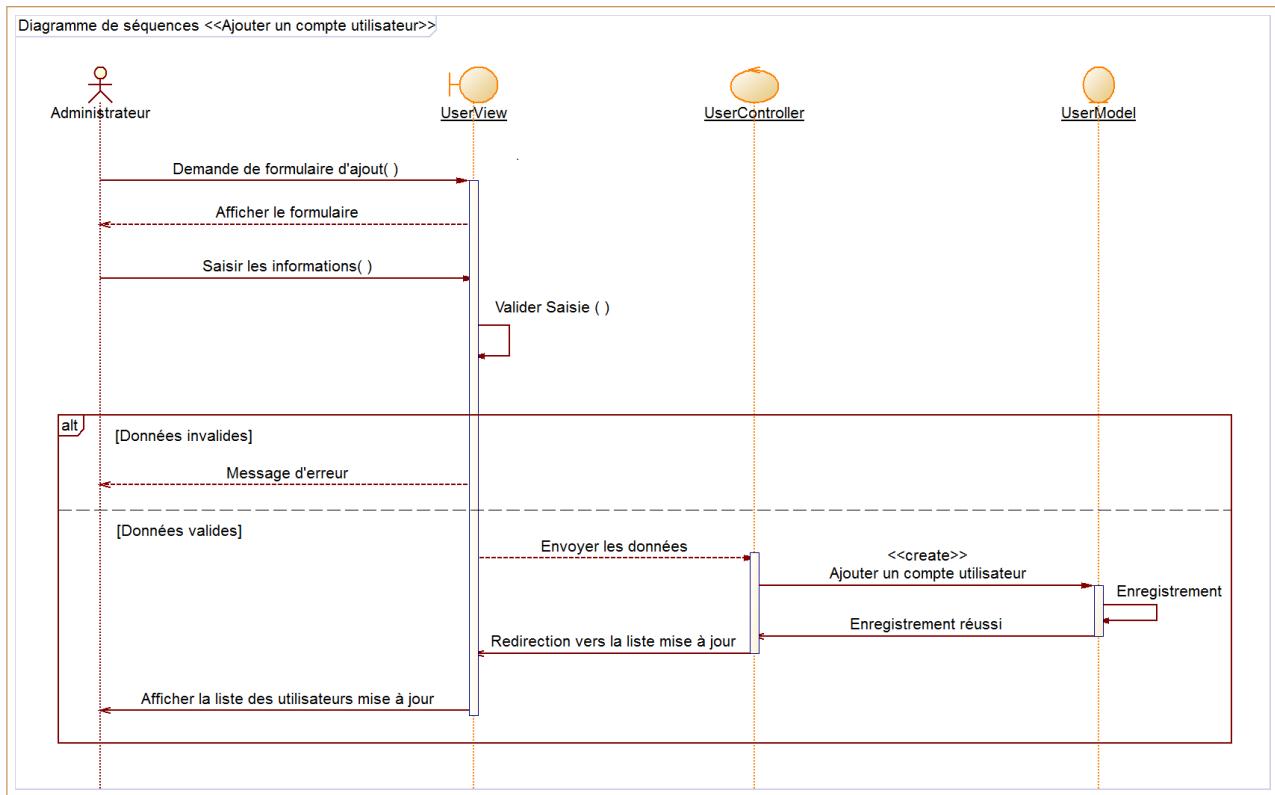


Figure 5.4: Diagramme de séquences objet « Ajouter un compte utilisateur »

Diagramme de séquences objet « Modifier un compte utilisateur »

L’administrateur peut modifier des données relatives à un compte utilisateur déjà pris en charge par l’application. Il sélectionne le compte à modifier à partir de l’interface « UserView » et prend en charge les modifications à apporter.

Une fois le compte est choisi, le contrôleur « UserController » récupère les données, ordonne l’affichage de la page du compte à modifier par l’administrateur et ordonne la mise à jour du compte utilisateur. La figure 5.5 expose le déroulement du processus de la réalisation de cette tâche.

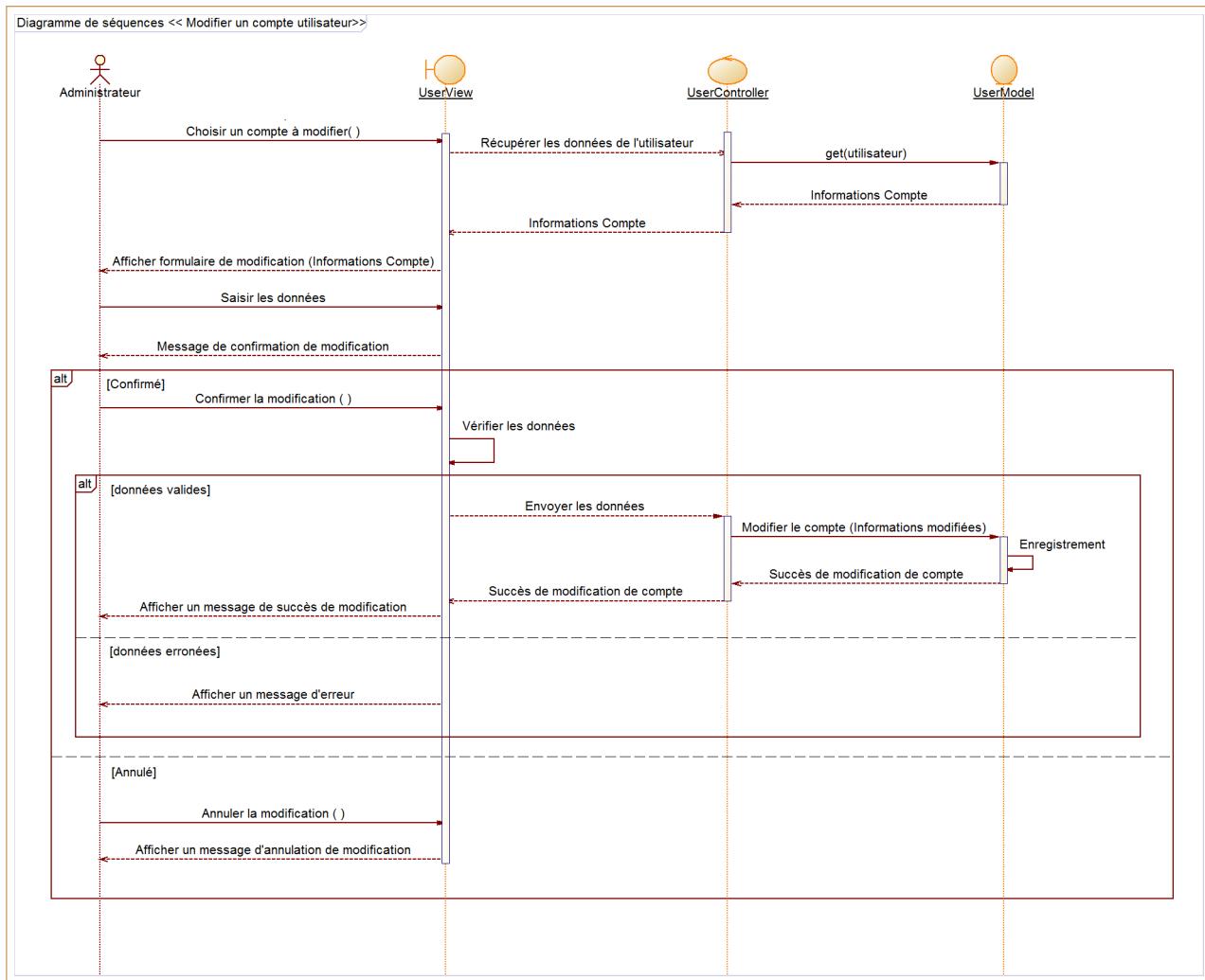


Figure 5.5: Diagramme de séquences objet « Modifier un compte utilisateur »

Diagramme de séquences objet « Supprimer un compte utilisateur »

La suppression d'un compte utilisateur est à la charge de l'administrateur de l'application. Ce dernier sélectionne le compte concerné à partir de l'interface « `UserView` ». Une fois le compte est choisi, le contrôleur « `UserController` » ordonne la suppression du compte utilisateur.

La suppression ne peut porter que sur un compte déjà pris en charge par l'application.

La figure 5.6 illustre le déroulement du processus de la réalisation de cette tâche.

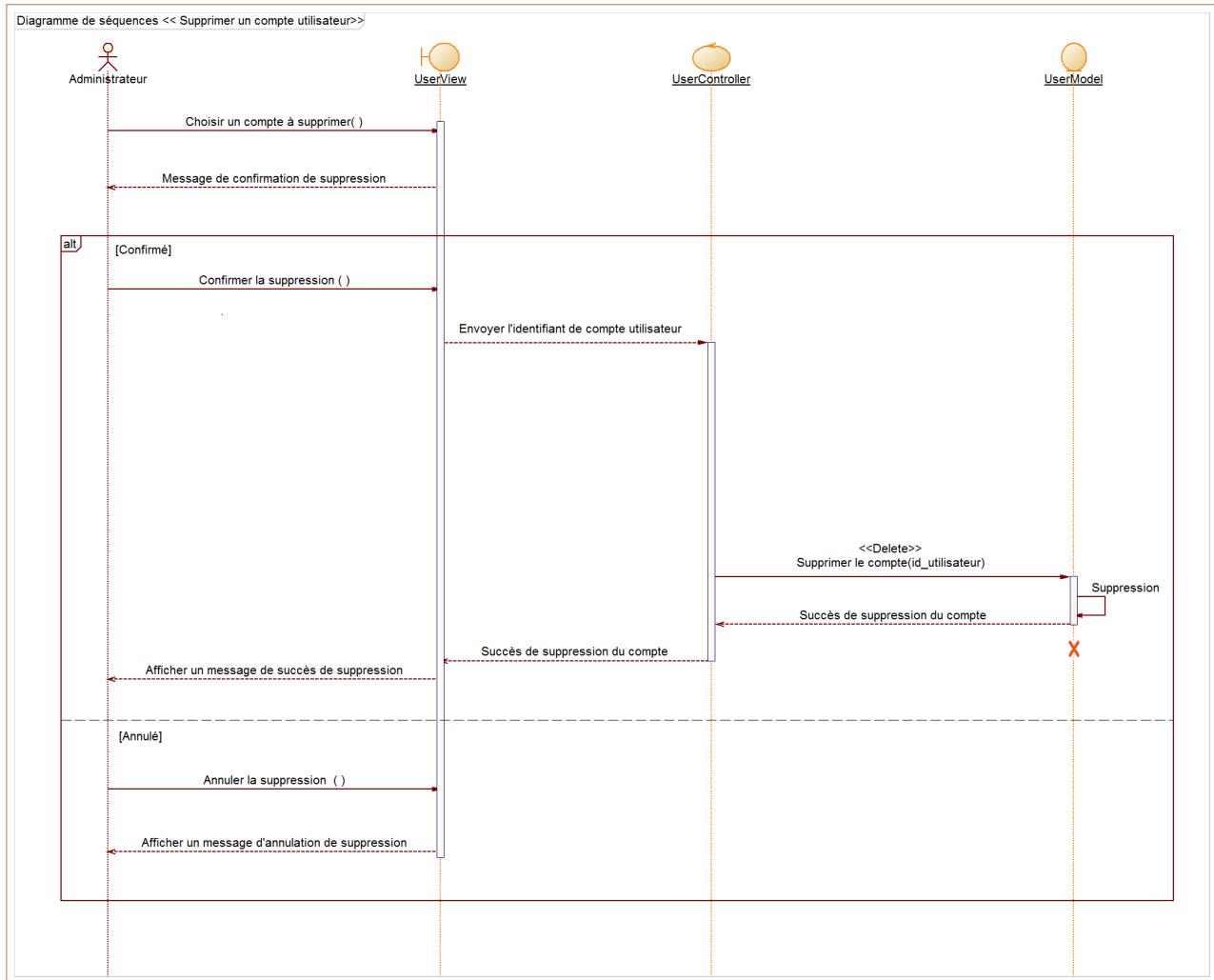


Figure 5.6: Diagramme de séquences objet « Supprimer un compte utilisateur »

5.1.4 Réalisation

Dans cette partie, nous allons exposer les différentes interfaces de notre application réalisées dans le troisième sprint.

Nous avons choisi l'administrateur comme utilisateur vu qu'il présente à travers ces interactions la majeure partie des principales fonctionnalités de ce sprint.

Interface de l'authentification

L'interface représentée dans la figure 5.7 permet à l'utilisateur de s'authentifier pour pouvoir accéder à l'application. L'utilisateur doit d'abord saisir son login et mot de passe.

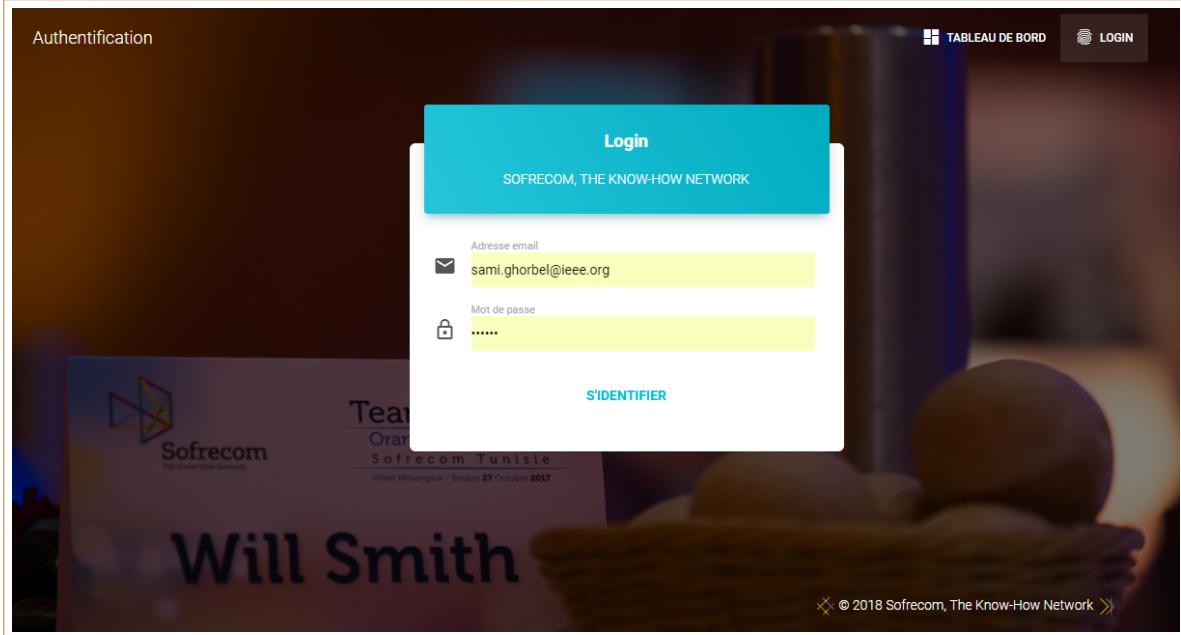


Figure 5.7: Interface de l'authentification

Interface de gestion des comptes utilisateurs

La figure 5.8 illustre la gestion des comptes utilisateurs qui interviennent dans l'application.

Chaque compte utilisateur est caractérisé par :

- Identifiant unique ;
- Nom de l'utilisateur ;
- Adresse électronique ;
- Rôle (administrateur ou utilisateur RH).

_id	username	email	role	
5b002844487982146cb98a80	sami	sami.ghorbel@ieee.org	admin	
5b00284f487982146cb98a81	Aymen	Aymen@sofrecom.com	admin	
5b00f37343dd180540dd100f	Khaled	khaled@sofrecom.com	admin	
5b00f38e43dd180540dd1010	Borhene	borhene@sofrecom.com	user	
5b00f3a943dd180540dd1011	Nidhal	nidhal@sofrecom.com	user	

Figure 5.8: Interface de gestion des comptes utilisateurs

Interface d'ajout d'un nouveau compte utilisateur

La figure 5.9 ci-dessous représente l'interface d'ajout d'un nouveau compte utilisateur.

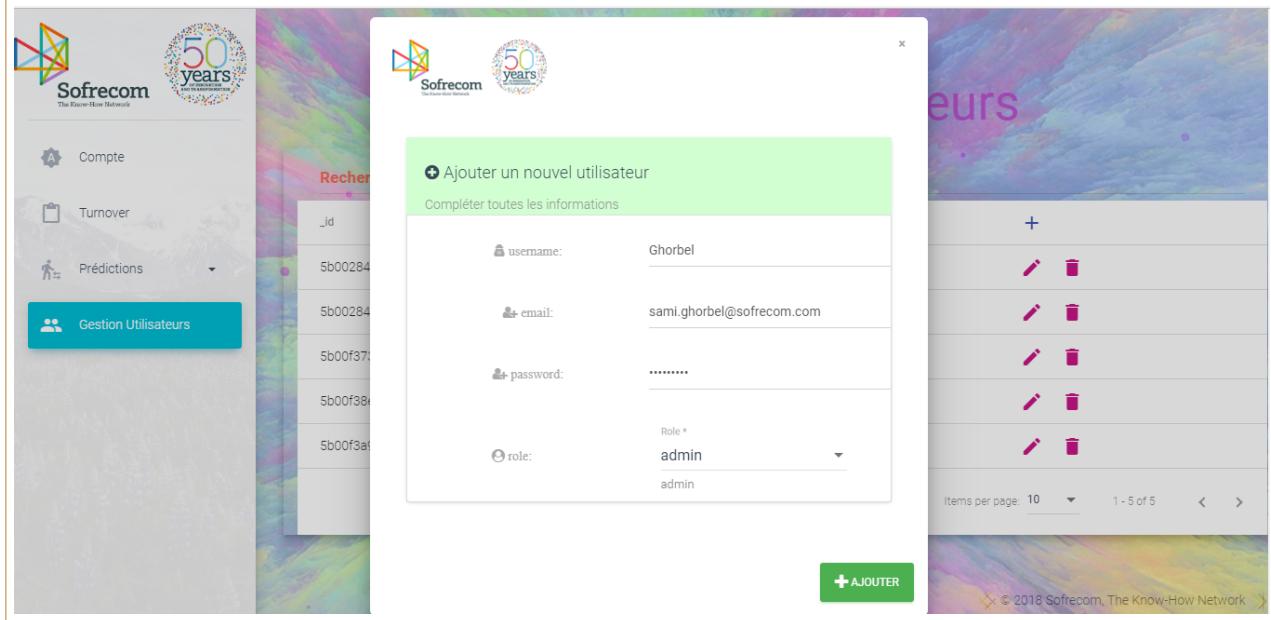


Figure 5.9: Interface d'ajout d'un compte utilisateur

Interface de modification des informations d'un compte utilisateur

Comme illustré par la figure 5.10, l'administrateur peut modifier les informations d'un compte utilisateur choisi.

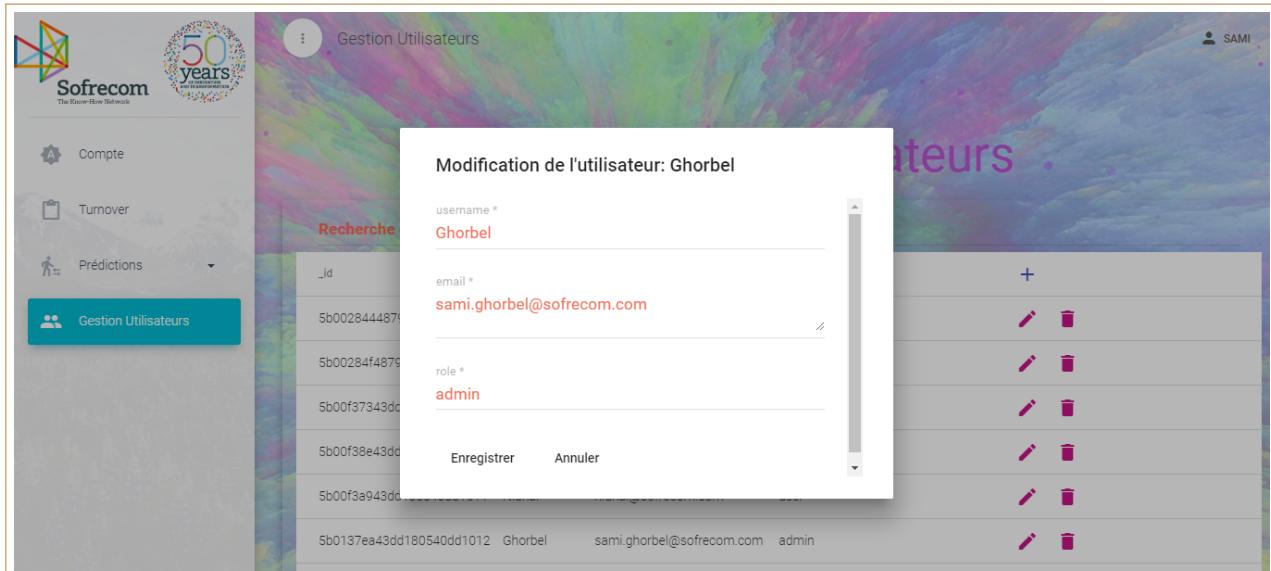


Figure 5.10: Interface de modification des informations d'un compte utilisateur

Interface de suppression d'un compte utilisateur

L'administrateur choisit un compte utilisateur à supprimer. Une alerte de confirmation sera déclenchée pour confirmer la suppression du compte concerné comme indiqué sur la figure 5.11.



Figure 5.11: Interface de suppression d'un compte utilisateur

Interface de succès de suppression d'un compte utilisateur

Dès que l'administrateur confirme la suppression du compte utilisateur, une alerte de succès sera affichée comme le montre la figure 5.12.



Figure 5.12: Interface de succès de suppression d'un compte utilisateur

5.1.4.1 Test et validation

Nous avons rassemblé dans le tableau 5.6 un ensemble de scénarios de cas de tests fonctionnels associés au sprint 3.

Tableau 5.6: Tests fonctionnels du sprint 3

Cas de test	Démarches	Comportement attendu	Résultat
S'authentifier	Demander la page d'authentification. Saisir et valider le login et le mot de passe.	Affichage de la page d'authentification. Redirection vers la page d'accueil.	Conforme
Ajouter un compte utilisateur	Demander la page de gestion des utilisateurs. Remplir et valider le formulaire d'ajout.	Affichage de la page de gestion des utilisateurs. Ajout effectué.	Conforme
Modifier un compte utilisateur	Demander la page de gestion des utilisateurs. Cliquer sur modifier.. Saisir et valider le formulaire de modification.	Affichage de la page de gestion des utilisateurs. Affichage de formulaire de modification. Modification effectuée.	Conforme
Supprimer un compte utilisateur	Demander la page de gestion des utilisateurs. Cliquer sur supprimer. Confirmer la suppression.	Affichage de la page de gestion des utilisateurs. Affichage d'un message de confirmation de suppression. Suppression effectuée.	Conforme

Après avoir vérifié le bon fonctionnement du code et des fonctionnalités, le troisième livrable a été testé et validé par le Product Owner.

5.2 Sprint 4 Visualisation du tableau de bord

5.2.1 Spécification fonctionnelle

5.2.1.1 Backlog du Sprint

5.2.1.2 Diagrammes de cas d'utilisations détaillés du sprint 4

5.2.2 Conception

5.2.2.1 Diagrammes de séquences

5.2.3 Réalisation

5.2.3.1 Test et validation

Conclusion

Conclusion générale et Perspectives

Bibliographie

- [1] SOFRECOM, [Dernière consultation le 20 Mars 2018]. adresse : <http://www.sofrecom.com/fr/sofrecom/a-propos-de-nous/notre-entreprise>.
- [2] S. TUNISIE, [Dernière consultation le 20 Mars 2018]. adresse : <https://fr.slideshare.net/NawresFarhat/rapport-pfe-refonte-et-dploiement-dune-solution-de-messagerie-en-utilisant-larchitecture-microservices>.
- [3] SCRUM, [Dernière consultation le 25 Mars 2018]. adresse : <https://www.thierry-pigot.fr/scrum-en-moins-de-10-minutes/>.
- [4] D. de cas UTILISATION, [Dernière consultation le 03 Avril 2018]. adresse : <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>.
- [5] ——, [Dernière consultation le 03 Avril 2018]. adresse : <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/use-case-diagramme>.
- [6] POWERAMC, [Dernière consultation le 13 Avril 2018]. adresse : http://www.01net.com/telecharger/windows/Programmation/base_de_donne/fiches/telecharger-50280.html.
- [7] INTELLIJ, [Dernière consultation le 16 Avril 2018]. adresse : <https://www.jetbrains.com/idea/>.
- [8] NODE.JS, [Dernière consultation le 19 Avril 2018]. adresse : <https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>.
- [9] HTML/CSS, [Dernière consultation le 19 Avril 2018]. adresse : <http://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3>.
- [10] JAVASCRIPT, [Dernière consultation le 20 Avril 2018]. adresse : <https://www.w3schools.com/js/default.asp>.
- [11] BOOTSTRAP, [Dernière consultation le 20 Avril 2018]. adresse : <https://www.w3schools.com/bootstrap/default.asp>.
- [12] ANGULAR, [Dernière consultation le 20 Avril 2018]. adresse : <https://angular.io/docs>.
- [13] MONGODB, [Dernière consultation le 20 Avril 2018]. adresse : <https://www.mongodb.com/what-is-mongodb>.

Bibliographie

- [14] PYTHON, [Dernière consultation le 18 Avril 2018]. adresse : <https://www.tutorialspoint.com/python/index.htm>.
- [15] TENSORFLOW, [Dernière consultation le 18 Avril 2018]. adresse : <https://blog.xebia.fr/2017/03/01/tensorflow-deep-learning-episode-1-introduction/>.
- [16] D. de DÉPLOIEMENT, [Dernière consultation le 15 Avril 2018]. adresse : <http://uml.free.fr/cours/%20i-p17.html>.
- [17] D. de COMPOSANTS, *Rahma NEFAI*, [Dernière consultation le 20 Avril 2018]. adresse : <http://www.isi.rnu.tn/BIBI/ing/P%20ING%20498>.
- [18] SPRINT, [Dernière consultation le 21 Avril 2018]. adresse : <http://www.isi.rnu.tn/BIBI/ing/P%20ING%20474>.
- [19] M. LEARNING, [Dernière consultation le 21 Avril 2018]. adresse : <https://openclassrooms.com/courses/initiez-vous-au-machine-learning/identifiez-les-differents-types-de-problemes-de-machine-learning>.
- [20] A. non SUPERVISÉ, [Dernière consultation le 21 Avril 2018]. adresse : <https://www.stat4decision.com/fr/machine-learning-pour-tous/>.
- [21] SVM, [Dernière consultation le 22 Avril 2018]. adresse : <https://www.xlstat.com/fr/solutions/fonctionnalites/machine-a-vecteur-support>.
- [22] A. de DÉCISION, [Dernière consultation le 22 Avril 2018]. adresse : <https://scaron.info/doc/intro-arbres-decision/>.
- [23] F. d'arbres DÉCISIONNELS, [Dernière consultation le 22 Avril 2018]. adresse : <https://social.technet.microsoft.com/wiki/contents/articles/25785.predict-wine-quality-with-azure-machine-learning.aspx>.
- [24] K. plus proches VOISINS, [Dernière consultation le 22 Avril 2018]. adresse : <http://www.lsta.upmc.fr/doct/auder/memoire.pdf>.
- [25] G. B. CLASSIFIER, [Dernière consultation le 22 Avril 2018]. adresse : http://rstudio-pubs-static.s3.amazonaws.com/201803_286a19ea7fa44e929b174b954730daef.html.
- [26] RÉGRESSION, [Dernière consultation le 22 Avril 2018]. adresse : <http://www.math.univ-angers.fr/~labatte/enseignement%20UFR/master%20MIM/classificationsupervisee.pdf>.

Bibliographie

- [27] R. de NEURONES, [Dernière consultation le 22 Avril 2018]. adresse : <https://www.irit.fr/publis/SAMOVA/TH/theseJorgeGUTIERREZ.pdf>.
- [28] perceptron MULTICOUCHE, [Dernière consultation le 22 Avril 2018]. adresse : <http://www.statsoft.fr/concepts-statistiques/reseaux-de-neurones-automatises/reseaux-de-neurones-automatises.htm#.WtySty7waCg>.

ملخص

نظرًا للمشاكل التي تواجهها SOFRECOM في أداء مهام التنبؤات التي من شأنها أن تساعد على اتخاذ قرار بشأن استقالة موظفيها في المستقبل، وقد بدأ مشروع لتحسين معالجة هذه المهام. والهدف من ذلك هو تنفيذ تطبيق تكنولوجيا المعلومات الذي يحسن معالجة التنبؤات في الوقت الحقيقي، ويسهل ويسهل تحسين الإدارية. وقد تحقق هذا المشروع ضمن شركة SOFRECOM من خلال تنفيذ الدراسة وتحليلها، وتصميم التطبيق ونماذجه عملية، وأخيراً تطوير واختبارات التطبيق.

كلمات مفاتيح :

.API ، H2O.AI ، MVC

Résumé

Face aux problèmes rencontrés par Sofrecom dans l'exécution des tâches des prédictions qui vont l'aider à prendre la décision sur la démissions de ses employés dans le futur, un projet d'optimisation du traitement de ces tâches a été engagé. L'objectif est de mettre en place une application informatique qui permet d'améliorer le traitement des prédictions en temps réel, de faciliter et d'améliorer leur gestion. Ce projet a été réalisé au sein de la société Sofrecom en procédant à l'étude et à l'analyse de l'existant, la conception de l'application et la modélisation de ses processus, et enfin le développement et les tests de l'application.

Mots clés : Prédiction, Apprentissage Automatique, MVC, H2O.AI, API.

Abstract

In front of problems met by the company Sofrecom in the execution of the tasks of the predictions that will help it to take a decision on the resignation of its employees in the future, a project to optimize the treatment of these tasks has been initiated. The objective is to set up an IT application that will improve the processing of real-time predictions, and facilitate and improve their management. This project was realized out within the company Sofrecom by proceeding to the study and to the analysis of the existing, the conception of the application and the modelling of its processes, and finally the development and the tests of the application.

Keywords : Prediction, Machine learning, MVC, H2O.AI, API.