



**FACULTE DES SCIENCES ECONOMIQUES  
ET DE GESTION DE TUNIS**

**Modélisation et développement d'une application**

**Web :**

**Gestion des ressources humaines**

**Elaboré par :**

**Rezgani Cyrine**

**Licence Appliquée en Informatique de Gestion**

**Encadrent universitaire :**

**Mme Dagdegui Aziza**

**Encadrent professionnel :**

**Mr Jazi Walid**

**2018/2019**

# Table des figures

1	Organigramme de l'ETAP	13
12	Organisation de la méthodologie Sacrum	20
13	Vue globale du processus Sacrum	24
4	Les acteurs de notre système	30
5	Diagramme de cas d'utilisation global	34
6	Architecture 3-tiers	36
7	Model View Controller schema global	37
8	Diagramme de cas d'utilisation du premier sprint	45
9	Raffinement de cas d'utilisation < s'authentifier >	46
10	Raffinement de cas d'utilisation « Accéder aux services »	48
11	Diagramme de séquence système du cas « s'authentifier »	51
12	Diagramme de séquence système du cas « demander congés »	52
13	Diagramme de séquence système du cas « consulter mes	53

	absences »	
14	Diagramme de séquence détaillé du cas d'utilisation « s'authentifier »	54
15	Diagramme de séquence détaille des cas d'utilisation « demander un congé »	54
16	Diagramme de séquence système du cas « consulter mes absences »	55
17	Diagramme de classe participantes du cas < S'authentifier >	56
18	Diagramme de classe de conception du cas < consulter mes absences >	56
19	Diagramme de classe du sprint 1	57
20	Interfaces d'authentification	59
21	Interface d'accès aux services	59
22	Interface de demande de matériel	59
23	Interface << consulter mes absences >>	60
24	Diagramme de cas d'utilisation du sprint 1	64
25	Raffinement du cas d'utilisation « Gérer dossier	64

	personnel »	
26	Raffinement du cas d'utilisation « gérer les absences »	68
27	Diagramme de séquence système du cas « Ajouter personnel »	70
28	Diagramme de séquence système du cas « modifier information personnel »	70
29	Diagramme de séquence système du cas « supprimer personnel »	71
30	Diagramme de séquence système du cas « lister les absences »	71
31	Diagramme de classes participantes du cas « gérer personnel »	72
32	Diagramme de classe de conception du cas « gérer les absences »	73
33	Diagramme de séquence détaillé du cas « gérer personnel »	73
34	Diagramme de classe du premier sprint	74
35	Interface << Ajouter personnel	74
36	Interface << modifier	75

	information personnel >>	
37	Interface « lister les absences »	75
38	Diagramme de cas d'utilisation du sprint 2	77
39	Raffinement de cas d'utilisation « gérer les congés »	77
40	Raffinement du cas d'utilisation « gérer les matériels »	79
41	Diagramme de séquence système du cas < Consulter les demandes des congés >	80
42	Diagramme de séquence système du cas « Accepter / Refuser congé »	8181
43	Diagramme de séquence système du cas < Consulter les demandes des matériels >	81
44	Diagramme de classe de conception du cas < gérer les congés >	82
45	Diagramme de séquences détaillées du cas « gérer les congé »	83
46	Diagramme de classe du sprint 2	84
47	interface « Consulter les demandes de matériel »	84

# Liste des tableaux

1	Comparaison entre les méthodes agiles	12
2	Mots clés dans Scrum	17
3	Backlog du Produit	36
4	Sprint Backlog	40
5	Description textuelle du cas d'utilisation « s'authentifier	42
6	Description textuelle du cas d'utilisation « demander un congé	43
7	Description textuelle du cas d'utilisation « consulter mes absences »	44
8	Description textuelle du cas d'utilisation « demander un matériel »	44
9	Backlog du sprint 1	59
10	Description textuelle du cas d'utilisation « Ajouter un personnel	62
11	Description textuelle du cas d'utilisation « modifier information personnel »	62
12	Description textuelle du cas d'utilisation « supprimer	62

	personnel »	
13	Description textuelle du cas d'utilisation « lister les absences »	63
14	Description textuelle du cas d'utilisation <rechercher l'absence par date	64
15	Backlog du sprint 2	71
16	Description textuelle du cas « consulter les demandes des congés	72
17	Description textuelle du cas < Accepter /refuser congé >	73
18	Description textuelle du cas « Consulter les demandes de matériels	74
19	Description textuelle du cas < Accepter / refuser demande matériel >	74

## Table des matières

1.1-	Cadre du projet	5
1.1.1-	Objectifs du PFE	5
1.2-	Présentation de l'organisme d'accueil	6
1.2.1-	Identité	6
1.2.2	La structure de l'ETAP	7

1.3- Problématique	7
2- Etude de l'existant	8
2.1- Analyse de l'existant	8
2.2- Critique de l'existant	9
2.3- Solution proposée	9
3- Choix méthodologique et conceptuel	10
3.1- Présentation des méthodes classiques	10
3.2- Les méthodes agiles	11
3.3- SCRUM	13
3.3.1- Principe de la méthode Sacrum	13
3.3.2- Le "package" Sacrum	14
3.3.3- Principe de Sacrum	16
4- Choix du langage de modélisation	19
4.1- Définition UML	19
4.2- Principe UML	19
Conclusion	19
Chapitre 2. Etude préliminaire : Sprint zéro	22
Introduction	22
2.1-Sprint 0	22
2.1.1- Définition	22
2.1.2- Objectifs	22



2.2- Spécification des besoins	23
2.2.1- Identification des acteurs	24
2.2.2- Les besoins fonctionnels	25
2.2.3- Les besoins non fonctionnels	27
2.2.4- Diagramme du cas d'utilisation général	27
2.2.4.1- Planning de traitement de cas d'utilisation	29
2.2.5- Architecture et modèle adoptés	29
2.2.5.1- Architecture 3-tiers	29
2.2.5.2- Modèle MVC	30
3-Pilotage du projet avec SCRUM	33
3.1- Les outils SCRUM utilisés	33
3.1.1- Equipe et rôles	33
3.1.2- Le back log du produit	33
4- Planification de releases	36
Conclusion	36
Chapitre 3 : Release 1	38
Introduction	38
3.1- Sprint 1	38
3.1.1- Backlog du sprint (Sprint Backlog)	38
3.2- La spécification fonctionnelle	39
3.2.1-Diagramme de cas d'utilisation du sprint	39

### 3.2.2- Raffinement des cas d'utilisation 40

#### 3.2.2.1- Raffinement du cas d'utilisation « s'authentifier » 40

#### 3.2.2.2- Raffinement du cas d'utilisation « Accéder aux services » 41

### 3.3-Analyse 43

#### 3.3.1- Diagrammes de séquences système 45

##### 3.3.1.1- Diagramme de séquence système du cas « s'authentifier » 45

##### 3.3.1.2-Diagramme de séquence système du cas « demander un congé » 46

##### 3.3.1.3- Diagramme de séquence système du cas « consulter mes absences » 47

#### 3.3.2- Diagrammes de séquences détaillés 47

##### 3.3.2.1- Diagramme de séquences détaillé du cas « s'authentifier » p48

##### 3.3.2.2- Diagramme de séquences détaillé du cas « demander un congé » p48

##### 3.3.2.3-Diagramme de séquences détaillées du cas « consulter mes absences » p49

#### 3.3.3-Diagramme de classes participantes p49

##### 3.3.3.1-Diagramme de classe participantes de conception du cas < S'authentifier > p50

##### 3.3.3.2- Diagramme de classe participantes de conception du cas < consulter mes absences > p50

#### 3.3.4-Diagramme de classe du sprint 1 p51

### 3.4- Réalisation p51

#### 3.4.1-Interfaces authentication utilisateur p52

#### 3.4.2-Interfaces d'accès aux services p53

#### 3.4.3- Interface de demande un matériel p53

3.4.4- Interface de consultation des absences p54

Conclusion p55

Chapitre 4. Release 2 p57

Introduction p57

4.1-Sprint 1 p57

4.1.1- Backlog du sprint 1 p57

4.2- Spécifications fonctionnelles p58

4.2.1- Diagramme de cas d'utilisation du sprint 1 p58

4.2.2- Raffinement des cas d'utilisation du sprint 1 p58

4.2.2.1- Raffinement du cas d'utilisation « gérer personnel » p58

4.2.2.2-Raffinement du cas d'utilisation « gérer les absences » p62

4.3- Analyse p63

4.3.1- Diagrammes de séquence système p63

4.3.2-Diagramme de séquence système du cas « Ajouter personnel » p64

4.3.3- Diagramme de séquence système du cas « modifier information personnel » p64

4.3.4- Diagramme de séquence système du cas « supprimer personnel » p65

4.3.5-Diagramme de séquence système du cas « lister les absences » p65

4.3.6-Diagrammes de classes participantes p66

4.3.6.1- Diagramme de classe de conception du cas « gérer personnel » p66

4.3.7- Diagramme de séquence détaillée p67

#### 4.3.8- Diagramme de classe du sprint 1 p68

#### 4.4- Réalisation p68

##### 4.4.1- Interface Ajouter personnel p68

##### 4.4.2- Interface modifier information personnel p69

##### 4.4.3- Interface « lister les absences » p69

#### 4.5-Test p70

#### 4.6- Sprint 2 71

##### 4.6.1- Backlog du sprint 2 p71

##### 4.6.2- Spécifications fonctionnelles p71

###### 4.6.2.1- Diagramme de cas d'utilisation du sprint 1 p71

###### 4.6.2.2-Raffinement des cas d'utilisation du sprint 2 p71

###### 4.6.2.3- Raffinement du cas d'utilisation « gérer les congés » p71

###### 4.6.2.4- Raffinement du cas d'utilisation « gérer les matériels » p74

#### 4.7- Analyse p77

##### 4.7.1-Diagrammes de séquence système p77

###### 4.7.1.1- Diagramme de séquence système du cas « Consulter les demandes des congé » p77

###### 4.7.1.2- Diagramme de séquence système du cas « Accepter /Refuser congé » p78

###### 4.7.1.3- Diagramme de séquence système du cas < Consulter les demandes des matériels > p79

#### 4.8-Diagrammes de classes participantes p79

##### 4.8.1- Diagramme de classe participantes du cas < gérer les congés > p79

## 4.9- Diagrammes de séquences détaillés p80

### 4.9.1- Diagramme de séquences détaillées du cas « gérer les congé » p80

## 4.10- Diagramme de classe du sprint 2 p81

### 4.11-Réalisation p81

### 4.12- Test p82

### Conclusion p82

## Chapitre 5. La phase de clôture

### Introduction p83

## 5.1-Environnement de développement p83

### 5.1.1- Environnement matériel p83

### 5.1.2- Environnement logiciel p83

### 5.1.3- Langages utilisés p86

## **Remerciements**

Avant tout développement de cette expérience, nous commençons notre rapport par des remerciements, à ceux qui nous ont beaucoup appris au cours de ce projet.

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier sincèrement Mme Dagdegui Aziza en tant que notre encadrant universitaire qui se montrée à l'écoute et très disponibles tout au long de cette expérience avec beaucoup de patience et de pédagogie ainsi que pour son inspiration.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

# Introduction générale

L'apparition des systèmes d'information a permis d'améliorer le traitement de l'information et l'interaction au sein des entreprises. Cette évolution n'est pas un phénomène éphémère.

Si nous, en tant que simple personnes, avons besoin d'avoir un minimum de coordination pour nos tâches journalières, alors que pouvons-nous dire des entreprises, qui gèrent quotidiennement des centaines de données, de processus et de traitements c'est pour cela que l'importance de l'informatique prend de l'ampleur et se présente comme étant un outil indispensable pour le bon fonctionnement des entreprises et touche plus en plus de domaine.

Du coup, les entreprises sont dans l'obligation d'être à la page avec l'amélioration de certaines de leurs activités afin d'optimiser l'organisation interne des personnels, sachant que la véritable richesse d'une entreprise se repose principalement sur les compétences et les contributions et atteindre des objectifs fixes.

Afin de maintenir cette richesse les entreprises doivent bien organiser et évoluer leurs système de la gestion des ressources humaines.

C'est dans ce cadre que nous avons effectué notre stage au sein de l'Entreprises Tunisienne d'Activités Pétrolières (ETAP) du 15 février au 15 mai dans le service des ressources humaines. Nous avons constaté que cette dernière poursuit des efforts considérables pour l'amélioration et l'intégration de nouvelles technologies au sein de son système d'information. C'est dans ce contexte que nous allons mettre en place une application web pour la gestion des ressources humaines.

Ce rapport présente l'ensemble des étapes suivies pour développer l'application. Il contient cinq chapitres organisés comme suit :

Le premier chapitre intitulé « *Cadre général du projet* » est consacré à la présentation du contexte du projet. Ensuite, nous définirons notre choix de la méthodologie de développement adoptée et nous finirons par présenter les outils utilisés ainsi que les différents environnements de développement. Le deuxième chapitre, « *Sprint 0 : Analyse Etude préliminaire* », s'articule autour de l'identification des acteurs et description des besoins fonctionnels et non fonctionnels. Puis, nous élaborons le Backlog du produit et nous terminons par la présentation des architectures physique et logique. Les chapitres 3, 4 et 5 décrivent respectivement la réalisation des Sprints 1, 2 et 3. Nous

commençons par la spécification du sprint, avant de présenter la phase conceptuelle qui sera matérialisée par l'élaboration du diagramme de classes ainsi que celle des diagrammes de séquence, puis nous présentons la phase d'implémentation par la description des interfaces de l'application et nous finissons par la réalisation des tests.



# Chapitre 1. Cadre général du projet

## Introduction

Ce premier chapitre sera dédié à la définition du cadre général de notre projet. La première partie sera réservée pour la présentation de l'organisme d'accueil où nous avons passé notre stage. La seconde partie s'occupera de l'étude de l'existant qui comporte l'évaluation des différents problèmes ainsi que la solution proposée. La dernière partie sera consacrée à la description du choix méthodologique et conceptuel suivi tout au long de notre projet.

## I. Cadre du projet

Notre stage s'est déroulé au sein de l'entreprise tunisienne d'activités pétrolières (ETAP) en vue de l'obtention d'une licence fondamentale à la faculté de science économie gestion de Tunis pour l'année universitaire 2018/2019.

### I.1. Objectifs du PFE

Ce projet de fin d'études de la 3<sup>ème</sup> année informatique appliquée à la gestion a été effectué sous l'encadrement de Mme AZIZA DEGDEGUI, enseignante à la FSEG et Mr. WALID JAZI, à l'ETAP

Il représente une belle occasion pour mettre en pratique toutes les connaissances et capacités analytiques, techniques et relationnelles acquises durant notre parcours académique dans un cadre professionnel

L'objectif de ce projet est d'instaurer et de développer une application web pour la gestion des ressources humaines.

## **I.2. Présentation de l'organisme d'accueil**

### **I.2.1. Identité**

L'Entreprise Tunisienne d'Activités Pétrolières (ETAP) est une entreprise publique à caractère non administratif qui a été créée par la loi 72-22 du 10 mars 1972 dans le but de permettre à l'état d'accroître son contrôle et sa participation active et directe dans des différentes activités de l'industrie pétrolière.

Son mandat initial était centré sur :

- la conduite de toutes les études à caractère pétrolier
- la formation et le perfectionnement des cadres tunisiens dans les différentes branches de l'industrie pétrolière
- l'intervention éventuelle dans toutes les opérations industrielles commerciales, financières, mobilières et immobilières se rattachant directement aux hydrocarbures.

Sa mission actuelle est axée sur :

- la promotion de l'exploration des hydrocarbures en Tunisie.
- la gestion des patrimoines nationaux pétroliers.
- la réalisation des études de développement des découvertes
- la gestion des exploitations de la concession pétrolière et gazière, dont l'ETAP est partenaire, et la promotion du développement des gisements marginaux.
- l'approvisionnement du pays en pétrole brut et gaz
- la réalisation des études technico économiques de projets ayant trait aux activités pétrolières et para pétrolières.
- la réalisation des prestations dans les domaines du traitement sismique, des études de simulation des gisements et des laboratoires exploration - production.

## I.2.2. La structure de l'ETAP

L'ETAP est dirigée par un président directeur générale (PDG) qui est secondé par un directeur général adjoint. Il y a aussi cinq directions centrales qui sont les suivantes :

- Direction centrale production
- Direction centrale exploration
- Direction centrale des stratégies et développement
- Direction centrale des ressources
- Direction centrale de contrôle et des finances

Ces directions sont divisées en départements qui sont scindées en services. La figure suivante présente l'organigramme de l'ETAP.

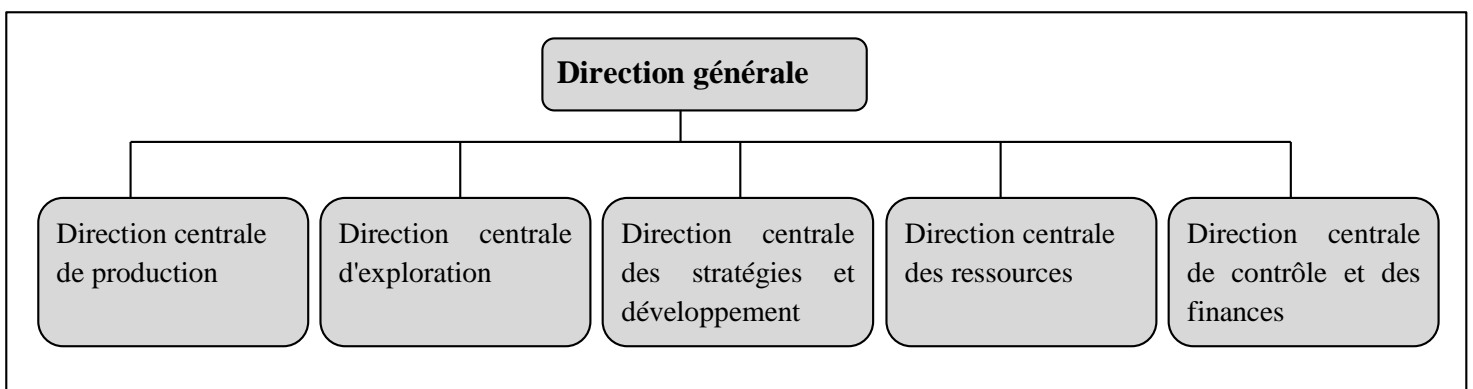


Figure 1: Organigramme de l'ETAP

## I.3. Problématique

L'objectif de notre projet est d'instaurer et de développer une application web pour la gestion des ressources humaine au sein de l'ETAP.

La technologie joue un rôle important dans la gestion des ressources humaines. Divers logiciels sont utilisés par les entreprises pour combiner un certain nombre de fonctions RH, telles que le stockage des données des employés, la gestion de la paie, les processus de recrutement, Il garantit que les processus de ressources humaines au quotidien sont et faciles d'accès et à gérer. [13]

Le système automatisé de gestion des ressources humaines aide également à gagner du temps, à augmenter le taux de productivité de l'équipe des ressources humaines et à les rendre sans problème. Ces logiciels fournissent des méthodes simples pour gérer tous les aspects dans une entreprise.

Grâce à la technologie avancée, les employés seront en mesure de gérer eux-mêmes leurs activités liées aux congés, ce qui réduira considérablement le fardeau. Ce logiciel innovant a permis aux professionnels des ressources humaines de consacrer moins de temps aux détails tactiques de leurs tâches et de se concentrer sur des éléments plus stratégiques. Les logiciels comprennent également des outils de mesure qui vous permettent d'évaluer avec précision des problèmes tels que le taux de rotation et les coûts d'embauche.

## **II. Etude de l'existant**

Dans cette section, nous commençons par l'analyse puis la critique de l'existant afin de proposer une solution permettant de remédier aux problèmes rencontrés.

### **II.1. Analyse de l'existant**

L'entreprise est composée de plusieurs départements et chaque département est divisé en services, chaque département a un directeur et peut avoir un sous-directeur, chaque service a un chef de service et comporte un ensemble des personnels.

L'échange d'information entre les différents services et départements se fait principalement à l'aide de support papier, ce qui résulte en plusieurs va et vient provoquant une perte de temps. La plupart des demandes sont sous la forme d'un formulaire en papier et doivent passer par un supérieur hiérarchique pour être approuvées, puis elles seront transmises au département concerné. Une telle opération peut prendre entre plusieurs heures et quelques jours d'où il s'agit d'une lenteur parfois inacceptable et des éventuelles interruptions du travail de l'employé

## II.2. Critique de l'existant

Le service de la gestion des ressources humaines est géré avec le logiciel oracle forms 6i. L'utilisation de ce logiciel engendre quelques problèmes qui rendent la réalisation des tâches demandées pénible. Parmi ces problèmes, nous citons :

- un gaspillage de temps : problème de modification et d'ajout et de suppression des fonctionnaires
- une anarchie dans le travail : problème d'échelon et promotion / l'architecture client-serveur très ancienne.
- une mauvaise coordination entre les services : perturbation dans les données des promotions et des échelons qui seront utilisées dans un autre service

## II.3. Solution proposée

Afin de remédier aux problèmes cités précédemment; nous proposons une solution qui se base sur l'automatisation et l'organisation de toutes les procédures du travail. Il s'agit donc de mettre en place un système d'information cohérent qui garantit :

- le bon encadrement du travail à réaliser
- la disponibilité et la cohérence de l'information
- la traçabilité afin de garantir l'évaluation
- la bonne organisation du travail
- l'optimisation du temps

Notre solution consiste à développer une application web capable de traiter les différentes tâches réalisées par le service des ressources humaines de l'ETAP.

### III. Choix méthodologique et conceptuel

Au cours du stage, nous avons eu le libre choix de la méthodologie à appliquer pour la gestion de notre projet. Pour cela nous avons effectué une étude comparative sur les méthodes agiles et les méthodes classiques afin de choisir celle que nous jugeons meilleure pour l'étude, l'analyse et la conception de notre application.

#### III.1. Présentation des méthodes classiques

La méthode classique est une succession de phases qui se déverse les unes dans les autres et on ne peut pas commencer une phase de l'approche classique tant que la précédente n'est pas terminée. Alors le plan du projet se fait par étape. [12]

- **Conception** : En général on commence par la phase de conception au cours de laquelle on va concevoir le produit qu'on souhaiterait réaliser, les spécifications détaillées mais avant on va réaliser le plan du projet la façon dont ça va se dérouler voilà pourquoi on l'appelle prédictif car on prédit comment le produit doit fonctionner, comment l'utilisateur doit l'utiliser.
- **Réalisation** : C'est le fait de construire le produit (logiciel), on peut aussi l'appeler phase de développement sur les projets de développement logiciel.
- **Recette** : Phase pendant laquelle on va vérifier la conformité du produit par rapport à ses spécifications qu'on a définies pendant la phase de conception. C'est seulement à l'issue de cette phase on va décider si on met en production ou à disposition des utilisateurs le produit en question (logiciel).

Lors de la phase de réalisation on parle souvent d'effet tunnel puisque le commanditaire du projet en général ne voit pas grande chose de concret sur le projet en train de se construire tant qu'on n'arrive pas à la phase de recette. C'est ainsi que cette approche a rapidement montré ses limites avec la complexité grandissante des projets liés à trois principaux facteurs qui sont : le travail en équipe, lié à la technologie vu sa grande évolution et anticiper les besoins de l'utilisateur.

[1]

## III.2. Les méthodes agiles

L'agilité se définit comme un grand principe de considérer que les besoins du cahier des charges peuvent changer et il faut s'y faire. Le propriétaire est responsable des changements, les développeurs travaillent pour réaliser les évolutions du besoin, dont le but est de faire un logiciel qui va correspondre aux attentes des utilisateurs. [1]

Le tableau suivant illustre les différences entre quelques méthodes de développement agiles les plus connus à savoir :

- **XP** : extrême Programming
- **RUP** : Rational Unified Process
- **Scrum**
- **FDD** : Feature Driven Development

Méthode	Points clés	Inconvénient
<b>XP</b>	Développement guidé par les besoins du client Equipes réduites, centrées sur les développeurs par binômes adaptabilité aux modifications	Risque d'anquer de contrôle et de structuration en laissant les développeurs trop libres de dériver par rapport à la fonction de l'application
<b>RUP</b>	Processus complet assisté par des outils Rôle bien définis	Lourd ; largement étendu, il peut être difficile à mettre en œuvre de façon spécifique Convient pour les gros projets qui génèrent beaucoup de documentation
<b>Scrum</b>	Petites équipes itérations de 30 jours réunions journalières	La mise en œuvre du développement n'est pas précisée seule compte la gestion des ressources humaines
<b>FDD</b>	Procédé bien défini et simple orienté objet et basé sur le développement itérations très courtes	Uniquement centré sur développement

Table 1. Comparaison entre les méthodes agiles

Les méthodes agiles sont aussi basées sur l'implication du client en lui permettant de donner son feedback sur le développement de l'application en cours.

Nous citons les principales caractéristiques de ces méthodes [2] :

- L'auto organisation des équipes de développement qui bénéficient d'une plus grande liberté de création et une plus grande part de responsabilité dans le projet.
- Grande visibilité sur l'avancement du projet : le client voit concrètement l'évolution du projet après chaque itération ce qui brise l'effet tunnel des méthodes classique.
- L'adaptabilité au changement : la planification initiale et la structure du logiciel sont flexibles pour que l'utilisateur puisse modifier ses exigences en cours du projet.
- Au-delà de l'effet sur la productivité et la qualité, les méthodes agiles ajoutent une dimension supplémentaire au succès d'un projet : la maximisation de la valeur produite par le logiciel. Cet argument, majeur d'un point de vue économique, s'explique par la mise en œuvre de quelques principes, comme la livraison fréquente l'implication des clients, la gestion des priorités basées sur des estimations de valeur et de coût

Le choix d'un processus et d'une méthodologie de travail est une étape essentielle afin de faciliter et d'accélérer la transformation des besoins des utilisateurs en un système logiciel.

De ce fait, nous avons opté pour **la méthode SCRUM des méthodologies agiles** qui est essentiellement dédiée à la gestion des projets informatiques en réduisant le cycle de développement du logiciel par la réalisation d'une version minimale, puis en intégrant les fonctionnalités par processus itératif.



Nous considérons que le modèle de développement agile pour les logiciels est révolutionnaire, car il permet une livraison plus rapide des projets et une possibilité d'apporter des modifications à des projets à tout moment. En effet les modes de gestion classique des projets informatiques limitaient l'intervention du client au démarrage du projet et au moment de la livraison finale. Cet effet tunnel conduit à un logiciel inadapté et souvent qui ne répond pas aux fonctionnalités requises et demandées par le client. Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles impliquent au maximum le client et permettent une grande réactivité à ses demandes. Elles visent la satisfaction réelle du client en priorité aux termes d'un contrat de développement.

### **III.3. SCRUM**

SCRUM est une méthode agile pour la gestion de projets informatiques. C'est une méthode itérative basée sur des itérations de courte durée appelées Sprints. Cette méthode a pour objectif d'améliorer au mieux la productivité tout en assurant la satisfaction du client. [4]

#### **III.3.1. Principe de la méthode Scrum**

La méthode SCRUM définit un cadre de travail permettant la réalisation de projets complexes. Initialement prévue pour le développement de projet type software, cette méthode peut être appliquée à tout type de projet, du plus simple au plus innovant, et ce de manière très simple. Scrum est l'une des méthodologies agiles qui permet de gérer des projets informatiques. Ces derniers seront subdivisés en modules à réaliser, qui seront programmés chacun sur une itération appelée *sprint*, et cela aboutira à un livrable valide [2].

Durant le projet, des réunions quotidiennes favorisant la communication et la résolution des problèmes sont menées à la fin de chaque sprint, le client et l'équipe réévaluent la spécification du logiciel afin de juger l'avancement du produit du sprint, ce dernier peut être livré.

Le succès de cette méthode repose sur le respect des rôles de chacun et permet de s'adapter facilement au changement d'un client.

La figure suivante explique le processus de déroulement de la méthode SCRUM :

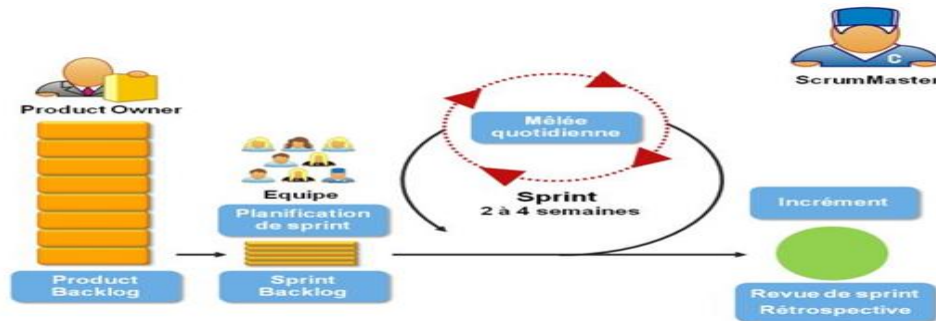


Figure 2 : organisation de la méthodologie Scrum

La durée d'un sprint varie de deux semaines à un mois, et à la fin de chaque sprint, l'équipe doit présenter un produit partiellement utilisable. Une auto évaluation doit être faite par l'équipe, pour déceler les obstacles rencontrés et comparer si les résultats obtenus, correspondant aux estimations escomptées, afin d'améliorer le rendement lors des futures itérations.

### III.3.2. Le "package" Scrum

Scrum est considéré comme un cadre ou « framework » de gestion de projet. Ce cadre est constitué d'une définition des rôles, de réunions et d'artefacts. [2]

Scrum définit 3 rôles :

- **Le Product owner :** C'est le maître d'ouvrage, également le propriétaire du produit il est responsable de maximiser le travail de l'équipe. Il est le seul gérant du carnet du produit il définit les exigences que doit satisfaire le produit, ajuste et règle les fonctionnalités du produit au cours de chaque itération. C'est lui également qui donne l'ordre d'approbation ou les refus de travail présenté.

- **L'équipe de développement :** Celle-ci est responsable de transformer les besoins exprimés dans le Product Backlog par le Product owner en fonctionnalités

concrètes et utilisables. Elle doit livrer un incrément, une version du logiciel, à la fin de chaque sprint. L'équipe de développement est auto – organisée. Nulle personne ne peut lui indiquer comment transformer les histoires du Product backlog. On dit qu'elle est autonome. Elle est pluridisciplinaire car elle peut contenir d'autres rôles tels que développeur, concepteur, designer.

▪ **Le Scrum master :** Le scrum master doit maîtriser SCRUM dans son intégralité. Il doit s'assurer que la méthode scrum doit être mise en œuvre correctement, que la démarche de l'équipe de développement s'adapte selon les pratiques de la méthode, généralement il est confondu avec le chef du projet. Il a comme mission de :

- ✓ diriger et aider les membres de l'équipe
- ✓ gérer efficacement le carnet du produit, afin qu'il soit concis et clair pour l'équipe
- ✓ aider l'équipe à éliminer les obstacles à son progrès
- ✓ s'assurer que l'équipe est productive

### III.3.3. Principe de Scrum

Afin de comprendre cette méthode agile, il est nécessaire de définir un ensemble de mots clés détaillés dans le tableau suivant :

Mots Clé	Description
<i>Product Backlog</i>	Définition des besoins fonctionnels sous forme de « <i>User Story</i> ».
<i>User Story</i>	Description littérale (non technique) d'une fonctionnalité donnée.
<i>Sprint Backlog</i>	Liste des « <i>User Story</i> » à implémenter durant un « <i>Sprint</i> ».
<i>Sprint</i>	Itération à courte durée (moyenne de 2 à 4 Semaines).  Chaque itération définit un objectif à atteindre, en lui associant une liste d'éléments du « <i>Product Backlog</i> » à réaliser.
<i>Task</i>	Un « <i>User Story</i> » se décompose en un ensemble de tâches élémentaires.  La durée maximale d'une tâche ne dépasse pas (obligatoirement) 2 Jours. (Dans les cas contraires la tâche est décomposée en autres tâches).
<i>Velocity</i>	Vitesse du progrès : Une vitesse réelle (Temps réel d'atteinte), et la Vitesse estimée (Temps au bout duquel l'atteinte est estimée).
<i>Story Point</i>	Estimation du travail nécessaire pour accomplir un « <i>User Story</i> ».
<i>Burndown Chart</i>	Graphique de ré-estimation du « Reste A faire ».

Tab2: Mots clés dans Scrum



La figure ci-après explique le principe de la méthode Scrum : [4]

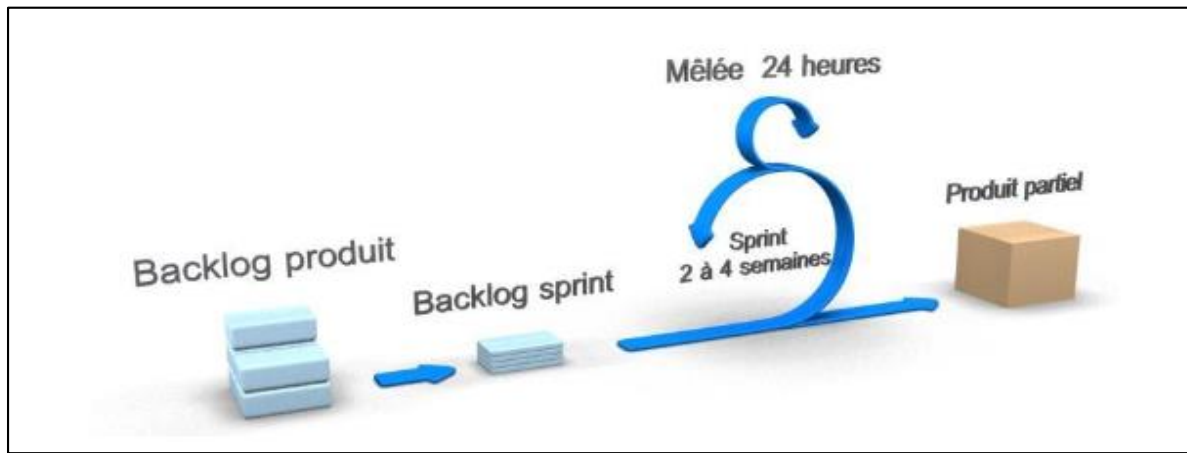


Figure 3 : Vue globale du processus Scrum

La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps (timeboxing):

- **Planification du Sprint** (Sprint = itération) : au cours de cette réunion, l'**équipe de développement** sélectionne les éléments prioritaires du « **Product Backlog** » (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du sprint (en accord avec le « **Product Owner** »).
- **Revue de Sprint** : au cours de cette réunion qui a lieu à la fin du sprint, l'**équipe de développement** présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du **Product Owner** et des utilisateurs finaux. C'est également le moment d'anticiper le périmètre des prochains sprints et d'ajuster au besoin la planification de release (nombre de sprints restants).
- **Rétrospective de Sprint** : la rétrospective qui a généralement lieu après la revue de sprint est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc) à la lueur du "vécu" sur le sprint écoulé (principe d'**amélioration continue**).
- **Mêlée quotidienne** : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout (elle est aussi appelée "stand up meeting") en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ? » [15]

## IV. Choix du langage de modélisation

Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer de l'information ou de la connaissance ou des systèmes dans une structure qui est définie par un ensemble cohérent de règles. Ces règles sont utilisées pour l'interprétation de la signification des composants dans la structure. [4]

Le langage de modélisation que nous avons choisi pour la modélisation de notre application est l'UML (Unified modeling language).

### IV.1. Définition UML

UML, Unified Modeling Language, langage de modélisation objet unifié est une démarche orientée objet. Elle est née de la fusion de trois méthodes orientées objet Booch, OMT Object Modeling Technique et OOSE Object Oriented Software Engineering, conçues respectivement par Grady Booch, James Rumbaugh et Ivar Jacobson. [5]

### IV.2 Principe UML

Les 3 experts d'UML ont focalisé leur attention sur les deux aspects modélisation et formalisation afin de concevoir un langage de modélisation standard et universel utilisé notamment pour le développement informatique en langage objet.

UML 2 est une évolution majeure du langage. UML 2.2, la dernière spécification, est supportée par l'OMG Object Management Group. La modélisation et la formalisation à l'aide d'un vocabulaire standardisé et de surcroît orienté objet confèrent à la méthode tout son intérêt. La formalisation et la modélisation facilitent, en effet, la définition du problème à traiter et la compréhension par l'ensemble des principales parties prenantes, après, il est vrai, un court apprentissage. Une fois le modèle bien défini, il est plus aisé de s'y référer lors du développement afin de s'assurer de la conformité de ce dernier. Un outil précieux qui explique à lui seul l'essor de la démarche UML. [5]

## Conclusion

Dans le présent chapitre, nous avons commencé par présenter le cadre général de notre projet, à savoir l'organisme d'accueil et le travail demandé. Ensuite, Nous avons fait une étude et une critique de l'existant afin de proposer une solution qui permet d'écarter les difficultés dégagées précédemment. Enfin, nous avons annoncé le choix méthodologique et conceptuel que nous allons

suivre pour réaliser notre application. Le prochain chapitre contiendra une planification du projet et une identification des fonctionnalités de notre application.





## Chapitre 2. Etude préliminaire : Sprint zéro

### Introduction

Le principe de base de Scrum est de se focaliser de façon itérative sur un ensemble de fonctionnalités à réaliser dans chaque itération. Dans ce chapitre, nous passerons à la phase la plus importante de cette méthodologie, communément appelée « Sprint Zéro » ou phase du pré-jeu, elle représente le travail le plus délicat et le plus significatif, mais elle même repose sur une bonne spécification des besoins qui n'est autre que la question que doit se poser tout informaticien au début de son travail " Qu'est-ce qu'on veut que nous fassions ? Et comment ? ", Afin d'identifier le Backlog du produit et en détaillant l'architecture adoptée pour la mise en place de le logiciel.

### I. Sprint 0

#### I.1 Définition

Le sprint 0, l'itération 0 ou l'exploration est un élément important pour mettre les bonnes bases d'un nouveau projet et pour permettre à une nouvelle équipe de se former et d'apprendre à travailler ensemble. Il ne se termine pas forcément par une livraison. Ceci est un des éléments qui le différencie par rapport aux autres itérations. [16]

#### I.2 Objectifs

Les objectifs impératifs de cette phase du projet :

- préparer l'environnement de développement ce qui nous permettra dans les prochaines itérations de se concentrer sur les aspects fonctionnels et techniques spécifiques au projet.
- partager une vision claire du projet. L'équipe a besoin de savoir les objectifs du client, le périmètre de l'application, les contraintes, les intervenants dans le projet, les utilisateurs finaux, une première version de l'architecture technique, etc.

- produire une première version du backlog du produit avec des estimations et des priorités associées aux tâches.
- réfléchir et se mettre d'accord sur une architecture globale du projet. On définit les grandes lignes d'architecture, nous décidons les outils et les technologies qui seront utilisés dans toutes les couches de l'architecture.
- roder l'équipe sur une partie du backlog initial. Le but est de faire un premier exercice pour les estimations, la division en tâches, la réalisation, une première démonstration, etc. [16]

Le sprint 0 n'apporte pas de valeur immédiate. Il faut le voir comme un investissement essentiel dans l'avenir du projet, qui va démarrer sur des bases solides, permettant d'éviter beaucoup de problèmes et de récupérer rapidement cet investissement initial.

La plupart du temps, le sprint 0 dure entre 3 et 4 jours.

## **II. Spécification des besoins**

La spécification des besoins est l'activité essentielle au début du processus de développement. Elle est menée avec l'étude de faisabilité et la planification. Son but principal est de garantir le développement d'un modèle du système à implémenter, adéquat suite à l'identification des différents intervenants : clients, utilisateur (acteurs).

La spécification des besoins répond aux questions suivantes:

- Que va faire le système pour les utilisateurs principaux?
- Quels services va-t-il rendre?
- Quelles vont être les propriétés du système liées à sa performance, sa fiabilité et son extensibilité?

## II.1. Identification des acteurs

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. [6]

La figure 4 permet de dresser la hiérarchie des acteurs au sein de notre système.

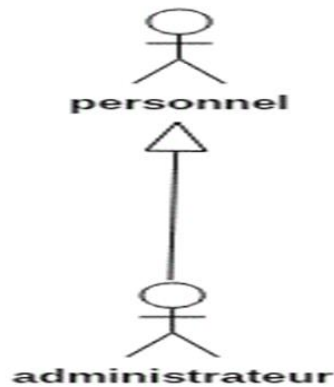


Figure 4 : Les acteurs de notre système

Dans notre application les acteurs invoqués dans le système dont nous chercherons à satisfaire sont les suivants :

**L'administrateur** : c'est un seul et unique acteur, il a comme rôle d'accorder les privilèges à chaque responsable de département et il gère les comptes des personnels, de plus il peut consulter tout l'ensemble de l'application.

**Personnel** : il représente plusieurs catégories d'employés (l'employées de l'ETAP). Il peut se bénéficier de l'application et accéder à ses services.

**Le système** : est dédié pour faire les tâches automatisées afin de satisfaire les besoins des acteurs. Par exemple calcule des salaires, envoi automatique des fiches de paies chaque mois...

## II.2. Les besoins fonctionnels

- Côté personnel :

Le personnel aura la possibilité d'accéder à l'application pour réaliser les tâches suivantes:

- ✓ Consulter ses absences
- ✓ Demande un matériel
- ✓ Demande un congé

- Côté administrateur :

L'administrateur aura la possibilité d'accéder à l'application pour réaliser les tâches suivantes:

- ✓ Gérer les comptes de personnels : consulter, ajouter, modifier ou supprimer des personnels.
- ✓ Gérer les absences du personnel : consulter l'interface des absences des personnels et rechercher par date les absences des personnels
- ✓ Gérer les congés du personnel : consulter la liste des personnes en congé et accepter ou refuser un congé
- ✓ Gérer la demande des matériels du personnel : consulter les demandes des matériels accepter ou refuser une demande de matériel

- Tous les utilisateurs du système

S'authentifier : C'est le processus de validation de l'identité d'un utilisateur. Chaque employé dispos d'un compte dans l'application doit passer par la page de connexion dans laquelle il va saisir ses paramètres d'accès.



## II.3. Les besoins non fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système. Les besoins non fonctionnels de notre avenir système sont :

- **Ergonomie et souplesse** : l'application doit offrir une interface conviviale et facile à manipuler par les différents utilisateurs et le menu doit être bien organisé.
- **La performance** : l'application doit être performante à travers ses fonctionnalités. Elle doit répondre à toutes les exigences des utilisateurs d'une manière optimale.
- **La sécurité** : l'application doit être confidentielle et sécurisée afin de protéger toutes les données. C'est pour cela un processus de validation de l'identité de chaque employé précède tout accès à l'application
- **L'extensibilité** : l'application pourra avoir une possibilité d'ajouter ou de modifier des nouvelles fonctionnalités selon les besoins des utilisateurs.

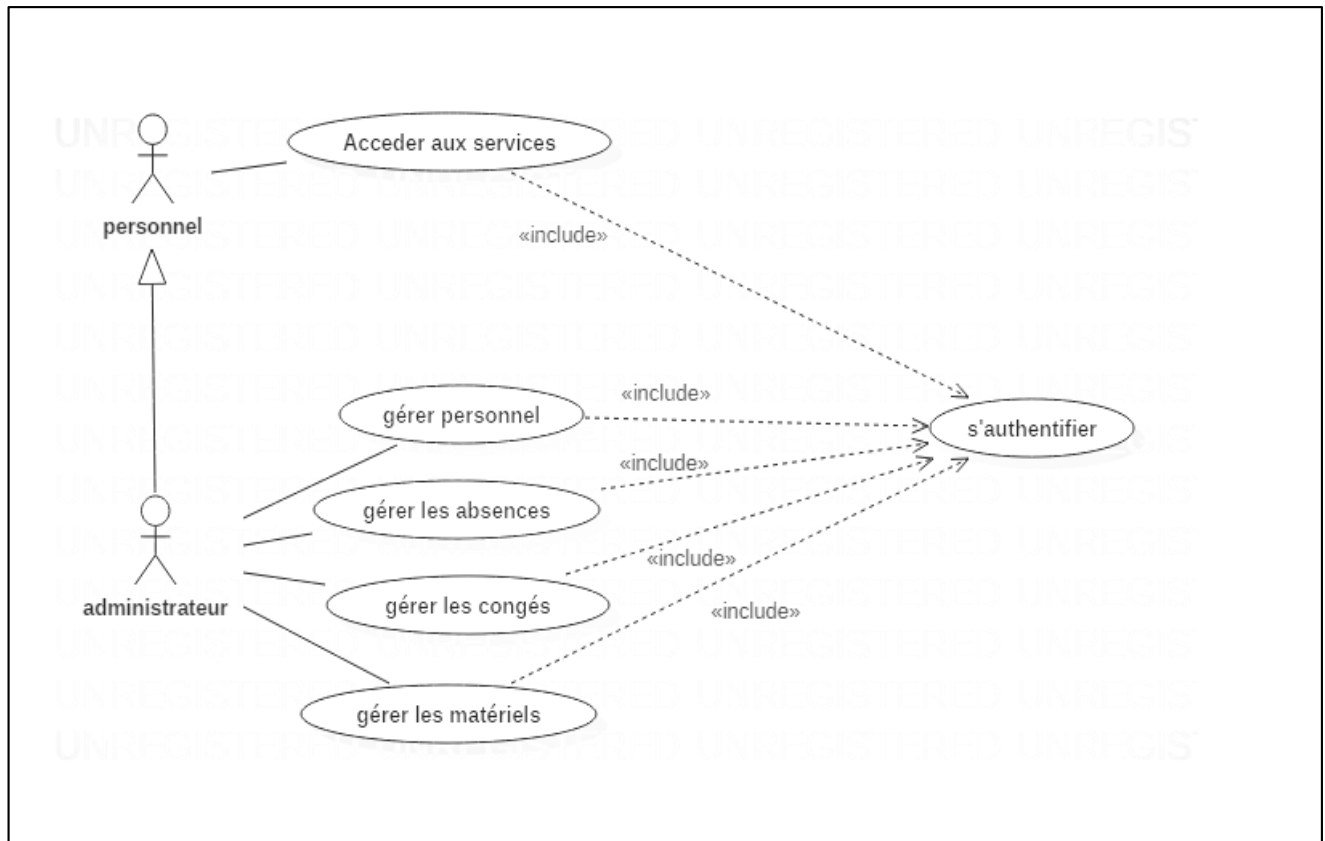
## II.4. Diagramme du cas d'utilisation général

Dans cette partie nous allons identifier et expliquer les différents cas d'utilisation du système.

Les cas d'utilisation décrivent les différentes fonctionnalités du système pour aider l'utilisateur à exécuter sa tâche. Les différentes fonctionnalités de notre système sont décrites dans le diagramme de cas d'utilisation global.

La modélisation de ces cas d'utilisation nous permet d'avoir une vue globale de haut niveau des besoins de notre système.

La figure ci-dessous présente le diagramme de cas d'utilisation global de notre application



**Figure 5: Diagramme de cas d'utilisation global**

Selon le diagramme de cas d'utilisation de notre application; un personnel aura la possibilité d'accéder aux services et ce afin de gérer ses demandes ainsi de consulter ses absences. Quant à l'administrateur; en plus des fonctionnalités offertes au personnel; aura la possibilité de gérer les personnels, gérer les absences des personnels, gérer les demandes de matériels et consulter les demandes de congés. Toutes ces fonctionnalités ne pourront être réalisées qu'après la phase d'authentification.



## **II.4.1. Planning de traitement de cas d'utilisation**

### **A. Importance et exigences**

Dans le cycle de développement d'une application web, lorsque nous disons qu'il existe des besoins et des fonctionnalités plus prioritaires que d'autre, c'est que la réalisation de ceux-ci est primordiale pour la continuité et pour la clarté de l'application. Il faut donc établir un ordre de travail, afin de coordonner, ce qui va être réalisé durant les premiers incréments, et ce qui peut être fait ultérieurement.

Généralement, les fonctionnalités qui seront travaillées au début vont servir à la réalisation de celles qui les succèdent. Nous pouvons, par exemple, définir un ordre d'importance au cours du cycle de vie de notre application : priorité sur 2 (puisque nous aurons deux releases)

### **B. Risque**

Les seuls risques pouvant être cités dans notre cas, c'est que pour passer de la release 1 à la release 2, il faut qu'ils soient bouclés de telle sorte à ce qu'ils ne soient plus sujets à des changements.

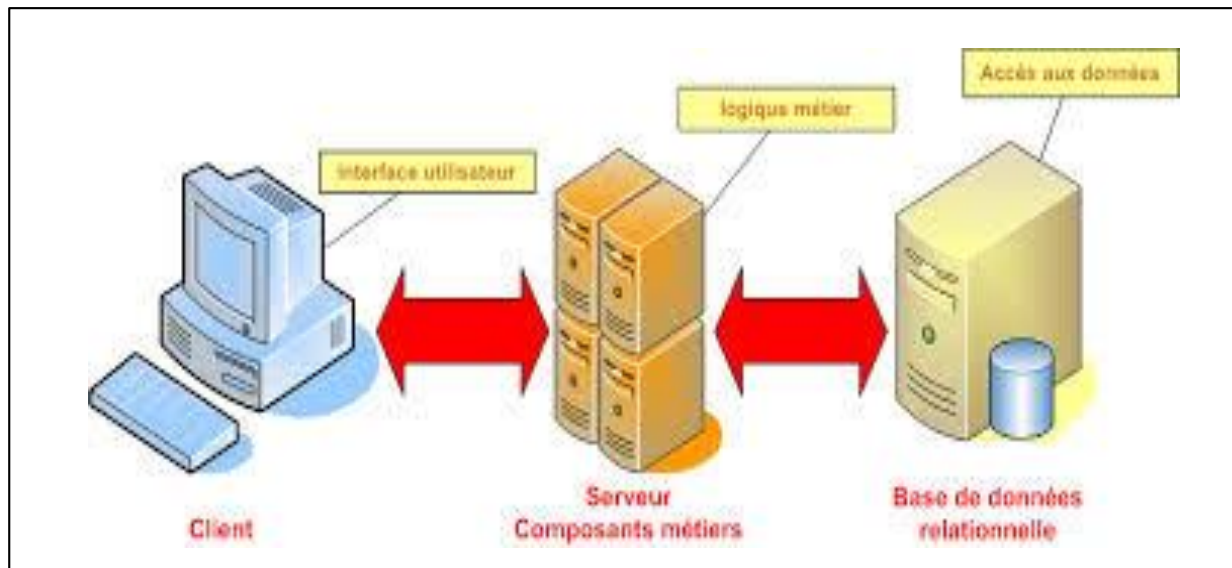
## **II.5. Architecture et modèle adoptés**

### **II.5.1. Architecture 3-tiers**

Une architecture est un modèle définissant comment sera le système. En ce sens, un style Architectural aide à avoir un aperçu du système avant son développement. Chaque système doit réaliser trois logiques :

- La logique applicative
- La logique de présentation
- La logique de stockage de données

La plupart des applications web adoptent une architecture 3-tiers. La figure suivante définit les différents composants de l'architecture 3-tiers [7]



**Figure 6 : Architecture 3-tiers**

Dans cette architecture trois composants sont mis en évidence :

- Le client : c'est le demandeur des ressources.
- Le serveur d'application : fournit les ressources demandées par le client par des traitements applicatifs avec l'appel à un serveur du niveau suivant.
- Le serveur de base de données : fournit les services demandés par le serveur opérant au sein de la couche applicative.

### II.5.2. Modèle MVC

Un patron de conception est un modèle de référence qui sert de source d'inspiration lors de la conception de l'architecture d'un système ou d'un logiciel informatique en sous-éléments plus simples [8]

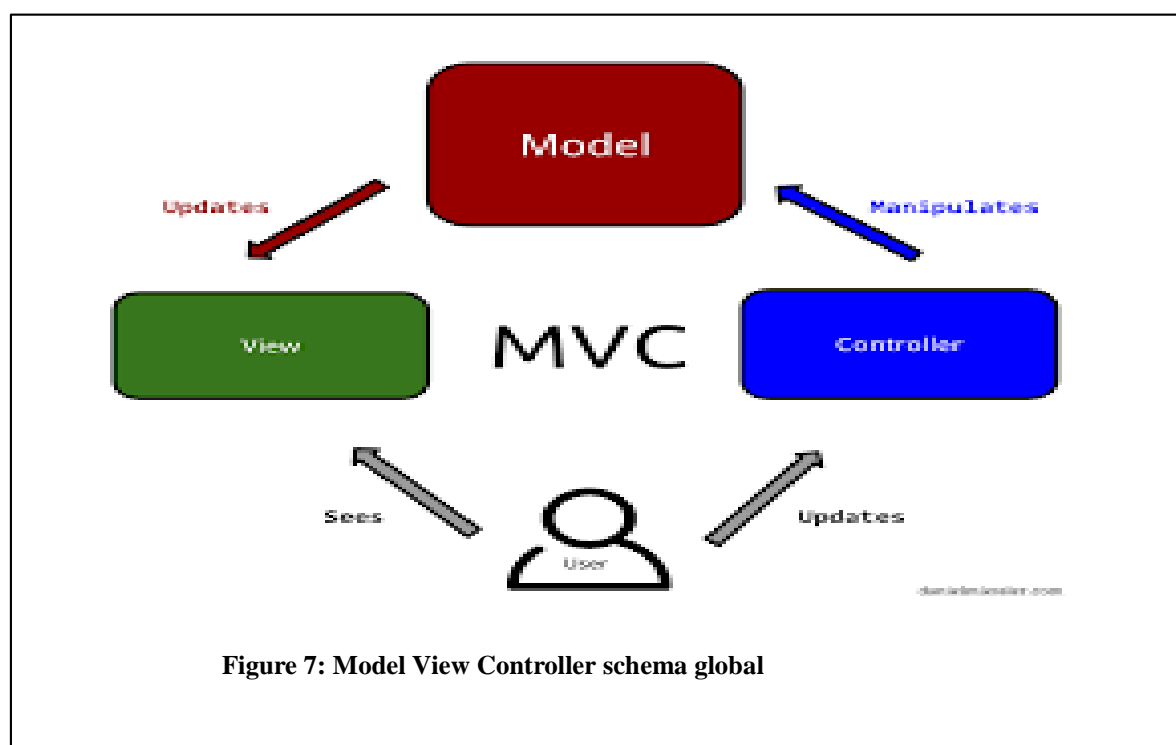
Le modèle MVC (Model–View–Controller) est un patron de conception de logiciel très populaire permettant d'être efficace et structurée lors du développement d'un projet.

- M : modèle (model) représente les données et ne fait rien d'autre, il ne dépend pas des vues et des contrôleurs.
- V : vue (view) affiche les données que le modèle fournit à l'utilisateur, ex : formulaire, bouton, etc. ... (Un modèle peut contenir plusieurs vues)
- C : contrôleur (Controller) traite les données de l'utilisateur, il dépend et interagit avec le modèle ainsi que la vue. [9]

Le modèle MVC nous permet d'être plus efficient en nous donnant la possibilité de réutiliser le même code plusieurs fois dans d'autres applications similaires; ainsi que de séparer notre structure en multiple calques. Cette combinaison nous offre une meilleure maintenance de notre projet et nous facilite la tâche durant les tests unitaires et nous permet de travailler à plusieurs sur un même projet (ex : le développeur back-end peut mettre en place la structure sans intervenir sur le front-end, vice versa).

Un autre avantage est le fait qu'il y a peu d'accrochage entre les vues, les modèles et les contrôleurs, ce qui nous permet d'avoir un code propre où les fonctions et les classes peuvent être facilement optimisées.

La figure suivante montre l'interaction entre les différents éléments du MVC.





## III. Pilotage du projet avec SCRUM

### III.1. Les outils SCRUM utilisés

#### III.1.1. Equipe et rôles

Avec une méthode de gestion de projet « classique », les rôles de différents intervenants sont généralement séparés. Les méthodes agiles, comme Scrum, brisent cette séparation et invitent tous les intervenants à travailler ensemble. Mais ce n'est pas pour autant qu'il n'y a pas d'organisation au sein du projet. Les **rôles Scrum** sont définis de manière bien précise, et nous allons les détailler ici.

- Le Product Owner : Il détient la vision du produit que l'on souhaite réaliser. Il représente les utilisateurs et le commanditaire. Dans notre prochain système c'est le propriétaire du produit (ETAP) : c'est un expert du domaine.
- Le Scrum Master : C'est le garant de la méthodologie Scrum. C'est le dirigeant de l'équipe : il a pour but de diriger, de guider et de rectifier le parcours de l'équipe tout au long du cycle de développement.
- Le scrum team : c'est l'équipe de développement (moi-même).

#### III.1.2. Le backlog du produit

C'est l'artefact le plus trivial dans le processus Scrum, puisqu'il contient l'ensemble des fonctionnalités du produit souhaité. Chaque fonctionnalité est représentée sous forme d'histoire utilisateur (user story) au sein du Backlog du produit, peut également cohabiter les caractéristiques techniques [10].

Le tableau suivant détaille le Backlog de notre application

N	Feature	ID S	User Story
1	<b>S'authentifier</b>	1.1	En tant qu'administrateur je peux accéder à mon compte grâce à un identifiant et un mot de passe
		1.2	En tant que personnel je peux accéder à mon compte grâce à un identifiant et un mot de passe
2	<b>Accéder aux services</b>	2.1	En tant que personnel je peux demander un congé
		2.2	En tant que personnel je peux consulter l'interface de mes absences
		2.3	En tant que personnel je peux demander un matériel
3	<b>Gérer personnel</b>	3.1	En tant qu'administrateur je peux ajouter un compte de personnel
		3.2	En tant qu'administrateur je peux modifier un compte de personnel
		3.3	En tant qu'administrateur je peux supprimer un compte de personnel
4	<b>Gérer les absences</b>	4.1	En tant qu'administrateur je peux consulter l'interface des absences des personnels
		4.2	En tant qu'administrateur je peux rechercher par date l'absence des personnels
5	<b>Gérer les congés</b>	5.1	En tant qu'administrateur je peux consulter les demandes des congés
		5.2	En tant qu'administrateur je peux accepter ou refuser un congé
5	<b>Gérer le matériel</b>	5.1	En tant qu'administrateur je peux consulter les demandes de matériel

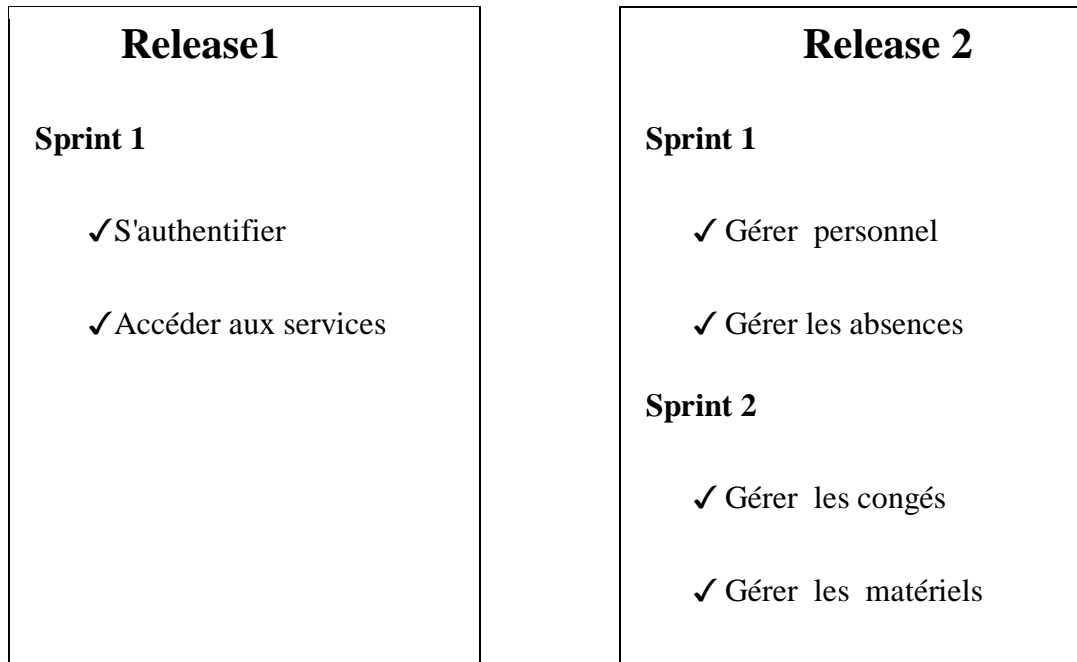
		5.2	En tant qu'administrateur je peux accepter ou refuser une demande de matériel
--	--	-----	---

**Table 3 : Backlog du Produit**

## IV. Planification de releases

L'étape de la planification est certes très importante pour une meilleure organisation et gestion du projet.

Afin de mener à bien la mise en place de notre application, nous allons découper les fonctionnalités à réaliser en deux releases. Ce découpage est basé en premier lieu sur la nature de l'acteur et en second lieu sur l'ordre de priorité de chaque fonctionnalité définie ci-avant (Product Backlog).



## Conclusion

Durant ce chapitre nous avons établi une étude préliminaire de notre travail. En premier lieu; nous avons identifié les besoins attendus de notre application. En second lieu; nous avons détaillé le premier artefact dans Scrum, qui est le Product Backlog comportant une liste exhaustive de fonctionnalités de notre système. En dernier lieu, nous avons préparé une planification de releases de notre application afin de mieux organiser notre travail et mener à bien la mise en place de notre application. Dans le chapitre suivant, nous allons entamer le développement du premier release.





# Chapitre 3 : Release 1

## Introduction

Dans ce chapitre, nous allons nous intéresser à la mise en place de la release 1 de notre système de l'analyse des besoins jusqu'au test. Nous traiterons en détails chacun des cas d'utilisation préalablement présentés.

## I. Sprint 1

Selon la planification établie au niveau du chapitre précédent, le premier Sprint porte principalement sur les cas d'utilisation « s'authentifier » et « accéder aux services » que nous allons décortiquer par la suite.

### I.1. Backlog du sprint (Sprint Backlog)

Le sprint est le cœur de Scrum. Il s'agit d'un bloc de temps durant lequel un incrément du produit sera réalisé. Tous les sprints d'une release ont une durée constante et ne se chevauchent jamais, c'est-à-dire qu'un sprint ne peut pas démarrer tant que le précédent n'est pas encore terminé.

Le tableau ci-dessous regroupe toutes les fonctionnalités qui seront développées au sein de ce sprint c'est ce qu'on appelle "**Sprint Backlog**"

User Story	Estimation
En tant qu'administrateur je souhaite m'authentifier	3
En tant que personnel je souhaite m'authentifier	3
En tant que personnel je peux demander un congé	2
En tant que personnel je peux consulter l'interface de mes absences	2
En tant que personnel je peux demander un matériel	2

Tableau 4 : Sprint Backlog

## I.2. La spécification fonctionnelle

La spécification fonctionnelle se traduit par le diagramme de cas d'utilisation du sprint en détaillant mieux le fonctionnement.

### I.2.1. Diagramme de cas d'utilisation du sprint



Figure 8 : Diagramme de cas d'utilisation du premier sprint

## I.2.2 Raffinement des cas d'utilisation

Afin de mieux assimiler les cas d'utilisation nous allons établir leurs raffinements pour livrer une description sur les différents scénarios possibles

### A. Raffinement du cas d'utilisation « s'authentifier »

Pour garantir la sécurité et la confidentialité des informations, il est impératif que chaque fonctionnalité du système soit précédée par une authentification.

A travers ce cas d'utilisation, l'utilisateur saisit son login et son mot de passe, le système va s'assurer de son existence en tant qu'utilisateur autorisé avant de lui donner l'accès au système.

La figure suivante présente le raffinement du diagramme du cas d'utilisation <s'authentifier>

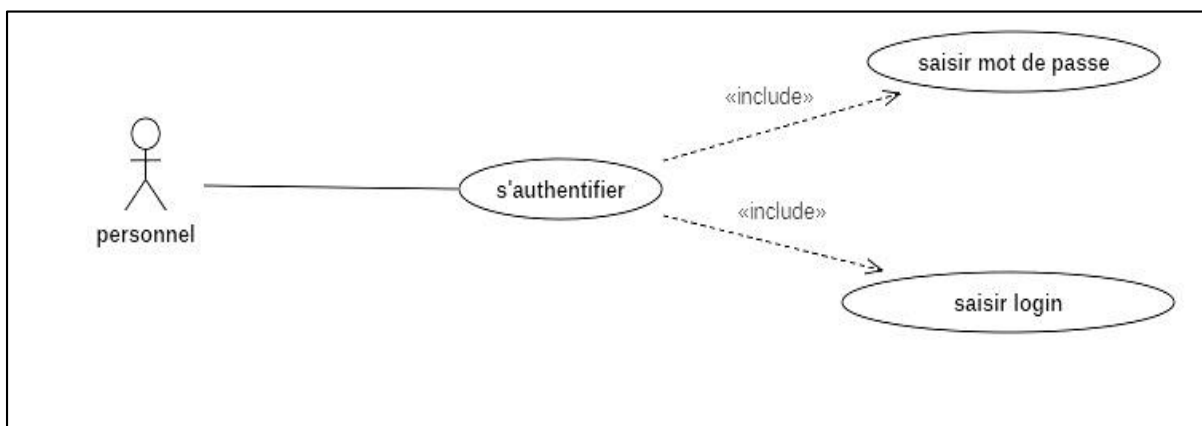


Figure 9 : raffinement de cas d'utilisation < s'authentifier >

❖ **Description textuelle du cas d'utilisation <s'authentifier >**

<b>Cas d'utilisation</b>	S'authentifier
<b>Acteur</b>	Utilisateur du système (Personnel / administrateur)
<b>Pré-condition</b>	Le login et le mot de passe saisis doivent être corrects
<b>Post-condition</b>	Utilisateur Authentifié
<b>Scénario principale</b>	1- L'utilisateur accède à l'interface de l'authentification 2- L'utilisateur saisit son login et son mot de passe 3-Il confirme en cliquant sur le bouton connexion 4-Le système affiche l'interface d'accueil propre à l'utilisateur
<b>Scénario alternatif</b>	Le système affiche un message d'erreur informant l'utilisateur que son login ou mot de passe sont incorrects

**Tableau5. Description textuelle du cas d'utilisation « s'authentifier"****B. Raffinement du cas d'utilisation « Accéder aux services »**

Une fois authentifié, le personnel a la possibilité d'accéder aux services. En effet, le personnel aura la possibilité de réaliser les tâches suivantes :

- demander un congé
- consulter l'interface de ses absences,
- demander un matériel.

La figure suivante présente le raffinement de diagramme de cas d'utilisation <Accéder aux services >

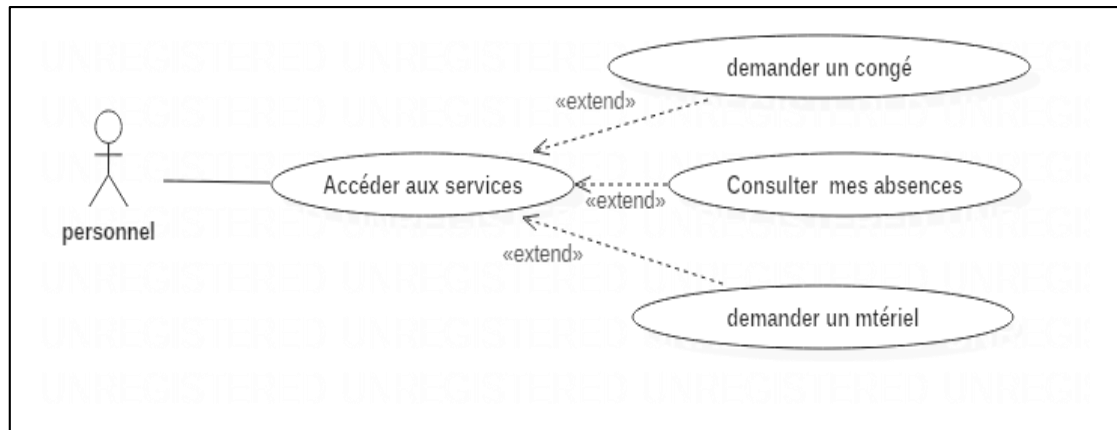


Figure 10 : raffinement de cas d'utilisation « Accéder aux services »

#### ❖ Description textuelle du cas « demander un congé »

<b>Cas d'utilisation</b>	Demander un congé
<b>Acteur</b>	Personnel
<b>Pré-condition</b>	Personnel authentifié interface de demande de congé affichée
<b>Post-condition</b>	Demande envoyée
<b>Scénario principal</b>	1- Le personnel clique sur le bouton demander un congé 2- Le système affiche l'interface de demande 3- Le personnel choisit le type de congé et sa durée 4- Le personnel confirme sa demande 5- Le personnel envoie sa demande
<b>Scénario alternatif</b>	Le système affiche un message d'erreur si la date est incorrecte
<b>Extensions</b>	Un administrateur peut demander un congé

Tableau6. Description textuelle du cas d'utilisation « demander un congé »

## ❖ Description textuelle du cas « Consulter mes absences »

<b>Cas d'utilisation</b>	Consulter mes absences
<b>Acteur</b>	Personnel
<b>Pré-condition</b>	Personnel authentifié
<b>Post-condition</b>	Affichage de l'interface de ses absences
<b>Scenario principal</b>	1- Le personnel clique sur le bouton consulter votre absences 2- Le système affiche l'interface des absences
<b>Scenario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion à la base de données
<b>Extensions</b>	Un administrateur peut consulter ses absences

❖

Tableau7. Description textuelle du cas d'utilisation « consulter mes absences »

## ❖ Description textuelle du cas « demander un matériel »

<b>Cas d'utilisation</b>	Demander un matériel
<b>Acteur</b>	Personnel
<b>Pré-condition</b>	Personnel authentifié
<b>Post-condition</b>	Affichage de l'interface des matériels
<b>Scenario principal</b>	1- Le personnel clique sur le bouton demande matériel 2- Le système affiche l'interface des matériels 3- Le personnel remplit formulaire 4-le personnel clique sur le bouton ajouter
<b>Scenario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion à la base de données
<b>Extensions</b>	Un administrateur peut demander un matériel

Tableau8. Description textuelle du cas d'utilisation « demander un matériel »





## I.3. Analyse

### I.3.1. Diagrammes de séquences système

Dans ce qui suit nous allons présenter les diagrammes de séquences systèmes des différentes fonctionnalités de notre premier sprint

#### A. Diagramme de séquence système du cas « s'authentifier »

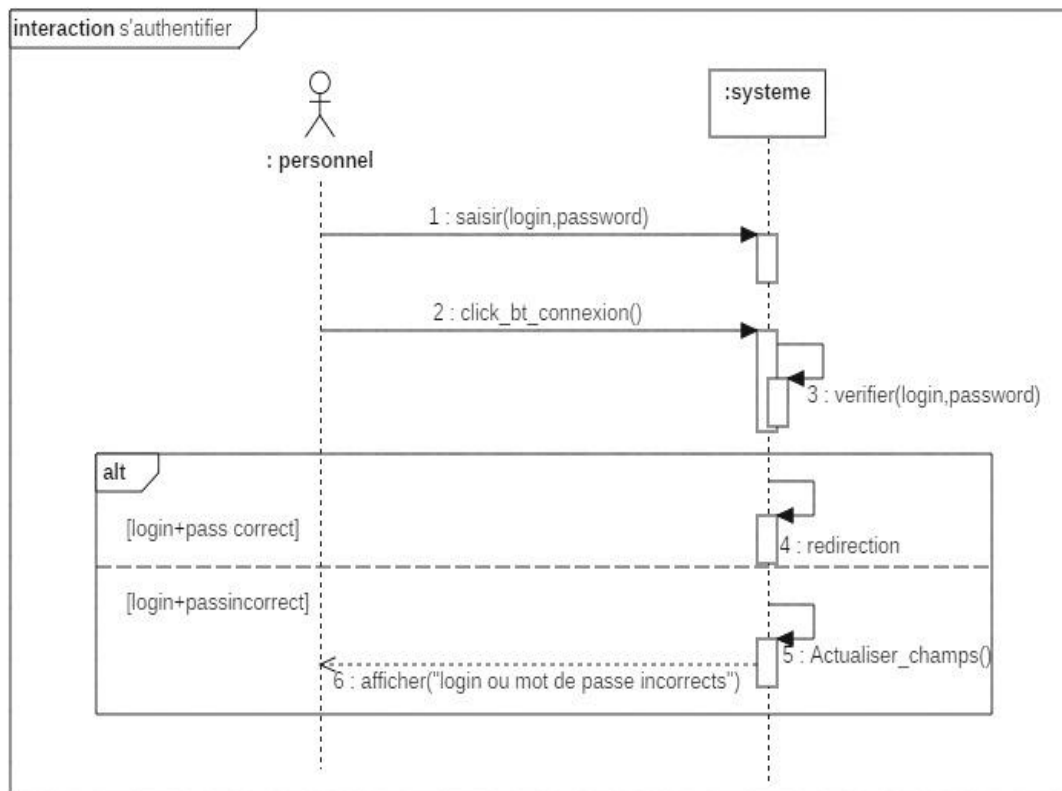


Figure 11 : diagramme de séquence système du cas « s'authentifier »

## B. Diagramme de séquence système du cas « demander un congé »

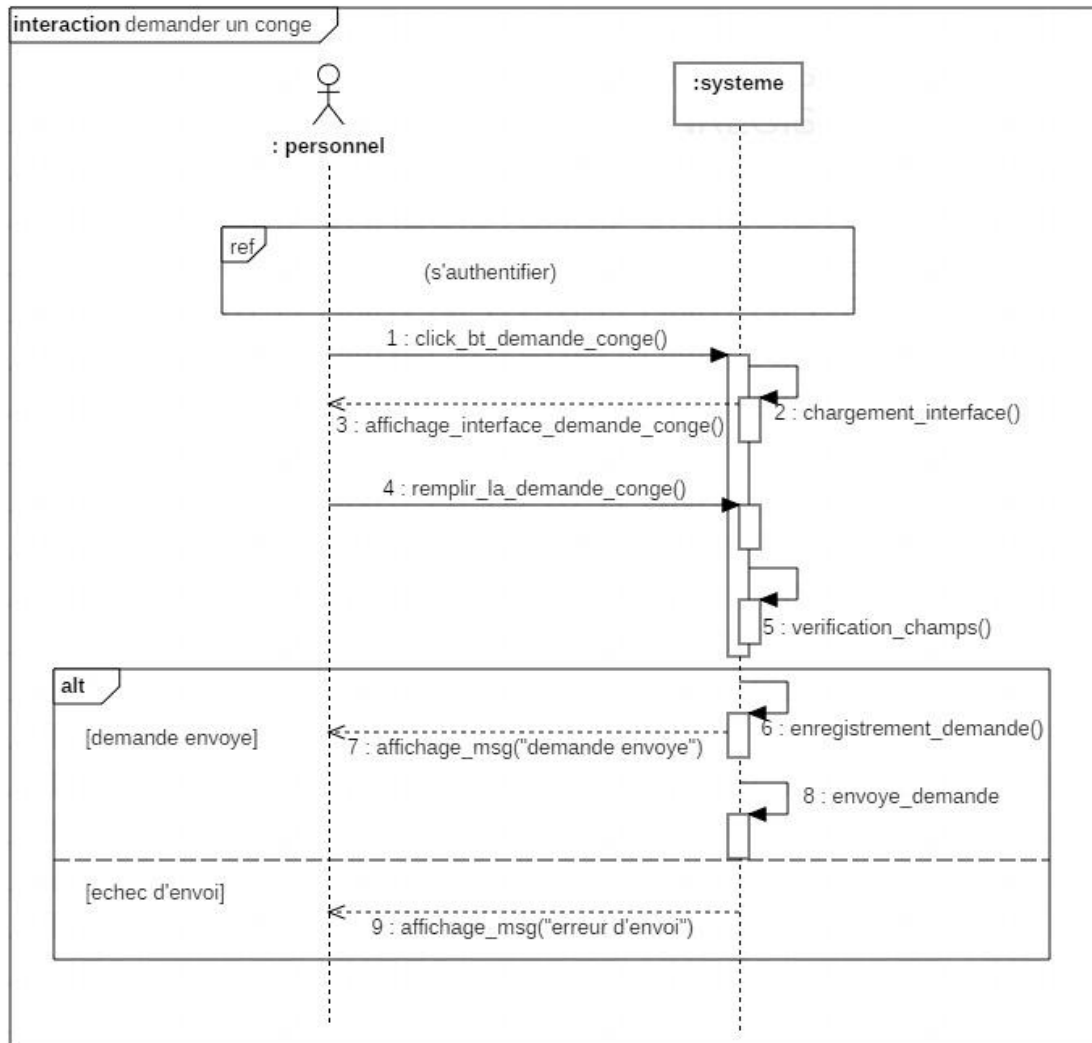


Figure 12 : Diagramme de séquence système du cas « demander congés »

### C. Diagramme de séquence système du cas « consulter mes absences »

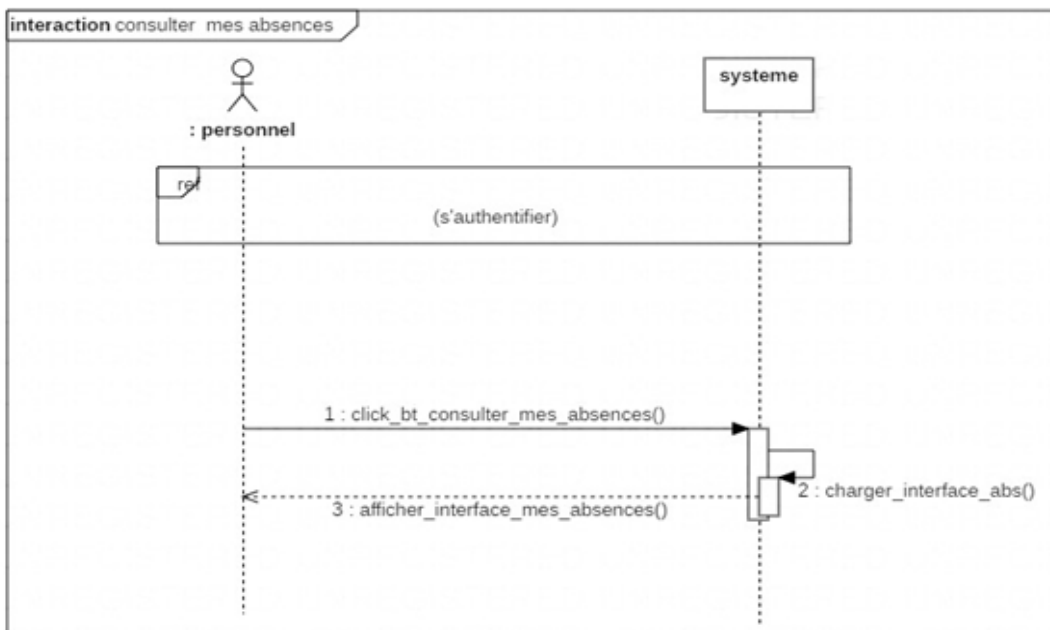


Figure 13 : diagramme de séquence système du cas « consulter mes absences »

#### 1.3.2. Diagrammes de séquences détaillés

Le diagramme de séquence détaillée permet de décrire les scénarios de chaque cas d'utilisation en interaction avec les objets tout mettant l'accent sur la chronologie des opérations. Les figures ci-après présentent les diagrammes de séquences détaillés de notre premier sprint.

## A. Diagramme de séquences détaillé du cas « s'authentifier »

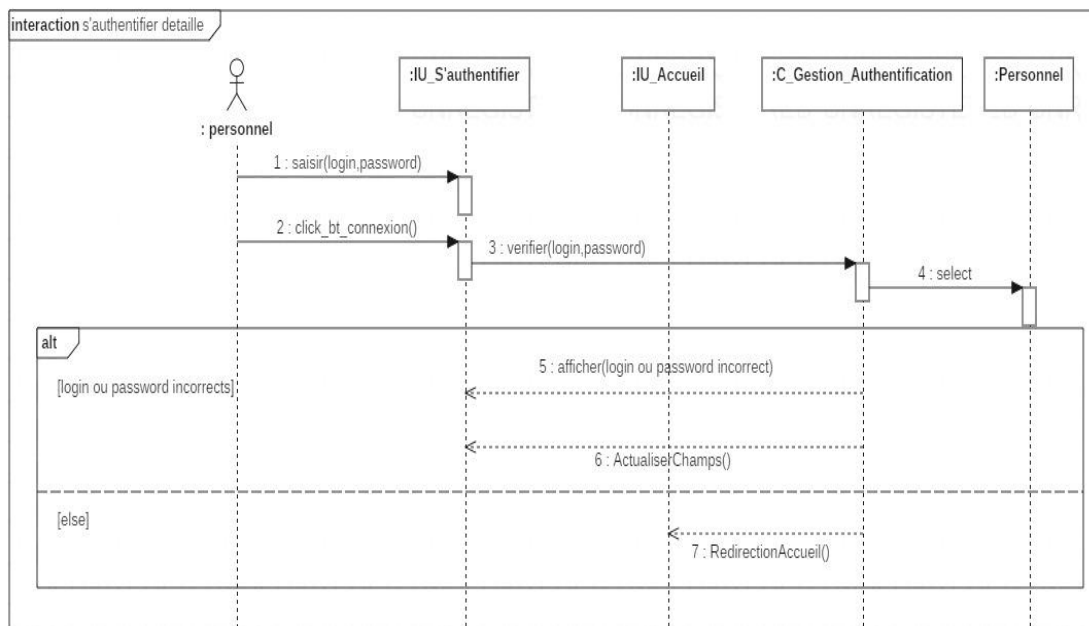


Figure14 : Diagramme de séquence détaillé du cas d'utilisation « s'authentifier »

## B. Diagramme de séquences détaillé du cas « demander un congé »

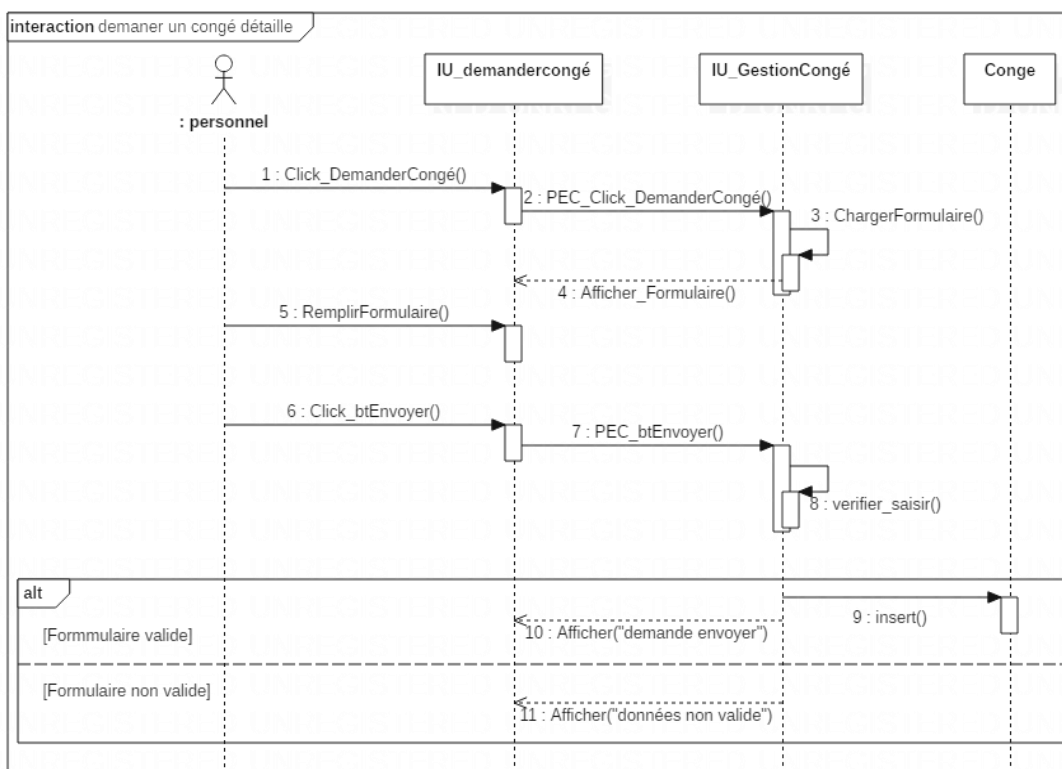


Figure 15 : Diagramme de séquence détaillé des cas d'utilisation « demander un congé »

### C. Diagramme de séquences détaillées du cas « consulter mes absences »

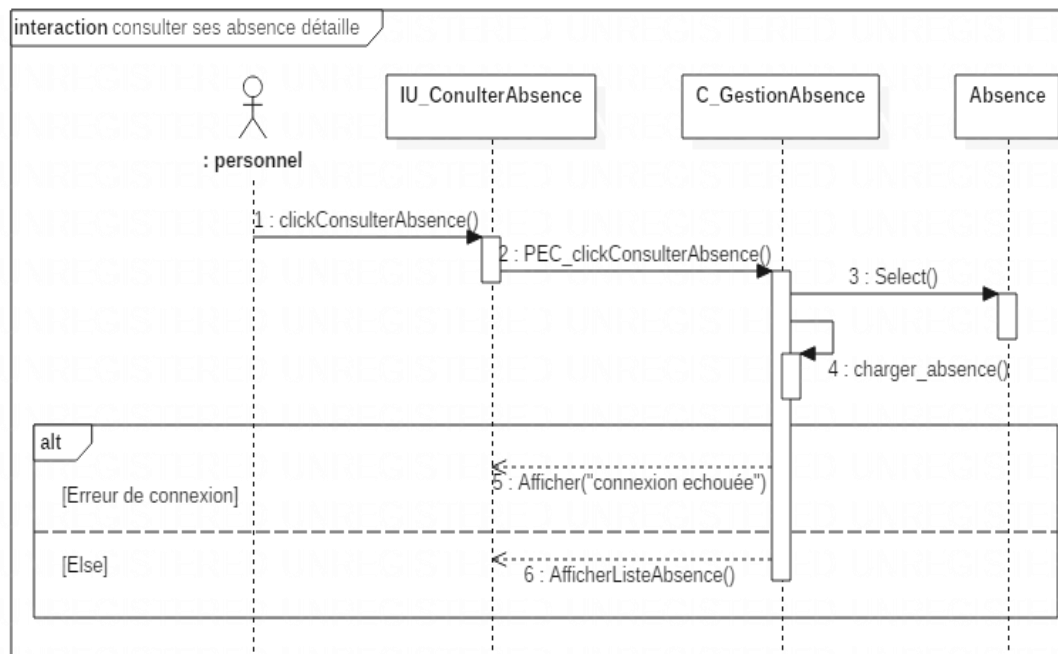


Figure 16 : Diagramme de séquence détaillé des cas d'utilisation « consulter mes absences »

### I.3.3. Diagramme de classes participantes

Le diagramme de classes participantes est important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, les modèles de la couche métiers et l'interface avec l'utilisateur. Il semble particulièrement important pour guider la phase de production du livrable final. On distingue trois stéréotypes de classes qui sont :

- **Boundary** : Les classes stéréotypées Boundary représentent les objets frontières du système. C'est la partie vue du pattern Modèle-Vue-Contrôleur.
- **control** : Les classes stéréotypées Control prennent en charge le traitement qui gère la dynamique du système lors de l'exécution d'une application. C'est la partie contrôle du pattern Modèle-Vue-Contrôleur.
- **entity** : Les classes stéréotypées Entity correspondent aux classes du domaine. Elles forment le cœur du système d'information du système. C'est la partie modèle du pattern Modèle-Vue-Contrôleur. [17]

Les figures ci-dessous illustrent les diagrammes de classes participantes de notre premier sprint.

#### A. Diagramme de classe participantes de conception du cas < S'authentifier >

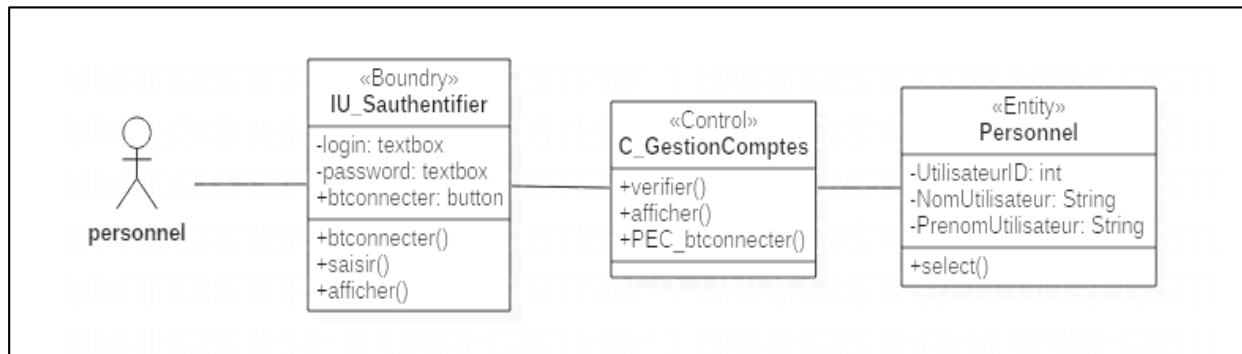


Figure 17 : Diagramme de classe participantes du cas < S'authentifier >

#### B. Diagramme de classe participantes de conception du cas < consulter mes absences >

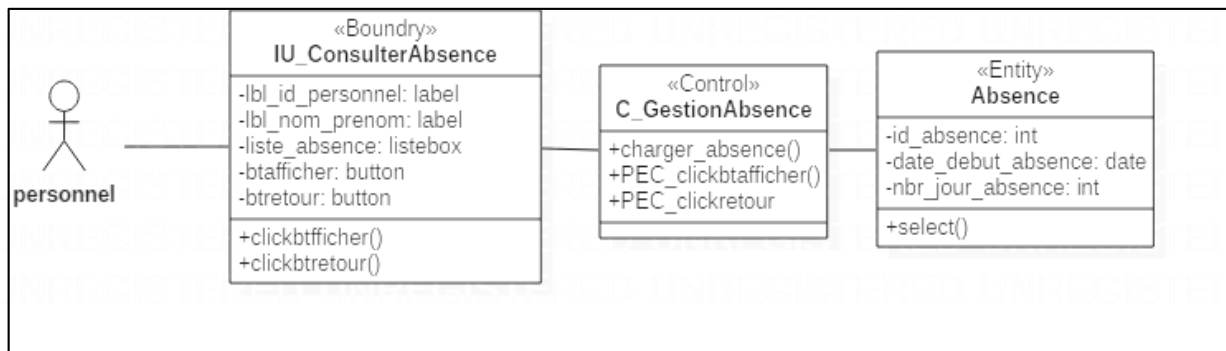
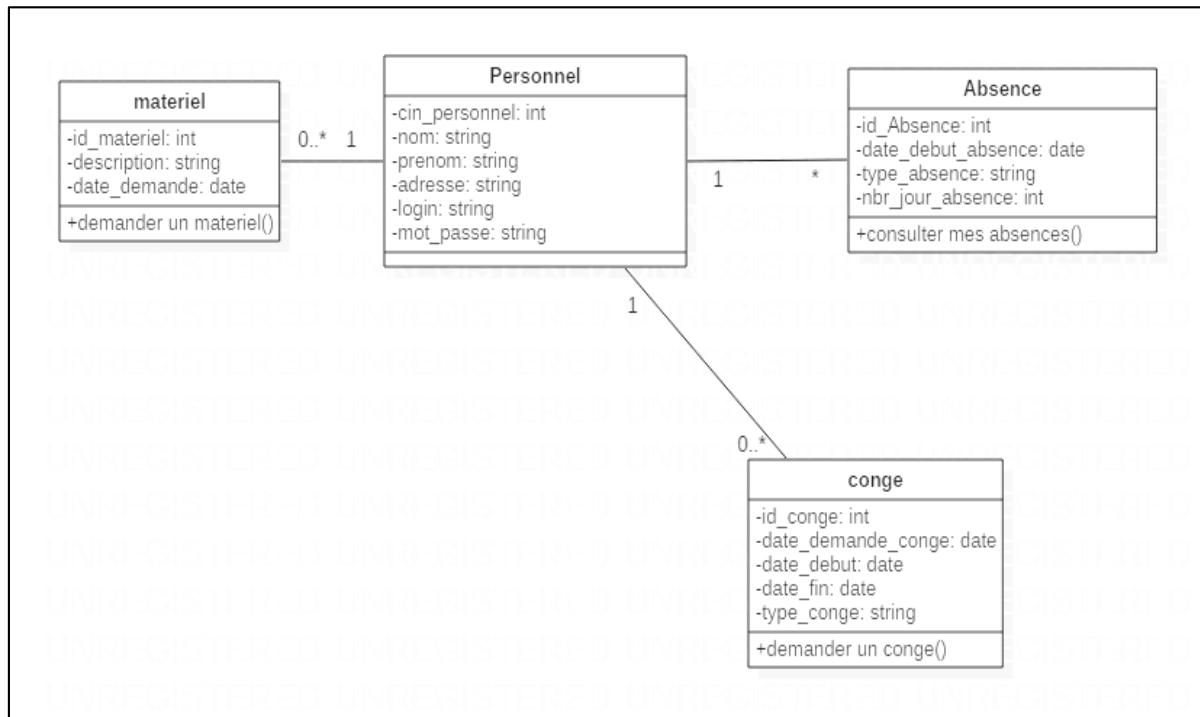


Figure 18 : diagramme de classe de conception du cas < consulter mes absences >

### I.3.4. Diagramme de classe du sprint 1



**Figure 19 : Diagramme de classe du sprint 1**

Les classes composant notre diagramme de classes sont décrites comme suit :

- La classe **Personnel** : Cette classe a comme attributs la clé primaire qui identifié le personnel (cin\_personnel : Int), le nom (nom : String), le prénom (prenom : String), l'adresse (adresse : string), le login (login : string) et le mot de passe (mot\_passe : string). Le personnel peut demander un matériel ; demander un conge et consulter ses absences.

- La classe **Absence** : Cette classe a comme attributs la clé primaire qui identifie l'absence (id\_Absence: Int), le date début d'absence (date\_debut\_absence : date), le type d'absence (type\_absence : string) le nombre de jour d'absence ( nbr\_jour\_absence : Int).

- La classe **Conge** : Cette classe a comme attributs la clé primaire qui identifié le congé (id\_conge : Int), le date de demande de conge (date\_demande\_conge : date) le date de debut de conge (date\_debut\_conge : date), le date fin de conge (date\_fin\_conge) et le type de congé (type\_conge : string)

- La classe **Matériel** : Cette classe a comme attributs la clé primaire qui identifié le matériels (id\_materiel: Int), description (description : String) et le date de demande d'une matériels (date\_demande : date).

### I.4. Réalisation

Dans cette partie, nous allons exposer quelques scénarios d'exécution à travers des captures d'écran.

### A. Interfaces authentification utilisateur

[Connexion](#)

## Inscription

Email :

Nom d'utilisateur :

Mot de passe :

Repetez Mot de passe :

[Créer un compte](#) [Annuler](#)

---

## Authentification

Nom d'utilisateur

Mot de passe

☒ **Se souvenir de moi**

[Connexion](#) [Annuler](#)

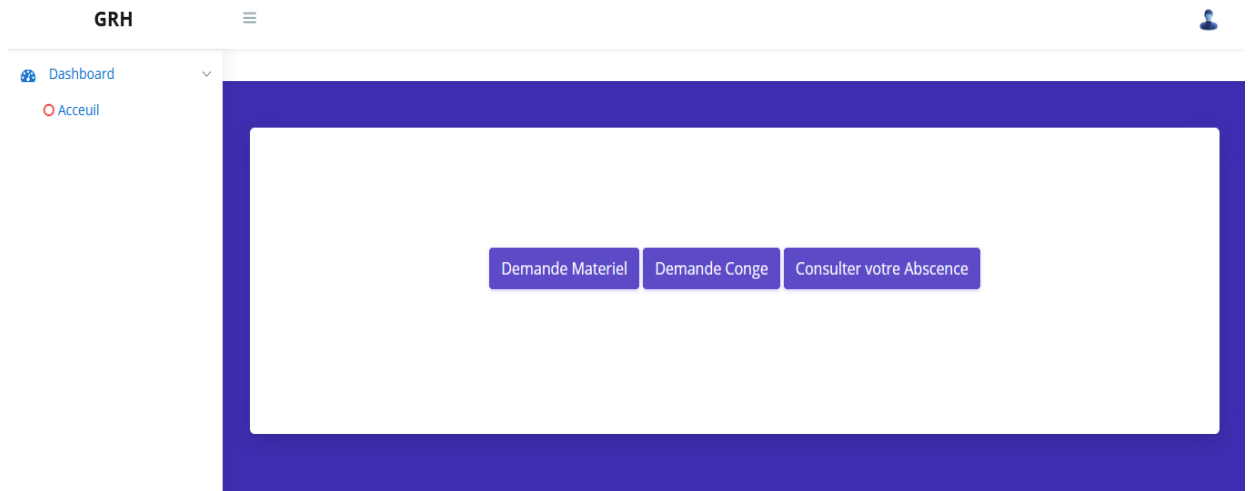
---

Figure 20 : Interfaces d'authentification

nous présentons les interfaces à partir desquelles l'utilisateur peut se connecter ainsi que le formulaire d'inscription pour la création d'un nouveau compte.

### B. Interfaces d'accès aux services





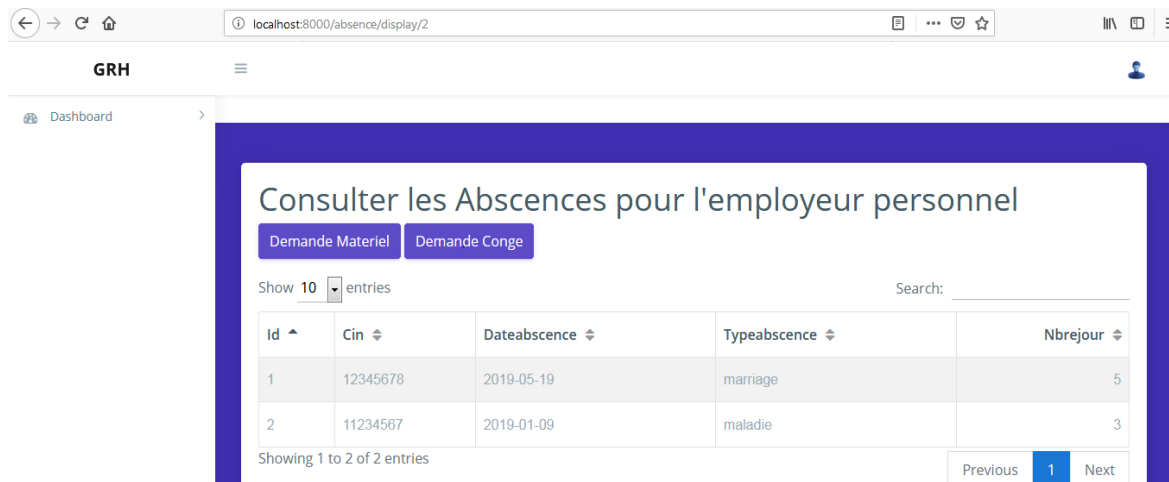
**Figure 21 : Interface d'accès aux services**

nous présentons l'interfaces à partir desquelles l'utilisateur peut accéder aux services (demande congé, demande matériel , consulter mes absence)

### C. Interface de demande un materiel

**Figure 22 : interface de demande de materiel**

### D. Interface consulter mes absences



**Figure 23 : interface << consulter mes absences >>**

## I.5 Tests

Le test d'un produit est un processus consistant qui vise à garantir le bon fonctionnement du système à travers une comparaison des comportements attendus et des résultats obtenus. Avant la fin de chaque Sprint nous devons tester les fonctionnalités du module, le tableau ci-après contient les différents tests réalisés.

Fonctionnalité	Démarche	Comportement attendu	Résultat
Test d'authentification de l'utilisateur	-demander la page d'authentification,  -saisir le login et le mot de passe,  -cliquer sur se connecter.	-le système affiche la page de connexion,  -l'utilisateur est redirigé vers la page d'accueil.	Conforme

Test de demande de materiel	-Demander la page d'accéder aux services  -Clique sur demande materiel  -remplir le formulaire convenablement  -cliquer sur Ajouter	-la page d'accéder aux services s'affiche,  - la page de demande de materiel s'affiche,  - le demande est ajouté	Conforme
Test de consulter mes absences	-Demander la page d'accéder aux services  -Clique sur consulter votre absence	- Affichage de la page d'accéder aux services  - Affichage de la page de consultation d'absence	Conforme

## Conclusion

A ce premier stade de notre projet, nous avons réussi à réaliser notre premier sprint. Nous disposons actuellement du premier incrément de notre logiciel. C'est une version qui est potentiellement exploitable. Dans la partie suivante, nos efforts seront consacrés à la réalisation de notre deuxième sprint.



## Chapitre 4. Release 2

### Introduction

Après avoir achevé le premier release de notre système informatique, nous pouvons maintenant nous lancer dans les travaux nécessaires pour produire la seconde. Ce deuxième incrément est composé de deux sprints comportant les fonctionnalités suivantes :

- Gérer personnel
- Gérer les absences
- Gérer les congés
- Gérer les matériels.

#### I. Sprint 1

Selon la planification établie précédemment, le premier Sprint porte principalement les cas d'utilisation « Gérer personnel » et « Gérer les absences ».

##### I.1. Backlog du sprint 1

Le tableau suivant présente le backlog du sprint 1 de la release 2

Histoire Utilisateur	Estimation
En tant qu'administrateur je souhaite ajouter un personnel	3
En tant qu'administrateur je souhaite modifier les informations des personnels	3
En tant qu'administrateur je souhaite supprimer un personnel	3
En tant qu'administrateur je peux consulter l'interface des absences des personnels	2

En tant qu'administrateur je peux rechercher par date l'absence des personnels	2
--	---

Tableau 9 : Backlog du sprint 1

I.2. Spécifications fonctionnelles

Dans cette partie nous allons détailler les cas d'utilisation inclus dans le sprint.

I.2.1. Diagramme de cas d'utilisation du sprint 1

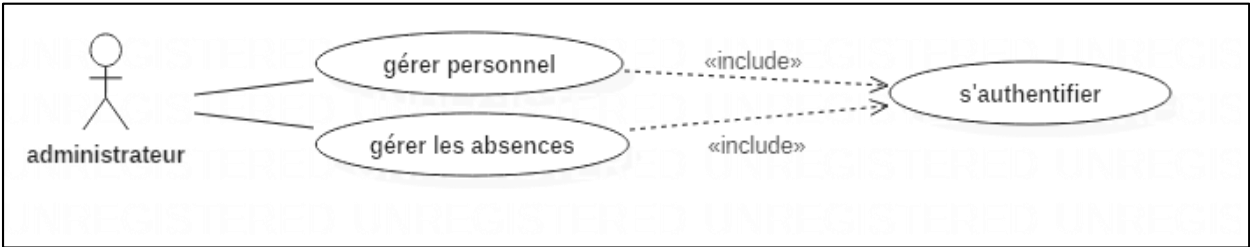


Figure 24 : Diagramme de cas d'utilisation du sprint 1

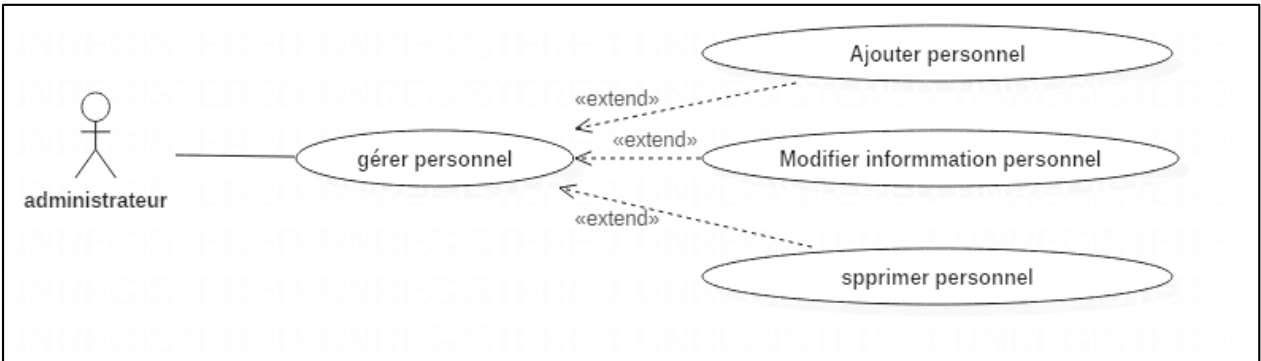
I.2.2. Raffinement des cas d'utilisation du sprint 1

Afin de mieux assimiler les cas d'utilisation de ce sprint, nous allons établir leur raffinement et ce afin d'avoir de prévoir sur les différents scenarios possible

A. Raffinement du cas d'utilisation « gérer personnel »

Une fois authentifié, l'administrateur a la possibilité de gérer personnel. En effet, l'administrateur peut ajouter, modifier ou supprimer un personnel.

La figure suivante présente le raffinement de diagramme de cas d'utilisation <Gérer dossier personnel >



**Figure 25 : Raffinement du cas d'utilisation « Gérer personnel »**

## ❖ Description textuelle du cas « Ajouter un personnel »

<b>Cas d'utilisation</b>	Ajouter personnel
<b>Acteur</b>	Administrateur
<b>Pré-condition</b>	Nouveau personnel ajouté
<b>Scénario principal</b>	1- L'administrateur clique sur le bouton ajouter 2- Le système affiche l'interface d'ajout 3- L'administrateur remplit les champs de l'interface 4- L'administrateur confirme en cliquant sur le bouton < ajouter> 5- Le système ajoute le nouveau compte à la liste du compte personnel
<b>Scénario alternatif</b>	Le système affiche un message informant l'utilisateur que certaines données sont dupliquées ou erronées

Table 10. Description textuelle du cas d'utilisation « Ajouter personnel »

## ❖ Description textuelle du cas « Modifier information personnel »

<b>Cas d'utilisation</b>	Modifier information personnel
<b>Acteur</b>	Administrateur
<b>Pré-condition</b>	Administrateur authentifié
<b>Post-condition</b>	Compte personnel modifié



<b>Scénario principal</b>	1- L'administrateur clique sur le bouton liste personnel 2- Le système affiche liste des compte existant 3- L'administrateur choisit le compte à modifier 4-L'administrateur clique sur le bouton modifier 5-Le système affiche l'interface de modification 6-L'administrateur modifie les champs 7- L'administrateur confirme en cliquant sur le bouton <modifier > 8- Le système modifie le compte personnel
<b>Scénario alternatif</b>	Le système affiche un message informant l'utilisateur que certains données sont dupliquées ou erronées

Tableau11. Description textuelle du cas d'utilisation « modifier information personnel »

## ❖ Description textuelle du cas « supprimer personnel »

<b>Cas d'utilisation</b>	Supprimer personnel
<b>Acteur</b>	Administrateur
<b>Pré –condition</b>	Administrateur authentifié
<b>Post-condition</b>	Compte personnel supprimé
<b>Scénario principal</b>	1- L'administrateur clique sur le bouton < lister compte> 2- Le système affiche la liste du compte existant 3- L'administrateur choisit le compte à supprimer 4- L'administrateur clique sur le bouton supprimer compte 5- le système supprime le compte choisi
<b>Scénario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion avec la base de données

Tableau12. Description textuelle du cas d'utilisation « supprimer personnel »

## B. Raffinement du cas d'utilisation « gérer les absences »

Une fois authentifié, l'administrateur a la possibilité de gérer les absences personnelles.

La figure suivante présente le raffinement de diagramme de cas d'utilisation < gérer les absences >

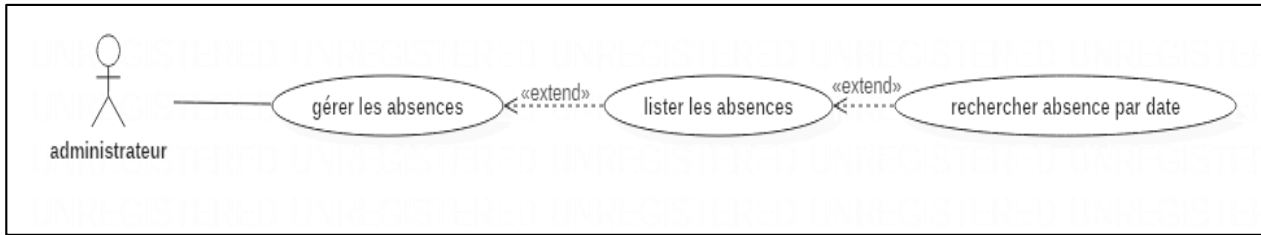


Figure 26 : Raffinement du cas d'utilisation « gérer les absences »

### ❖ Description textuelle du cas « lister les absences »

<b>Cas d'utilisation</b>	Lister l'absence
<b>Acteur</b>	Administrateur
<b>Pré-condition</b>	Administrateur authentifié
<b>Post-condition</b>	Affichage de la liste de personnels absents
<b>Scénario principal</b>	1- L'administrateur clique sur le bouton < gestion absentéisme > 2- Le système affiche la liste des absences
<b>Scénario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion à la base de données.

Tableau13. Description textuelle du cas d'utilisation « lister les absences »

## ❖ Description textuelle du cas &lt; rechercher l'absence par date &gt;

<b>Cas d'utilisation</b>	Rechercher l'absence par date
<b>Acteur</b>	Administrateur
<b>Pré-condition</b>	Administrateur authentifié
<b>Post-condition</b>	Affichage de la liste de personnels absents pendant une date saisie
<b>Scénario principal</b>	1- L'administrateur clique sur le bouton <gestion absentéisme > 2- Le système affiche la liste de personnels absents 3- L'administrateur choisit la date 4- Le système réaffiche la liste des personnels actualisée
<b>Scénario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion à la base de données.

Tableau14. Description textuelle du cas d'utilisation &lt;rechercher l'absence par date &gt;

## I.3. Analyse

### I.3.1. Diagrammes de séquence système

Dans ce qui suit nous allons présenter les diagrammes de séquences systèmes des différentes fonctionnalités de notre sprint à savoir :

- diagramme de séquence système du cas « Ajouter personnel »
- diagramme de séquence système du cas « Modifier informations personnelles »
- diagramme de séquence système du cas « Supprimer personnel »
- diagramme de séquence système du cas « Lister les absences »

### A. Diagramme de séquence système du cas « Ajouter personnel »

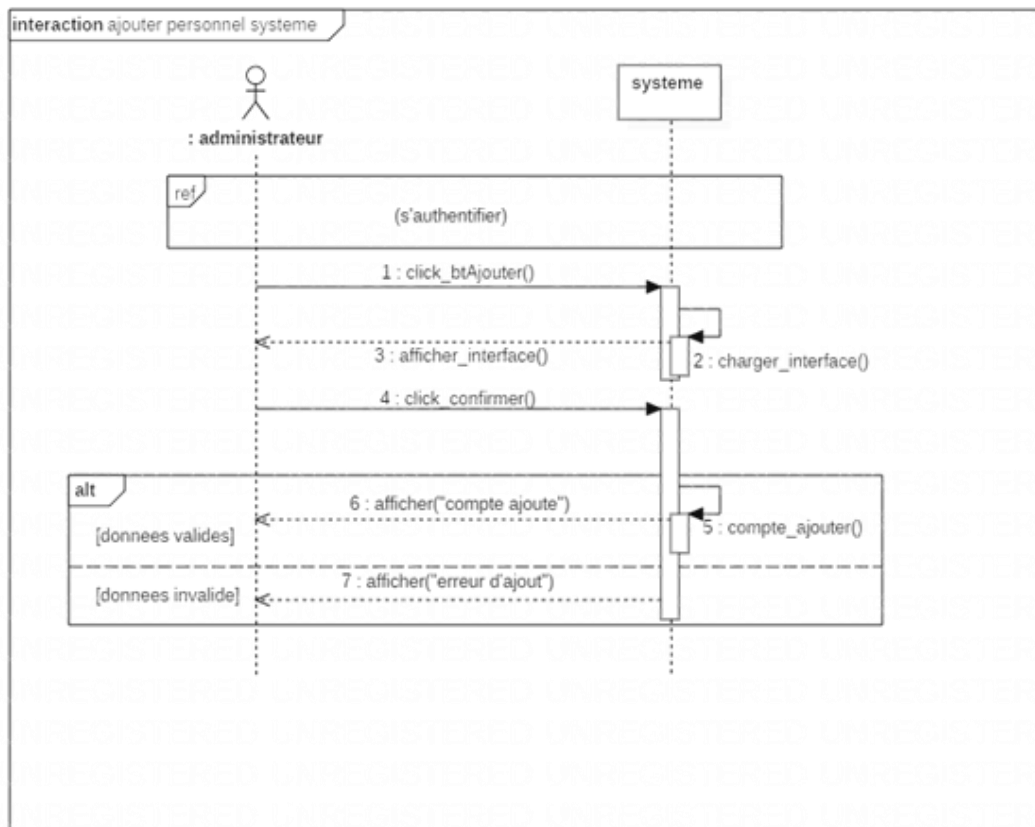


Figure 27 : Diagramme de séquence système du cas « Ajouter personnel »

### B. Diagramme de séquence système du cas « modifier information personnel »

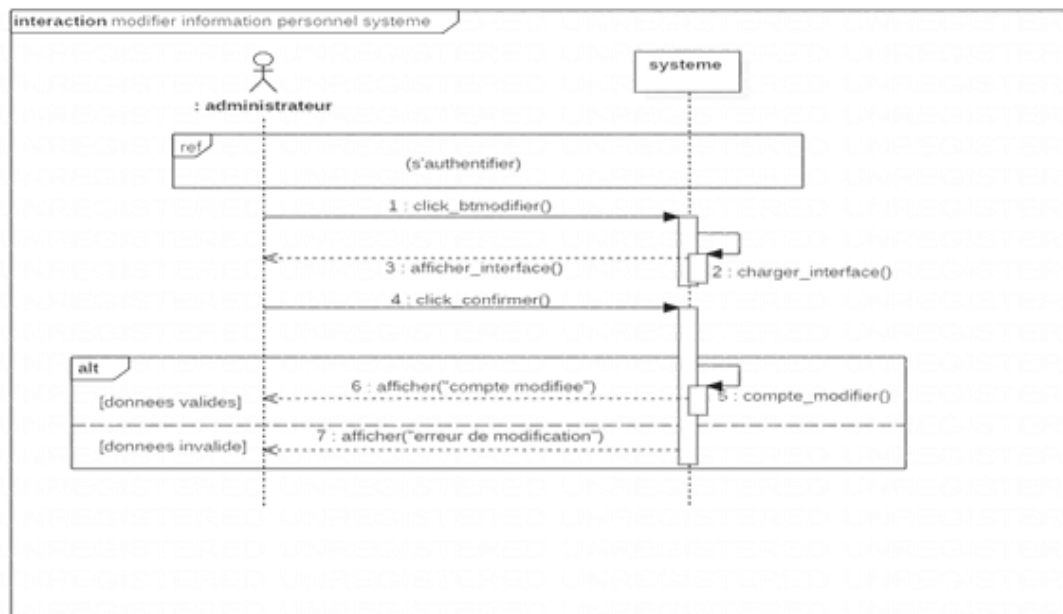


Figure 28 : Diagramme de séquence système du cas « modifier information personnel »

### C. Diagramme de séquence système du cas « supprimer personnel »

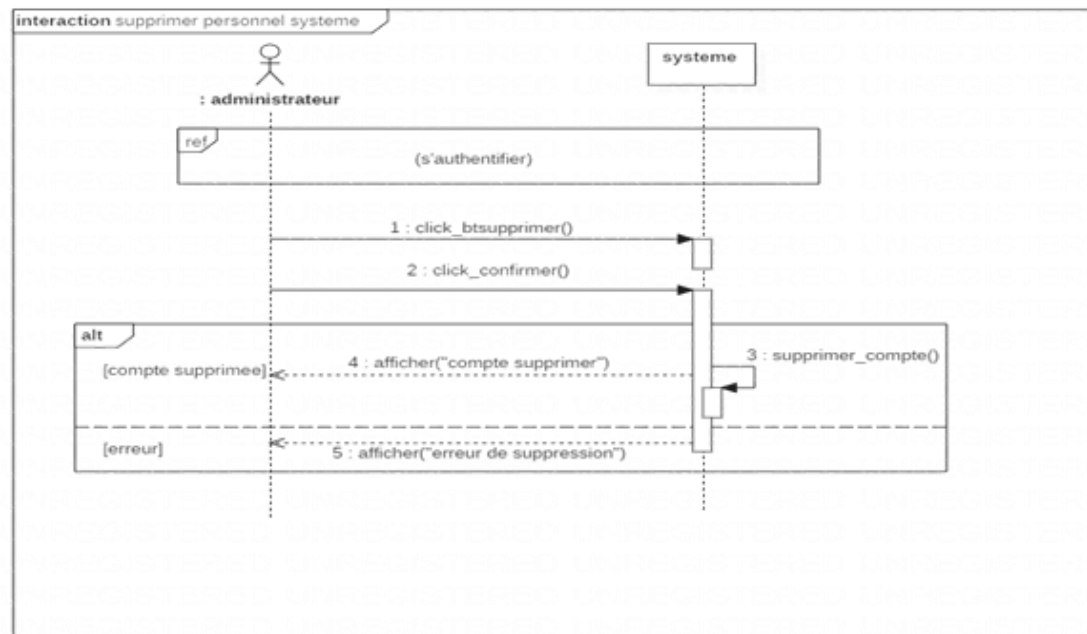


Figure 29 : Diagramme de séquence système du cas « supprimer personnel »

#### D. Diagramme de séquence système du cas « lister les absences »

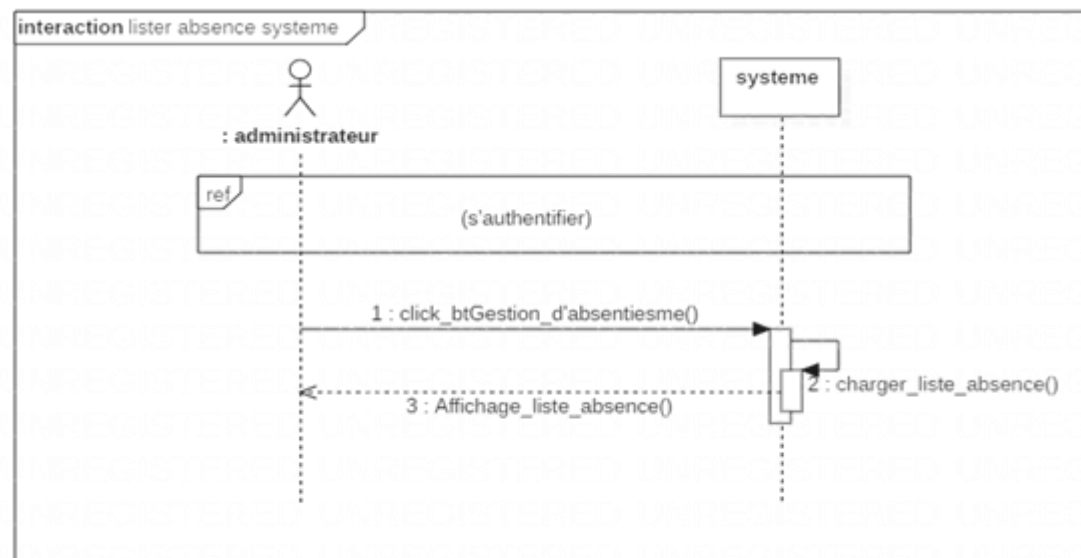


Figure 30 : Diagramme de séquence système du cas « lister les absences »

### I.3.2. Diagrammes de classes participantes

Dans ce qui suit nous allons présenter les diagrammes de classes participantes des cas d'utilisation du sprint actuel.

#### A. Diagramme de classe de conception du cas « gérer personnel »

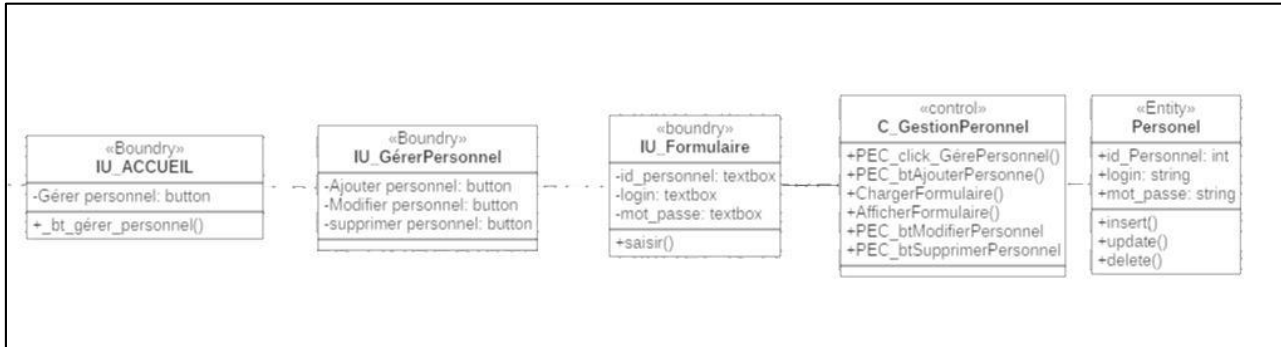


Figure 31: Diagramme de classes participantes du cas « gérer personnel »

#### B. Diagramme de classe de conception du cas « gérer les absences »

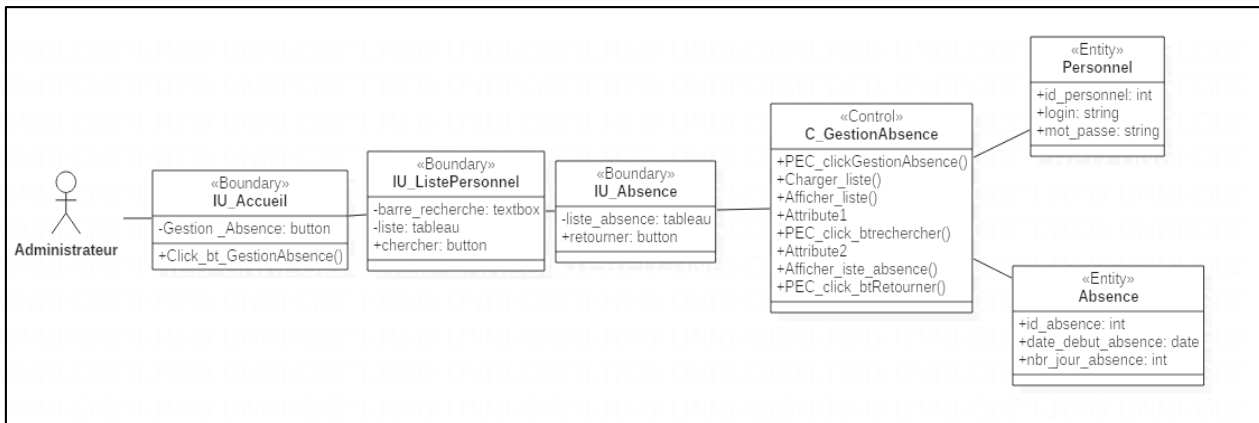


Figure 32 : Diagramme de classe de conception du cas « gérer les absences »

### 1.3. 3. Diagramme de séquence détaillée

Dans ce qui suit nous allons détailler les différentes interactions dans les cas d'utilisation de notre sprint actuel et ce via les diagrammes de séquence détaillés.

Afin de pouvoir réaliser cette fonctionnalité, l'administrateur doit commencer par s'authentifier.

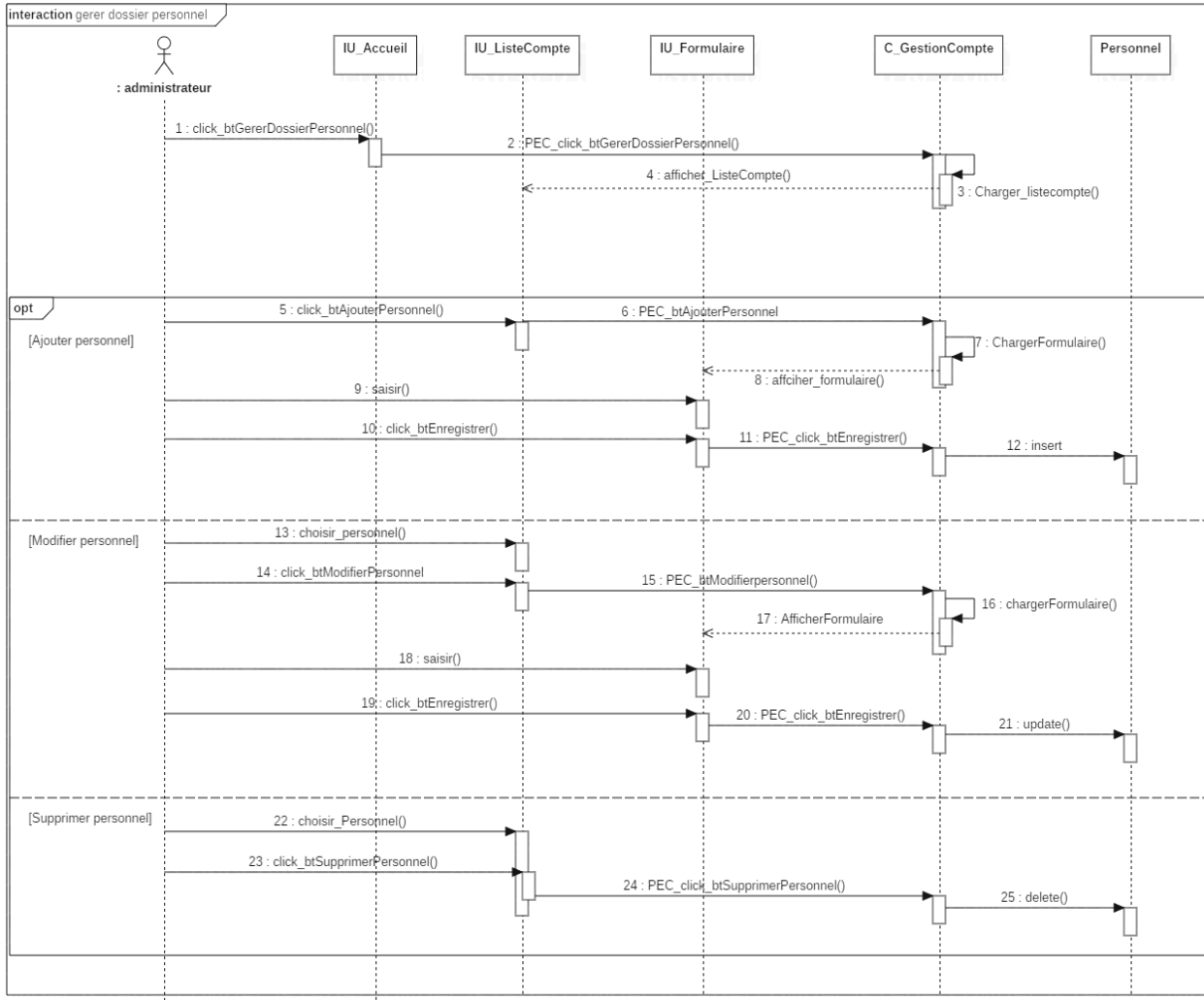


Figure 33 : Diagramme de séquence détaillé du cas « gérer personnel »

### I.3.4. Diagramme de classe du sprint 1

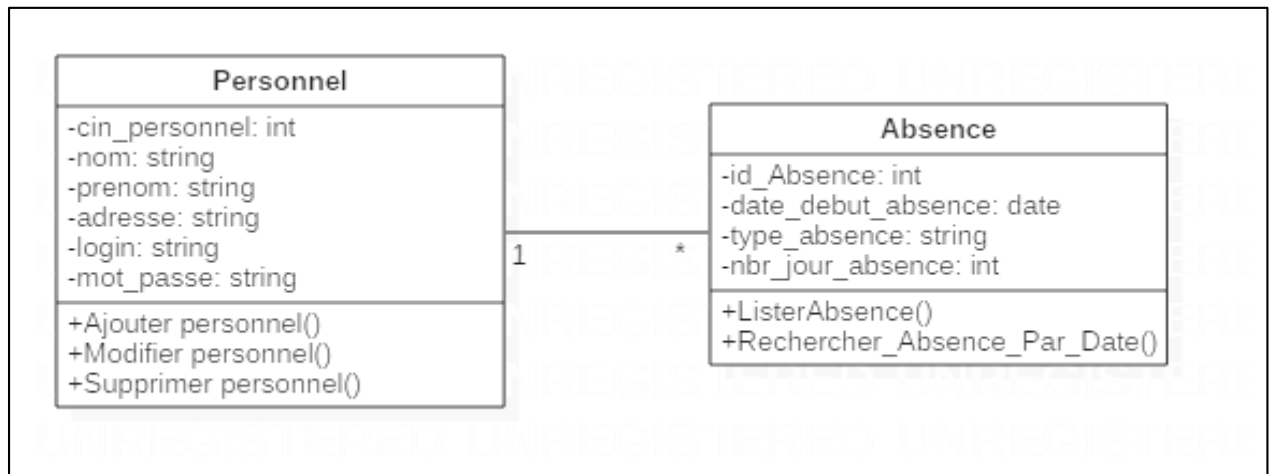


Figure34 : Diagramme de classe du premier sprint

## I.4. Réalisation

### A. Interface Ajouter personnel

**Ajouter un utilisateur**  
 Les champs marqués par un astérisque (\*) sont obligatoires

username\* :

email\* :

password\* :

Figure 35 : Interface << Ajouter personnel >>

### B. Interface modifier information personnel



User edit

username \* : admin

email \* : admin@live.com

password \* : 12345

Modifier

Figure 36: Interface &lt;&lt; modifier information personnel &gt;&gt;

### C. Interface « lister les absences »

liste des Absences

Ajouter

Show 10 entries

Search: \_\_\_\_\_

Id ▲	Cin ◆	Dateabsence ◆	Typeabsence ◆	Nbrejour ◆	
1	12345678	2019-05-19	marriage	5	Afficher
2	11234567	2019-01-09	maladie	3	Afficher

Showing 1 to 2 of 2 entries

Previous

Figure 37 : interface « lister les absences »

## I.5. Test

Fonctionnalité	Démarche	Comportement attendu	Résultat

Test Ajout de personnel	-Demander la page gérer personnel -Clique sur ajouter personnel -remplir le formulaire convenablement -cliquer sur Ajouter	-la page gérer personnel - la page d'ajout de personnel s'affiche, - le personnel est ajouté	Conforme
Test lister les absences	-Demander la page gérer absence -Clique sur lister les absences	- Affichage de la page gérer absence - Affichage de la page lister les absences	Conforme

## II. Sprint 2

Une fois le sprint 1 est validé, nous passons maintenant au sprint 2 de la release 2. Ce second sprint porte principalement les cas d'utilisation « Gérer les congés » et « Gérer les matériels ».

### II.1. Backlog du sprint 2

User Story	Estimation
En tant qu'administrateur je peux consulter les demandes de congés de personnels	2
En tant qu'administrateur je peux accepter ou refuser une demande de congé	2
En tant qu'administrateur je peux consulter les demandes de matériels de personnels	2
En tant qu'administrateur je peux accepter ou refuser une demande de matériel	2

Table 15. Backlog du sprint 2

### II.2. Spécifications fonctionnelles

Dans cette partie nous allons détailler les cas d'utilisation inclus dans le sprint.

### 2.2.1. Diagramme de cas d'utilisation du sprint 1

La figure ci-après présente le diagramme de cas d'utilisation du sprint actuel

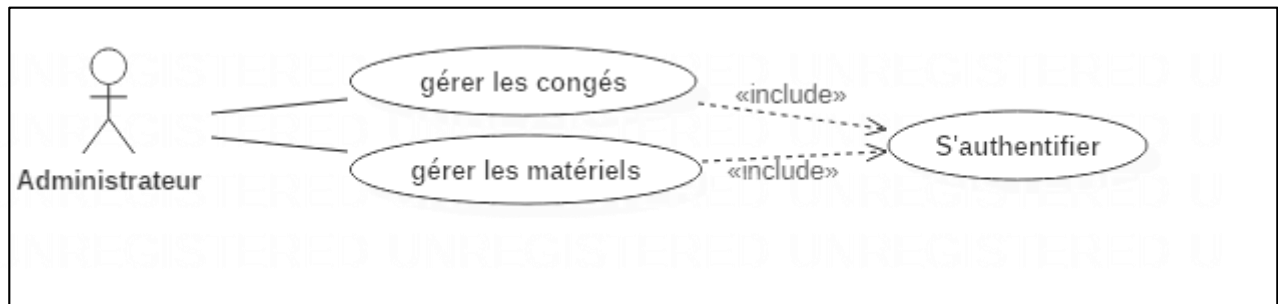


Figure38 : Diagramme de cas d'utilisation du sprint 2

### II.2.2. Raffinement des cas d'utilisation du sprint 2

Afin de mieux assimiler les cas d'utilisation de ce sprint, nous allons établir leurs raffinements et ce afin de prévoir les différents scénarios possibles

#### A. Raffinement du cas d'utilisation « gérer les congés »

Une fois authentifié, l'administrateur a la possibilité de gérer les congés. En effet, l'administrateur peut accepter ou refuser une demande de congé.

La figure suivante présente le raffinement de diagramme de cas d'utilisation « gérer congés »

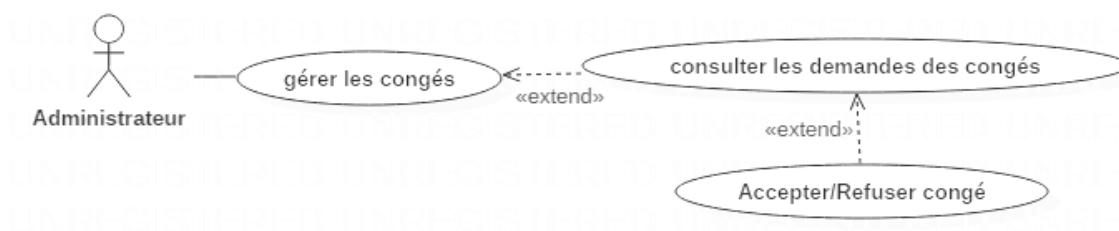


Figure 39 : Raffinement de cas d'utilisation « gérer les congés »

#### ❖ Description textuelle du cas « consulter les demandes des congés »

<b>Cas d'utilisation</b>	Consulter les demandes des congés
<b>Acteur</b>	Administrateur
<b>Pré condition</b>	Administrateur authentifié
<b>Post condition</b>	Affichage de la liste des demandes des congés

<b>Scénario principal</b>	1-l'administrateur clique sur le bouton Gestion congés 2-Le système affiche une liste des demandes des congés
<b>Scénario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion avec la base de données

Table 16. Description textuelle du cas « consulter les demandes des congés »

## ❖ Description textuelle du cas &lt; Accepter /refuser congé &gt;

<b>Cas d'utilisation</b>	Accepter /refuser congé
<b>Acteur</b>	Administrateur
<b>Pré condition</b>	Administrateur authentifié
<b>Post condition</b>	Demande accepté ou refusé
<b>Scénario principal</b>	1- L'administrateur clique sur le bouton Gestion congés 2- Le système affiche une liste des demandes des congés 3- L'administrateur choisit la demande 4- L'administrateur accepte ou refuse un demande
<b>Scénario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion avec la base de données

Table 17. Description textuelle du cas &lt; Accepter /refuser congé &gt;

**B. Raffinement du cas d'utilisation « gérer les matériels »**

Une fois authentifié, l'administrateur a la possibilité de gérer les matériels. En effet, l'administrateur peut accepter ou refuser une demande de matériel

La figure suivante présente le raffinement de diagramme de cas d'utilisation < gérer les matériels >

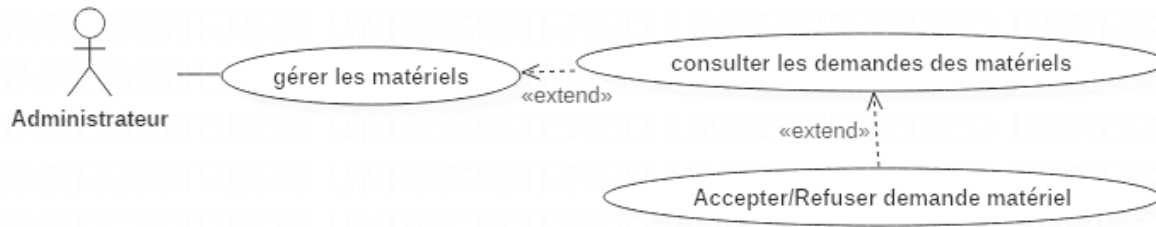


Figure 40 : Raffinement du cas d'utilisation « gérer les matériels »

## ❖ Description textuelle du cas « Consulter les demandes de matériel »

<b>Cas d'utilisation</b>	Consulter les demandes des matériels
<b>Acteur</b>	Administrateur
<b>Pré condition</b>	Administrateur authentifié
<b>Post condition</b>	Affichage de la liste des demandes des matériels
<b>Scénario principal</b>	1-l'administrateur clique sur le bouton Gestion matériel 2-le système affiche une liste des demandes des matériels
<b>Scénario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion avec la base de données

Table 18.Description textuelle du cas « Consulter les demandes de matériel »

## ❖ Description textuelle du cas &lt; Accepter / refuser demande matériel &gt;

<b>Cas d'utilisation</b>	Accepter demande matériel
<b>Acteur</b>	Administrateur
<b>Pré condition</b>	Administrateur authentifié
<b>Post condition</b>	Demande accepté ou refusé

<b>Scénario principal</b>	1-l'administrateur clique sur le bouton Gestion matériel 2-le système affiche une liste des demandes des congés 3-l'administrateur choisit la demande 4-l'administrateur accepte la demande
<b>Scénario alternatif</b>	Le système affiche un message d'erreur s'il existe un problème de connexion avec la base de données

Table 19. Description textuelle du cas &lt; Accepter / refuser demande matériel &gt;

## II.3. Analyse

### II.3.1. Diagrammes de séquence système

Dans ce qui suit nous allons présenter les diagrammes de séquences systèmes des différentes fonctionnalités de notre sprint.

#### A. Diagramme de séquence système du cas « Consulter les demandes des congé »

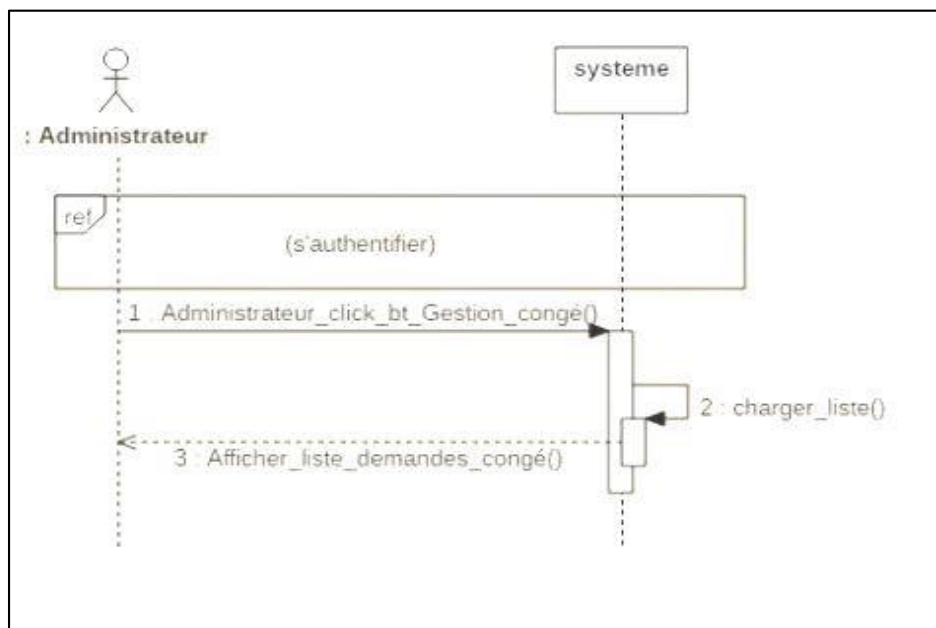


Figure 41: Diagramme de séquence système du cas &lt; Consulter les demandes des congés &gt;

### B. Diagramme de séquence système du cas « Accepter /Refuser congé »

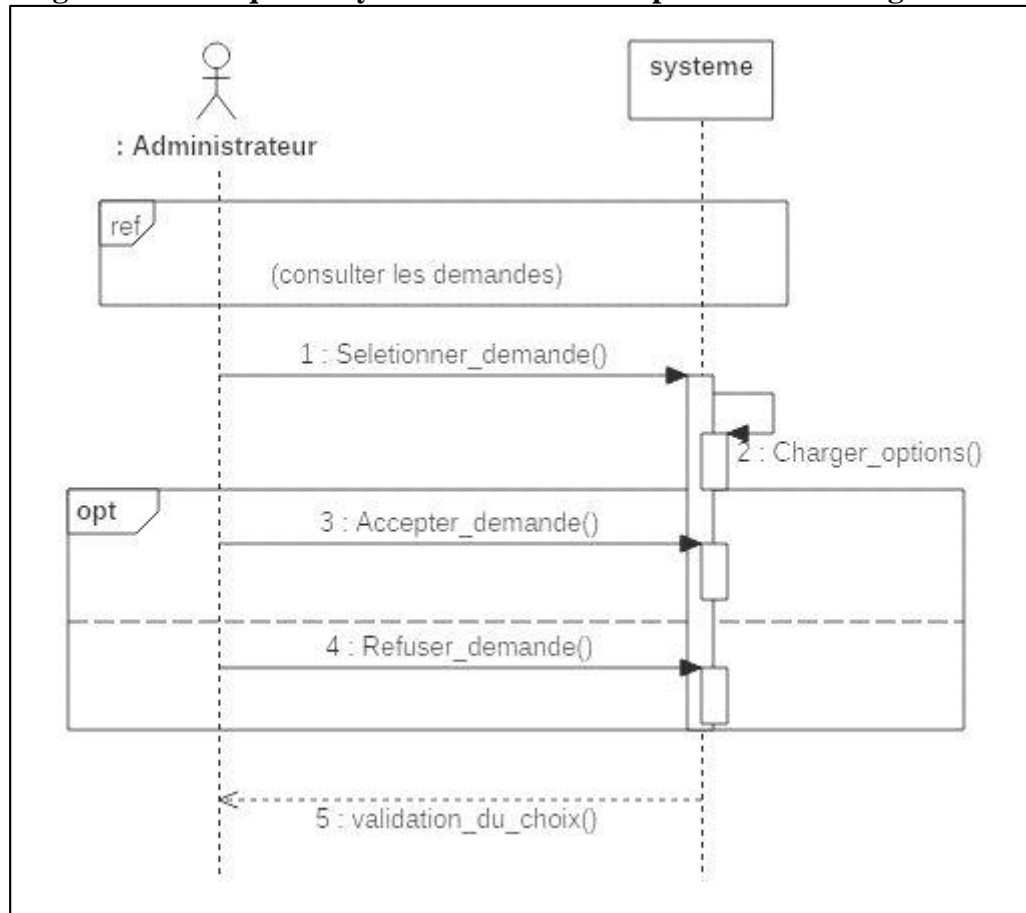


Figure42: Diagramme de séquence système du cas « Accepter / Refuser congé »

### C. Diagramme de séquence système du cas < Consulter les demandes des matériels >

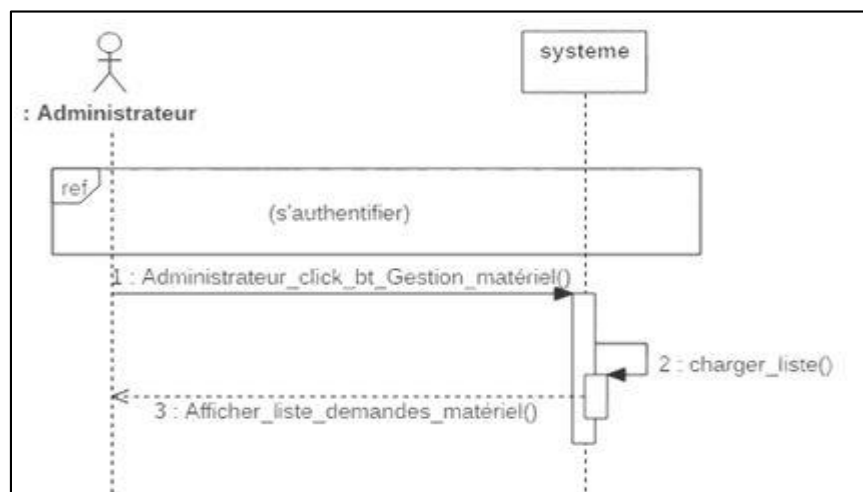
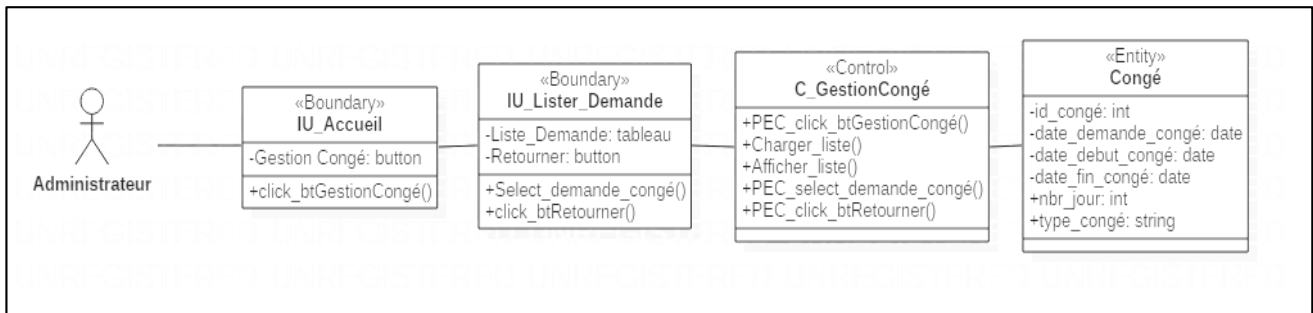


Figure 43 : Diagramme de séquence système du cas < Consulter les demandes des matériels >

## II.3.2. Diagrammes de classes participantes

Dans ce qui suit nous allons présenter les diagrammes de classes participantes des cas d'utilisation composant le sprint actuel.

### A. Diagramme de classe participantes du cas < gérer les congés >



**Figure 44 : Diagramme de classe de conception du cas < gérer les congés >**



### II.3.3 Diagrammes de séquences détaillés

Le diagramme de séquence détaillée permet de décrire les scénarios de chaque cas d'utilisation en interaction avec les objets tout mettant l'accent sur la chronologie des opérations. Les figures ci-après présentent les diagrammes de séquences détaillés de notre premier sprint.

#### A. Diagramme de séquences détaillées du cas « gérer les congé »

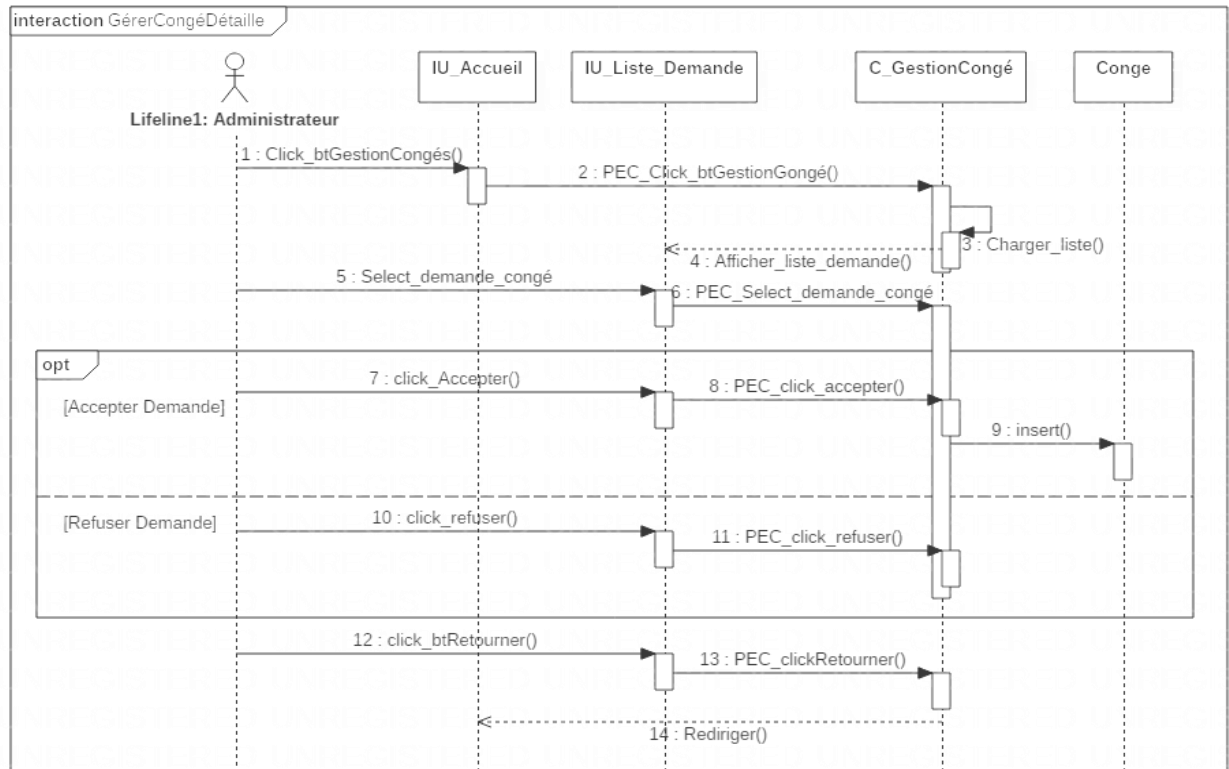


Figure 45 : Diagramme de séquences détaillées du cas « gérer les congé »

II.3.4. Diagramme de classe du sprint 2

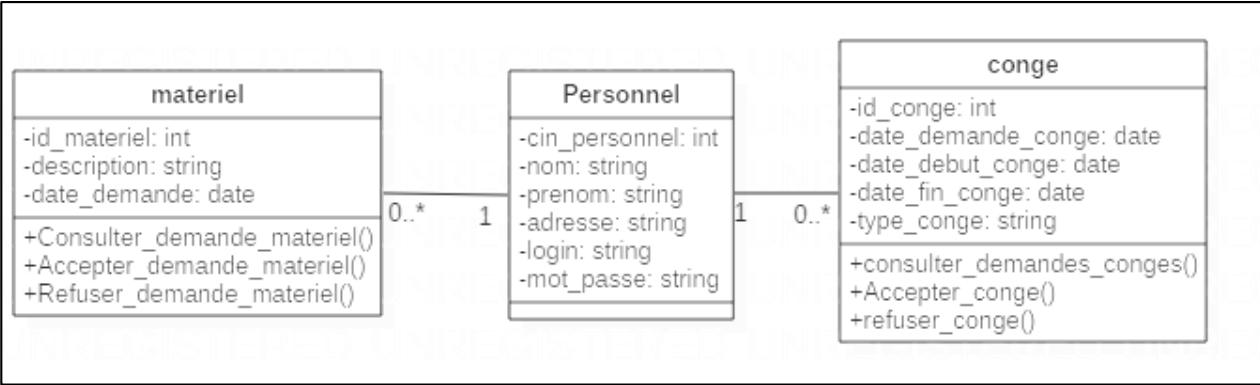


Figure 46 : Diagramme de classe du sprint 2

II.4. Réalisation

A. Interface consulter les demandes des matériels

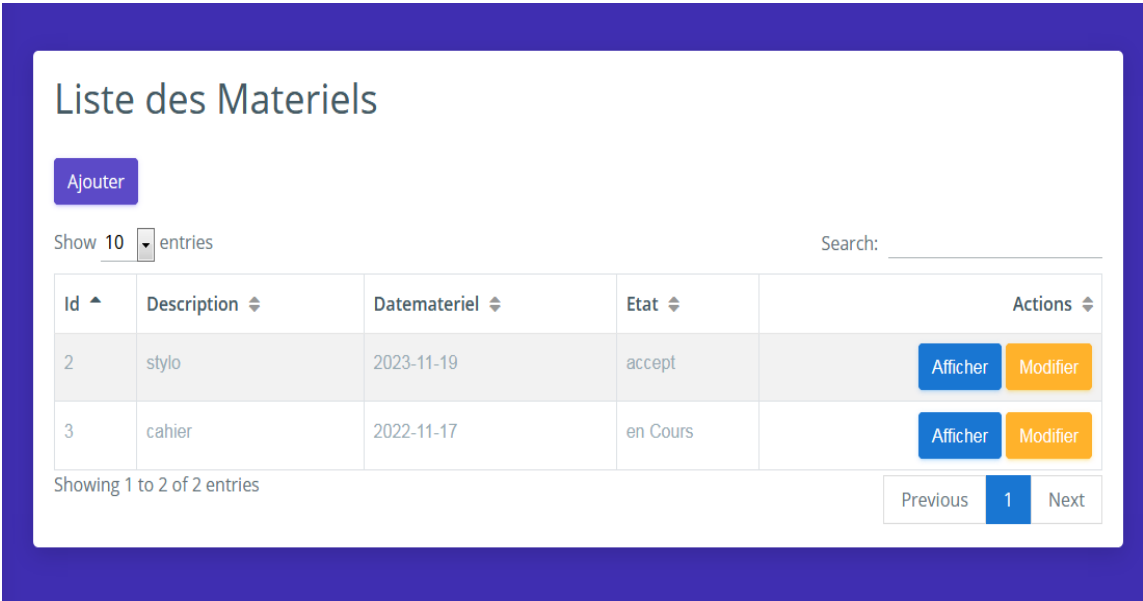


Figure 47 : interface « Consulter les demandes de matériel »

II.5. Test

Fonctionnalité	Démarche	Comportement	Résultat

		<b>attendu</b>	
Tester consulter les demandes des matériels	-Demander la page gérer matériel  -Cliquer sur consulter les demandes des matériels  -choisir une demande  -cliquer sur Accepter/refuser demande	-la page gérer matériel s'affiche  - la page consulter les demandes des matériels s'affiche  - le demande accepté ou bien refuser	Conforme

## Conclusion

A la fin de ce chapitre nous avons réussi à produire notre deuxième release .Dans la partie qui suit, nous allons détailler toutes les ressources que l'on a utilisées pour la réalisation de ce projet.



# Chapitre 5. La phase de clôture

## Introduction

La phase de clôture appelée également phase de réalisation, correspond au cinquième et ultime chapitre de notre rapport de de notre cycle de développement d'une application Scrum .

Dans cette partie, nous recenserons les différents outils technologiques et bureautiques ainsi que l'environnement de développement ayant été utilisé pour la mise en place de notre application.

## I. Environnement de développement

Cette section présente l'environnement matériel ainsi que l'environnement logiciel qui a été employés pour la mise en œuvre de notre application.

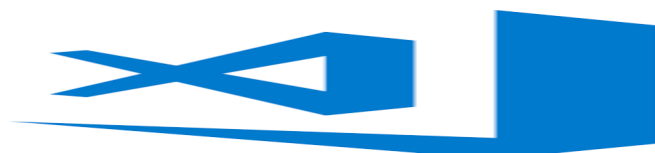
### I.1. Environnement matériel

Voici les caractéristiques techniques des machines que nous avons utilisées

<b>Ordinateur</b>	1
<b>Propriétaire</b>	Rezgani cyrine
<b>Processeur</b>	Core i5
<b>RAM</b>	6 GO
<b>Système d'exploitation</b>	Windows 7 64 bits

### I.2. Environnement logiciel

**Visual studio code** : est un éditeur de code open-source, gratuit et multiplateforme (Windows, Mac et Linux), développé par Microsoft, à ne pas confondre avec Visual Studio, l'IDE propriétaire de Microsoft.



**Wamp Server** : C'est est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMysqlAdmin pour gérer plus facilement vos bases de données.



**Composer** : est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin



**StarUml** : est un logiciel de modélisation UML, qui a été "cédé comme open source par son éditeur



### 1.2.3 Langages utilisés

**Symphony** : C'est est un ensemble de composants PHP ainsi qu'un Framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.





# Netographie

- [1]: <https://www.supinfo.com/articles/single/5131-imites-methodes-classiques-origine-methode-agile>
- [2] <https://www.supinfo.com/articles/single/3093-comparatif-methodes-agiles>
- [3] <https://medium.com/les-cahiers-agiles/au-del%C3%A0-de-l%C3%A9quipe-scrum-les-r%C3%B4les-dans-l-entreprise-agile-961d737a9e28>
- [4] <https://docplayer.fr/5330777-1-processus-de-developpement-methodologie-scrum.html>
- [5] <https://www.piloter.org/projet/methode/uml.htm>
- [6]: <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>
- [7] : <https://www.supinfo.com/articles/single/6437-fonctionnement-une-architecture-trois-tiers>
- [8] : [https://fr.wikipedia.org/wiki/Patron\\_d'architecture](https://fr.wikipedia.org/wiki/Patron_d'architecture)
- [9] : <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/MasterInfo/Cours/Swing/mvc.html>
- [10] : <https://www.nutcache.com/fr/blog/quest-ce-quun-backlog-scrum/>
- [11] : <http://www.aubryconseil.com/post/Release-mise-en-production-et-deploiement>
- [12] <https://www.supinfo.com/articles/single/5131-imites-methodes-classiques-origine-methode-agile>
- [13] <http://www.umangsoftware.com/fr/human-resource-management-system-industry-experience/>
- [14] [https://fr.wikipedia.org/wiki/Langage\\_de\\_modelisation](https://fr.wikipedia.org/wiki/Langage_de_modelisation)
- [16] <https://www.pentalog.fr/blog/sprint-0-element-projet-agile/>
- [17] <https://medium.com/@manurnx/utilisation-d-uml-pour-l-analyse-d-application-cd0936b3a39>
- [18] <https://agiliste.fr/introduction-methodes-agiles/#L-organisation-g-n-r%C3%A8le>