

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ DE TUNIS  
INSTITUT SUPÉRIEUR DE GESTION



## **TABLEAU DE BORD DE SUIVI ET AUTOMATISATION DES PLANNINGS D'UN PROJET**

**Linedata**

**ELABORÉ PAR**

Mohamed Ali Tourir

**FILIÈRE**

Licence Fondamentale en Informatique de Gestion

**ENCADRÉ PAR**

**Encadrant pédagogique**  
Mlle. Zeineb Chelly

**Encadrant professionnel**  
Mr. Adnen Sadfi

Année universitaire  
2016-2017

# Dédicaces

Je dédis ce travail :

À ma famille, mes amis et mes collègues qui m'ont encouragé et soutenu tout au long de la réalisation du projet.

À tous ceux qui m'ont aidé dans la société ou en ligne et dans les forums et qui m'ont fourni l'aide et les outils ou bibliothèques tierces nécessaires à la réalisation du projet.

Mohamed Ali Tourir

# Remerciements

Au terme de ce travail, je tiens à exprimer ma gratitude et mes remerciements pour toutes les personnes qui ont contribué à sa réalisation.

Tout d'abord, j'adresse mes remerciements à **Mademoiselle Zaineb Chelly** pour m'avoir accompagné durant toutes les étapes de la réalisation du projet. Ses conseils, son soutien et son sens du sérieux m'ont permis de faire face aux difficultés rencontrées et d'aboutir à un projet réussi.

Je remercie également **Monsieur Adnen Sadfi**, **Monsieur Ghazi Souelmi** et **Mademoiselle Ons Bouzouida** pour m'avoir donné l'opportunité de travailler sur ce projet. Cette expérience m'a permis d'acquérir des connaissances techniques et m'a fait découvrir la vie professionnelle en me traitant comme étant un employé et non pas un externe.

Enfin, je tiens à remercier toute personne qui à cru en moi, qui m'a encouragé et qui m'a inspiré durant les moments fastidieux de ce projet.

# Table des figures

1.1	Logo Linedata . . . . .	2
1.2	Organigramme de Linedata Tunisie . . . . .	5
1.3	Requête SQL utilisée . . . . .	6
1.4	Feuille Excel utilisée . . . . .	7
1.5	Exemple de rapport d'indicateurs : Ratio tâches par mois . . . . .	7
1.6	Itération selon la méthode Scrum . . . . .	10
2.1	Représentation des acteurs . . . . .	12
2.2	Diagramme de cas d'utilisation global . . . . .	14
2.3	Prototype de la page d'authentification . . . . .	16
2.4	Prototype du tableau de bord personnel . . . . .	17
2.5	Prototype du tableau de bord de l'équipe . . . . .	17
2.6	Prototype de la gestion de compte . . . . .	18
2.7	Prototype de la gestion des objectifs ICP . . . . .	18
2.8	Prototype de la gestion des équipes . . . . .	19
2.9	Prototype de la modification d'une équipe . . . . .	19
2.10	Prototype de la gestion des membres . . . . .	20
2.11	Prototype de la planification . . . . .	20
2.12	Planification de release de notre projet . . . . .	21
3.1	Logo Eclipse . . . . .	23
3.2	Logo Sublime Text . . . . .	23
3.3	Logo Oracle SQL Developer . . . . .	23
3.4	Logo Edraw max . . . . .	23
3.5	Logo Enterprise Architect . . . . .	24
3.6	Logo Google Chrome . . . . .	24
3.7	Logo Postman . . . . .	24
3.8	Logo Overleaf . . . . .	24
3.9	Logo Sharelatex . . . . .	25
3.10	Logo Trello . . . . .	25
3.11	Logo Ollert . . . . .	25
3.12	Logo DB Designer . . . . .	25
3.13	Logo Jira . . . . .	26
3.14	Logo Apache Tomcat . . . . .	26
3.15	Logo Yeoman . . . . .	26
3.16	Logo Dozer . . . . .	26

3.17	Logo NodeJS . . . . .	27
3.18	Logo Bower . . . . .	27
3.19	Logo Grunt . . . . .	27
3.20	Logo Maven . . . . .	27
3.21	Logo Hibernate . . . . .	28
3.22	Logo Spring . . . . .	28
3.23	Logo AngularJS . . . . .	28
3.24	Logo Java . . . . .	28
3.25	Logo Javascript . . . . .	29
3.26	Logo HTML5 . . . . .	29
3.27	Logo CSS3 . . . . .	29
3.28	Logo JQuery . . . . .	29
3.29	Logo Ajax . . . . .	30
3.30	Logo Bootstrap . . . . .	30
3.31	Logo Oracle Database . . . . .	30
3.32	Logo SQL . . . . .	30
3.33	Logo Mockflow . . . . .	30
3.34	Modélisation du système . . . . .	31
3.35	Pattern Architectural . . . . .	32
4.1	Diagramme de classes d'analyse de la fonctionnalité "s'authentifier" . . . . .	34
4.2	Diagramme de classes d'analyse de la fonctionnalité "Ajouter un membre" . . . . .	35
4.3	Diagramme de classes d'analyse de la fonctionnalité "Consulter tout les tableaux de bord" . . . . .	35
4.4	Diagramme de séquence de la fonctionnalité "S'authentifier" . . . . .	36
4.5	Diagramme de séquence de la fonctionnalité "Ajouter un membre" . . . . .	37
4.6	Diagramme de séquence de la fonctionnalité "Consulter tout les tableaux de bord" . . . . .	37
4.7	Capture d'écran : mot de passe incorrecte . . . . .	38
4.8	Capture d'écran : ajouter membre . . . . .	39
4.9	Capture d'écran : partie d'un tableau de bord d'un membre . . . . .	39
4.10	Burndown Chart du sprint 1 : Tableaux de bord . . . . .	40
4.11	Diagramme de classes d'analyse de la fonctionnalité "créer une équipe" . . . . .	42
4.12	Diagramme de classes d'analyse de la fonctionnalité "supprimer une équipe" . . . . .	43
4.13	Diagramme de classes d'analyse de la fonctionnalité "modifier un membre" . . . . .	43
4.14	Diagramme de classes d'analyse de la fonctionnalité "supprimer un membre" . . . . .	44
4.15	Diagramme de classes d'analyse de la fonctionnalité "consulter le tableau de bord de l'équipe" . . . . .	44
4.16	Diagramme de classes d'analyse de la fonctionnalité "modifier les valeurs des objectifs ICP" . . . . .	45
4.17	Diagramme de séquence de la fonctionnalité "créer une équipe" . . . . .	45
4.18	Diagramme de séquence de la fonctionnalité "supprimer une équipe" . . . . .	46
4.19	Diagramme de séquence de la fonctionnalité "modifier un membre" . . . . .	46
4.20	Diagramme de séquence de la fonctionnalité "supprimer un membre" . . . . .	47
4.21	Diagramme de séquence de la fonctionnalité "consulter le tableau de bord de l'équipe" . . . . .	48
4.22	Diagramme de séquence de la fonctionnalité "modifier les valeurs des objectifs ICP" . . . . .	49

4.23	Capture d'écran : créer une équipe . . . . .	50
4.24	Capture d'écran : supprimer une équipe . . . . .	50
4.25	Capture d'écran : modifier un membre . . . . .	51
4.26	Capture d'écran : supprimer un membre . . . . .	51
4.27	Capture d'écran : tableau de bord de l'équipe . . . . .	52
4.28	Capture d'écran : tableau de bord de l'équipe . . . . .	52
4.29	Burndown Chart du sprint 2 : Tableaux de bord de l'équipe . . . . .	53
4.30	Diagramme de classes d'analyse de la fonctionnalité "modifier le mot de passe" . . . . .	55
4.31	Diagramme de classes d'analyse de la fonctionnalité "consulter le tableau de bord personnel" . . . . .	55
4.32	Diagramme de classes d'analyse de la fonctionnalité "éditer les informations rela- tives à la planification et consulter le résultat" . . . . .	56
4.33	Diagramme de séquence de la fonctionnalité "modifier le mot de passe" . . . . .	57
4.34	Diagramme de séquence de la fonctionnalité "consulter le tableau de bord personnel" . . . . .	58
4.35	Diagramme de séquence de la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat" . . . . .	59
4.36	Capture d'écran : modifier le mot de passe . . . . .	60
4.37	Capture d'écran : consulter le tableau de bord personnel . . . . .	60
4.38	Capture d'écran : éditer les informations relatives à la planification et consulter le résultat . . . . .	61
4.39	Burndown Chart du sprint 3 : Planification . . . . .	62
4.40	Diagramme de déploiement de notre application . . . . .	63
A.1	Architecture de l'application . . . . .	66

# Liste des tableaux

2.1	Backlog produit de l'application . . . . .	15
3.1	Caractéristiques techniques de la machine . . . . .	22
4.1	Backlog du sprint1 : "Tableaux de bords" . . . . .	33
4.2	Backlog du sprint2 : "Tableau de bord de l'équipe" . . . . .	41
4.3	Backlog du sprint3 : "Planification" . . . . .	54

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Cadre général du projet</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Présentation de l'organisme d'accueil . . . . .	2
1.2.1 Historique . . . . .	2
1.2.2 Domaines d'expertise . . . . .	3
1.2.3 Présentation de Linedata Tunisie . . . . .	4
1.3 Contexte du projet . . . . .	6
1.3.1 Etude de l'existant . . . . .	6
1.3.2 Critique de l'existant . . . . .	8
1.3.3 Présentation du projet . . . . .	8
1.4 Langage de modélisation UML . . . . .	8
1.5 Méthodologie de travail . . . . .	9
1.5.1 Approche agile . . . . .	9
1.5.2 Méthode Scrum . . . . .	9
1.6 Conclusion . . . . .	10
<b>2 Analyse des besoins</b>	<b>11</b>
2.1 Introduction . . . . .	11
2.2 Capture des besoins . . . . .	11
2.2.1 Acteurs . . . . .	11
2.2.2 Besoin fonctionnel . . . . .	12
2.2.3 Besoin non fonctionnel . . . . .	13
2.3 Diagramme de cas d'utilisation global . . . . .	13
2.4 Backlog produit . . . . .	14
2.4.1 Notion de priorité . . . . .	14
2.4.2 Notion de complexité . . . . .	15
2.4.3 Backlog produit . . . . .	15
2.4.4 Prototypage des interfaces . . . . .	16
2.5 Planification de release . . . . .	21
2.6 Conclusion . . . . .	21



<b>3</b>	<b>Conception et réalisation de l'application</b>	<b>22</b>
3.1	Introduction . . . . .	22
3.2	Environnement de travail . . . . .	22
3.2.1	Environnement matériel . . . . .	22
3.2.2	Environnement logiciel . . . . .	22
3.3	Choix architecturaux . . . . .	31
3.3.1	Style architectural . . . . .	31
3.3.2	Pattern architectural . . . . .	31
3.4	Conclusion . . . . .	32
<b>4</b>	<b>Organisation des sprints</b>	<b>33</b>
4.1	Introduction . . . . .	33
4.2	Sprint 1 : Tableaux de bords (du 13-02-2017 au 13-03-2017) . . . . .	33
4.2.1	Sprint planning . . . . .	33
4.2.2	Modélisation statique . . . . .	34
4.2.3	Modélisation dynamique . . . . .	36
4.2.4	Développement du sprint . . . . .	38
4.2.5	Test et revue du sprint . . . . .	40
4.2.6	Rétrospective . . . . .	40
4.3	Sprint 2 : Tableau de bord de l'équipe (du 13-03-2017 au 10-04-2017) . . . . .	40
4.3.1	Sprint planning . . . . .	41
4.3.2	Modélisation statique . . . . .	42
4.3.3	Modélisation dynamique . . . . .	45
4.3.4	Développement du sprint . . . . .	49
4.3.5	Test et revue du sprint . . . . .	53
4.3.6	Rétrospective . . . . .	53
4.4	Sprint 3 : Planification (du 10-04-2017 au 08-05-2017) . . . . .	53
4.4.1	Sprint planning . . . . .	54
4.4.2	Modélisation statique . . . . .	54
4.4.3	Modélisation dynamique . . . . .	56
4.4.4	Développement du sprint . . . . .	59
4.4.5	Test et revue du sprint . . . . .	61
4.4.6	Rétrospective . . . . .	62
4.5	Déploiement de l'application . . . . .	62
4.6	Conclusion . . . . .	63
	<b>Conclusion générale</b>	<b>63</b>
<b>A</b>	<b>Flux d'exécution</b>	<b>65</b>
A.1	Introduction . . . . .	65
A.2	Flux d'exécution d'une requête client . . . . .	65
A.3	Conclusion . . . . .	66
	<b>Bibliographie</b>	<b>67</b>

# Introduction générale

Le domaine de gestion de projet grandi de plus en plus et ceci à cause d'un besoin critique qui est présent depuis l'apparition des civilisations. Quel que soit le projet, vous rencontrerez toujours les mêmes contraintes :

- Qualité : votre produit doit répondre aux attentes
- Délai : votre produit doit être livré dans les délais
- Coûts : la réalisation du projet ne doit pas dépasser les limites budgétaires.

Ces derniers sont liés et chaque action sur l'un entraîne des répercussions sur les autres. Si on réduit les délais, il faut soit augmenter les coûts en mettant plus de mains-d'œuvre sur le projet, soit de réduire la qualité du produit pour finir dans les temps. Et ceci sans parler des imprévus. Sur ce, il faut une excellente planification du projet et une bonne gestion de risque pour le bon déroulement du projet.

Un chef de projet est un membre de l'équipe qui gère tous ces contraintes. Pour cela, il doit non seulement avoir des connaissances techniques de base, mais en plus il doit connaître les compétences des membres de son équipe. Et ceci pour pouvoir calculer le temps nécessaire pour finir un scénario ou une tâche dans un projet et pouvoir distribuer les tâches sur les membres.

Dans le cadre de mon projet de fin d'études et dans cet aspect, Linedata m'a proposé le sujet intitulé "Tableau de bord de suivi et automatisation des plannings d'un projet". Cette dernière a voulu faciliter la tâche d'un chef d'équipe en automatisant quelques traitements qu'il fait et en mettant à sa disposition des tableaux de bords pour avoir une vue globale sur le rendement de son équipe.

Notre projet de fin d'études comporte les chapitres suivants :

- Le premier chapitre "Cadre général du projet", présente l'organisme d'accueil, la méthodologie de travail utilisé et le contexte du projet.
- Le deuxième chapitre "Analyse des besoins", présente les acteurs et besoins du projet en plus du backlog produit et la planification de release.
- Le troisième chapitre "Conception et réalisation de l'application", présente l'environnement de travail et les choix architecturaux.
- Le quatrième chapitre "Organisation des sprints", présente la réalisation du projet.

Et nous finirons le rapport par une conclusion générale et la bibliographie.

# Chapitre 1

## Cadre général du projet

### 1.1 Introduction

Dans ce chapitre, nous allons présenter le cadre du projet. En premier lieu, nous proposerons une présentation de l'entreprise d'accueil Linedata. Ensuite nous présenterons une critique de la solution existante ainsi que les solutions proposées. Et nous clôturons ce chapitre par une exposition de la méthodologie suivie tout au long de la réalisation du projet.

### 1.2 Présentation de l'organisme d'accueil

Linedata est un éditeur de solutions globales qui travaille dans le secteur de management, de l'assurance et du crédit. Chaque jour, plus de 63000 professionnels financiers opérant dans 50 pays font confiance à ses technologies afin de gérer leurs activités. Le logo de l'organisme d'accueil est présenté par la Figure 1.1



FIGURE 1.1 – Logo Linedata

#### 1.2.1 Historique

Linedata est né en janvier 1998 du rapprochement de trois sociétés : **Générale de service informatique (GSI) Division des Banques, Line Data** et **BDB Participation**.

GSI Division des banques a été rachetée majoritairement par ses salariés au mois de décembre 1997. L'acquisition ultérieure de Line Data et de BDB Participation a permis la constitution d'une nouvelle société : **Linedata Services**.

Cette dernière n'est pas donc une création de toutes pièces mais elle avait dès ses débuts du savoir-faire humain et technologique ce qui lui a permis en 17 mai 2000 d'être introduit en bourse sur le Nouveau Marché de la Bourse de Paris. Linedata est ainsi depuis 10 ans une société cotée sur Euronext Paris.

Linedata s'est rapidement développé pour accompagner ses clients en élargissant son offre produit tout en ciblant de nouveaux marchés. Afin d'accompagner sa croissance, Linedata a mis en place une stratégie d'acquisition et d'intégration réussie depuis plus de 15 ans :

- Février 2000 : Les sociétés Bimaco Finance, Pen Lan et EKIP/Ingénétudes qui sont basées à Paris ont permis à Linedata d'appuyer sa leadership dans le domaine des crédits et financement.
- Février 2003 : Acquisition des solutions Icon et Préview qui permettent la gestion de portefeuilles. Ils ont permis au groupe de rejoindre la tête du classement au Royaume-Uni et de consolider sa position de numéro 1 en Europe.
- Décembre 2003 : Acquisition de Economic and Social Data Service Solutions (ESDS) qui est un des spécialistes français dans le domaine des progiciels d'assurance individuelle. Ceci a permis à Linedata Services d'être en 2004 un acteur majeur dans le domaine d'Epargne Retraite et la Prévoyance tout en se positionnant significativement sur le segment en très forte croissance des assurances de personnes.
- Mars 2013 : Acquisition de l'activité CapitalStream auprès de Hindustan Computers Limited Technologies (HCL), basée à Seattle et à Irvine, qui a permis à Linedata d'offrir une solution globale front to back sur les marchés des crédits et financements en Amérique du Nord et en Europe.
- Avril 2016 : Acquisition de Derivation qui est l'acteur du tout premier plan spécialisé dans la gestion des risques, les données analytiques et la gestion de portefeuille pour les gérants institutionnels et alternatifs à l'échelle mondiale. Cette dernière a permis à Linedata de proposer désormais une plate-forme globale et complète sur toute la chaîne d'investissement de ses clients, et ce pour tout type d'actif et tout type de structure.

Linedata a su acquérir et intégrer de nombreuses sociétés en quelques années. Cette politique proactive lui permet d'accompagner ses clients à travers des solutions Front to Back intégrées et modulaires pour les professionnels de la gestion d'actifs, de l'assurance et du crédit [1].

### 1.2.2 Domaines d'expertise

Linedata est un prestataire de solutions mondial et indépendant dédié à la communauté internationale des professionnels de l'Asset Management et du Crédit. Ayant plus de 900 collaborateurs répartis dans le monde, Linedata comprend les enjeux de ses clients et leur propose des solutions et des services innovants et adaptés à l'évolution de leur cœur de métier. Linedata met à disposition le meilleur de la technologie logicielle et du service pour répondre aux besoins des :

- Sociétés de gestion et intermédiaires financiers : Sociétés de gestion de portefeuilles institutionnels et collectifs, administrateurs de fonds, agents de transfert, teneurs de compte d'épargne entreprise, filiales de banques ou indépendants. Linedata propose toute une gamme de produits pour gérer l'ensemble des processus d'investissement.
- Compagnies d'assurance et mutuelles, caisses de retraite et instituts de prévoyance : Linedata propose un progiciel unique sur le marché européen permettant de gérer tous les types de contrats d'assurance de personnes depuis l'épargne individuelle jusqu'à la prévoyance en passant par la retraite collective.
- Entreprises commerciales et industrielles : Linedata intervient après des moyennes et grandes sociétés pour la gestion de leur actionnariat salarié.
- Etablissements de crédits spécialisés : Linedata met à la disposition de ces spécialistes financiers une solution progicielle unique, globale et reconnue qui couvre l'ensemble des activités de crédit et de financement des particuliers ou des entreprises [2].

### 1.2.3 Présentation de Linedata Tunisie

Linedata Services (LDS) Tunisie s'est installée à Tunis en 2000 et elle compte actuellement 300 employés. Elle a choisi de se spécialiser dans les domaines Epargne et Assurance, Crédit et Financement.

On remarque que Linedata Tunisie se décompose en sept grands départements. Executive Assistance est le bureau des directeurs et dirigeants de la société. Ensuite vient celui des Ressources humaines qui est chargé des recrutements. En troisième est le bureau Maintenance qui est chargé du maintien du matériel au sein de la société. En quatrième est le bureau TM qui est décomposé en Information sécurité qui est chargé des contrôles de sécurité dans la société et le bureau Corporate IT qui est chargé des réseaux sociaux (publications, annonces). Les deux derniers bureaux sont spécialisés dans les produits et services que propose Linedata que ce soit du développement, de l'assurance qualité ou même de la maintenance et stabilisation. La Figure 1.2 représente l'organigramme de LDS.

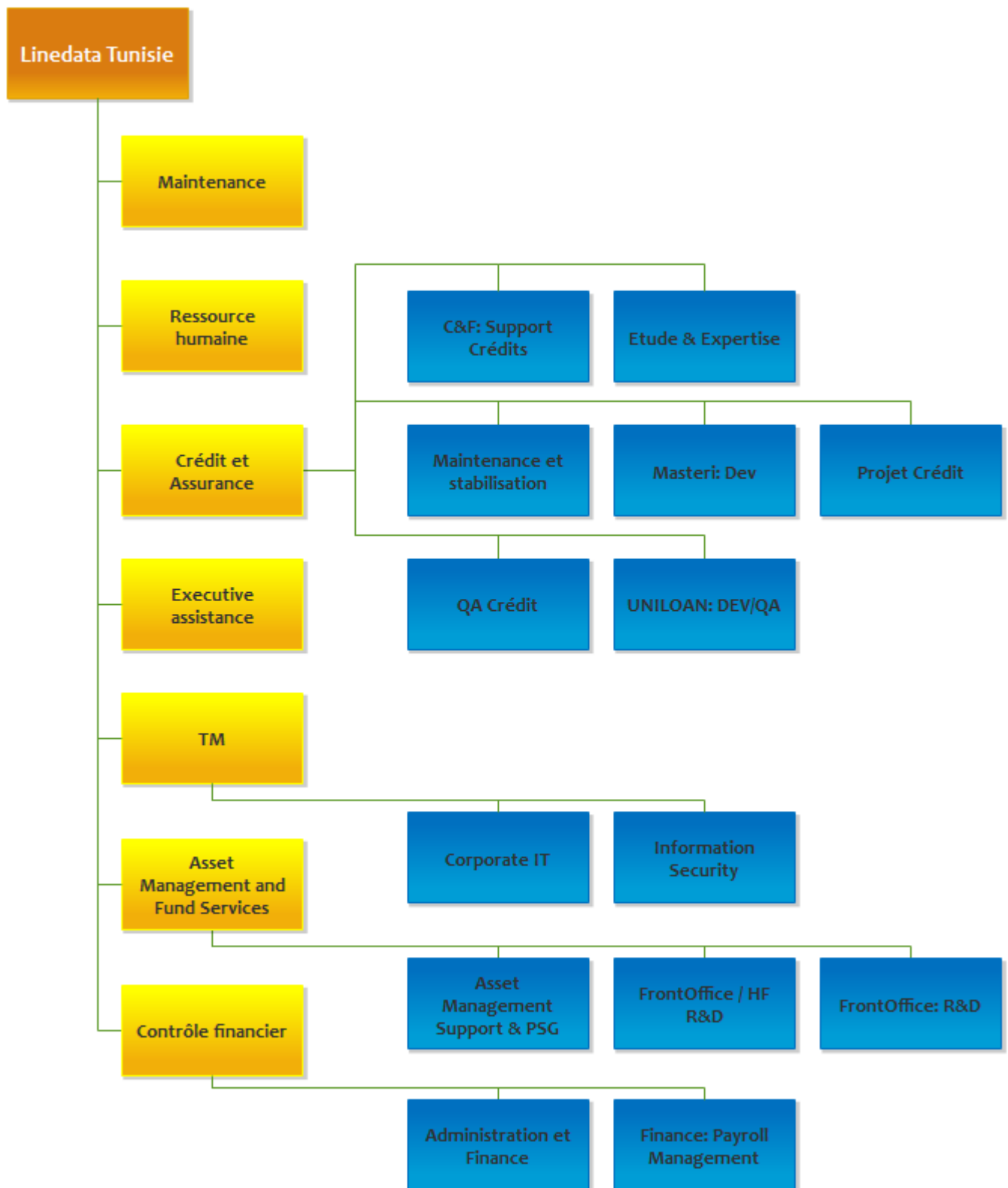


FIGURE 1.2 – Organigramme de Linedata Tunisie

## 1.3 Contexte du projet

Dans cette section, nous allons détailler les objectifs du projet ainsi que les différentes tâches à réaliser.

### 1.3.1 Etude de l'existant

Linedata est une société qui offre des solutions globales de qualité. De ce fait, cette dernière donne une importance majeure aux avis de ses clients. Chaque jour, elle reçoit de multiples demandes d'ajout de fonctionnalités ou des réclamations de bogue que ce soit du client ou de leurs assurances qualité. Ces réclamations sont donc triées puis distribuées aux équipes chargées de la maintenance du produit concerné. Le chef d'équipe doit alors distribuer les tâches aux membres de son équipe. Pour pouvoir les distribuer, ce dernier doit se baser sur plusieurs indicateurs de performance. Il doit en premier lieu utiliser des requêtes SQL<sup>1</sup> pour extraire les données nécessaires sur chaque membre de l'équipe d'une base de données déjà existante. La figure 1.3 représente un exemple d'une requête SQL utilisée.

```
SELECT w.author utilisateur,
       (select pkey || '-' from jira.project p WHERE p.id = i.project) ||
       i.issuenum fiche,
       i.summary libelle,
       t.pname type,
       (SELECT o.customvalue
        FROM jira.customfieldvalue e, jira.customfieldoption o
        WHERE o.customfield = e.customfield
              AND to_char(o.id) = e.stringvalue
              AND e.customfield = 10020
              AND issue = i.id) epic,
       round(w.timeworked / 3600 / 8, 3) consomsee,
       to_char(w.startdate, 'IW') semaine,
       to_char(w.startdate, 'DD/MM/YYYY') date_tache,
       to_char(w.updated, 'DD/MM/YYYY hh:MM:ss') date_saisie,
       (SELECT pname
        FROM jira.priority
        WHERE id = i.priority) priorite,
       (SELECT o.customvalue
        FROM jira.customfieldvalue e, jira.customfieldoption o
        WHERE o.customfield = e.customfield
              AND to_char(o.id) = e.stringvalue
              AND e.customfield = 12430
              AND issue = i.id) severite
FROM jira.worklog w, jira.jiraissue i, jira.issuetype t
WHERE w.issueid = i.id
      and t.id = i.issuetype
      --and w.author in ('[REDACTED]') -- utilisateurs
      and to_char(w.startdate, 'J') between
        to_char(to_date('09/01/2017', 'DD/MM/YYYY'), 'J') and
        to_char(to_date('13/01/2017', 'DD/MM/YYYY'), 'J')
      and i.project in (10920,10621)
      --and to_char(w.startdate, 'IW')='43' --Semaine
ORDER BY to_number(to_char(w.startdate, 'IW')), w.author;
```

FIGURE 1.3 – Requête SQL utilisée

Puis les données sont saisies et enregistrées manuellement sous format Excel. La Figure 1.4 représente un exemple de feuille Excel.

1. Structured Query Language, en français langage de requête structurée. C'est un langage informatique normalisé servant à exploiter des bases de données relationnelles [3]

Étiquettes de lignes	Somme de TIMESPENT	Somme de TIMEESTIMATE	consommé par rapport à	À	Remarques
	3,5	2,5	140%		Epic partagée par les différents membres de
	1,625	2	81%		
	5,1875	4,25	122%		Epic partagée par les différents membres de
	0,5	0,5	100%		
	0,875	0,875	100%		
	2,9375	5,5	53%		
	0,375	0,375	100%		
Total général	15	16	100%		
Étiquettes de lignes	Epic	Somme de Reouverture	Raisons d'réouverture		
		0			
		0			
			réouverture suite a une modification des paramètres d'entrés par le QA		
		2	il s'agit d'un problème autre que le problème décrit initialement dans la		
		0			
		0			
		0			
		0			
		0			
		0			
		0			

FIGURE 1.4 – Feuille Excel utilisée

Ensuite, le chef d'équipe calcule quelques indicateurs de performance à partir de la feuille Excel tel que le nombre de tâches traitées qui ont une certaine complexité sur le nombre de jours passés, le nombre de tâches traitées sur le nombre de jours passés regroupé par mois et le nombre de retours<sup>2</sup> sur le total des tâches traitées. La figure 1.5 représente un exemple de rapport après le calcul de l'indicateur de performance ratio des tâches par mois.

## Ratio / mois

Mois	Nombre CR	Nombre Jours	Ratio (J/CR)
JANUARY	2	2.25	1.13
FEBRUARY	4	5.94	1.48
MARCH	4	7.88	1.97
APRIL	7	12.81	1.83
MAY	10	21.44	2.14
JUNE	11	19.63	1.78
JULY	7	10.38	1.48
AUGUST	8	14.53	1.82
SEPTEMBER	10	18.12	1.81
OCTOBER	9	17.61	1.96
NOVEMBER	13	19.45	1.50
DECEMBER	9	17.10	1.90

FIGURE 1.5 – Exemple de rapport d'indicateurs : Ratio tâches par mois

2. Un retour sur une tâche est la non-validation de la tâche traitée et la demande de correction de cette dernière



Enfin, le chef d'équipe distribue les tâches aux membres en estimant le temps nécessaire pour les finir et en s'appuyant sur les indicateurs de performance.

### 1.3.2 Critique de l'existant

Le responsable, ne disposant d'aucun outil d'aide pour la collecte et le calcul des indicateurs, mettait trop de temps dans la sélection des membres qui vont s'occuper d'une réclamation. En plus de cela, le risque d'erreur est toujours présent puisque le calcul se faisait de façon manuelle. Sans parler du fait que le suivi du rendement était une tâche difficile puisqu'il n'existe aucun outil de collecte d'informations sur les employés et de traitement de ces derniers dans la société.

### 1.3.3 Présentation du projet

Afin de faciliter la tâche du chef de projet, nous proposons une solution qui va être décomposée en deux parties : la première qui consiste à créer des tableaux de bords pour le suivi des membres de l'équipe (suivi de performances des membres avec les différents indicateurs de performance évoqués). Ces tableaux de bords vont être basés sur plusieurs indicateurs de performance tels que le nombre de jours passés dans le traitement de fiches regroupé par complexité ou le nombre de fiches traité dans une période déterminée. La seconde partie va tourner autour de l'optimisation des plannings. Et ceci en automatisant le calcul des indicateurs de performance des membres de l'équipe et en optimisant le choix des membres qui vont traiter les tâches. Cette solution va considérablement réduire le temps que prend l'étape de sélection des membres. Elle va aussi éliminer le risque d'erreurs de calcul et de collecte de données erronées.

## 1.4 Langage de modélisation UML

La notation UML<sup>3</sup> est un **langage visuel** constitué d'un ensemble de schémas, appelé des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous permet donc de représenter le logiciel à développer sous forme de diagrammes qui définissent son fonctionnement.[4]

UML est constitué de 13 diagrammes qui représentent chacun un concept du système. Ces derniers peuvent être représentés par le schéma de 4 vues qui est axées sur les besoins des utilisateurs et qu'on appelle 4+1 vues.

Les besoins des utilisateurs représente le cœur de l'analyse. On y décrit le contexte, les acteurs du projet, les fonctionnalités et les interactions entre ces acteurs et ces fonctionnalités. Les besoins peuvent être représentés à l'aide de deux diagrammes : le diagramme de packages et le diagramme de cas d'utilisation.

La vue logique a pour but d'identifier les éléments du domaine et les relations et interactions entre ces éléments. Cette vue organise les éléments du domaine en catégories. Deux diagrammes peuvent être utilisés : le diagramme de classes et le diagramme d'objets.

---

3. Unified Modeling Language en anglais qui signifie langage de modélisation unifié

La vue des processus démontre la décomposition du système en processus et actions, les interactions entre les processus et la synchronisation et la communication des activités parallèles. La vue des processus peut être représentée par plusieurs diagrammes : le diagramme de séquence, le diagramme d'activité, le diagramme de collaboration, le diagramme d'état-transition, le diagramme global d'interaction et le diagramme de temps.

La vue des composants (vue de réalisation) permet de mettre en évidence les composants du futur système (fichiers sources, bibliothèques, base de données ou autre). Cette vue comprend deux diagrammes : le diagramme de structure composite et le diagramme de composants.

La vue de déploiement décrit les ressources matérielles et la répartition des parties du logiciel sur ces éléments. Cette vue contient qu'un diagramme : le diagramme de déploiement[5].

## 1.5 Méthodologie de travail

Après des recherches sur les méthodes de gestion de projets nous avons choisi les méthodes agiles pour piloter le nôtre.

### 1.5.1 Approche agile

Le mouvement des méthodes agiles a vu le jour en 2001 aux États-Unis [6]. Des experts dans le domaine de l'informatique se sont réunis afin de créer ces méthodes suite à un taux d'échec important des projets informatiques dans les années 1990.

Les méthodes agiles se reposent sur des cycles de développement itératifs et incrémentaux en fonction des besoins évolutifs du client. Ces méthodes permettent donc de mieux répondre aux attentes du client en un temps limité (grâce à l'implication de ce dernier) tout en faisant monter les collaborateurs en compétences. Ces méthodes constituent donc un gain en productivité. Il existe plusieurs méthodes agiles telles que eXtreme Programming (XP), Crystal, Kanban ou encore Rapid Application Development (RAD). Nous avons choisi d'utiliser Scrum pour notre projet.

### 1.5.2 Méthode Scrum

La méthode Scrum est une méthode agile créée en 2002 dont le nom signifie littéralement "la mêlée". Elle consiste à découper le projet en des itérations qu'on nomme sprints. Un sprint a une durée qui varie entre deux à quatre semaines. Avant chaque sprint, les tâches sont estimées en temps et en complexité et ceci pour planifier les livraisons et estimer les coûts auprès du client. Les fonctionnalités qui font objet d'un sprint (appelé user stories) constituent ce qu'on appelle sprint backlog du produit qui va être livré à la fin du sprint. Il est nécessaire de distinguer entre sprint backlog et product backlog qui est l'ensemble des fonctionnalités du produit final.

La méthode Scrum est caractérisée par une mêlée quotidienne, dans laquelle les membres de l'équipe indiquent les tâches qu'ils ont finies, les difficultés rencontrées et ce qu'il leur reste à faire.

Cela permet d'évaluer l'avancement du projet et surtout de venir en aide à ceux qui rencontrent des difficultés qu'on a déjà rencontrées auparavant.

La figure 1.6 présente une itération selon la méthode Scrum

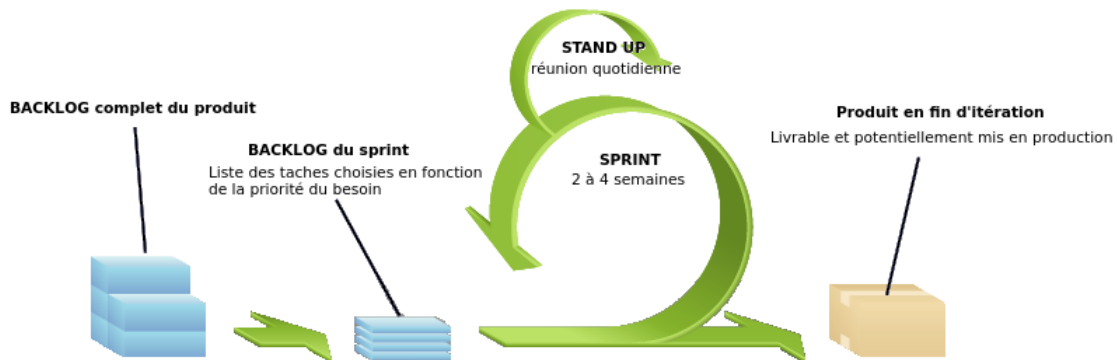


FIGURE 1.6 – Itération selon la méthode Scrum

### Les rôles de la méthode Scrum

La méthode Scrum définit trois rôles :

- **Product Owner** : Il s'agit du représentant du client dans un projet Scrum. Il définit les besoins et les spécifications. Il est aussi chargé de définir les user stories.
- **Scrum master** : Il est chargé de veiller à la mise en application de la méthode et le respect de ses objectifs. C'est une personne chargée de lever les obstacles qui ralentissent l'avancement de l'équipe et du projet.
- **L'équipe** : Ce sont les personnes chargées de la réalisation des sprints. Il peut s'agir d'un architecte, développeur ou testeur.

## 1.6 Conclusion

Dans ce chapitre nous avons présenté l'organisme d'accueil. Puis, nous avons énoncé le contexte du projet. Après, nous avons défini le langage de modélisation UML et à la fin, nous avons présenté la méthode qu'on va utiliser pour la réalisation du projet. Dans le prochain chapitre, nous allons détailler les besoins de notre projet.



**Chef d'équipe**

Le chef d'équipe est chargé de gérer toute les informations relative à son équipe (distribution des tâches, planification d'un projet ...)

**Membre de l'équipe**

Le membre de l'équipe est un consommateur qui peut consulter le tableau de bord personnel.

**Membre de l'équipe sénior**

Le membre de l'équipe est un consommateur qui peut consulter le tableau de bord de l'équipe en plus du tableau de bord personnel.

FIGURE 2.1 – Représentation des acteurs

### 2.2.2 Besoin fonctionnel

Les besoins fonctionnels représentent ce que le système doit être en mesure d'effectuer. Notre application doit offrir à ces utilisateurs les fonctionnalités suivantes :

- Le membre de l'équipe doit pouvoir :
  - s'authentifier
  - consulter le tableau de bord personnel
- Le membre de l'équipe senior doit pouvoir :
  - s'authentifier
  - consulter les tableaux de bord de l'équipe
  - consulter le tableau de bord personnel
- Le chef d'équipe doit pouvoir :
  - s'authentifier
  - gérer les membres
  - gérer les équipes
  - consulter les tableaux de bord de l'équipe

- gérer les objectifs des ICP<sup>2</sup>
- utiliser la fonctionnalité de planification

### 2.2.3 Besoin non fonctionnel

Les besoins non fonctionnels peuvent être considérés comme des besoins fonctionnels spéciaux. Parfois, ils ne sont pas rattachés à un cas d'utilisation particulier, mais ils caractérisent tout le système (l'architecture, la sécurité, le temps de réponse, etc.).

Le système doit garantir les besoins opérationnels suivants :

- La sécurité et la confidentialité des données : Les droits d'accès à l'application doivent être bien attribués aux différents partenaires, afin d'assurer la sécurité des données.
- Maintenabilité et évolutivité : Le code de l'application doit être bien lisible et compréhensible pour pouvoir le maintenir facilement et rapidement. En outre, l'application doit être évolutive afin de répondre aux changements des besoins fonctionnels.
- Performance : L'application doit garantir un temps de réponse et un temps de traitement réduit.
- Ergonomie : L'application doit avoir une interface conviviale et facile à utiliser et ceci en limitant le nombre des éléments dans l'écran, en utilisant des couleurs qui vont ensemble et en créant des interfaces qui offrent une bonne expérience utilisateur.

## 2.3 Diagramme de cas d'utilisation global

Après avoir identifié les besoins et les acteurs, nous traçons le diagramme de cas d'utilisation global qui sera après la base pour la rédaction du backlog produit.

La figure 2.2 présente le diagramme de cas d'utilisation global.

---

2. Indicateur clé de performance (ICP), en anglais Key Performance Indicator (KPI), est un indicateur mesurable d'aide décisionnelle [9].

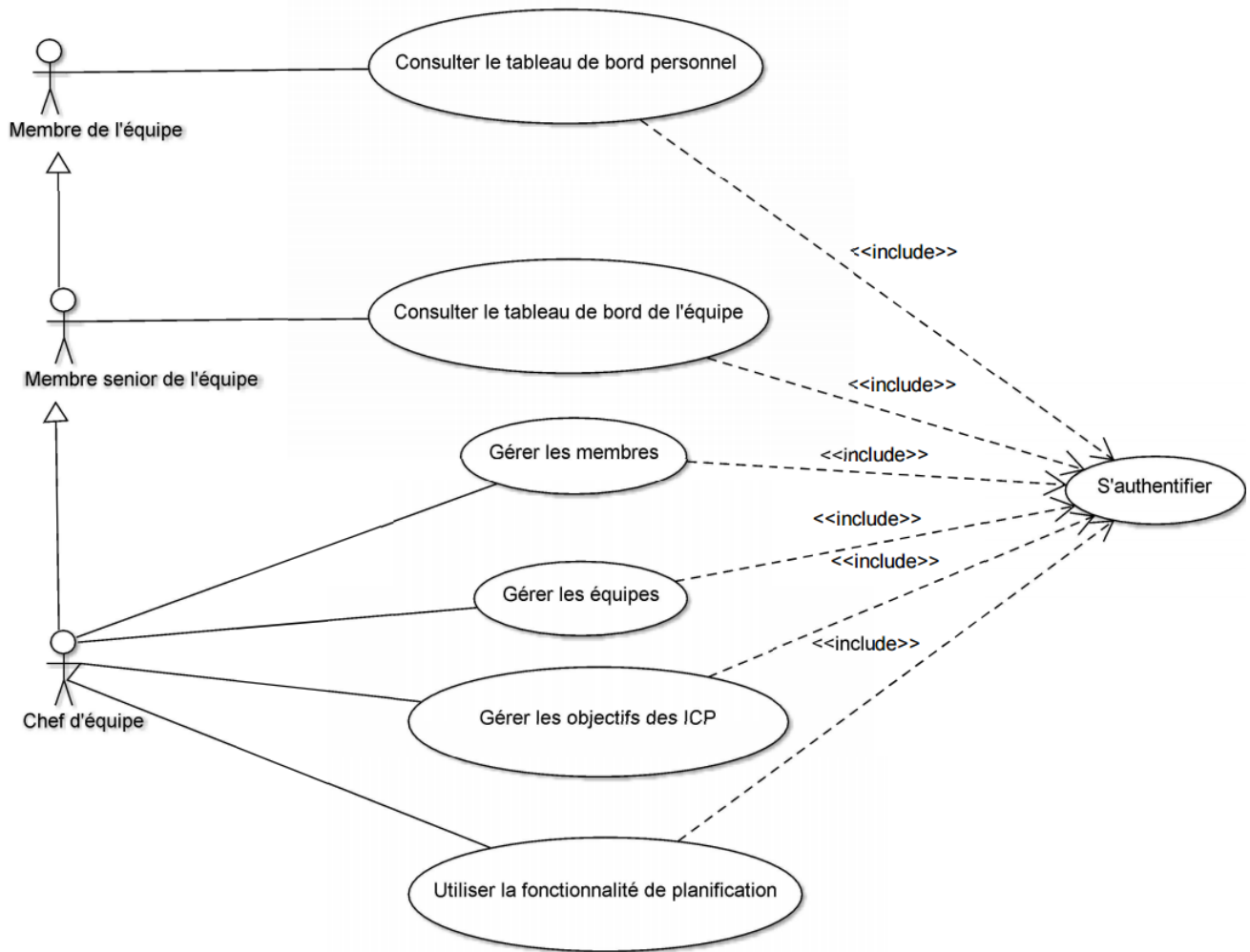


FIGURE 2.2 – Diagramme de cas d'utilisation global

## 2.4 Backlog produit

Dans Scrum, le backlog produit est la liste des fonctionnalités attendues d'un produit. Il contient donc tous les éléments qui vont nécessiter du travail pour l'équipe. Les éléments y sont classés par priorité ce qui permet de définir l'ordre de réalisation et par complexité ce qui permet de définir la taille d'un backlog sprint. Nous allons donc présenter les notions de priorité et de complexité. Après nous allons présenter le backlog produit de notre application. Et nous proposerons enfin la planification de release.

### 2.4.1 Notion de priorité

La priorité représente l'ordre de développement des user stories. Elle permet donc de connaître l'ordre selon lequel les user stories vont être sélectionnées pour être développé dans les backlogs sprint. Les niveaux de priorité qui vont être utilisés dans notre backlog sont :

- Élevée : L'exigence est essentielle. Si elle n'est pas faite le projet échoue.

- Moyenne : Il s'agit d'une exigence essentielle, qu'il faut faire dans la mesure du possible.
- Faible : Il s'agit d'une exigence souhaitable. Elle pourrait être faite dans la mesure où elle n'a pas d'impact sur les autres tâches.
- Très faible : Il s'agit d'une exigence "Luxe". Elle ne sera pas faite cette fois mais plus tard, mais intéressante et à garder pour la prochaine version [10].

### 2.4.2 Notion de complexité

La notion de complexité(ou Story point) sert à estimer l'effort nécessaire à une équipe pour implémenter une fonctionnalité. Cette estimation prend en compte l'effort pour le développement, la complexité du développement et le risque [11].

Nous allons utiliser les six premiers nombres de la suite de Fibonacci comme points de complexités dans notre backlog produit (1,2,3,5,8,13) [12].

### 2.4.3 Backlog produit

Le tableau 2.1 représente le backlog produit de notre application.

TABLE 2.1 – Backlog produit de l'application

ID	Fonctionnalité	ID story	User Story	Priorité	Complexité
1	s'authentifier	1.1	En tant que membre, membre senior ou chef d'équipe, je souhaite pouvoir m'authentifier	Moyenne	5
		1.2	En tant que membre, membre senior ou chef d'équipe, je souhaite pouvoir modifier mon mot de passe	Faible	3
2	gérer les équipes	2.1	En tant que chef d'équipe, je souhaite pouvoir créer une nouvelle équipe	Moyenne	3
		2.2	En tant que chef d'équipe, je souhaite pouvoir supprimer une équipe	Moyenne	3
		2.3	En tant que chef d'équipe, je souhaite pouvoir assigner un chef d'équipe	Elevé	3
3	gérer les membres	3.1	En tant que chef d'équipe, je souhaite pouvoir ajouter un membre dans l'application	Moyenne	3
		3.2	En tant que chef d'équipe, je souhaite pouvoir supprimer un membre de l'application	Moyenne	3
		3.3	En tant que chef d'équipe, je souhaite pouvoir modifier un membre de l'application	Moyenne	3



ID	Fonctionnalité	ID story	User Story	Priorité	Complexité
4	consulter les tableaux de bord de l'équipe	4.1	En tant que membre senior ou chef d'équipe, je souhaite pouvoir consulter le tableaux de bord de l'équipe	Elevée	8
5	consulter les tableaux de bord personnels	5.1	En tant que membre ou membre senior de l'équipe, je souhaite pouvoir consulter le tableau de bord me représentant	Faible	5
		5.2	En tant que membre senior ou chef d'équipe, je souhaite pouvoir consulter les tableaux de bord des membres de mon équipe	Elevée	13
6	gérer les objectifs des ICP	6.1	En tant que chef d'équipe, je souhaite pouvoir modifier les valeurs des objectifs ICP	Moyenne	5
7	utiliser la fonctionnalité de planification	7.1	En tant que chef d'équipe, je souhaite pouvoir saisir les informations relatives à la planification et consulter le résultat obtenu	Elevée	13

#### 2.4.4 Prototypage des interfaces

La figure 2.3 présente le prototype de l'interface de la page d'authentification.

Login to your account

Login :

Password :

**Login**

FIGURE 2.3 – Prototypage de la page d'authentification

La figure 2.4 présente le prototype du tableau de bord personnel où nous retrouverons toutes les statistiques concernant l'utilisateur connecté.

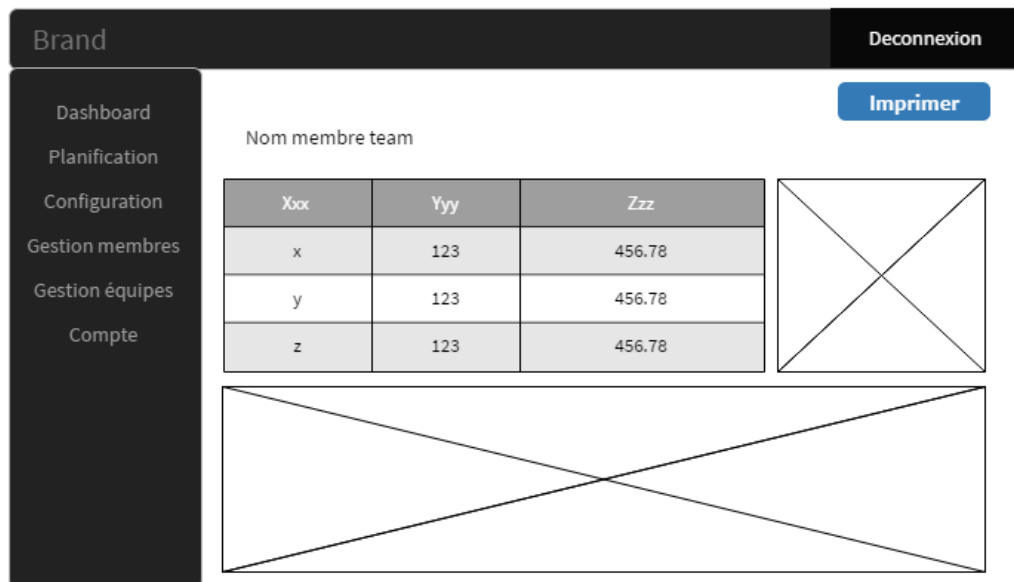


FIGURE 2.4 – Prototype du tableau de bord personnel

La figure 2.5 présente le prototype du tableau de bord de l'équipe où en retrouve des statistiques qui concernent toute l'équipe.

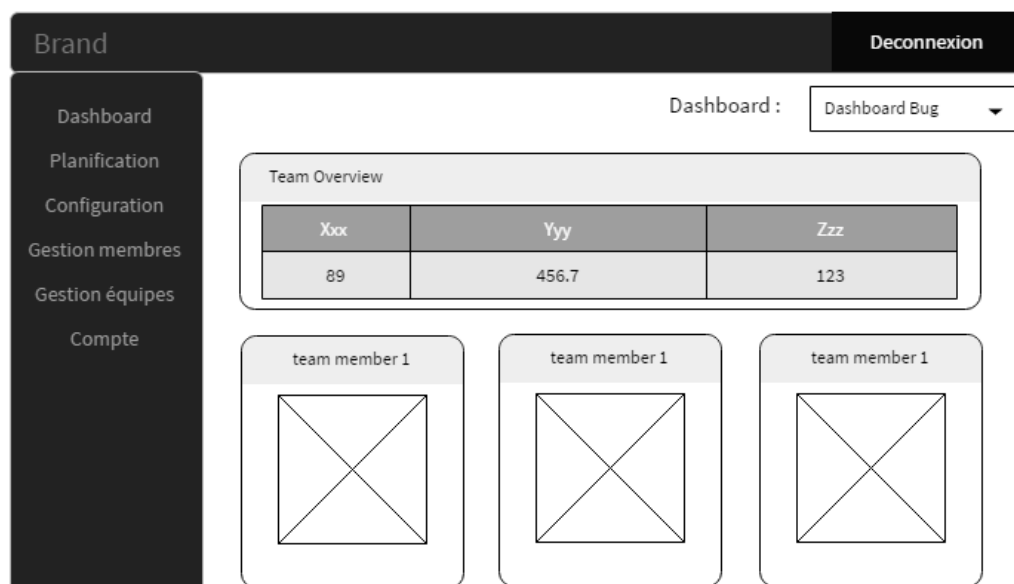


FIGURE 2.5 – Prototype du tableau de bord de l'équipe

La figure 2.6 présente le prototype de l'interface de gestion de compte où l'utilisateur peut changer son mot de passe.

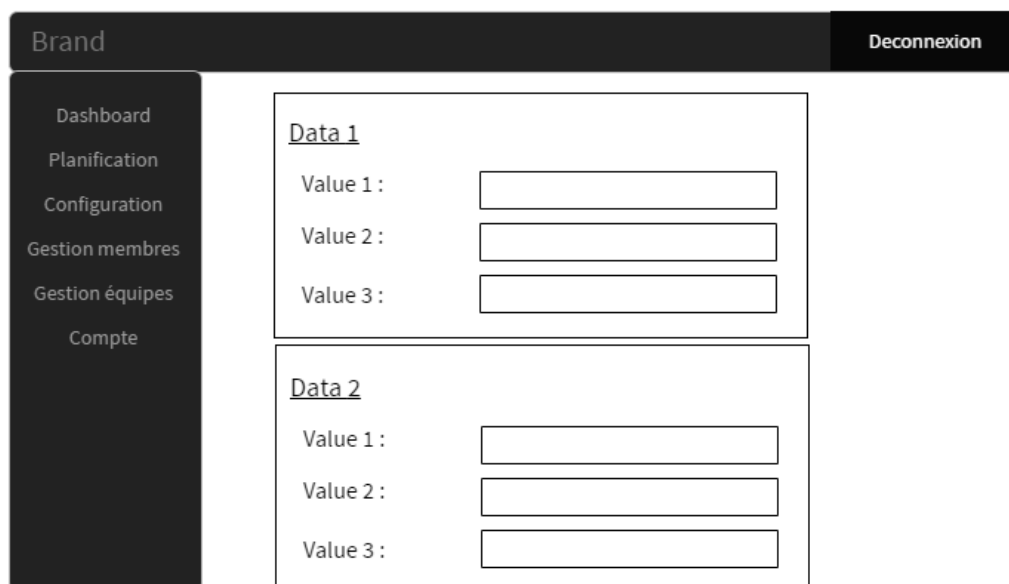


Le prototype de l'interface de gestion de compte se compose d'une barre d'en-tête, d'un menu latéral et d'une zone de contenu principale.

- Barre d'en-tête :** Contient le logo "Brand" à gauche et le lien "Deconnexion" à droite.
- Menu latéral :** Une liste de liens : Dashboard, Planification, Configuration, Gestion membres, Gestion équipes, et Compte (surligné).
- Zone de contenu principale :**
  - Contient trois champs de saisie étiquetés "Ancien mot de passe :", "Nouveau mot de passe :", et "Répéter mot de passe :".
  - Un bouton "Valider" est positionné en bas à droite de la zone de saisie.

FIGURE 2.6 – Prototype de la gestion de compte

La figure 2.7 présente le prototype de l'interface de gestion des objectifs ICP où l'utilisateur peut modifier les objectifs.

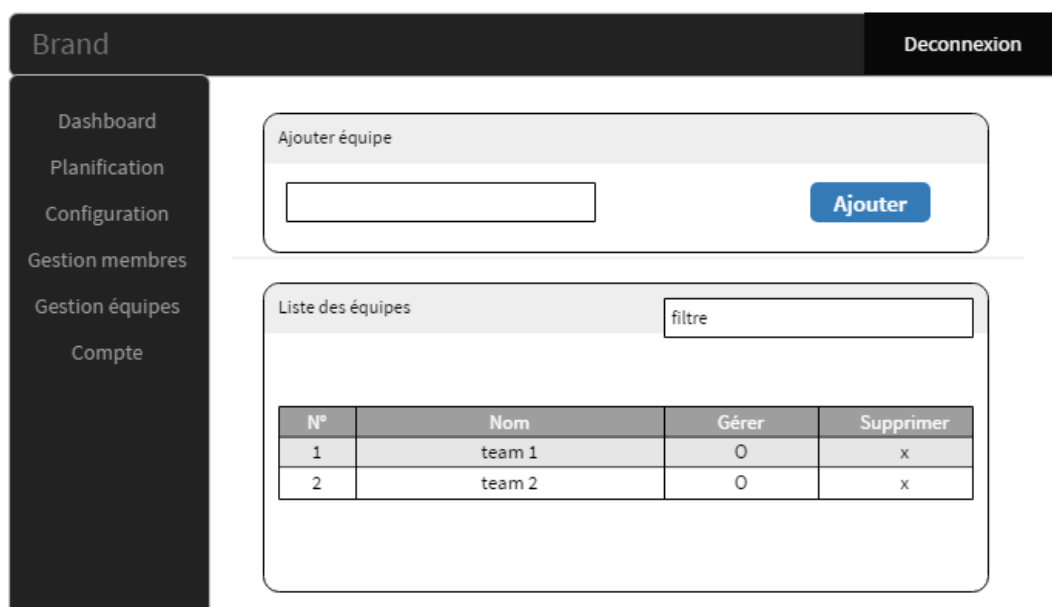


Le prototype de l'interface de gestion des objectifs ICP se compose d'une barre d'en-tête, d'un menu latéral et d'une zone de contenu principale.

- Barre d'en-tête :** Contient le logo "Brand" à gauche et le lien "Deconnexion" à droite.
- Menu latéral :** Une liste de liens : Dashboard, Planification, Configuration, Gestion membres, Gestion équipes, et Compte.
- Zone de contenu principale :**
  - Contient deux sections distinctes, "Data 1" et "Data 2", chacune avec une sous-entête surlignée.
  - Chaque section contient trois champs de saisie étiquetés "Value 1 :", "Value 2 :", et "Value 3 :".

FIGURE 2.7 – Prototype de la gestion des objectifs ICP

La figure 2.8 présente le prototype de l'interface de gestion d'équipe où l'utilisateur peut ajouter, modifier ou supprimer une équipe.



Brand Deconnexion

- Dashboard
- Planification
- Configuration
- Gestion membres
- Gestion équipes
- Compte

Ajouter équipe

Ajouter

Liste des équipes

N°	Nom	Gérer	Supprimer
1	team 1	O	x
2	team 2	O	x

FIGURE 2.8 – Prototype de la gestion des équipes

La figure 2.9 présente le prototype de l'interface de modification d'équipe où l'utilisateur peut changer le chef d'équipe ou enlever un utilisateur de cette dernière.



Brand Deconnexion

- Dashboard
- Planification
- Configuration
- Gestion membres
- Gestion équipes
- Compte

Ajouter équipe

Nom de l'équipe

Team leader	Nom	Remove
<input checked="" type="checkbox"/>	membre 1	X
<input type="checkbox"/>	membre 2	X
<input type="checkbox"/>	membre 3	X

FIGURE 2.9 – Prototype de la modification d'une équipe

La figure 2.10 présente le prototype de l'interface de gestion des membres où l'utilisateur peut ajouter, modifier ou supprimer un utilisateur de l'application.

N°	Nom	Active
1	membre 1	<input type="checkbox"/>
2	membre 2	<input checked="" type="checkbox"/>

FIGURE 2.10 – Prototype de la gestion des membres

La figure 2.11 présente le prototype de l'interface de planification de tâche.

nom	Xxx	Yyy	Zzz
member 1	25%	123.45	111
member 2	32%	167.15	121
member 3	100%	183.42	113
Total	45%	201.12	152

FIGURE 2.11 – Prototype de la planification

## 2.5 Planification de release

La planification de release est conçu pour décomposer le projet en plusieurs sprints de durée fixe. Pour pouvoir le faire, il faut estimer la complexité totale de user stories qui peuvent être réalisés dans une durée fixe du sprint. Cette valeur estimée s'appelle la vélocité [13] [14]. La durée d'un sprint pour notre projet va être fixé à 4 semaines et nous avons conclu que la vélocité sera égale à 21. La figure 2.12 représente la planification de release pour notre projet.



FIGURE 2.12 – Planification de release de notre projet

## 2.6 Conclusion

Dans ce chapitre nous avons présenté les besoins fonctionnels, les besoins non fonctionnels et les acteurs de notre application. Puis, nous avons présenté le diagramme de cas d'utilisation initiale. Après, nous avons défini la notion de priorité et celle de complexité. Ensuite, nous avons présenté le product backlog. Et enfin, nous avons présenté le prototypage et la planification de release qu'on va suivre tout au long de notre projet. Dans le prochain chapitre nous allons détailler nos choix architecturaux ainsi que l'environnement de travail matériel et logiciel.

# Chapitre 3

## Conception et réalisation de l'application

### 3.1 Introduction

Dans ce chapitre, nous allons présenter l'environnement de travail et les choix architecturaux. En premier lieu, nous parlerons de l'environnement matériel et logiciel. Après, nous proposerons nos choix architecturaux.

### 3.2 Environnement de travail

Cette section va être dédiée à la présentation de l'environnement de travail que ce soit matériel ou logiciel.

#### 3.2.1 Environnement matériel

Le tableau 3.1 présente les caractéristiques techniques de la machine utilisé.

TABLE 3.1 – Caractéristiques techniques de la machine

<b>Constructeur</b>	Dell Inc.
<b>Processeur</b>	Intel(R) Core(TM) i3-2100 CPU @ 3.10GHz
<b>RAM</b>	8.0 GO
<b>Disque dur</b>	500.0 GO
<b>Système d'exploitation</b>	Microsoft Windows 10 Entreprise (x64)

#### 3.2.2 Environnement logiciel

Dans cette partie, nous allons présenter les logiciels utilisés en plus des technologies.

## Eclipse

Eclipse est un environnement de développement intégré qui est open source. Cet outil a été utilisé pour développer la partie back-office<sup>1</sup> de l'application qui est en Java.



FIGURE 3.1 – Logo Eclipse

## Sublime Text 3

Sublime Text 3 est un éditeur de texte qui facilite la saisie de code grâce à son auto-complétion, aux différents raccourcis et à la coloration de syntaxe. Cet outil a été utilisé pour développer la partie front-office<sup>2</sup> de l'application qui est en Javascript, HTML et CSS.



FIGURE 3.2 – Logo Sublime Text

## Oracle SQL Developer

Oracle SQL Developer est environnement de développement intégré permettant d'interroger les bases de données Oracle grâce au langage SQL. Cet outil a été utilisé pour consulter la base de données distante de l'application Jira en plus de la base de données de notre application.



FIGURE 3.3 – Logo Oracle SQL Developer

## Edraw Max

Edraw Max est un logiciel permettant de créer des diagrammes de toutes sortes. Ce logiciel a été utilisé pour la création de l'organigramme de la société.



FIGURE 3.4 – Logo Edraw max

---

1. Back-office désigne les applications qui ne sont pas accessibles au client final  
2. Front-office désigne les applications accessibles au client final



### Enterprise Architect

Enterprise Architect est un logiciel permettant de créer des diagrammes UML. Ce logiciel a été utilisé pour la conception de notre application afin de créer les diagrammes UML utilisés dans le rapport.



FIGURE 3.5 – Logo Enterprise Architect

### Google Chrome

Google Chrome est un navigateur web disposant d'une console permettant de debug le code de notre application front-office. Cet outil a été utilisé pour la visualisation de l'application web.



FIGURE 3.6 – Logo Google Chrome

### Postman

Postman est une application permettant de tester les services web. Cette application a été utilisée pour tester tous les services web de notre application back-office.



FIGURE 3.7 – Logo Postman

### Overleaf

Overleaf est un éditeur de rapport Latex en ligne. Il dispose d'outil de gestion de version en plus de sa facilité d'utilisation. Cet outil a été utilisé pour éditer le rapport de notre projet de fin d'études.



FIGURE 3.8 – Logo Overleaf

### Sharelatex

Sharelatex est aussi un éditeur de rapport Latex en ligne. En plus de sa facilité d'utilisation, ce dernier n'a pas de limitations si vous allez l'utiliser sans collaborateurs. Cet outil a été utilisé pour éditer le rapport de notre projet de fin d'études.



FIGURE 3.9 – Logo Sharelatex

### Trello

Trello est un outil de gestion de projet en ligne. Nous l'avons utilisé pour gérer les sprints de notre projet.



FIGURE 3.10 – Logo Trello

### Ollert

Ollert est un outil de génération de rapports sur l'avancement d'un projet géré par Trello. Cet outil a été utilisé pour visualiser notre avancement tout au long du projet.



FIGURE 3.11 – Logo Ollert

### DB Designer

DB Designer est un outil de création d'un schéma de base de données en ligne. Cet outil permet d'exporter le schéma sous format SQL. Nous avons utilisé ce dernier pour pouvoir créer le schéma de base de données de notre application afin de pouvoir en discuter lors des réunions hebdomadaires.

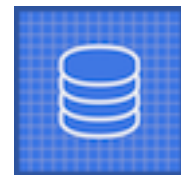


FIGURE 3.12 – Logo DB Designer

## Jira

Jira est un outil de gestion de projet piloté par Scrum ou Kanban<sup>3</sup>. Cet outil est utilisé dans la société pour piloter les projets et sa base de données contient des informations sur les employés de cette dernière. Nous avons utilisé Jira pour pouvoir en extraire les données sur les employés afin de calculer les indicateurs de performance.



FIGURE 3.13 – Logo Jira

## Apache Tomcat

Apache Tomcat est un serveur HTTP et conteneur web libre de servlet [15]. Cet outil a été utilisé pour héberger notre application back-office.

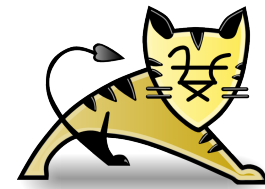


FIGURE 3.14 – Logo Apache Tomcat

## Yeoman

Yeoman est un générateur de code utilisé pour générer des projets web. Cet outil a été utilisé pour générer le squelette initial de notre application web en plus des composants de cette dernière.



FIGURE 3.15 – Logo Yeoman

## Dozer

Dozer est un outil de mise en correspondance des données en Java / JEE. Cet outil a été utilisé pour mettre en correspondance les modèles et les entités.



FIGURE 3.16 – Logo Dozer

---

3. C'est une méthode de travail japonaise connu pour son efficacité dans la gestion de projet

## NodeJs

NodeJs est un runtime Javascript permettant d'exécuter du code Javascript côté serveur. Cet outil dispose aussi d'un manager de paquetage Javascript Open Source appelé Npm. Cet outil a été utilisé pour installer différents paquetages pour notre projet front-office.



FIGURE 3.17 – Logo NodeJS

## Bower

Bower est un manager de paquetage comme Npm. Cet outil a aussi été utilisé pour installer différents paquetages pour notre projet front-office.



FIGURE 3.18 – Logo Bower

## Grunt

Grunt est un outil d'automatisation de code Javascript. Cet outil a été utilisé comme serveur http pour héberger notre application front-office et pour gérer le changement de code.



FIGURE 3.19 – Logo Grunt

## Maven

Maven est un moteur de production pour les applications Java / JEE. Cet outil a été utilisé pour gérer les dépendances du projet en plus de la génération de codes.



FIGURE 3.20 – Logo Maven

## Hibernate

Hibernate est un ORM<sup>4</sup> servant à la gestion de persistance de la base de données relationnelle. Cet outil a été utilisé pour gérer la base de données de notre application back-office.



FIGURE 3.21 – Logo Hibernate

## Spring

Spring est un framework Java / JEE conçu pour faciliter le développement d'applications. Ce framework a plusieurs avantages tels que la facilité d'utilisation ou son architecture robuste et facile à comprendre. Ce framework a été utilisé pour développer notre application back-office.



FIGURE 3.22 – Logo Spring

## AngularJS

AngularJS est un framework MVC<sup>5</sup> Javascript conçu pour développer des applications web. Ce framework a été utilisé pour développer notre application front-office.



FIGURE 3.23 – Logo AngularJS

## Java

Java est un langage de programmation orienté objet qui permet de développer une application indépendante des architectures logicielles [16]. Ce langage a été utilisé pour développer l'application back-office.



FIGURE 3.24 – Logo Java

---

4. Object Relationship Mapping ou en français Mapping objet relationnel est un outil permettant de gérer la persistance des données entre l'application et la base de données relationnelle

5. Model-View-Controller ou en français modèle, vue et contrôleur est une architecture logicielle populaire pour les applications web

## Javascript

Javascript est un langage de programmation conçu pour rendre les pages web dynamiques. Ce langage a été utilisé pour développer notre application front-office.



FIGURE 3.25 – Logo Javascript

## HTML5

HTML5 est un langage de balisage qui permet de concevoir une interface IHM<sup>6</sup> d'un site web. Ce langage a été utilisé pour créer les interfaces graphiques de notre application front-office.



FIGURE 3.26 – Logo HTML5

## CSS3

CSS3 est un langage qui permet de décrire le style d'un document HTML. Ce dernier a été utilisé pour définir le style de notre application.



FIGURE 3.27 – Logo CSS3

## JQuery

JQuery est un framework Javascript qui permet de simplifier le développement en Javascript. Ce framework a été utilisé pour développer notre application front-office.



FIGURE 3.28 – Logo JQuery

---

6. Interface Homme Machine

## Ajax

Ajax est une bibliothèque Javascript qui permet de modifier une partie d'une page web sans la recharger. Cette dernière a été utilisée pour consommer les services web côté client fournis par notre application back-office.



FIGURE 3.29 – Logo Ajax

## Bootstrap

Bootstrap est un framework Html/Css/Javascript qui fournit à son utilisateur un ensemble d'outils pour créer des pages web dynamiques et ergonomiques.



FIGURE 3.30 – Logo Bootstrap

## Oracle Database

Oracle Database est un système de gestion de base de données (SGBD) pour le stockage des données de l'application. Il a été utilisé pour le stockage des données de notre application en plus de la consultation des données déjà stockées dans la base de données Jira (externe).



FIGURE 3.31 – Logo Oracle Database

## SQL

SQL est un langage de requêtes utilisé pour communiquer avec les SGBD. Ce dernier a été utilisé pour interagir avec nos bases de données.



FIGURE 3.32 – Logo SQL

## Mockflow

Mockflow est un outil de dessin de maquette en ligne. Il a été utilisé pour réaliser les prototypes des interfaces.



FIGURE 3.33 – Logo Mockflow

### 3.3 Choix architecturaux

Cette section va être dédiée à la présentation du pattern et style architecturaux qui ont été utilisés dans notre application.

#### 3.3.1 Style architectural

Un style architectural est une modélisation du système et sa répartition. Ce dernier est conçu pour aider à avoir un aperçu du système. Notre système va être décomposé en plusieurs parties :

- Deux couches d'accès aux données : La première pour accéder aux données que nous utiliserons pour calculer les ICP, et la seconde est pour gérer les données de notre application comme les données des utilisateurs ou les données en relation avec les équipes.
- Deux couches métiers : La première va servir d'intermédiaire entre les couches d'accès aux données et la couche métier de contrôle, et la seconde va servir de contrôleur des vues de la couche présentation.
- Une couche présentation qui va s'occuper de l'interface graphique.

Nous avons choisi donc d'utiliser le style architectural N-tiers. La figure 3.34 représente la modélisation de notre système.

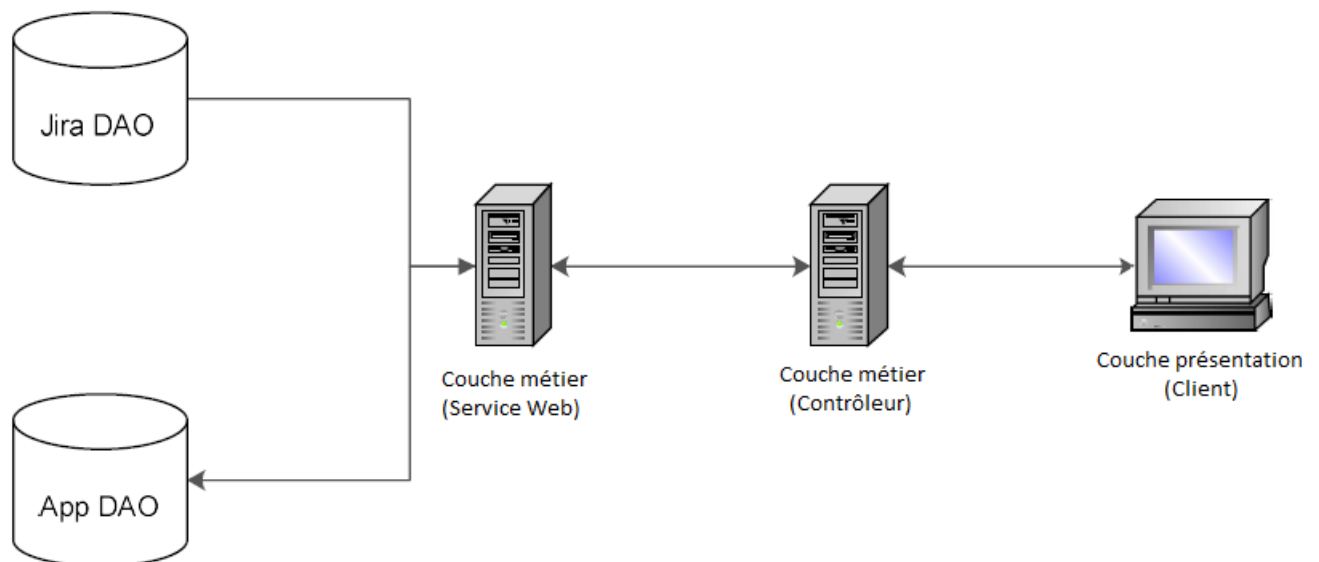


FIGURE 3.34 – Modélisation du système

#### 3.3.2 Pattern architectural

Un pattern architectural est un modèle de référence qui sert à concevoir un système d'information en donnant les sous-systèmes qui le composent, les relations et un ensemble de règles à suivre afin



d'organiser les relations entre ces derniers [17]. Notre système suit deux patterns architecturaux :

- L'inversion de contrôle (IoC) qui est un pattern architectural qui donne le contrôle du flux d'exécution au framework utilisé. Il existe plusieurs représentations de l'IoC, le framework Spring utilise l'injection de dépendance qui permet de découpler les dépendances entre les objets du projet [18].
- Modèle - vue - vue modèle (MVVM) qui est un pattern architectural qui permet de séparer entre le modèle et la vue en mettant une vue-modèle comme intermédiaire entre les deux. Ce dernier joue le rôle de convertisseur entre la vue et le modèle. AngularJS utilisent ce qu'on appelle liaison de données (data-binding en anglais) qui permet de modifier la vue si le modèle change et vis versa [19].

La figure 3.35 présente le pattern architectural de notre application.

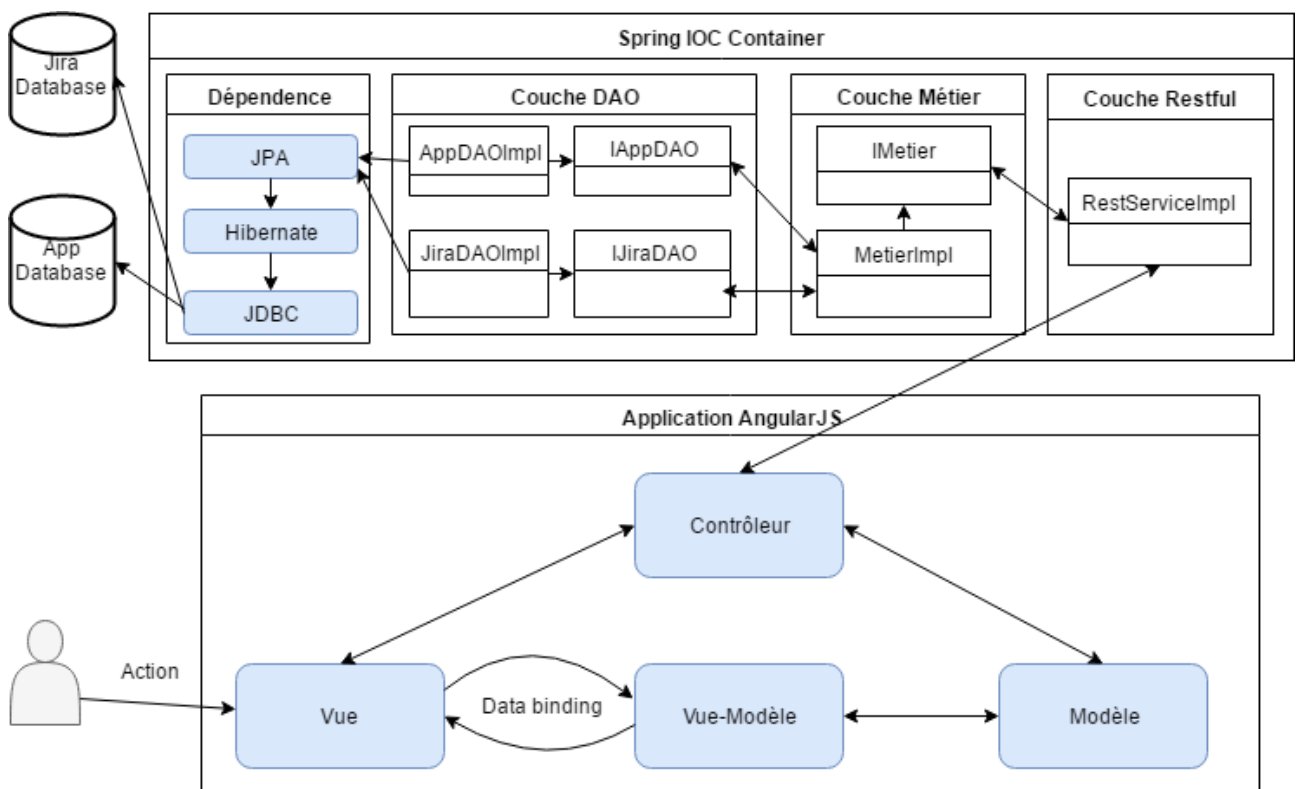


FIGURE 3.35 – Pattern Architectural

## 3.4 Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel et logiciel de notre application. Ensuite nous avons détaillé notre style architectural. Et enfin, nous avons présenté le pattern architectural de notre projet. Dans le prochain chapitre, nous allons présenter l'organisation de nos sprints.

# Chapitre 4

## Organisation des sprints

### 4.1 Introduction

Dans ce chapitre, nous allons présenter les sprints de notre projet. En premier lieu, nous allons spécifier le planning pour chaque sprint. Après, nous présenterons la modélisation statique et dynamique. Ensuite, nous allons détailler le développement du sprint. Et enfin, nous allons présenter le test, la revue du sprint et la rétrospective.

### 4.2 Sprint 1 : Tableaux de bords (du 13-02-2017 au 13-03-2017)

Dans cette section, nous allons détailler la réalisation du sprint 1. En premier lieu, nous allons présenter la planification du sprint en présentant le but du sprint et le backlog sprint. Ensuite nous allons spécifier la modélisation statique et la modélisation dynamique. Puis, nous allons présenter le développement du sprint sous forme de capture d'écrans. Et nous finirons, enfin, avec le test, la revue et la rétrospective du sprint.

#### 4.2.1 Sprint planning

Le but de ce sprint est de créer des tableaux de bords fonctionnels qui utilisent des données de notre base de données Jira. De ce fait, nous proposons le backlog sprint qui est représenté dans la table 4.1.

TABLE 4.1 – Backlog du sprint1 : "Tableaux de bords"

User storyID	User Story	Tâche ID	Tâche
1.1	En tant que membre, membre senior ou chef d'équipe, je souhaite pouvoir m'authentifier.	1.1.1	Réaliser les diagrammes UML de la fonctionnalité "s'authentifier".
		1.1.2	Développer la fonctionnalité "s'authentifier".
		1.1.3	Tester la fonctionnalité "s'authentifier".

User story ID	User Story	Tâche ID	Tâche
3.1	En tant que chef d'équipe, je souhaite pouvoir ajouter un membre dans l'application.	3.1.1	Réaliser les diagrammes UML de la fonctionnalité "Ajouter un membre".
		3.1.2	Développer la fonctionnalité "Ajouter un membre".
		3.1.3	Tester la fonctionnalité "Ajouter un membre".
5.2	En tant que membre senior ou chefd'équipe, je souhaite pouvoir consulter les tableaux de bord des membres de mon équipe.	5.2.1	Réaliser les diagrammes UML de la fonctionnalité "Consulter tout les tableaux de bord".
		5.2.2	Développer la fonctionnalité "Consulter tout les tableaux de bord".
		5.2.3	Tester la fonctionnalité "Consulter tout les tableaux de bord".

### 4.2.2 Modélisation statique

La modélisation statique permet d'avoir une idée sur les objets et les relations entre ces derniers. Nous présentons donc les diagrammes de classes d'analyse pour chaque fonctionnalité.

#### S'authentifier

La figure 4.1 représente le diagramme de classes d'analyse de la fonctionnalité "s'authentifier".

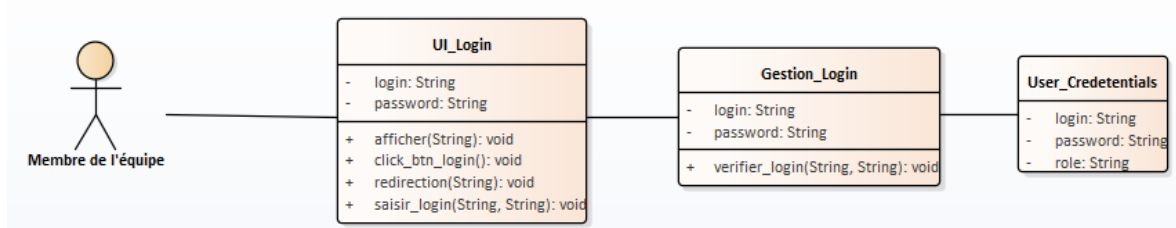


FIGURE 4.1 – Diagramme de classes d'analyse de la fonctionnalité "s'authentifier"

#### Ajouter un membre

La figure 4.2 représente le diagramme de classes d'analyse de la fonctionnalité "ajouter un membre".

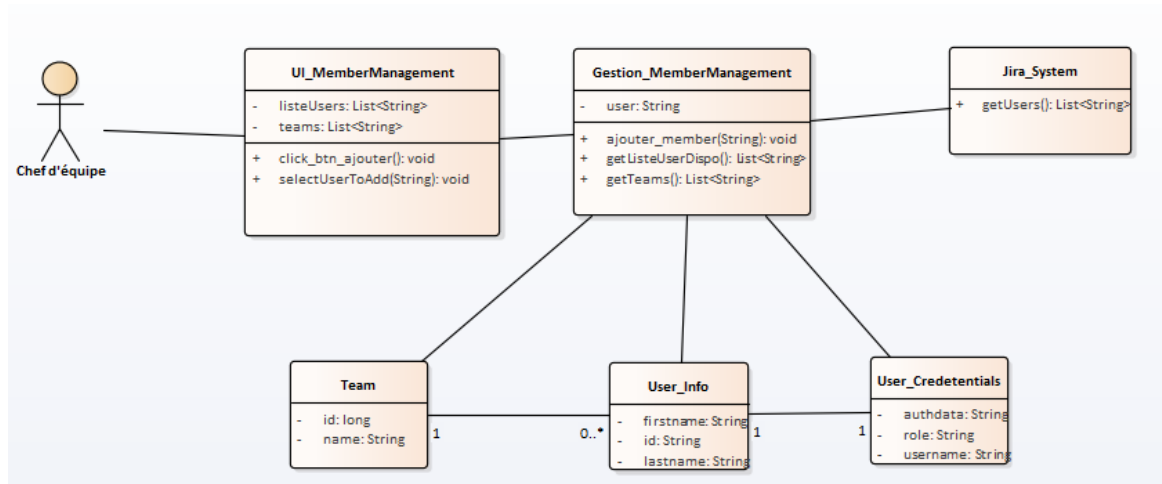


FIGURE 4.2 – Diagramme de classes d'analyse de la fonctionnalité "Ajouter un membre"

### Consulter tout les tableaux de bord

La figure 4.3 représente le diagramme de classes d'analyse de la fonctionnalité "Consulter tout les tableaux de bord".

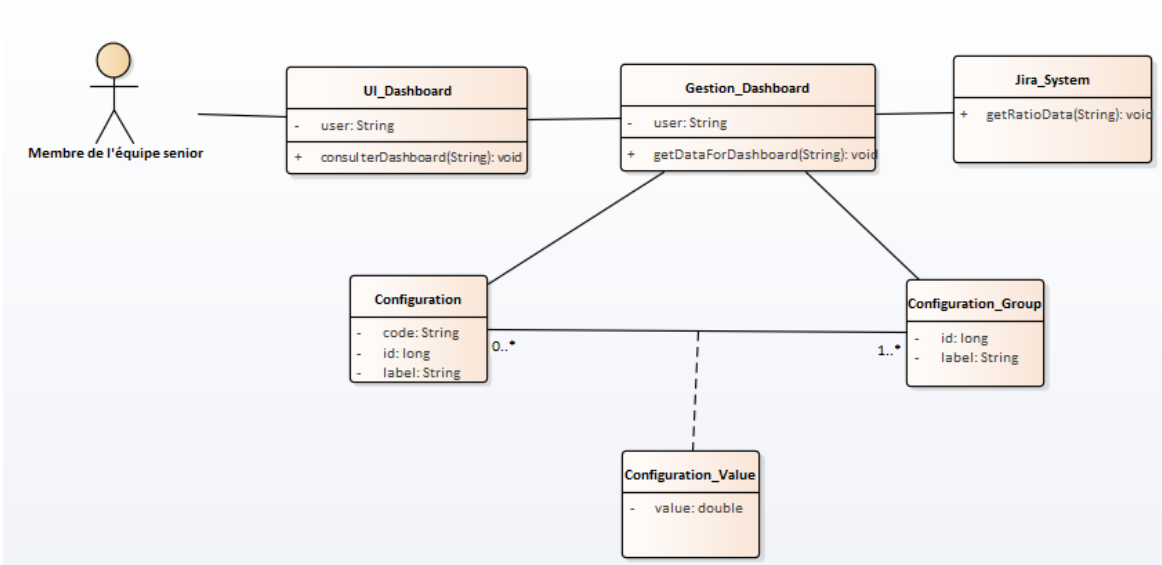


FIGURE 4.3 – Diagramme de classes d'analyse de la fonctionnalité "Consulter tout les tableaux de bord"

### 4.2.3 Modélisation dynamique

La modélisation dynamique permet d'avoir une idée sur le flux d'exécution de notre application. Nous présentons donc les diagrammes de séquences pour chaque fonctionnalité.

#### S'authentifier

Pour s'authentifier, le membre de l'équipe saisie son nom d'utilisateur et son mot de passe. Puis, il valide par le bouton login. Si la combinaison est incorrecte, un message d'erreur apparaît. Sinon, l'utilisateur est redirigé vers la page d'accueil. La figure 4.4 représente le diagramme de séquence de la fonctionnalité "s'authentifier".

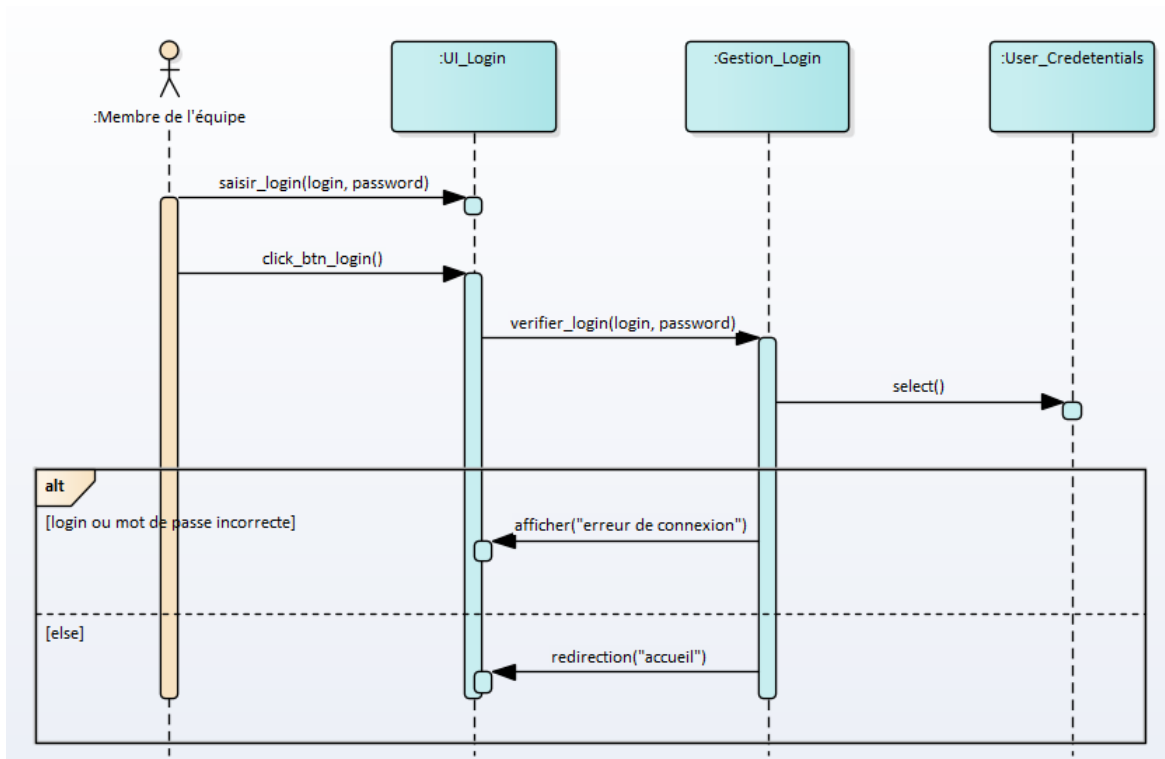


FIGURE 4.4 – Diagramme de séquence de la fonctionnalité "S'authentifier"

#### Ajouter un membre

Pour ajouter un membre, le chef d'équipe doit sélectionner le nom du membre à ajouter. Puis, il valide par le bouton ajouter. La figure 4.5 représente le diagramme de séquence de la fonctionnalité "ajouter un membre".

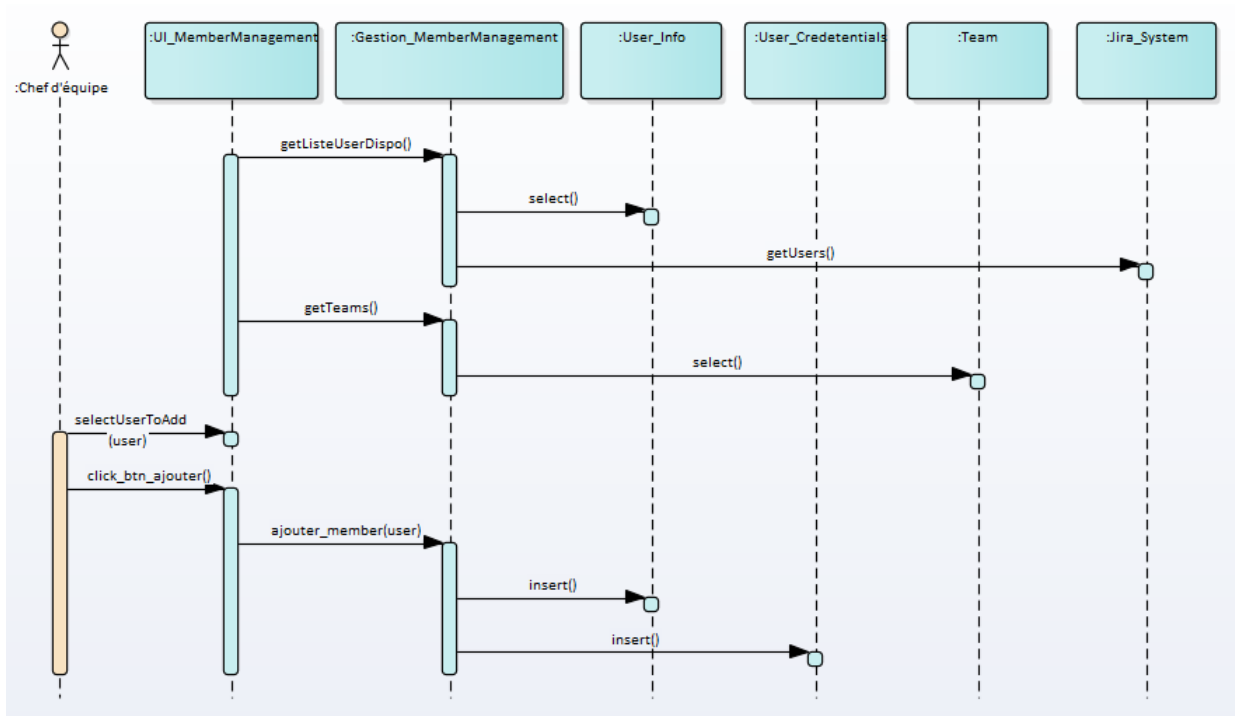


FIGURE 4.5 – Diagramme de séquence de la fonctionnalité "Ajouter un membre"

### Consulter tout les tableaux de bord

Pour consulter tout les tableaux de bord, le membre de l'équipe senior doit cliquer sur le bouton consulter tableau de bord. La figure 4.6 représente le diagramme de séquence de la fonctionnalité "consulter tout les tableaux de bord".

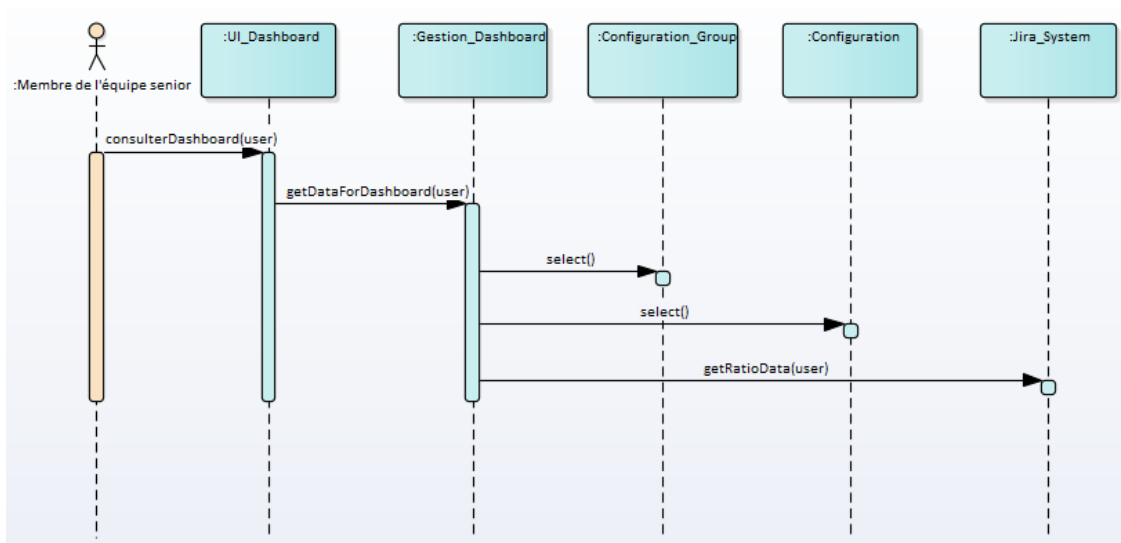


FIGURE 4.6 – Diagramme de séquence de la fonctionnalité "Consulter tout les tableaux de bord"

### 4.2.4 Développement du sprint

Le but de cette section est de montrer l'acheminement du sprint en présentant des captures d'écrans des fonctionnalités réalisés dans ce sprint.

#### S'authentifier

L'utilisateur saisit son nom d'utilisateur et son mot de passe. S'il se trompe dans l'un d'eux, un message d'erreur apparaît. Sinon l'écran d'accueil apparaît. La figure 4.7 représente le message d'erreur reçu lorsqu'on se trompe de nom d'utilisateur ou de mot de passe.

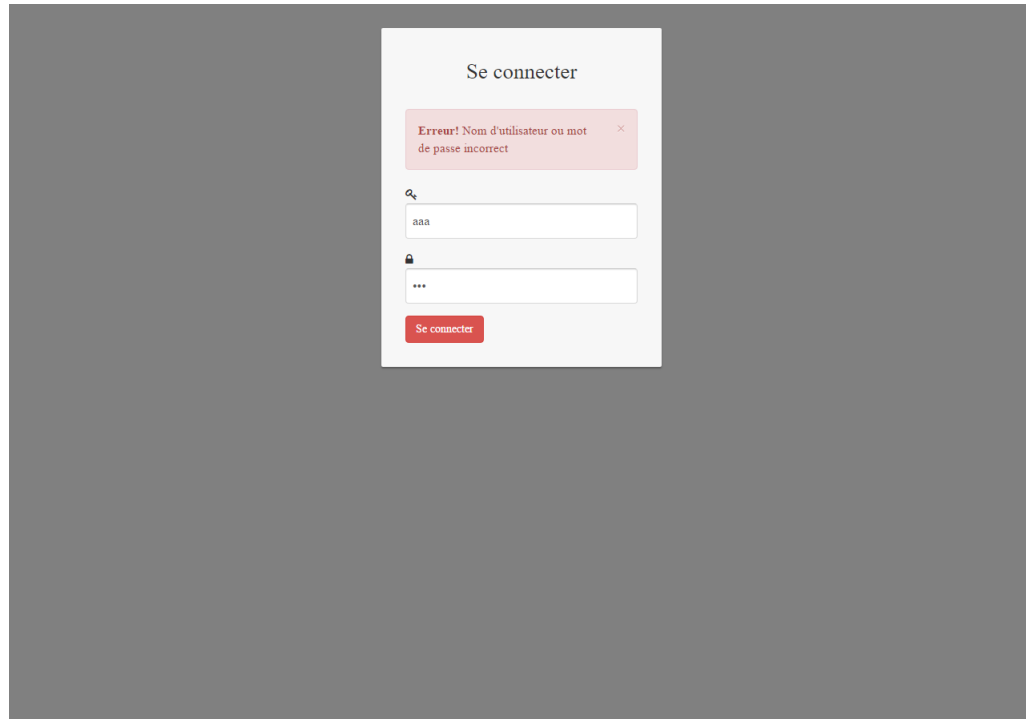


FIGURE 4.7 – Capture d'écran : mot de passe incorrecte

Si le mot de passe est correcte, l'utilisateur se retrouve à l'écran d'accueil.

#### Ajouter un membre

Les utilisateurs qui ont un compte jira mais qui n'ont pas un compte dans notre application sont automatiquement chargé dans la liste des utilisateurs à ajouter. On remplis les champs privilège et équipe puis on valide avec le bouton ajouter. Le message de validation d'ajout est représenté par la figure 4.8.

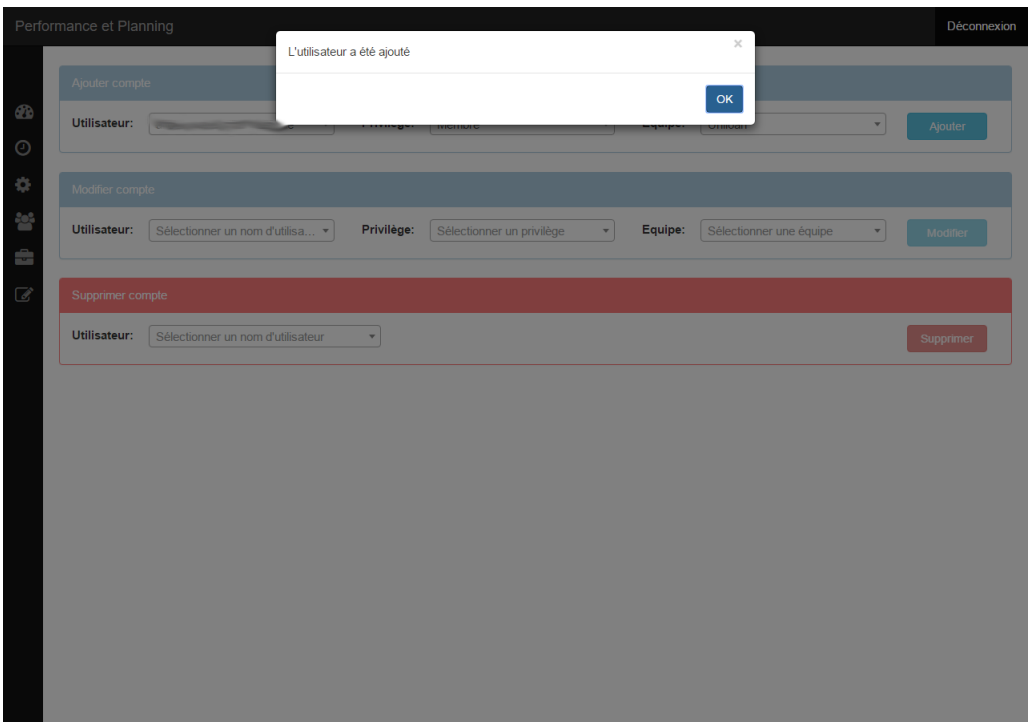


FIGURE 4.8 – Capture d’écran : ajouter membre

**Consulter tout les tableaux de bord**

Le membre de l’équipe senior ou le chef d’équipe peuvent consulter tout les tableaux de bord de leur équipe. Une partie du tableau de bord d’un membre de l’équipe est représenté dans la figure 4.9.

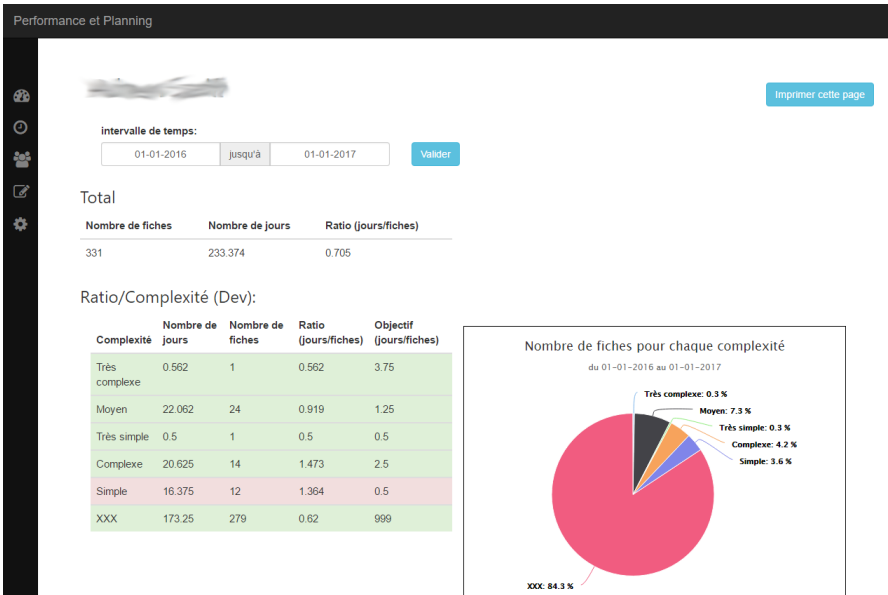


FIGURE 4.9 – Capture d’écran : partie d’un tableau de bord d’un membre



### 4.2.5 Test et revue du sprint

Dans cette section, nous allons présenter l'avancement de notre sprint sous forme de burndown chart. La figure 4.10 représente la burndown chart de notre sprint.

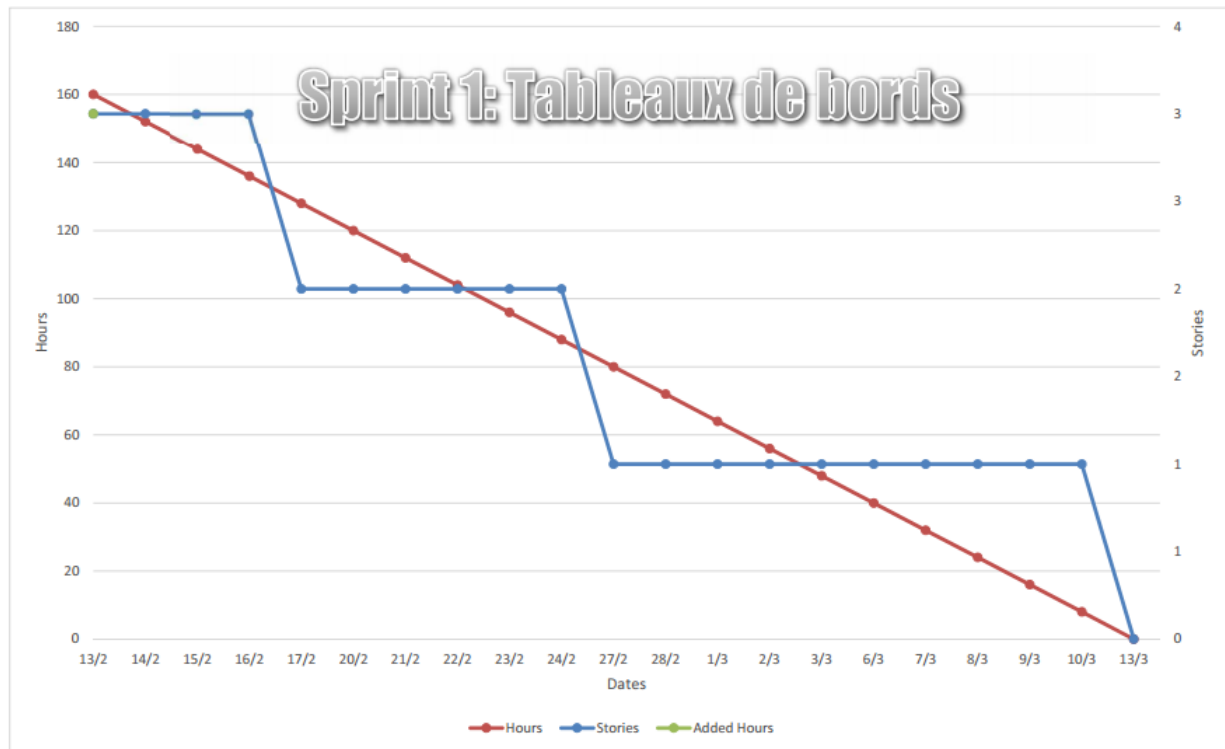


FIGURE 4.10 – Burndown Chart du sprint 1 : Tableaux de bord

### 4.2.6 Rétrospective

Cette section est dédiée à l'énumération des problèmes rencontrés dans ce sprint. Ces derniers se résument comme suit :

- Se familiariser avec la base de données Jira qui contient trop de tables sans clés étrangères.
- Se familiariser avec les techniques de cryptage et hashage pour pouvoir sécuriser l'authentification.
- Comprendre les différents indicateurs de performances pour pouvoir créer les tableaux de bord.

## 4.3 Sprint 2 : Tableau de bord de l'équipe (du 13-03-2017 au 10-04-2017)

Dans cette section, nous allons détailler la réalisation du sprint 2. En premier lieu, nous allons présenter la planification du sprint en présentant le but du sprint et le backlog sprint. Ensuite nous allons spécifier la modélisation statique et la modélisation dynamique. Puis, nous allons présenter

le développement du sprint sous forme de capture d'écrans. Et nous finirons, enfin, avec le test, la revue et la rétrospective du sprint.

### 4.3.1 Sprint planning

Le but de ce sprint est de créer des tableaux de bords de l'équipe fonctionnels qui utilisent des données de notre base de données Jira. De ce fait, nous proposons le backlog sprint qui est représenté dans la table 4.2.

TABLE 4.2 – Backlog du sprint2 : "Tableau de bord de l'équipe"

User storyID	User Story	Tâche ID	Tâche
2.1	En tant que chef d'équipe, je souhaite pouvoir créer une nouvelle équipe.	2.1.1	Réaliser les diagrammes UML de la fonctionnalité "créer une équipe".
		2.1.2	Développer la fonctionnalité "créer une équipe".
		2.1.3	Tester la fonctionnalité "créer une équipe".
2.2	En tant que chef d'équipe, je souhaite pouvoir supprimer une équipe.	2.2.1	Réaliser les diagrammes UML de la fonctionnalité "supprimer une équipe".
		2.2.2	Développer la fonctionnalité "supprimer une équipe".
		2.2.3	Tester la fonctionnalité "supprimer une équipe".
3.2	En tant que chef d'équipe, je souhaite pouvoir supprimer un membre de l'application.	3.2.1	Réaliser les diagrammes UML de la fonctionnalité "supprimer un membre".
		3.2.2	Développer la fonctionnalité "supprimer un membre".
		3.2.3	Tester la fonctionnalité "supprimer un membre".
3.3	En tant que chef d'équipe, je souhaite pouvoir modifier un membre de l'application.	3.3.1	Réaliser les diagrammes UML de la fonctionnalité "modifier un membre".
		3.3.2	Développer la fonctionnalité "modifier un membre".
		3.3.3	Tester la fonctionnalité "modifier un membre".

User storyID	User Story	Tâche ID	Tâche
4.1	En tant que membre senior ou chef d'équipe, je souhaite pouvoir consulter le tableaux de bord de l'équipe.	4.1.1	Réaliser les diagrammes UML de la fonctionnalité "consulter le tableau de bord de l'équipe".
		4.1.2	Développer la fonctionnalité "consulter le tableau de bord de l'équipe".
		4.1.3	Tester la fonctionnalité "consulter le tableau de bord de l'équipe".
6.1	En tant que chef d'équipe, je souhaite pouvoir modifier les valeurs des objectifs ICP.	6.1.1	Réaliser les diagrammes UML de la fonctionnalité "modifier les valeurs des objectifs ICP".
		6.1.2	Développer la fonctionnalité "modifier les valeurs des objectifs ICP".
		6.1.3	Tester la fonctionnalité "modifier les valeurs des objectifs ICP".

### 4.3.2 Modélisation statique

Nous présentons dans cette partie la modélisation statique de ce sprint sous forme de diagrammes de classes d'analyse pour chaque fonctionnalité.

#### Créer une équipe

La figure 4.11 représente le diagramme de classes d'analyse de la fonctionnalité "créer une équipe".

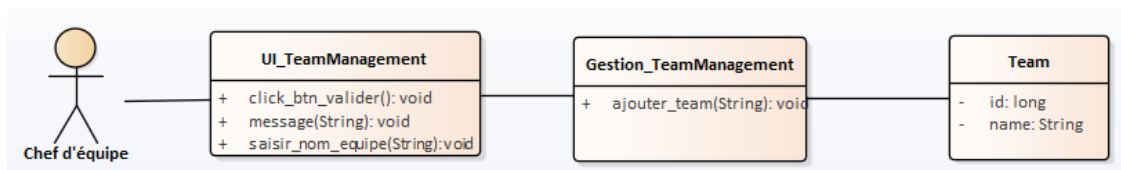


FIGURE 4.11 – Diagramme de classes d'analyse de la fonctionnalité "créer une équipe"

#### Supprimer une équipe

La figure 4.12 représente le diagramme de classes d'analyse de la fonctionnalité "supprimer une équipe".

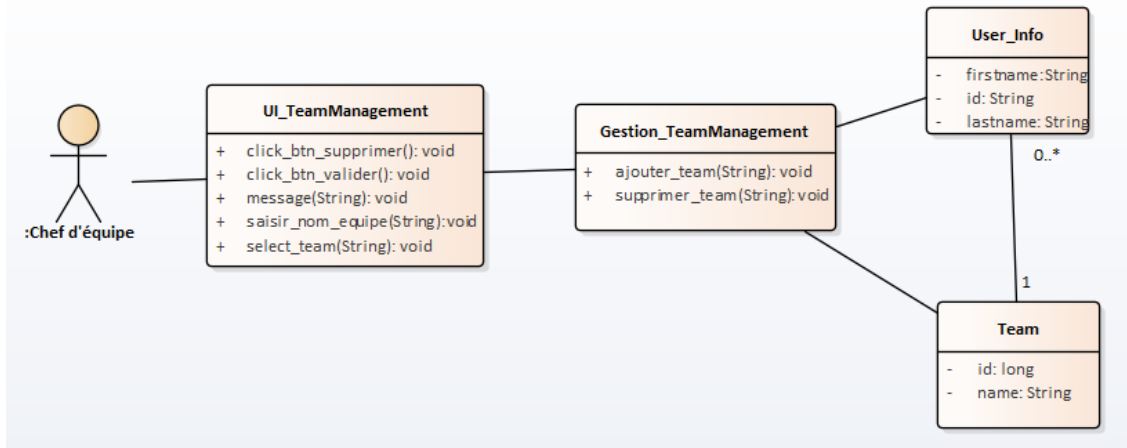


FIGURE 4.12 – Diagramme de classes d'analyse de la fonctionnalité "supprimer une équipe"

### Modifier un membre

La figure 4.13 représente le diagramme de classes d'analyse de la fonctionnalité "modifier un membre".

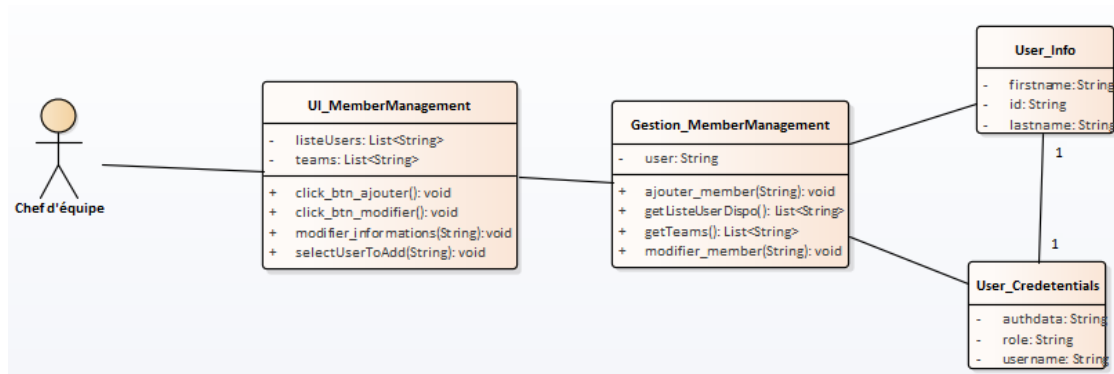


FIGURE 4.13 – Diagramme de classes d'analyse de la fonctionnalité "modifier un membre"

### Supprimer un membre

La figure 4.14 représente le diagramme de classes d'analyse de la fonctionnalité "supprimer un membre".

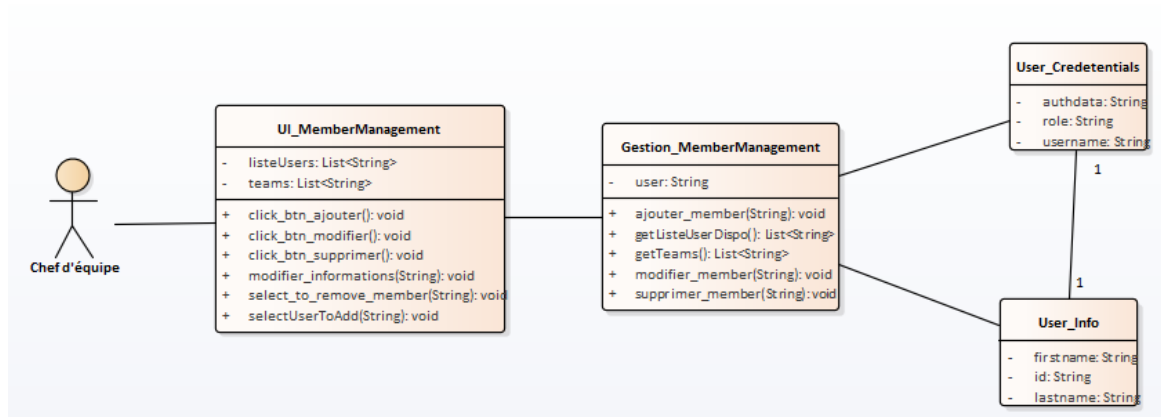


FIGURE 4.14 – Diagramme de classes d'analyse de la fonctionnalité "supprimer un membre"

### Consulter le tableau de bord de l'équipe

La figure 4.15 représente le diagramme de classes d'analyse de la fonctionnalité "consulter le tableau de bord de l'équipe".

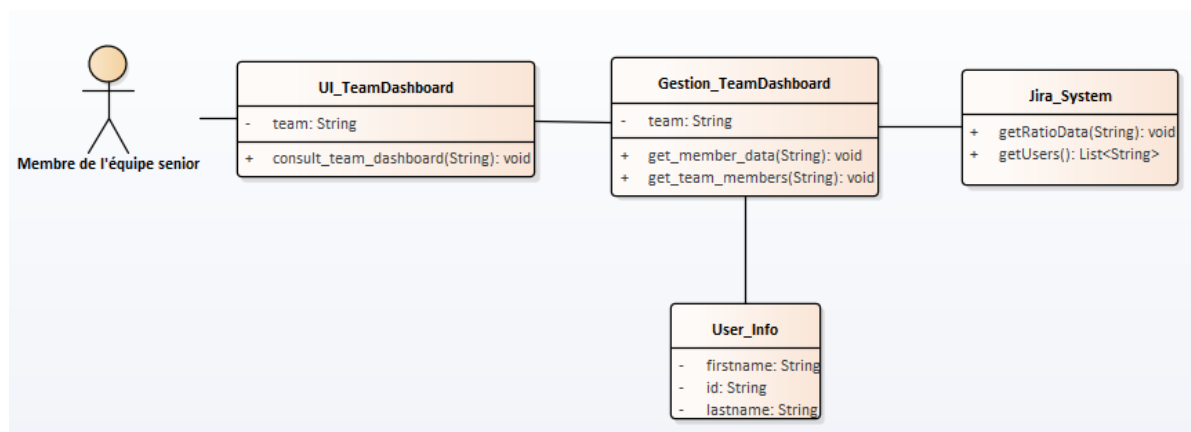


FIGURE 4.15 – Diagramme de classes d'analyse de la fonctionnalité "consulter le tableau de bord de l'équipe"

### Modifier les valeurs des objectifs ICP

La figure 4.16 représente le diagramme de classes d'analyse de la fonctionnalité "modifier les valeurs des objectifs ICP".

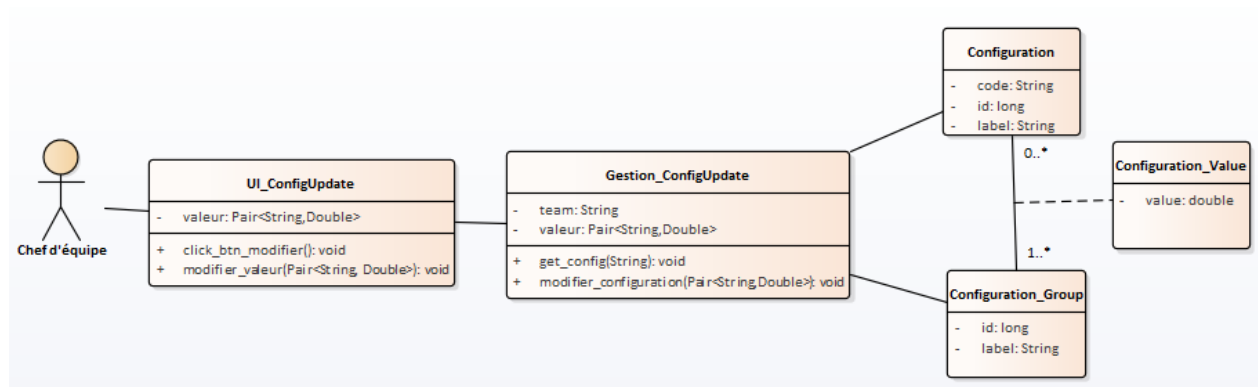


FIGURE 4.16 – Diagramme de classes d'analyse de la fonctionnalité "modifier les valeurs des objectifs ICP"

### 4.3.3 Modélisation dynamique

Nous présentons dans cette partie la modélisation dynamique de ce sprint sous forme de diagrammes de séquences pour chaque fonctionnalité.

#### Créer une équipe

Pour créer une équipe, le chef d'équipe saisie le nom d'une équipe à ajouter. Puis, il clique sur le bouton valider. Si le nom n'existe pas dans la base de données, l'équipe est automatiquement ajoutée et un message validant l'ajout apparaît. Sinon, un message d'erreur apparaît sur l'écran. La figure 4.17 représente le diagramme de séquence de la fonctionnalité "créer une équipe".

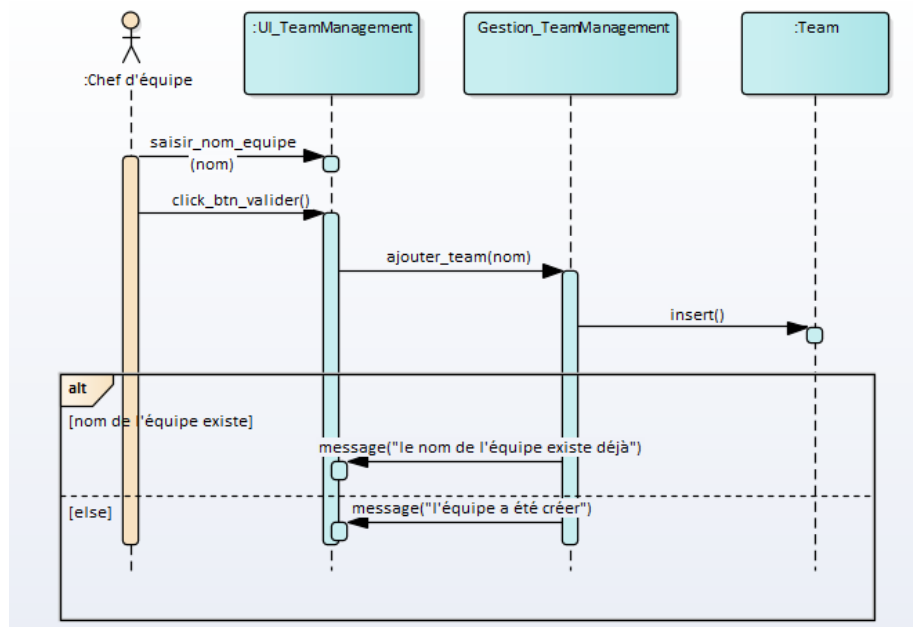


FIGURE 4.17 – Diagramme de séquence de la fonctionnalité "créer une équipe"

### Supprimer une équipe

Pour supprimer une équipe, le chef d'équipe sélectionne le nom de l'équipe à supprimer. Puis, il valide par le bouton valider. l'équipe est alors supprimée. La figure 4.18 représente le diagramme de séquence de la fonctionnalité "supprimer une équipe".

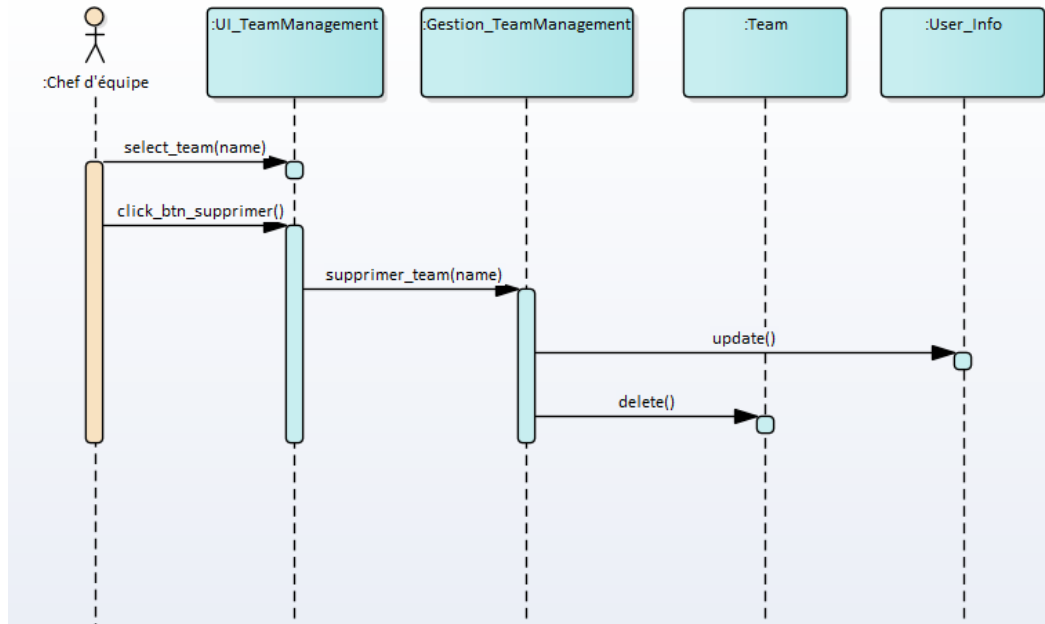


FIGURE 4.18 – Diagramme de séquence de la fonctionnalité "supprimer une équipe"

### Modifier un membre

Pour modifier un membre, le chef d'équipe sélectionne le nom du membre à modifier. Puis, il fait les changements nécessaires et il valide par le bouton valider. La figure 4.19 représente le diagramme de séquence de la fonctionnalité "modifier un membre".

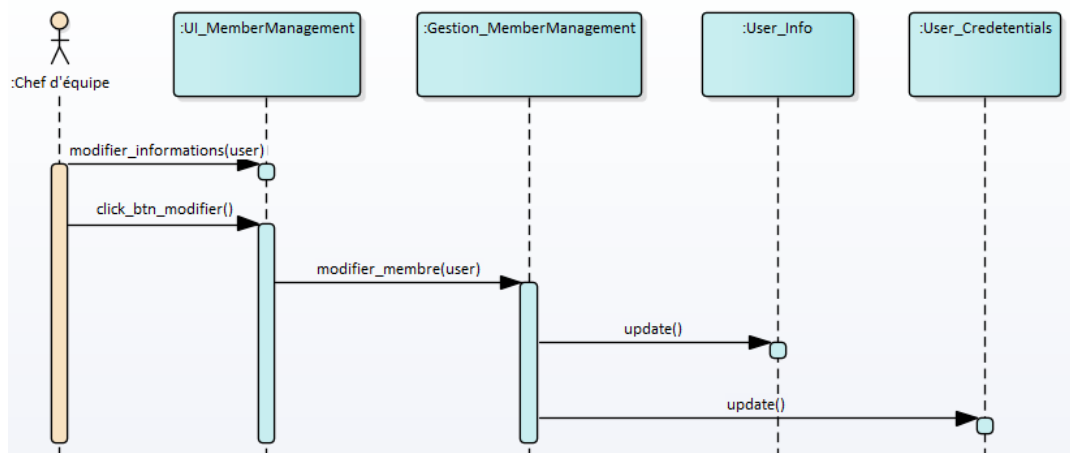


FIGURE 4.19 – Diagramme de séquence de la fonctionnalité "modifier un membre"

### Supprimer un membre

Pour supprimer un membre, le chef d'équipe sélectionne le nom du membre à supprimer. Puis, il valide par le bouton supprimer. La figure 4.20 représente le diagramme de séquence de la fonctionnalité "supprimer un membre".

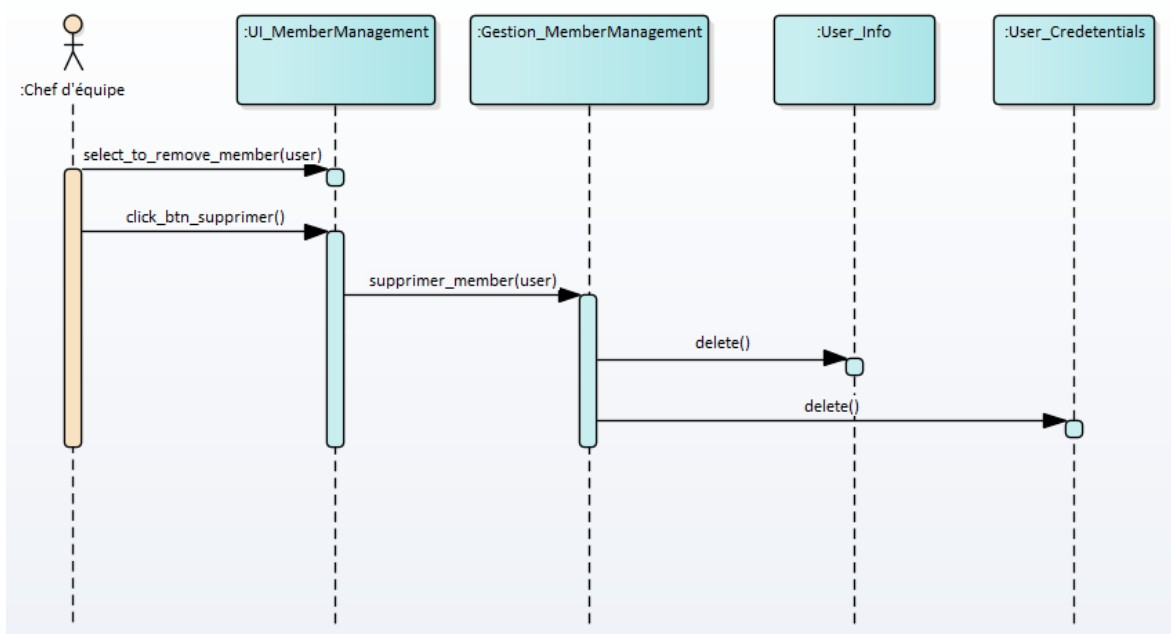


FIGURE 4.20 – Diagramme de séquence de la fonctionnalité "supprimer un membre"

### Consulter le tableau de bord de l'équipe

Pour consulter le tableau de bord de l'équipe, le membre de l'équipe senior clique sur le bouton consulter. La figure 4.21 représente le diagramme de séquence de la fonctionnalité "consulter le tableau de bord de l'équipe".



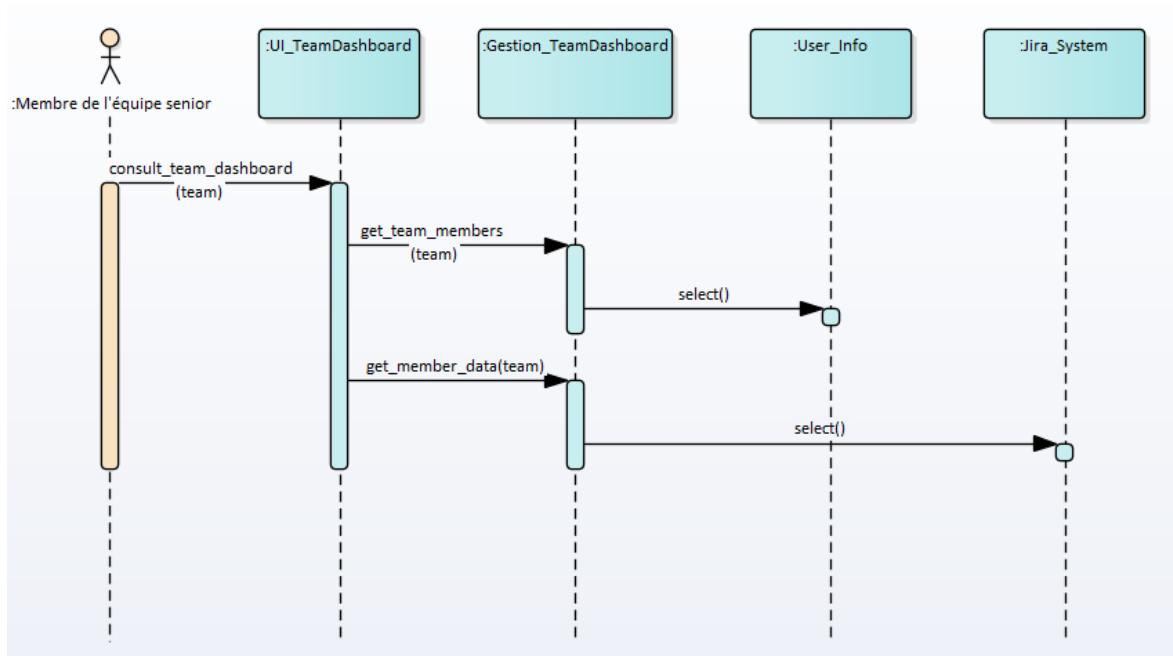


FIGURE 4.21 – Diagramme de séquence de la fonctionnalité "consulter le tableau de bord de l'équipe"

### Modifier les valeurs des objectifs ICP

Pour modifier les valeurs des objectifs ICP, le chef d'équipe modifie les valeurs des champs dans la page. Puis, il valide par le bouton modifier. La figure 4.22 représente le diagramme de séquence de la fonctionnalité "modifier les valeurs des objectifs ICP".

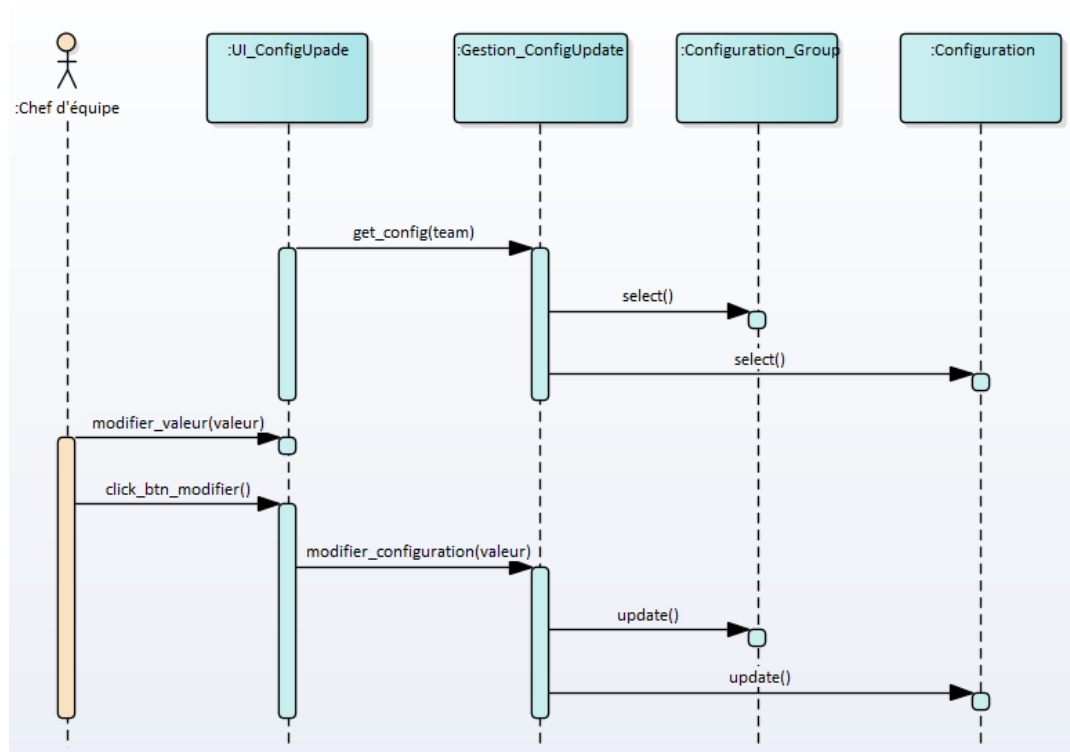


FIGURE 4.22 – Diagramme de séquence de la fonctionnalité "modifier les valeurs des objectifs ICP"

#### 4.3.4 Développement du sprint

Le but de cette section est de montrer l'acheminement du sprint en présentant des captures d'écrans des fonctionnalités réalisés dans ce sprint.

##### Créer une équipe

L'utilisateur saisit le nom de l'équipe qu'il veut ajouter. Si le nom existe dans la base de données, un message d'avertissement apparaît et le bouton d'ajout se désactive. Sinon l'équipe est ajoutée dans la base de données. La figure 4.23 représente le message de validation d'ajout d'une équipe.

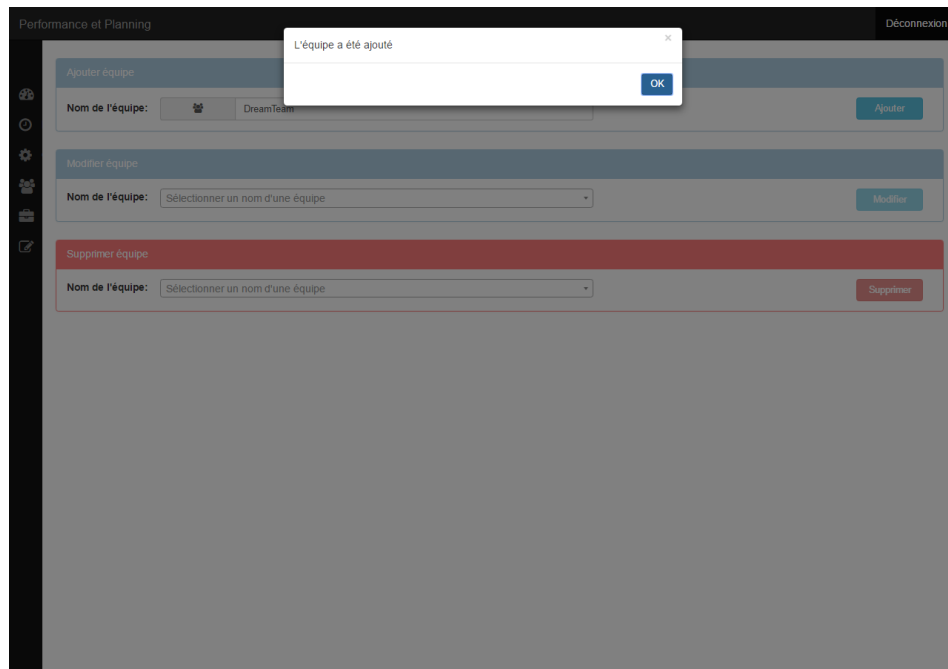


FIGURE 4.23 – Capture d’écran : créer une équipe

### Supprimer une équipe

L'utilisateur sélectionne l'équipe à supprimer puis il clique sur le bouton supprimer. L'équipe est alors supprimée de la base de données. La figure 4.24 représente le message de validation de suppression d'une équipe.

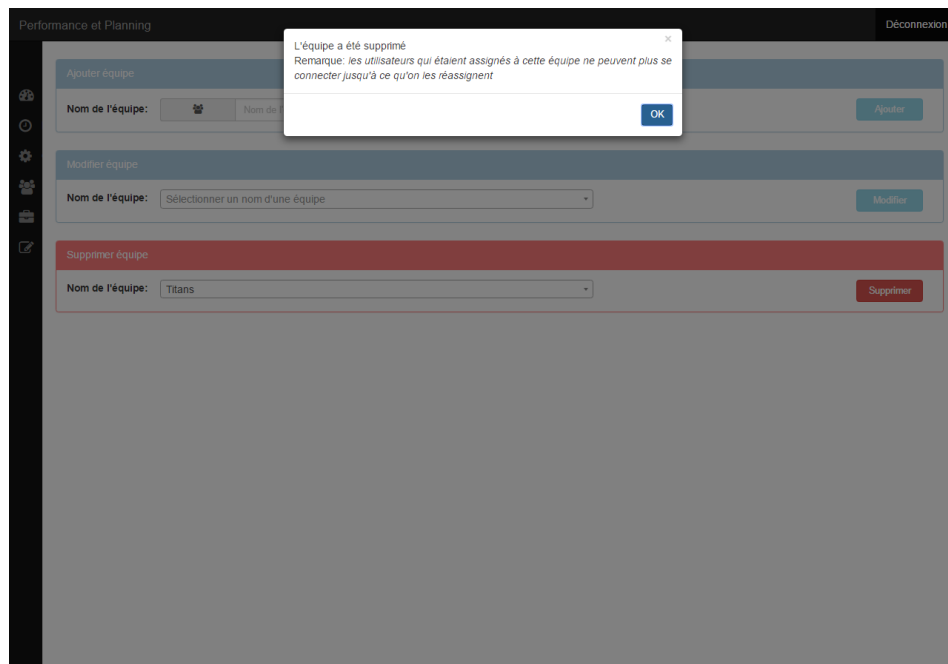


FIGURE 4.24 – Capture d’écran : supprimer une équipe

### Modifier un membre

L'utilisateur sélectionne le membre à modifier. Puis il fait les changements nécessaires et il valide par le bouton modifier. La figure 4.25 représente le message de validation de modification d'un membre.

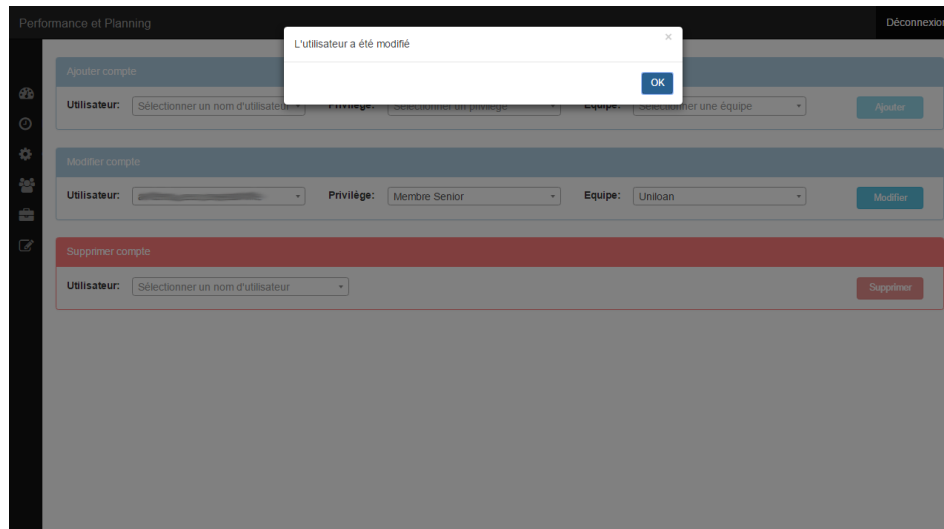


FIGURE 4.25 – Capture d'écran : modifier un membre

### Supprimer un membre

L'utilisateur sélectionne le membre à supprimer puis il clique sur le bouton supprimer. Le membre est alors supprimée de la base de données. La figure 4.26 représente le message de validation de suppression d'un membre.

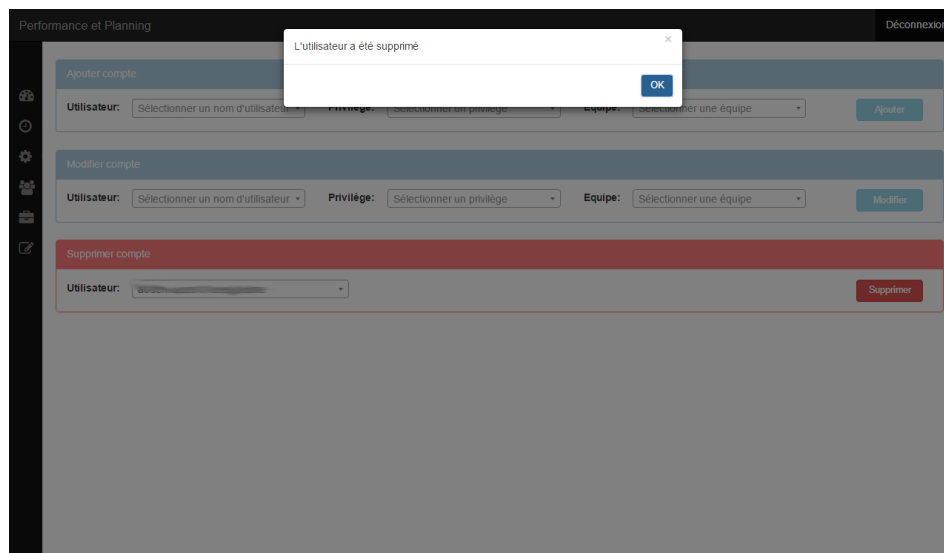


FIGURE 4.26 – Capture d'écran : supprimer un membre

### Consulter le tableau de bord de l'équipe

L'utilisateur peut accéder au tableau de bord de l'équipe s'il clique sur le bouton "Tableau de bord" du menu vertical à gauche. La figure 4.27 représente le tableau de bord de l'équipe.

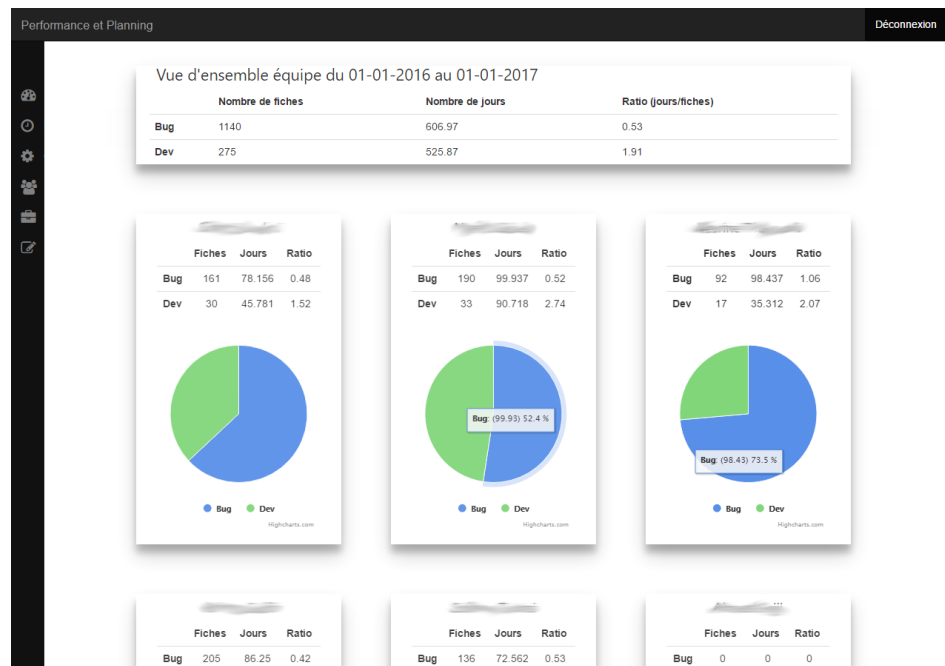


FIGURE 4.27 – Capture d'écran : tableau de bord de l'équipe

### Modifier les valeurs des objectifs ICP

L'utilisateur peut modifier les valeurs des objectifs ICP puis il valide la modification en cliquant sur le bouton "Enregistrer les modifications". La figure 4.28 représente le message de validation des modifications.

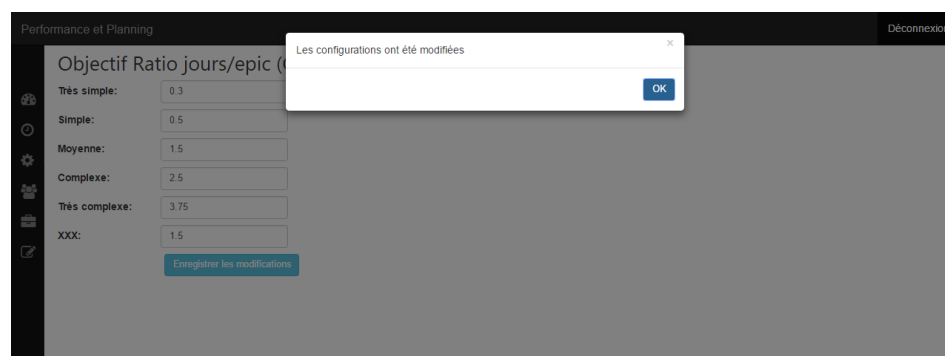


FIGURE 4.28 – Capture d'écran : tableau de bord de l'équipe

### 4.3.5 Test et revue du sprint

Dans cette section, nous allons présenter l'avancement de notre sprint sous forme de burndown chart. La figure 4.29 représente la burndown chart de notre sprint.

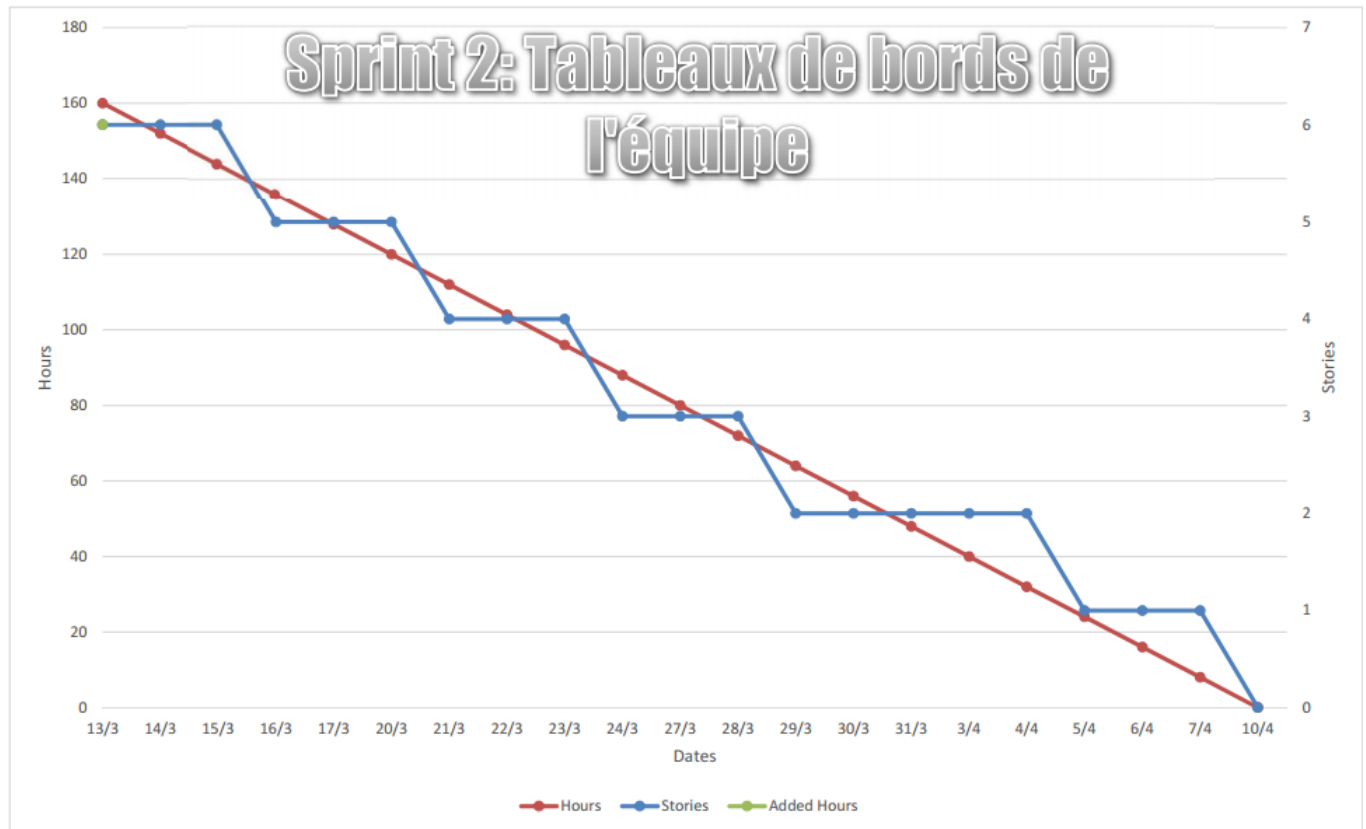


FIGURE 4.29 – Burndown Chart du sprint 2 : Tableaux de bord de l'équipe

### 4.3.6 Rétrospective

Cette section est dédiée à l'énumération des problèmes rencontrés dans ce sprint. Ces derniers se résument comme suit :

- Pouvoir gérer les privilèges de chaque utilisateur de notre application
- Trouver une disposition relativement ergonomique pour afficher la liste de tout les membres de l'équipe.

## 4.4 Sprint 3 : Planification (du 10-04-2017 au 08-05-2017)

Dans cette section, nous allons détailler la réalisation du sprint 3. En premier lieu, nous allons présenter la planification du sprint en présentant le but du sprint et le backlog sprint. Ensuite nous allons spécifier la modélisation statique et la modélisation dynamique. Puis, nous allons présenter le développement du sprint sous forme de capture d'écrans. Et nous finirons, enfin, avec le test, la revue et la rétrospective du sprint.

### 4.4.1 Sprint planning

Le but de ce sprint est de créer des tableaux de bords de l'équipe fonctionnels qui utilisent des données de notre base de données Jira. De ce fait, nous proposons le backlog sprint qui est représenté dans la table 4.3.

TABLE 4.3 – Backlog du sprint3 : "Planification"

User storyID	User Story	Tâche ID	Tâche
1.2	En tant que membre, membre senior ou chef d'équipe, je souhaite pouvoir modifier mon mot de passe.	2.1.1	Réaliser les diagrammes UML de la fonctionnalité "modifier le mot de passe".
		2.1.2	Développer la fonctionnalité "modifier le mot de passe".
		2.1.3	Tester la fonctionnalité "modifier le mot de passe".
5.1	En tant que membre ou membre senior de l'équipe, je souhaite pouvoir consulter le tableau de bord me représentant.	2.2.1	Réaliser les diagrammes UML de la fonctionnalité "consulter le tableau de bord personnel".
		2.2.2	Développer la fonctionnalité "consulter le tableau de bord personnel".
		2.2.3	Tester la fonctionnalité "consulter le tableau de bord personnel".
7.1	En tant que chef d'équipe, je souhaite pouvoir saisir les informations relatives à la planification et consulter le résultat obtenu.	3.2.1	Réaliser les diagrammes UML de la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat".
		3.2.2	Développer la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat".
		3.2.3	Tester la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat".

### 4.4.2 Modélisation statique

Nous présentons dans cette partie la modélisation statique de ce sprint sous forme de diagrammes de classes d'analyse pour chaque fonctionnalité.

### Modifier le mot de passe

La figure 4.30 représente le diagramme de classes d'analyse de la fonctionnalité "modifier le mot de passe".

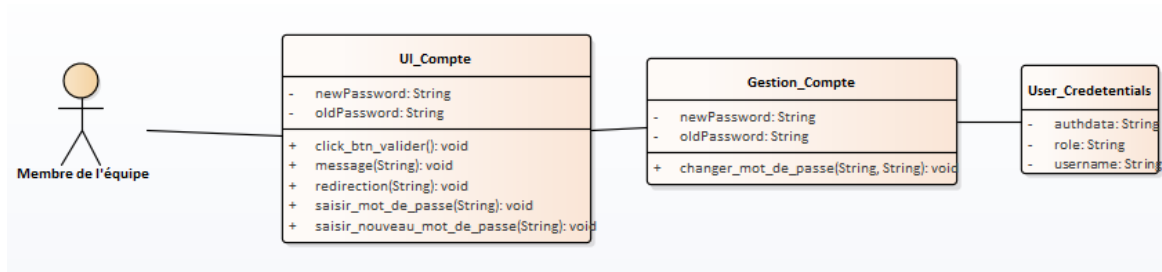


FIGURE 4.30 – Diagramme de classes d'analyse de la fonctionnalité "modifier le mot de passe"

### Consulter le tableau de bord personnel

La figure 4.31 représente le diagramme de classes d'analyse de la fonctionnalité "consulter le tableau de bord personnel".

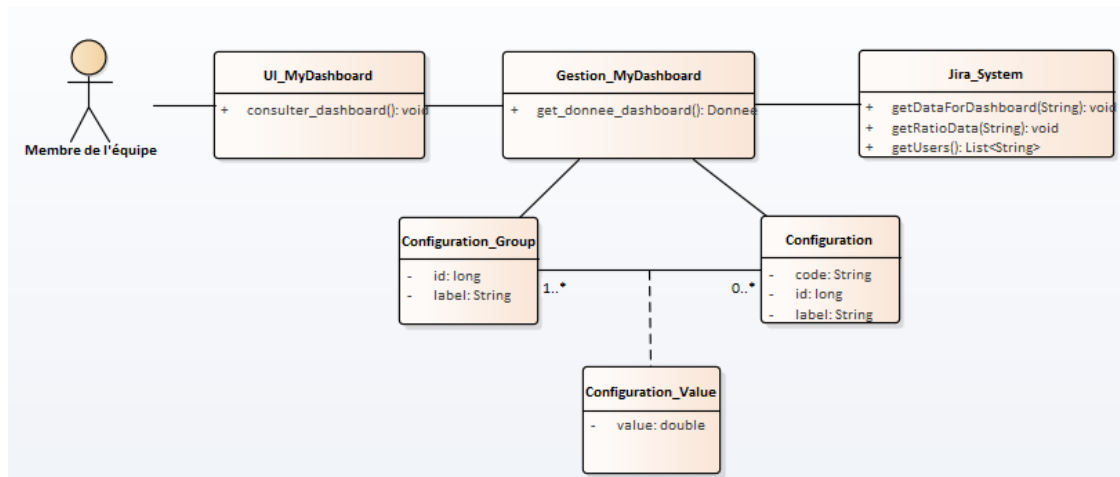


FIGURE 4.31 – Diagramme de classes d'analyse de la fonctionnalité "consulter le tableau de bord personnel"

### Editer les informations relatives à la planification et consulter le résultat

La figure 4.32 représente le diagramme de classes d'analyse de la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat".



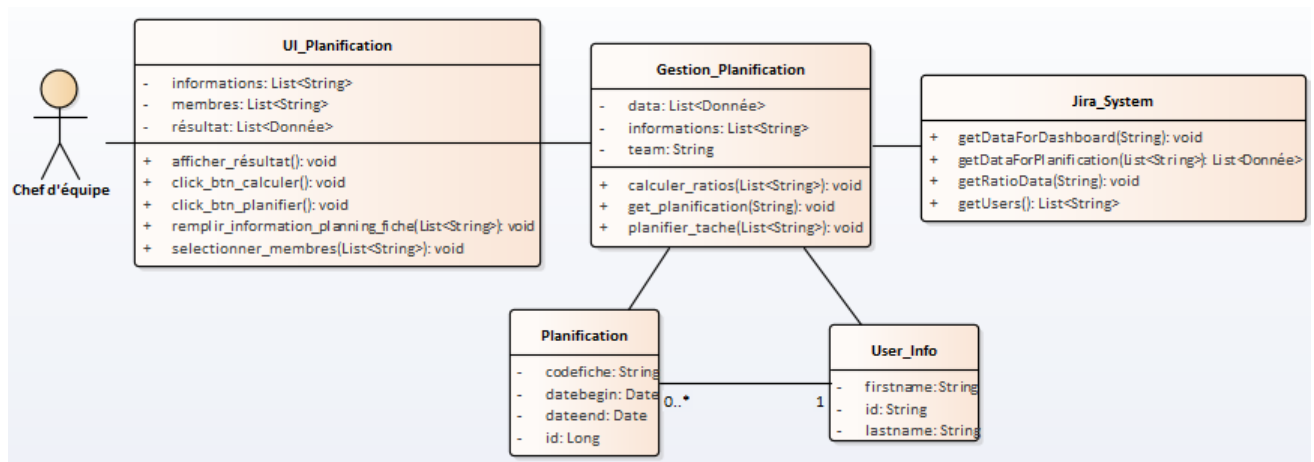


FIGURE 4.32 – Diagramme de classes d’analyse de la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat"

### 4.4.3 Modélisation dynamique

Nous présentons dans cette partie la modélisation dynamique de ce sprint sous forme de diagrammes de séquences pour chaque fonctionnalité.

#### Modifier le mot de passe

Pour modifier le mot de passe, le membre de l’équipe doit saisir l’ancien et le nouveau mot de passe. Puis, il valide par le bouton valider. Si le mot de passe est incorrecte, un message d’erreur apparaît. Sinon, un message de validation apparaît pour confirmer le changement. La figure 4.33 représente le diagramme de séquence de la fonctionnalité "modifier le mot de passe".

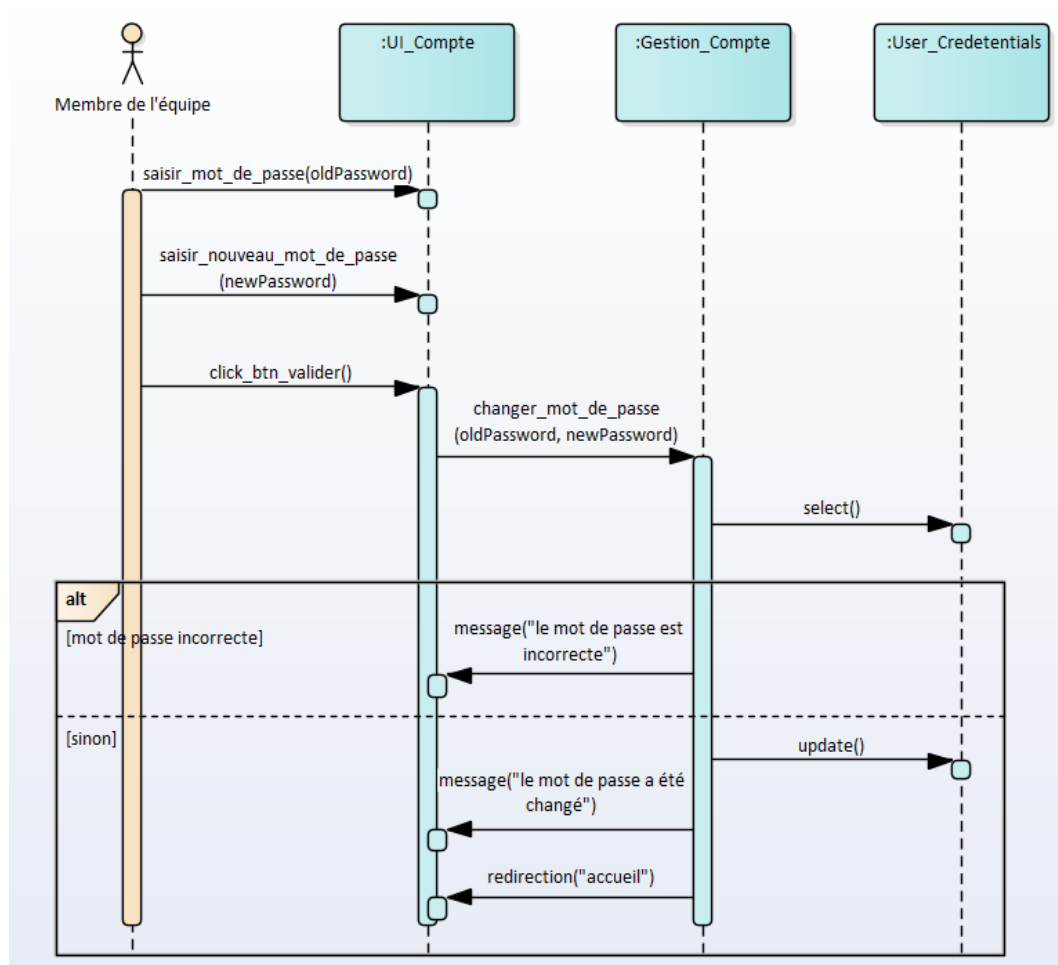


FIGURE 4.33 – Diagramme de séquence de la fonctionnalité "modifier le mot de passe"

### Consulter le tableau de bord personnel

Pour consulter le tableau de bord personnel, le membre de l'équipe doit cliquer sur le bouton tableau de bord personnel. La figure 4.34 représente le diagramme de séquence de la fonctionnalité "consulter le tableau de bord personnel".

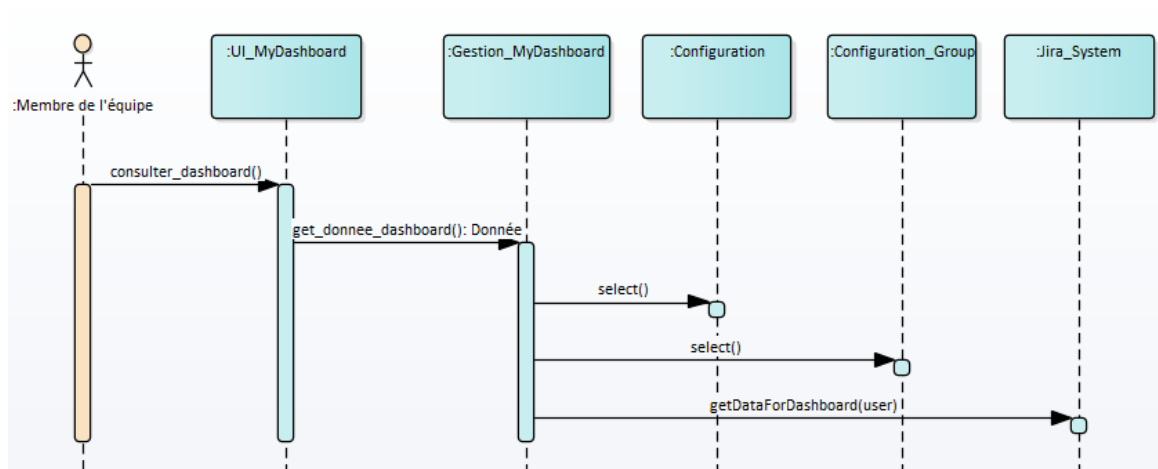


FIGURE 4.34 – Diagramme de séquence de la fonctionnalité "consulter le tableau de bord personnel"

### Editer les informations relatives à la planification et consulter le résultat

Pour éditer les informations relatives à la planification et consulter le résultat, le chef d'équipe doit remplir les informations relatives à la fiche. Puis, il clique sur le bouton calculer. Après l'affichage du résultat, le chef d'équipe sélectionne les membres. Et enfin, il clique sur le bouton planifier. La figure 4.35 représente le diagramme de séquence de la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat".

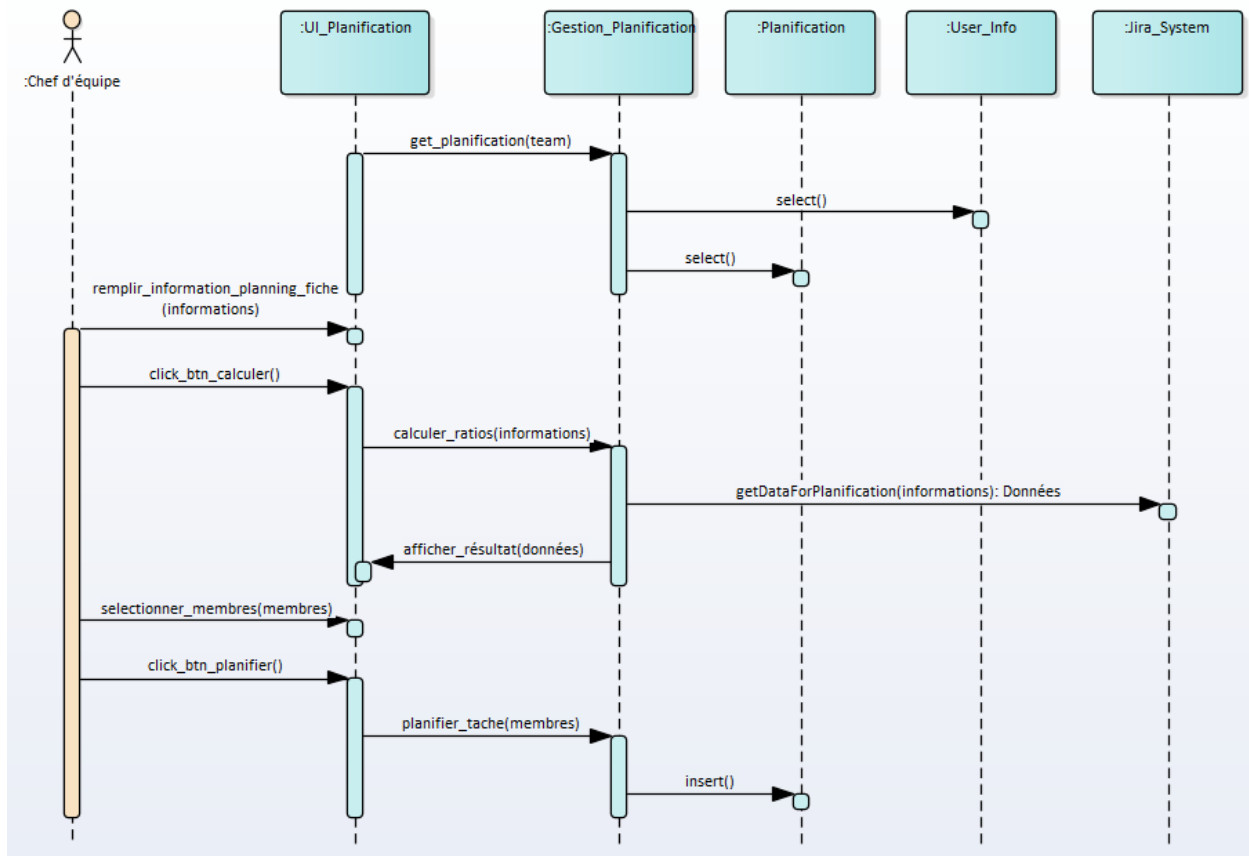


FIGURE 4.35 – Diagramme de séquence de la fonctionnalité "éditer les informations relatives à la planification et consulter le résultat"

#### 4.4.4 Développement du sprint

Le but de cette section est de montrer l'acheminement du sprint en présentant des captures d'écrans des fonctionnalités réalisés dans ce sprint.

##### Modifier le mot de passe

L'utilisateur saisit son ancien mot de passe. Puis il saisit deux fois le nouveau mot de passe et il clique sur le bouton "Enregistrer les modifications". S'il saisit l'ancien mot de passe incorrectement, un message d'erreur apparaît. Sinon, un message de validation de changement de mot de passe apparaît. La figure 4.36 représente le message de validation du changement de mot de passe.

Performance et Planning Déconnexion

Le mot de passe a été changé

Ancien mot de passe:

Nouveau mot de passe:

Répéter mot de passe:

Champ requis\*

Champ requis\*

Champ requis\*

Champ requis\*

Enregistrer les modifications

FIGURE 4.36 – Capture d’écran : modifier le mot de passe

### Consulter le tableau de bord personnel

L'utilisateur peut consulter son tableau de bord. Ceci en cliquant sur le bouton tableau de bord du menu vertical situé à gauche. La figure 4.37 représente le tableau de bord d'un simple membre.

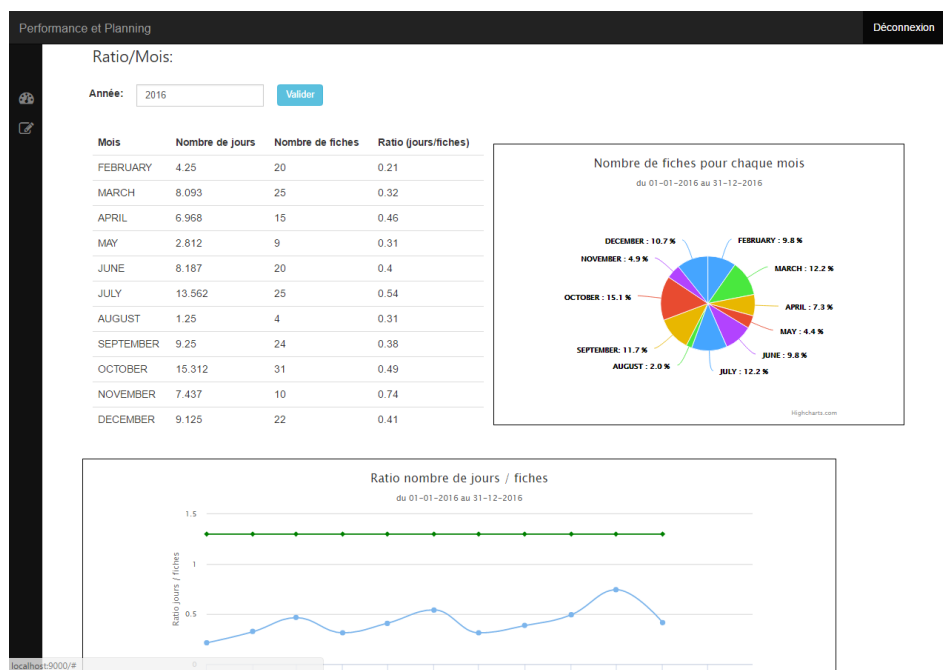


FIGURE 4.37 – Capture d’écran : consulter le tableau de bord personnel

### Editer les informations relatives à la planification et consulter le résultat

L'utilisateur peut rechercher une tâche déjà existante dans la base de données Jira pour remplir les champs de la planification ou les sélectionner lui-même. Il clique après sur le bouton calculer pour avoir la liste des membres de l'équipe et leurs ratios. Il sélectionne ceux qui vont réaliser la tâche et le ratio total s'actualise. Si l'utilisateur clique sur le bouton Valider, la planification est enregistré dans la base de données. La figure 4.38 représente le message de validation de la planification.

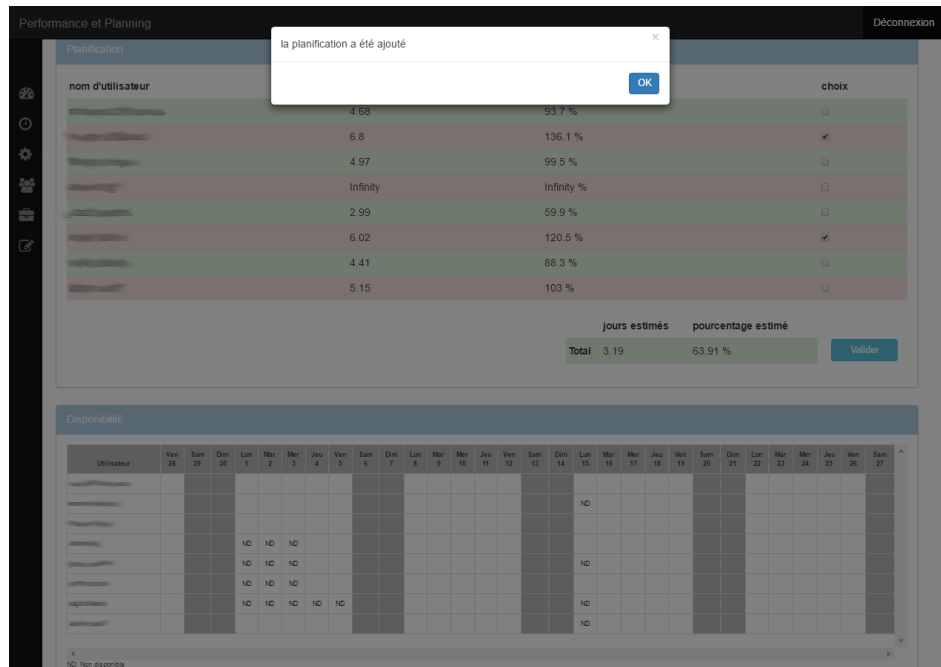


FIGURE 4.38 – Capture d'écran : éditer les informations relatives à la planification et consulter le résultat

#### 4.4.5 Test et revue du sprint

Dans cette section, nous allons présenter l'avancement de notre sprint sous forme de burndown chart. La figure 4.39 représente la burndown chart de notre sprint.

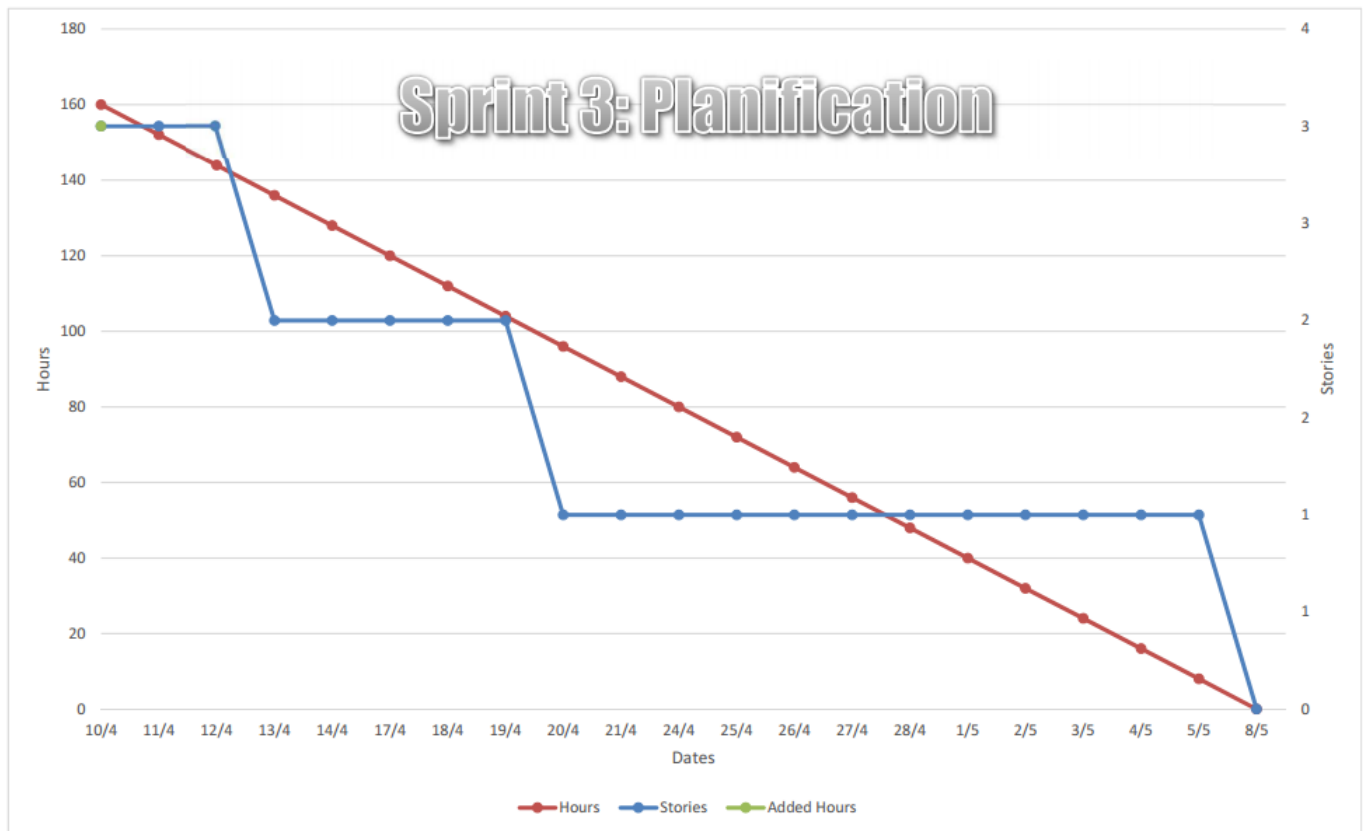


FIGURE 4.39 – Burndown Chart du sprint 3 : Planification

#### 4.4.6 Rétrospective

Cette section est dédiée à l'énumération des problèmes rencontrés dans ce sprint. Ces derniers se résument comme suit :

- Comprendre les ratios et trouver comment les extraire.
- Trouver comment calculer le ratio total à partir des ratios des membres sélectionnés.
- Se familiariser avec la manipulation de dates et ceci sur trois niveaux : AngularJS, Spring et base de données Oracle.

### 4.5 Déploiement de l'application

Dans cette section, nous allons présenter le déploiement de notre application. Cette dernière est composée de deux grands modules qui sont l'application back office qui est accessible via un service web et qui interagit avec nos deux bases de données via le protocole HTTP, et l'application front office qui représente l'interface de communication avec le client. Ces deux modules communiquent entre eux grâce au protocole HTTP. Pour pouvoir déployer notre application nous avons généré en premier lieu un fichier .war pour chaque partie. Puis, nous les avons hébergé sur nos deux serveur Tomcat. La figure 4.40 représente le diagramme de déploiement pour notre application.

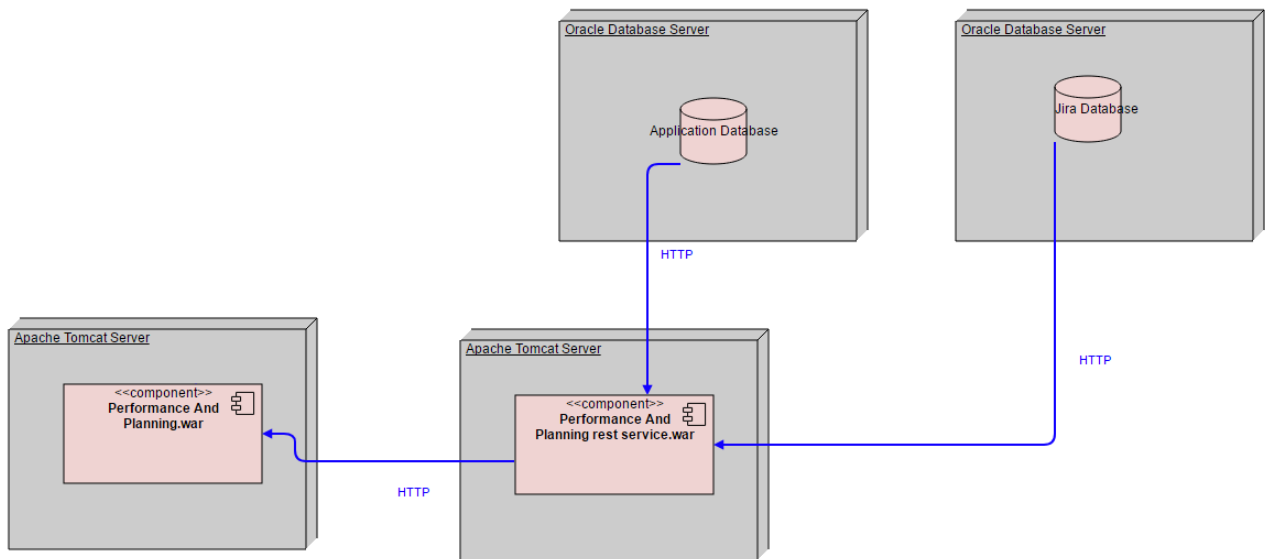


FIGURE 4.40 – Diagramme de déploiement de notre application

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté les sprints de notre projet. En premier lieu, nous avons spécifié le planning pour chaque sprint. Après, nous avons présenté la modélisation statique et dynamique. Ensuite, nous avons détaillé le développement du sprint. Et enfin, nous avons présenté le test, la revue du sprint et la rétrospective.



# Conclusion générale

Nous avons été accueillis, pour la réalisation de notre projet de fin d'études, par la société Linadata. Cette dernière est engagée dans plusieurs grands projets dans le domaine de la bourse et des finances. Afin d'aider les chefs d'équipes chargés de piloter ce type de projets nous avons été chargé de travailler sur une application web qui permet de faciliter la tâche de planification des tâches d'un projet.

Le projet consiste à développer une application web qui fournit des tableaux de bords contenant des informations sur les membres de l'équipe. En plus, elle doit aider le chef d'équipe dans l'étape de planification en calculant le temps nécessaire à un ou plusieurs membres pour finir une tâche.

Nous avons eu du plaisir à développer cette application et ceci en apprenant à utiliser de nouvelles technologies et en étant constamment en communication avec ceux qui ont proposé le sujet. Et nous avons du feedback quotidien sur le produit donc nous avons pu l'améliorer au fur et à mesure du déroulement du projet.

Nous avons choisi la méthode Scrum pour piloter le projet pour sa flexibilité puisque les besoins sont constamment ajoutés et changés au cours du projet.

Afin d'améliorer le projet plus encore, nous envisageons d'ajouter d'autres fonctionnalités telles que :

- Séparer le tableau de bord en deux plus spécifiques encore.
- Créer un module d'intelligence artificielle qui permet de distinguer entre un retour bas niveau ou non.
- Ajouter d'autres indicateurs dans la planification pour rendre le calcul plus précis.

Ce projet nous a permis d'acquérir de l'expérience dans le domaine informatique ainsi que dans le domaine de gestion de projet. En plus de nous avoir permis d'améliorer nos compétences de prise de parole.

# Annexe A

## Flux d'exécution

### A.1 Introduction

Dans cette annexe, nous allons parler du flux d'exécution d'une requête du client dans le serveur en partant du client jusqu'à la base de données et en revenant pour envoyer une réponse.

### A.2 Flux d'exécution d'une requête client

Notre application est divisé en deux parties, la première étant la partie front office implémentée avec AngularJS, et la seconde est la partie back office qui est implémentée avec le framework Spring. Si le client demande l'accès à une certaine information, l'information sera livré comme suit :

1. AngularJS envoie une requête HTTP au service Rest.
2. Le service Rest fait appel à une fonction "getResult()" par exemple par l'intermédiaire d'une interface IMetier.
3. La classe Metier, qui implémente les méthodes de cette dernière, fait appel à une fonction qu'on va appeler "getData()" par l'intermédiaire d'une interface IDAO.
4. La classe DAO, qui implémente les méthodes de cette dernière, envoie une requête SQL Select spécifique à la donnée demandé à la base de données Oracle qui contient les données de l'application Jira.
5. La base de données Jira renvoie une réponse contenant la donnée demandé à la classe DAO.
6. La classe DAO instancie un objet spécifique à la donnée demandé qu'on va appelé "Data-Model" puis elle le renvoi à la classe Metier.
7. La classe Metier fait le mapping entre l'objet DataModel et un objet "DataEntity" puis elle le renvoi à la classe qui implémente notre service Rest.
8. Le service Rest renvoie alors l'objet DataEntity comme réponse à la requête HTTP provenant du client AngularJS.
9. AngularJS affiche alors le résultat de la requête.

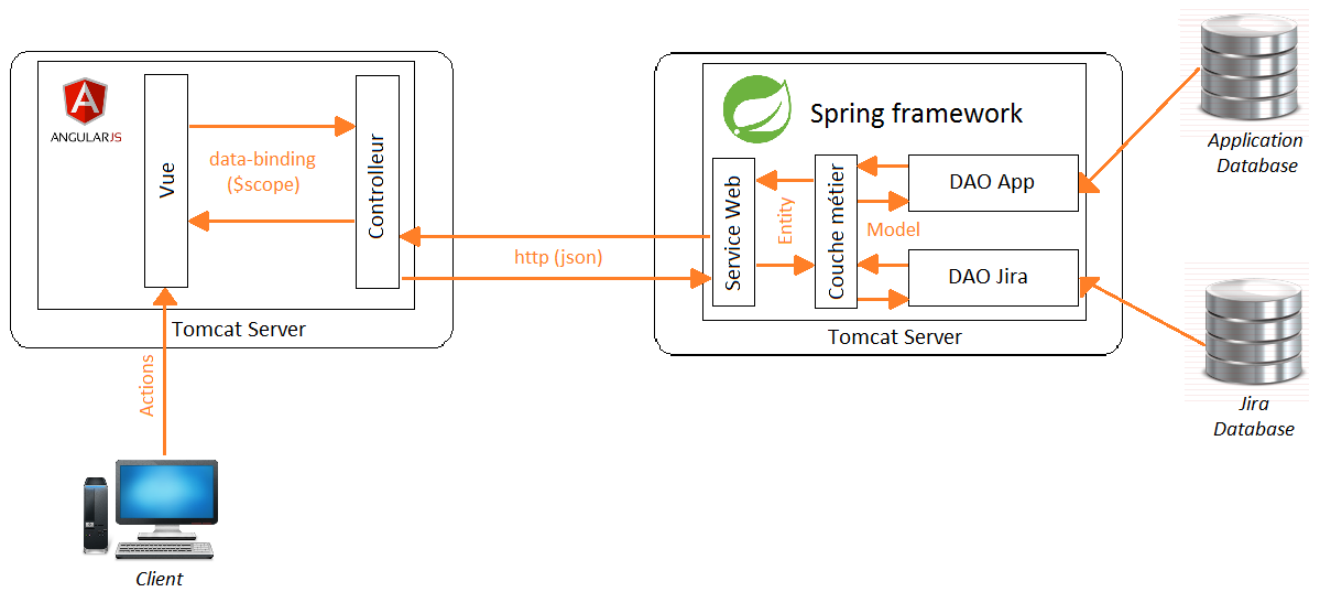


FIGURE A.1 – Architecture de l'application

### A.3 Conclusion

Dans cette annexe, nous avons parlé du flux d'exécution d'une requête du client dans le serveur en partant du client jusqu'à la base de données et en revenant pour envoyer une réponse.

# Bibliographie

- [1] Linedata. Linedata history. <http://fr.linedata.com/profil/notre-histoire/>. [En ligne; date d'accès 26-02-2017].
- [2] Linedata. Linedata solutions. <http://fr.linedata.com/solutions/>. [En ligne; date d'accès 26-02-2017].
- [3] SQL. [https://fr.wikipedia.org/wiki/Structured\\_Query\\_Language](https://fr.wikipedia.org/wiki/Structured_Query_Language). [En ligne; date d'accès 26-02-2017].
- [4] UML, c'est quoi? <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/uml-c-est-quoi>. [En ligne; date d'accès 26-02-2017].
- [5] Les différents types de diagrammes. <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/les-differents-types-de-diagrammes>. [En ligne; date d'accès 27-02-2017].
- [6] Méthode classique vs agile. <http://www.access-dev.com/access-dev/la-gestion-de-projet-methodes-classiques-vs-methodes-agiles/>. [En ligne; date d'accès 30-03-2017].
- [7] Jira. <https://fr.wikipedia.org/wiki/Jira>. [En ligne; date d'accès 07-04-2017].
- [8] UML2 - de l'apprentissage à la pratique. <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>. [En ligne; date d'accès 20-03-2017].
- [9] Indicateur clé de performance. [https://fr.wikipedia.org/wiki/Indicateur\\_cl%C3%A9\\_de\\_performance](https://fr.wikipedia.org/wiki/Indicateur_cl%C3%A9_de_performance). [En ligne; date d'accès 20-03-2017].
- [10] Backlog produit. [salihayacoub.com/420-ka6/2013Theorie/Backlog.pdf](http://salihayacoub.com/420-ka6/2013Theorie/Backlog.pdf). [En ligne; date d'accès 20-03-2017].
- [11] Story points. <http://blog.neoxia.com/story-points-5-minutes-pour-comprendre/>. [En ligne; date d'accès 20-03-2017].
- [12] Suite de Fibonacci. [https://fr.wikipedia.org/wiki/Suite\\_de\\_Fibonacci](https://fr.wikipedia.org/wiki/Suite_de_Fibonacci). [En ligne; date d'accès 20-03-2017].
- [13] Planning sprint. <http://henrik-kniberg.developpez.com/livre/scrum-xp/?page=planning-sprint>. [En ligne; date d'accès 09-04-2017].
- [14] Estimation et planification Agile. <https://fr.slideshare.net/yquenechdu/agile-session-2>. [En ligne; date d'accès 09-04-2017].
- [15] Tomcat. [https://fr.wikipedia.org/wiki/Apache\\_Tomcat](https://fr.wikipedia.org/wiki/Apache_Tomcat). [En ligne; date d'accès 10-04-2017].

- [16] Java. <https://fr.wikipedia.org/wiki/Java>. [En ligne; date d'accès 11-04-2017].
- [17] Pattern Architectural. <http://masig.ensg.eu/PDF/Patrons\%20d\%27architecture.pdf>. [En ligne; date d'accès 19-04-2017].
- [18] IoC. <https://en.wikipedia.org/wiki/Model\%E2\%80\%93view\%E2\%80\%93viewmodel>. [En ligne; date d'accès 19-04-2017].
- [19] MVVM. [https://en.wikipedia.org/wiki/Inversion\\_of\\_control](https://en.wikipedia.org/wiki/Inversion_of_control). [En ligne; date d'accès 19-04-2017].