
DÉDICACE

“ *Je dédie ce travail ..*

À Texte ici

À Texte ici

À Texte ici

À Texte ici

À Texte ici

À Texte ici

”

- *Nom et prénom*

REMERCIEMENTS

Mes vifs remerciements s'adresse

.....

.....

.....

TABLE DES MATIÈRES

Dédicace	i
Remerciements	ii
Introduction générale	6
1 Étude Préalable	7
Introduction	8
1 Présentation de Première Consulting	8
2 Présentation du sujet	8
3 La gestion électronique des documents	9
3.1 Étape GED	9
3.2 Avantages GED	10
4 Analyse concurrentielle	10
4.1 Étude de solutions existantes	10
4.2 Bilan et synthèse	13
4.3 Solution proposée	13
5 Méthodologie adoptée	13
5.1 Méthodologies de développement de logiciel	13
5.1.1 Cycle en V	13
5.1.2 Méthode Scrum	15
5.1.3 Étude comparative	16
5.2 Langage de modélisation	17

6	Architecture de l'application	19
7	Étude Technologique	20
7.1	Développement Back-End	20
7.2	Développement Front-End	21
7.3	Logiciel de modélisation	22
	Conclusion	23
2	Planification et capture des besoins	24
	Introduction	25
1	Capture des besoins	25
1.1	Identification des acteurs	25
1.2	Les besoins fonctionnels	26
1.3	Les besoins non fonctionnels	27
1.4	Modèle de cas d'utilisation	28
2	Réalisation des maquettes	28
3	Pilotage du projet avec Scrum	30
3.1	Équipe et rôles	30
3.2	Backlog de produit	31
3.3	La planification de la release	32
	Conclusion	33
3	Étude et réalisation de la release	34
	Introduction	35
1	Étude et réalisation du Sprint1	35
1.1	Backlog Sprint1	35
1.2	Analyse et spécification fonctionnelle	36
1.2.1	Diagramme de cas d'utilisation du sprint1	36
1.2.2	Raffinement des cas d'utilisation du sprint1	36
1.3	Conception	43
1.3.1	Diagrammes de séquences de conception	43
1.3.2	Diagramme de classes du sprint1	47
1.4	Réalisation	47
1.5	Revue du sprint1	49
2	Étude et réalisation du Sprint2	50

2.1	Backlog Sprint2	50
2.2	Analyse et spécification fonctionnelle	50
2.2.1	Diagramme de cas d'utilisation du sprint2	50
2.2.2	Raffinement des cas d'utilisation du sprint2	51
2.3	Conception	55
2.3.1	Diagrammes de séquences	55
2.3.2	Diagramme de classes du sprint2	56
2.4	Diagramme d'état transition	56
2.5	Réalisation	56
2.6	Revue du sprint2	58
	Conclusion	58
	Références bibliographiques	59

TABLE DES FIGURES

1.1	Logo de la société Première Consulting	8
1.2	Les étapes de GED	10
1.3	Solution Alfresco	11
1.4	Solution Dokmee	11
1.5	Solution KnowledgeTree	12
1.6	cycle en v	13
1.7	Cycle en V	16
1.8	Historique d'UML	18
1.9	Architecture de l'application	20
1.10	20
1.11	20
1.12	21
1.13	21
1.14	21
1.15	21
1.16	22
1.17	22
1.18	22
1.19	22
1.20	23
1.21	23
1.22	23

2.1	Modèle de cas d'utilisation	28
2.2	Caption	29
2.3	Caption	29
2.4	Caption	29
2.5	Caption	29
2.6	Caption	30
2.7	Planification de la release	33
3.1	Diagramme de cas d'utilisation du sprint1	36
3.2	gérer les comptes	37
3.3	Raffinement du CU « gérer les catégories »	40
3.4	Raffinement du CU « gérer les départements »	42
3.5	Diagramme de séquences CU « S'authentifier »	43
3.6	Diagramme de séquences CU « Créer un compte »	44
3.7	Diagramme de séquences CU « chercher un compte »	44
3.8	Diagramme de séquences CU « Modifier les rôles d'accès un droit d'accès »	45
3.9	Diagramme de séquences CU « Créer une catégorie »	46
3.10	Diagramme de séquences CU « Supprimer une catégorie »	46
3.11	Diagramme de séquences CU « Créer un département »	47
3.12	Diagramme de classes du sprint1	47
3.13	Capture	48
3.14	Capture	48
3.15	Capture	48
3.16	Capture	48
3.17	Capture	49
3.18	Capture	49
3.19	Revue du sprint1	49
3.20	Diagramme de cas d'utilisation du sprint2	51
3.21	Raffinement du CU « gérer un document	51
3.22	Raffinement du CU « valider un document	53
3.23	Raffinement du CU « valider un document	54
3.24	Diagramme de séquences CU « créer un document »	55
3.25	Diagramme de séquences CU « valider un document »	55
3.26	Diagramme de séquences CU « aviser un document »	56

3.27 Diagramme de classes du sprint2	56
3.28 Diagramme d'état transition d'un document	56
3.29 Capture	56
3.30 Capture	57
3.31 Capture	57
3.32 Capture	57
3.33 Capture	57
3.34 Capture	58
3.35 Revue du sprint2	58

LISTE DES TABLEAUX

1.1	Comparatif entre V et SCRUM	17
2.1	Équipe et rôles	30
2.2	Backlog de produit	31
3.1	Backlog Sprint1	35
3.2	Description textuelle du CU « S'authentifier »	37
3.3	Description textuelle du CU « Créer un compte »	38
3.4	Description textuelle du CU « consulter un agent »	39
3.5	Description textuelle du CU « Créer une catégorie »	40
3.6	Description textuelle du CU « Supprimer une catégorie »	41
3.7	Description textuelle du CU « Créer un département »	42
3.8	Backlog Sprint2	50
3.9	Description textuelle du CU « créer un document »	52
3.10	Description textuelle du CU « indexer un document »	52
3.11	Description textuelle du CU « valider un document »	53
3.12	Description textuelle du CU « aviser un document »	54

LISTE DES ABRÉVIATIONS

- **CSS** = **Les feuilles de style en cascade**
- **CU** = **Cas d'utilisation**
- **ECM** = **....**
- **HTML5** = **HyperText Markup Language**
- **MVC** = **Modèle-Vue-Contrôleur**
- **MYSQL** = **My Structured Query Language**
- **PHP** = **Hypertext Preprocessor**
- **UML** = **Le Langage de Modélisation Unifié**

INTRODUCTION GÉNÉRALE

T^{exte}

CHAPITRE 1

ÉTUDE PRÉALABLE

Introduction	8
1 Présentation de Première Consulting	8
2 Présentation du sujet	8
3 La gestion électronique des documents	9
3.1 Étape GED	9
3.2 Avantages GED	10
4 Analyse concurrentielle	10
4.1 Étude de solutions existantes	10
4.2 Bilan et synthèse	13
4.3 Solution proposée	13
5 Méthodologie adoptée	13
5.1 Méthodologies de développement de logiciel	13
5.2 Langage de modélisation	17
6 Architecture de l'application	19
7 Étude Technologique	20
7.1 Développement Back-End	20
7.2 Développement Front-End	21
7.3 Logiciel de modélisation	22
Conclusion	23

Introduction

...

1 Présentation de Première Consulting

Première Consulting est un cabinet d'étude, de Conseil et de Formation. Dès sa création il a fait le choix essentiel d'offrir à ses clients du conseil et de la formation à forte valeur ajoutée, basé sur une approche pragmatique, fortement orienté résultat.



FIGURE 1.1 – Logo de la société Première Consulting

Première Consulting s'appuie sur un réseau de collaborateurs, d'experts et de consultants hautement qualifiés bénéficiant d'une expérience significative dans le domaine afin de répondre efficacement aux besoins des clients.

En effet, les clients attendent de nous des solutions opérationnelles et pragmatiques pour aller droit au but que des théories sans résultats.

Convaincu de la pertinence de son positionnement, Première Consulting Sarl s'inscrit dans une logique de fort développement et a pour ambition de devenir un des leaders indépendants du conseil en Tunisie et Maghreb.

2 Présentation du sujet

Tenant compte de l'importance croissante des nouvelles technologies et de l'évolution importante des services que l'informatique offre à ses utilisateurs, Première Consulting décide de migrer ses fonctions classiques en gestion de documents vers une nouvelle dimension qui se base sur l'outil informatique. Notre sujet consiste donc en le développement d'un système automatisé de **gestion électronique de documents** au sein de Première Consulting dans le but de rendre les documents disponibles aux collaborateurs concernés. Les objectifs du système visé à travers ce sujet peuvent être résumés comme suit :

- * Réduction des volumes et des coûts de stockage;
- * Gain du temps qui est souvent perdu dans la recherche des documents mal classés;

- * Intégration des informations administratives, techniques dans des systèmes cohérents;
- * Archivage et sécurisation des documents.

3 La gestion électronique des documents

La gestion électronique des documents désigne le processus de gestion des documents dans une organisation. Ce procédé comprend l'acquisition des documents, **l'indexation et le classement, le stockage d'informations, l'accès et la diffusion des documents.**

La GED nous permet d'exploiter l'ensemble de ces documents au format numérique et nous simplifie le travail au quotidien. Le contrôle des procédures se retrouve renforcé par la mise en œuvre de « **workflows** », circuits de validation des documents.

Cette solution permet l'harmonisation des processus dans l'entreprise, la centralisation des informations, l'amélioration de la collaboration entre les services et facilite l'accès à l'information aux collaborateurs. Cela concerne surtout les archives courantes, c'est-à-dire les documents d'utilisation quotidienne ou régulière, qui doivent pouvoir circuler, être modifiés ou même supprimés si besoin.

3.1 Étape GED

- * **La numérisation :** Cette opération consiste à scanner un document papier puis à appliquer un traitement de reconnaissance de texte pour le transformer en document numérique, via un logiciel OCR.
- * **Le traitement :** Cette étape est consacrée à la classification, au référencement et à l'indexation. Il s'agit de décrire le document (type de fichier, date de création, responsable de la création) et son contenu (recherche en texte intégral) afin de faciliter son exploitation.
- * **Le stockage des données :** Il doit être adapté au volume des documents et doit permettre un temps d'accès rapide. Des copies de sauvegarde sont nécessaires car il existe toujours des risques d'accidents, comme les incendies ou les pannes de matériel.
- * **L'archivage :** Il concerne particulièrement les documents qui doivent être légalement conservés, mais qui ne servent pas dans les opérations quotidiennes.

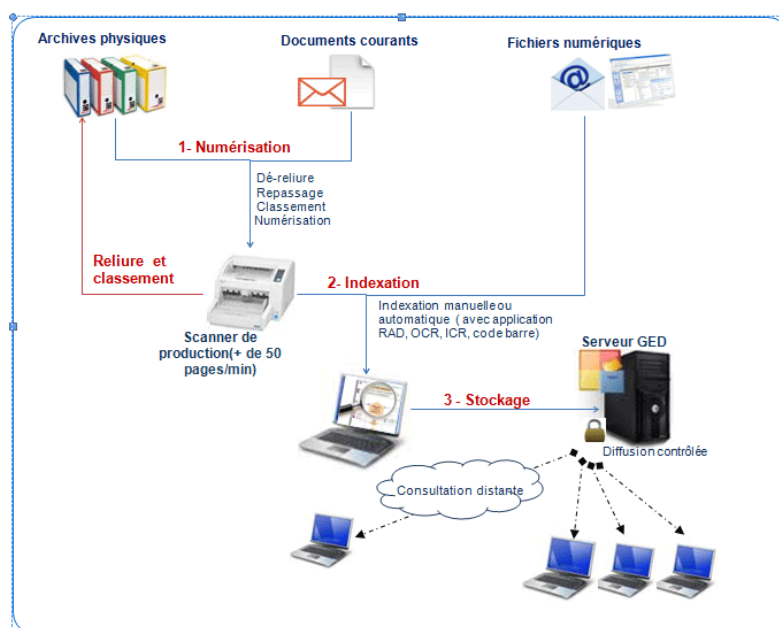


FIGURE 1.2 – Les étapes de GED

3.2 Avantages GED

Les avantages de GED sont :

- * **Le gain de temps considérable** : pas de file d'attente pour la recherche, pas de classement ;
- * **La sécurité des données** : il n'y a plus de risque que les données se perdent. L'accès est sécurisé et vous pouvez contrôler le droit d'accès ;
- * **Les coûts liés à la diffusion sont réduits** ;
- * **Le contrôle des versions** : il est possible de récupérer les dossiers en cas de fausse manipulation ;
- * **Le gain d'espace** : libération de l'espace dans les locaux en éliminant les documents en papier ;
- * **L'efficacité au travail**.

4 Analyse concurrentielle

4.1 Étude de solutions existantes

....

- * **Alfresco** : Alfresco est l'une des solutions GED open source les plus connues, disponible sous deux versions : Community (licence libre et gratuite) et Entreprise (avec abonnement).

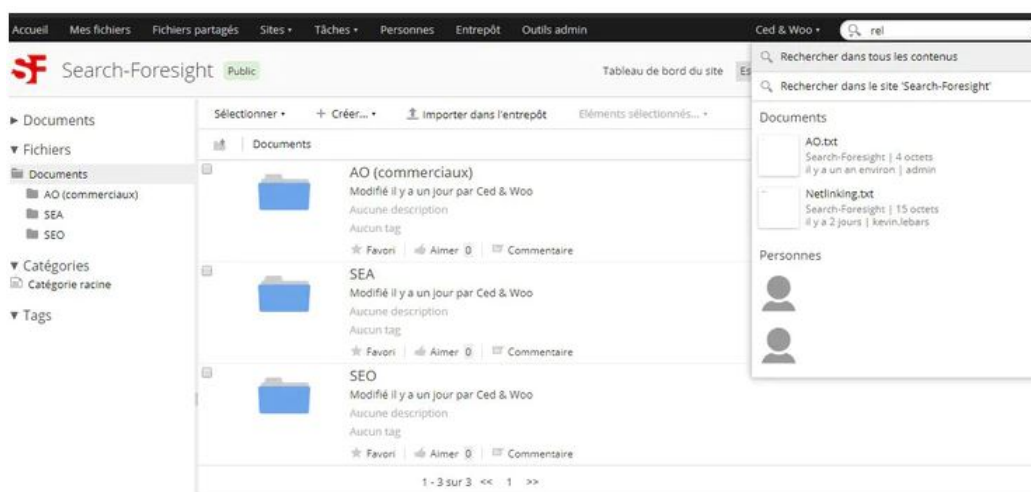


FIGURE 1.3 – Solution Alfresco

Cette solution propose :

- * La numérisation de documents statiques pour une exploitation des informations pertinentes;
 - * L'organisation de des documents;
 - * Synchronisation des dossiers facilitant le travail offline;
 - * Une analyse basée sur l'IA et l'apprentissage automatique pour l'extraction des données.
- * **Dokmee** Dokmee propose également une solution GED open source. L'outil professionnel offre des fonctionnalités particulièrement poussées de sécurité, de confidentialité et d'intégration.

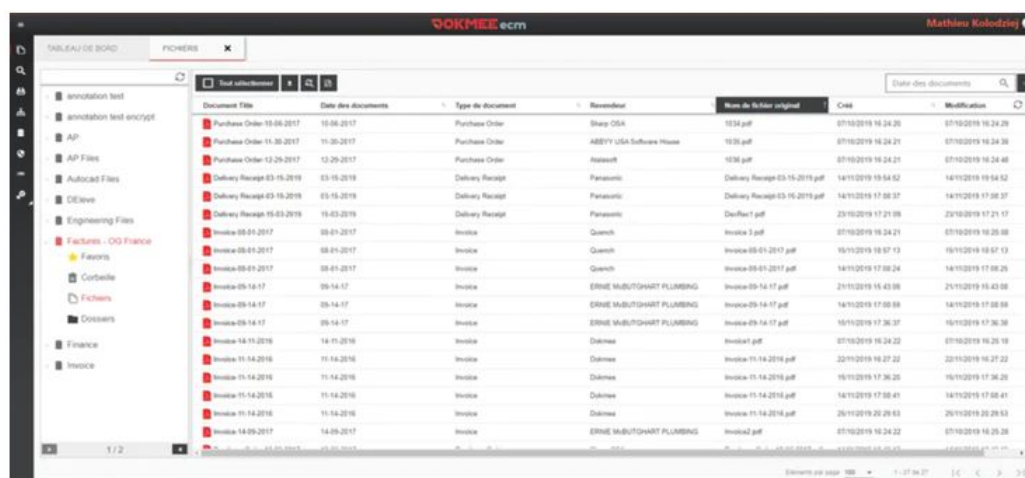


FIGURE 1.4 – Solution Dokmee

Dokmee vous permet de :

- * Organiser et gérer les fichiers;
 - * Accéder à un bureau virtuel;
 - * Automatiser les processus et la saisie de données;
 - * Partager les fichiers et collaborer simultanément;
 - * Générer des rapports.
- * **KnowledgeTree** : KnowledgeTree est une autre référence en gestion de documents open source, très appréciée des services marketing et commerciaux.

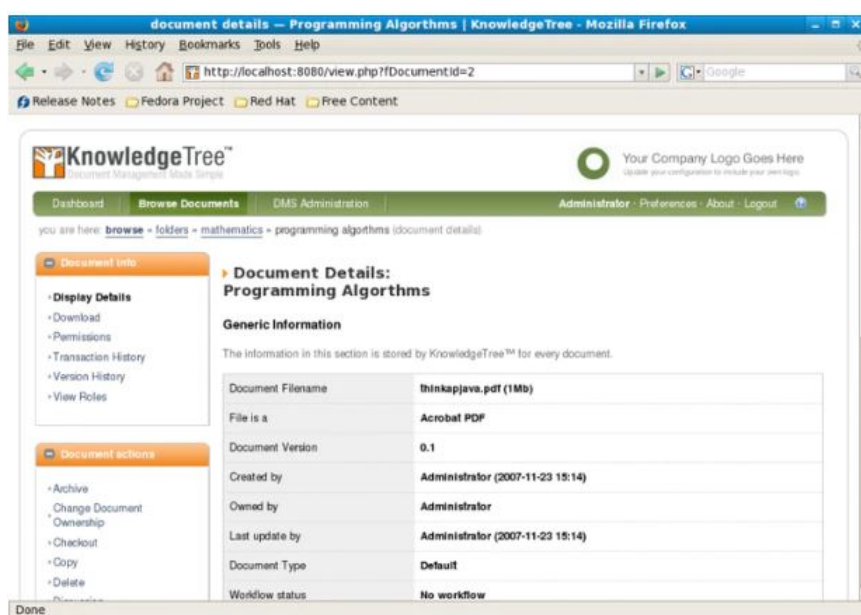


FIGURE 1.5 – Solution KnowledgeTree

Comme beaucoup de logiciels open source, il est disponible en version Enterprise (payante) et Community (gratuite), et permet aux utilisateurs de sécuriser, partager, suivre et gérer des documents.

Ses fonctionnalités (pour la version Enterprise) incluent :

- * Plusieurs modules permettant une bonne intégration dans l'environnement bureau-tique, notamment avec MS Office;
- * La comparaison de documents;
- * La signature électronique;
- * La recherche dans le contenu du document;
- * La gestion du flux de travail.

4.2 Bilan et synthèse

Texte...

4.3 Solution proposée

....

* ;

* ;

* ;

* ;

* ;

5 Méthodologie adoptée

...

5.1 Méthodologies de développement de logiciel

...

5.1.1 Cycle en V

Le cycle en V se définit comme un modèle de gestion de projets composé d'une phase descendante, puis d'une phase ascendante. Issu du modèle en cascade, ou waterfall model, il implique le même principe de gestion séquentielle et linéaire. le modèle de cycle en V est figuré comme le montre la figure 1.6.

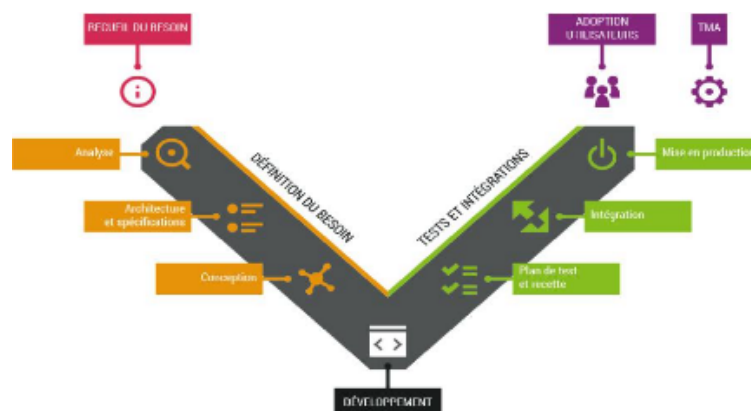


FIGURE 1.6 – cycle en v

Nous pouvons y distinguer 3 grandes parties : La phase de conception, la phase de réalisation (codage) et la phase de validation.

✱ **La phase de conception se réduit à 2 étapes :**

- ✱ **Les spécifications fonctionnelles**, qui représentent l'ensemble des besoins du client et/ou définissent ce que doit faire le produit fini;
- ✱ **spécifications techniques**, qui détaillent comment le produit va être réalisé techniquement.

✱ **La phase de réalisation :**

- ✱ **Codage**, c'est la phase de réalisation à proprement parler, pendant laquelle sont développées des briques qui sont ensuite assemblées pour créer le produit fini;
- ✱ **Tests unitaires**, assurent que les briques logicielle modélisée et codée durant les étapes précédentes respectent de manière individuelle la cahier des charges.

✱ **La phase de validation contient 3 étapes :**

- ✱ **Les tests d'intégration**, pendant lesquels on vérifie que l'intégralité du produit est valide techniquement;
- ✱ **Les tests de validation**, qui sont un mélange de tests techniques et fonctionnels, et sur lesquels le client se base souvent pour décider du lancement du produit;
- ✱ **La recette**, qui est utilisée pour vérifier que le produit est valide par rapport aux spécifications fonctionnelles, mais qui a tendance à n'intervenir qu'après la mise en production.

Ce modèle présente de multiples avantages :

- ✱ Il évite les allers-retours durant le **cycle de vie** du projet : si des problèmes sont rencontrés, chaque étape de la partie ascendante peut s'appuyer sur la documentation produite lors de l'étape de la partie descendante correspondante (voir l'illustration ci-dessus);
- ✱ Il apporte plus de précisions durant sa phase de test.

D'autre part, le cycle en V paraît assez intuitif et simple à mettre en pratique :

- ✱ Il nécessite juste quelques réunions régulières pour le pilotage du projet et le suivi budgétaire. Quant à la documentation, elle peut être créée à partir de templates déjà existants;
- ✱ Il requiert moins de formation et de pré-requis pour son application que d'autres méthodes telles que Scrum;

- * Il s'adapte facilement aux projets impliquant des structures multisites, contrairement aux modèles de gestion de projet nécessitant des réunions quotidiennes.

Ce modèle présente aussi quelques inconvénients :

- * Il **tolère mal les changements**. De par sa construction séquentielle et linéaire, le retour en arrière est impossible. Pourtant, il n'est pas rare de rencontrer des problèmes conceptuels lors de la phase de réalisation et de validation. Faut-il alors reprendre le cycle en V depuis le début ? Ou attendre le prochain cycle en V pour procéder aux changements ?
- * Il **nécessite une documentation importante**, perçue par certains comme une lourde perte de temps. De plus, si elle s'avère imparfaite, nous ne pouvons pas la rectifier lors d'étapes intermédiaires prévues à cet effet.
- * Il **s'adapte difficilement à certains types de projets**. Le développement logiciel, par exemple, supporte difficilement le manque de réactivité et la séparation entre la conception et la réalisation des activités.
- * Il **peut être long**. On court alors le risque que le produit dans sa version finale ne soit pas adapté aux évolutions apparues au cours de sa conception. C'est là tout le paradoxe d'un modèle qui n'admet pas le changement, alors que sa durée ne permet pas de l'éviter.

5.1.2 Méthode Scrum

Scrum signifie « mêlée » en anglais. Et pour cause, la méthode Scrum s'apparente à une équipe de rugby avançant ensemble, et pouvant se réorienter au fil de sa progression en fonction des besoins.

Mais à quoi ressemble une gestion de projet avec la méthode Scrum ?

Premièrement, Scrum entraîne deux formes de découpage :

- * Le **découpage du temps** : la réalisation du projet est découpée en sprints (ou itérations) de deux à quatre semaines. Au terme de chaque sprint, un produit est livré, même si le projet global n'est pas terminé. Le client peut alors faire ses retours, proposer des axes d'amélioration, etc. ;
- * Le **découpage des tâches** : les « tâches » deviennent des user stories. Une user story énonce un besoin auquel une réponse devra être apportée au cours du sprint.

Deuxièmement, Scrum implique la présence de trois grands rôles :

- * Le **Product Owner** : sorte de chef de projet, il porte la vision du produit. Par conséquent, il établit les priorités, définit les spécificités fonctionnelles et les valide. Ce rôle peut être

endossé par le client lui-même ;

- * Le **Scrum Master** : il s'assure de faire respecter la méthode et les valeurs Scrum. Il favorise la communication et le bon déroulement du travail ;
- * L'**équipe** : elle accomplit les tâches, réalise le produit.

Troisièmement, Scrum repose sur une série de réunions rythmant les **sprints** :

- * La **planification** : les user stories sont présentées, et le prochain sprint s'organise ;
- * La **daily** : l'équipe se réunit quotidiennement pour échanger sur l'avancement du projet, le sprint et les éventuelles difficultés rencontrées ;
- * La **review** : une démonstration des fonctionnalités livrables s'effectue au terme de chaque sprint. C'est le moment de recueillir le feedback du client ;
- * La **rétrospective** : dans une démarche d'**amélioration continue**, l'équipe se réunit pour faire le bilan du sprint et se perfectionner pour le suivant.

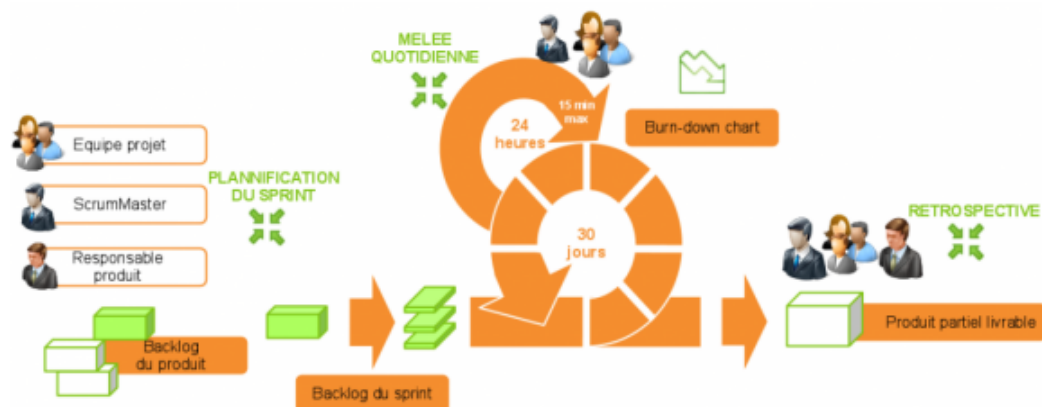


FIGURE 1.7 – Cycle en V

5.1.3 Étude comparative

Bien que la méthode SCRUM semble plus efficace, elle n'est pas adaptée à tout type de projet. La comparaison des deux méthodes permet donc de cibler la bonne méthode pour la conception et la réalisation du sujet.

TABLE 1.1 – Comparatif entre V et SCRUM

Thème	Cycle en v	SCRUM
Cycle de vie	Phases séquentielles	Processus itératif
Livraison	À la fin de la réalisation de toutes les fonctionnalités → livraison tardive	Utilisation partielle du produit suite à la priorisation des besoins → livraison plus rapide
Contrôle Qualité	À la livraison finale (fin du cycle de développement) → effet tunnel	À chaque livraison partielle au client
Spécification	Pas de changement possible sans revenir à la phase de spécifications et repasser par toutes les autres phases → délais et coûts supplémentaires	Spécifications plus souples en ajoutant/modifiant les fonctionnalités aux sprints suivants qui n'étaient pas prévues au départ → principal atout de la méthode Agile
Planification	Plans détaillés basés sur des exigences stables définies dès le début du projet	Planification adaptative et ajustements si nécessaires en fonction des nouvelles demandes
Équipe	Intervention uniquement dans la phase de développement, pas de vision globale du projet	Engagements, échanges et prises de décisions collectives par l'équipe
Documentation	Quantité importante	Strict nécessaire

....

5.2 Langage de modélisation

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet (figure 1.8).

Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

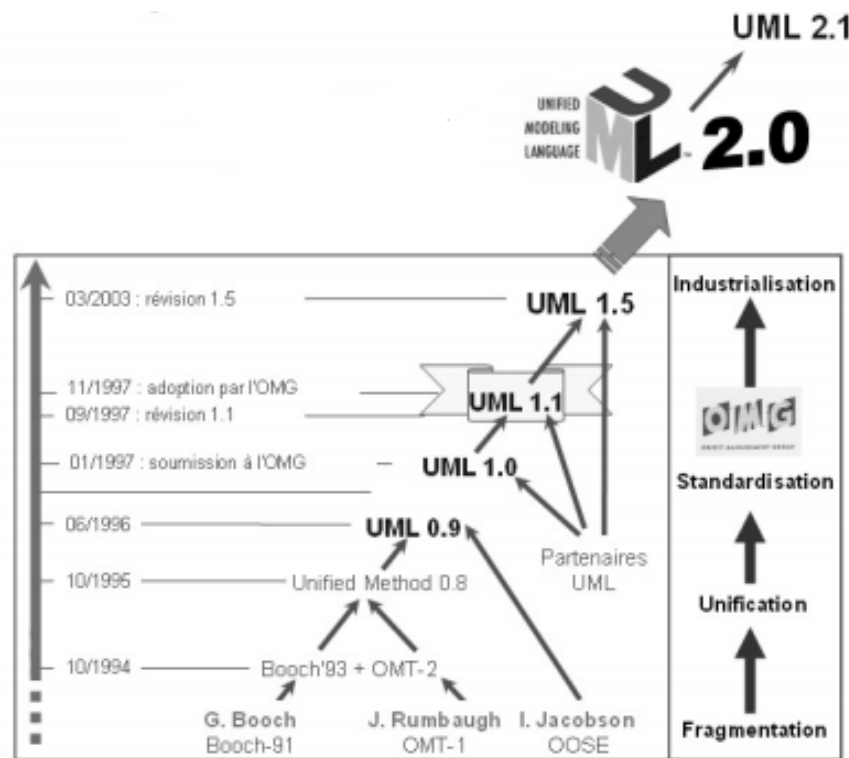


FIGURE 1.8 – Historique d'UML

UML2 s'articule autour de treize types de diagrammes , chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel . Ces types de diagrammes sont répartis en **deux grands groupes** :

*** Six diagrammes structurels**

- * Diagramme de classes – Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.
- * Diagramme d'objets - Il montre les instances des éléments structurels et leurs liens à l'exécution.
- * Diagramme de packages- Il montre l'organisation logique du modèle et les relations entre packages.
- * Diagramme de structure composite – Il montre l'organisation interne d'un élément statique complexe.
- * Diagramme de composants – Il montre des structures complexes, avec leurs interfaces fournies et requises.

- * Diagramme de déploiement – Il montre le déploiement physique des « artefacts » sur les ressources matérielles.

*** Sept diagrammes comportementaux**

- * Diagramme de cas d'utilisation - Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.
- * Diagramme de vue d'ensemble des interactions - Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.
- * Diagramme de séquence - Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.
- * Diagramme de communication - Il montre la communication entre objets dans le plan au sein d'une interaction.
- * Diagramme de temps – Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.
- * Diagramme d'activité - Il montre l'enchaînement des actions et décisions au sein d'une activité.
- * Diagramme d'états – Il montre les différents états et transitions possibles des objets d'une classe.

UML utilise des méthodes objet en présentant un langage de description commun. Il permet grâce à un ensemble de diagrammes très clairs, de représenter l'architecture et les fonctions des systèmes informatiques complexes et prennent en compte le concept utilisé et l'implémentation qui en résulte.

UML est caractérisé par :

- * Sa notation graphique qui permet d'exprimer visuellement une solution objet , ce qui facilite la comparaison et l'évaluation de solutions.
- * L'aspect formel de sa notation qui limite les ambiguïtés et les incompréhensions.

6 Architecture de l'application

Texte

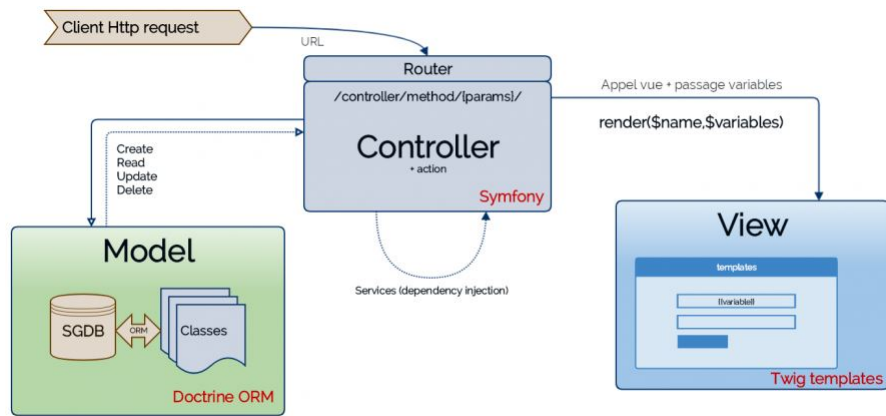


FIGURE 1.9 – Architecture de l'application

7 Étude Technologique

7.1 Développement Back-End

* Symfony 5.4 Release



FIGURE 1.10

* Symfony CLI



FIGURE 1.11

* View composer



FIGURE 1.12

*** XAMPP**



FIGURE 1.13

*** Visual Studio Code**



FIGURE 1.14

7.2 Développement Front-End

*** Bootstrap**



FIGURE 1.15

*** CSS3**



FIGURE 1.16

*** JQuery**



FIGURE 1.17

*** HTML5**



FIGURE 1.18

7.3 Logiciel de modélisation

*** Visual Paradigm for UML**



FIGURE 1.19

*** Pacestar UML Diagrammer**



FIGURE 1.20

*** Enterprise Architect**



FIGURE 1.21

*** Balsamiq Wireframes**



FIGURE 1.22

Conclusion

...

CHAPITRE 2

PLANIFICATION ET CAPTURE DES BESOINS

Introduction	25
1 Capture des besoins	25
1.1 Identification des acteurs	25
1.2 Les besoins fonctionnels	26
1.3 Les besoins non fonctionnels	27
1.4 Modèle de cas d'utilisation	28
2 Réalisation des maquettes	28
3 Pilotage du projet avec Scrum	30
3.1 Équipe et rôles	30
3.2 Backlog de produit	31
3.3 La planification de la release	32
Conclusion	33

Introduction

Dans ce chapitre , nous débutons par établir la liste des exigences fonctionnelles et non fonctionnelles du système. Nous énumérons par la suite les différentes étapes de la démarche adoptée afin d'aboutir à la construction du modèle des cas d'utilisation.

Une fois le modèle est prêt, et suite à des négociations avec les différents intervenants du projet nous planifions les sprints et nous les priorisons selon les attentes et les besoins du client.

En définitive, nous exposons le plan de la release présentant les différentes séquences de sprints à venir en matière de semaine.

1 Capture des besoins

Nous allons procéder selon la méthode UML qui consiste à recenser et modéliser les différents processus métiers afin de migrer facilement vers une architecture objet d'un point de vue statique et dynamique. Cette analyse présente une abstraction totale étant indépendante de toute technologie ou implémentation.

La spécification des besoins va nous permettre d'avoir une meilleure approche des utilisateurs, des fonctionnalités et de la relation entre les deux. Elle est sous forme de cas d'utilisation. Pour cela nous allons procéder ainsi :

- * Identification des acteurs du nouveau système;
- * Identification des cas d'utilisations;
- * Regroupement en modèle de cas d'utilisation .

1.1 Identification des acteurs

Dans UML nous n'utilisons pas le terme d'utilisateurs mais d'acteurs. Un acteur d'un système est une entité externe à ce système qui interagit (saisie de données, réceptions d'informations,...) avec lui. Les acteurs permettent de cerner l'interface que le système va offrir à son environnement.

Un acteur regroupe plusieurs utilisateurs qui ont le même rôle. Et pour trouver un acteur il faut identifier les différents rôles que vont devoir jouer ses utilisateurs.

Les différents acteurs du système étudié sont :

- * **Administrateur;**
- * **Approbateur;**

*** Validateur;**

*** Rédacteur.**

1.2 Les besoins fonctionnels

Pour chaque acteur identifié précédemment, nous recherchons les différentes intentions « métier » selon lesquelles il utilise le système.

*** Administrateur :**

- * Authentification;
- * Gestion des comptes (rédacteur, approbateur et validateur);
- * Gestion des rôles d'accès;
- * Gestion des départements;
- * Gestion des catégories;
- * Gestion des documents;
- * Indexation des documents;
- * Validation des documents;
- * Traitement des document;
- * Archivage des documents;
- * Diffusion des documents;
- * Gestion la listes de diffusion;
- * Gestion des processus.

*** Approbateur :**

- * Authentification;
- * Gestion des documents;
- * Indexation des documents;
- * Validation des documents;
- * Traitement des document;
- * Archivage des documents;
- * Diffusion des documents;

*** Validateur :**

- * Authentification;
- * Gestion des documents;
- * Indexation des documents;
- * Validation des documents;

*** Rédacteur :**

- * Authentification;
- * Gestion des documents;
- * Indexation des documents;

1.3 Les besoins non fonctionnels

A part les exigences fonctionnels, les exigences non-fonctionnelles sont les besoins qui caractérisent les propriétés (qualités) désirées du système telles que :

*** Besoin d'utilisabilité**

- * Font référence aux aspects généraux de l'interface utilisateur;

*** Besoins de performance**

- * Le système doit pouvoir gérer un nombre importants de comptes utilisateurs;
- * Le système doit supporter un nombre importants de connexions simultanées.
- * Le système doit pouvoir comprendre un nombre importants de documents;
- * Les opérations de recherche ne doit prendre plus de 2 secondes.

*** Besoins de disponibilité/fiabilité**

- * S'assurer que le système doit être toujours disponible sauf pendant les périodes de maintenance.

*** Besoins de sécurité**

- * Définir les niveaux d'accès possibles au système pour les utilisateurs du système;
- * Besoins d'établissement d'une connexion sécurisés via des codes d'accès;
- * Besoins de mot de passe (longueur, caractères spéciaux);

*** Besoins matériels**

- * Définissent les configurations matérielles minimales nécessaires au fonctionnement du système.

1.4 Modèle de cas d'utilisation

Le modèle des cas d'utilisation général donné par la figure 2.1 présente les principales fonctionnalités que le système peut offrir.



FIGURE 2.1 – Modèle de cas d'utilisation

2 Réalisation des maquettes

La réalisation des maquettes est une étape déterminante dans la création d'un site internet. Pour que le site web soit agréable, il faut naturellement soigner le design de l'interface graphique, mais surtout, pour que le site internet soit efficace, il faut construire une mise en page intelligente et adaptée à nos objectifs et à nos contenus.

La réalisation des maquettes doit passer par plusieurs étapes, du design fonctionnel en noir et blanc, au design graphique intégrant votre identité et vos couleurs.

Le but est de provoquer des retours de la part des utilisateurs et de bien cerner les fonctionnalités du système attendues et observables.

Dans ce qui suit, nous allons donner un aperçu sur quelques maquettes réalisées.

*



FIGURE 2.2 – Caption

*



FIGURE 2.3 – Caption

*



FIGURE 2.4 – Caption

*



FIGURE 2.5 – Caption

*



FIGURE 2.6 – Caption

3 Pilotage du projet avec Scrum

3.1 Équipe et rôles

Comme le montre le tableau 2.1, L'équipe Scrum est composée de trois rôles : le Scrum Master, Product Owner et l'équipe de développement.

Tous les membres de l'équipe Scrum, indépendamment de leurs rôles, sont situés au même niveau, sans rapport hiérarchique entre eux, et agissent pour la réussite du projet.

TABLE 2.1 – Équipe et rôles

Rôle	Personne	Tâches demandées
Scrum Master	Mme Bouaziz Amani	<ul style="list-style-type: none">→ Animer et faciliter le travail de l'équipe de développement;→ Acquitter de toutes les tâches administratives et s'assurer que la méthodologie Scrum est correctement appliquée.
Product Owner	Bouaziz Amani	<ul style="list-style-type: none">→ Participer à l'élaboration des besoins des clients;→ Porter la vision client du produit final;→ Avoir la charge du backlog produit;→ Prendre la décision des priorités à donner aux développements des différentes user stories.

Équipe	Mlle Jlassi Molka Mlle Msolli Rihab	→ Mettre en œuvre les solutions techniques; → Réaliser les développements; → Travailler de façon incrémentale; → Livrer une partie du produit final utilisable et testable à la fin de chaque sprint ou itération.
--------	--	---

3.2 Backlog de produit

Comme le montre le tableau 2.2, Le backlog Scrum est destiné à recueillir tous les besoins du client que nous devons réaliser. Il contient donc la liste des fonctionnalités nécessitant notre intervention.

Tous les éléments inclus dans le backlog Scrum sont classés par priorité indiquant l'ordre de leur réalisation. Cet ordonnancement est effectué en fonction de la valeur ajoutée que les fonctionnalités apportent au produit.

TABLE 2.2 – Backlog de produit

En tant que		Je voudrais (User Story)	Priorité
1	Administrateur	gérer les comptes	Haute
2	Administrateur	modifier les rôles d'accès	Moyenne
3	Administrateur	gérer les catégories	Haute
4	Administrateur	gérer les départements	Haute
5	Administrateur	gérer les listes de diffusion	Haute
6	Administrateur	gérer les processus	Haute
7	Administrateur Approbateur Valideur Rédacteur	m'authentifier	Haute
» Page suivante			

TABLE 2.2 – Backlog de produit

En tant que		Je voudrais (User Story)	Priorité
8	Administrateur Approbateur Valideur Rédacteur	gérer son compte	Basse
9	Administrateur Approbateur Valideur Rédacteur	Gérer un document	Haute
10	Administrateur Approbateur Valideur Rédacteur	indexer un document	Haute
11	Administrateur Approbateur Valideur	valider les documents	
12	Administrateur Approbateur	aviser un document	Haute
13	Administrateur Approbateur	diffuser un document	Haute
14	Administrateur Approbateur	archiver un document	Haute

3.3 La planification de la release

Un plan de release schématisé par la figure 2.7 est une séquence de sprints à venir, avec une vision du contenu prévu (les éléments de backlog de produit) de ces sprints.

Les sprints sont présentés de façon séquentielle de gauche à droite, avec pour chacun son numéro, ses buts et les dates de réalisation prévues.

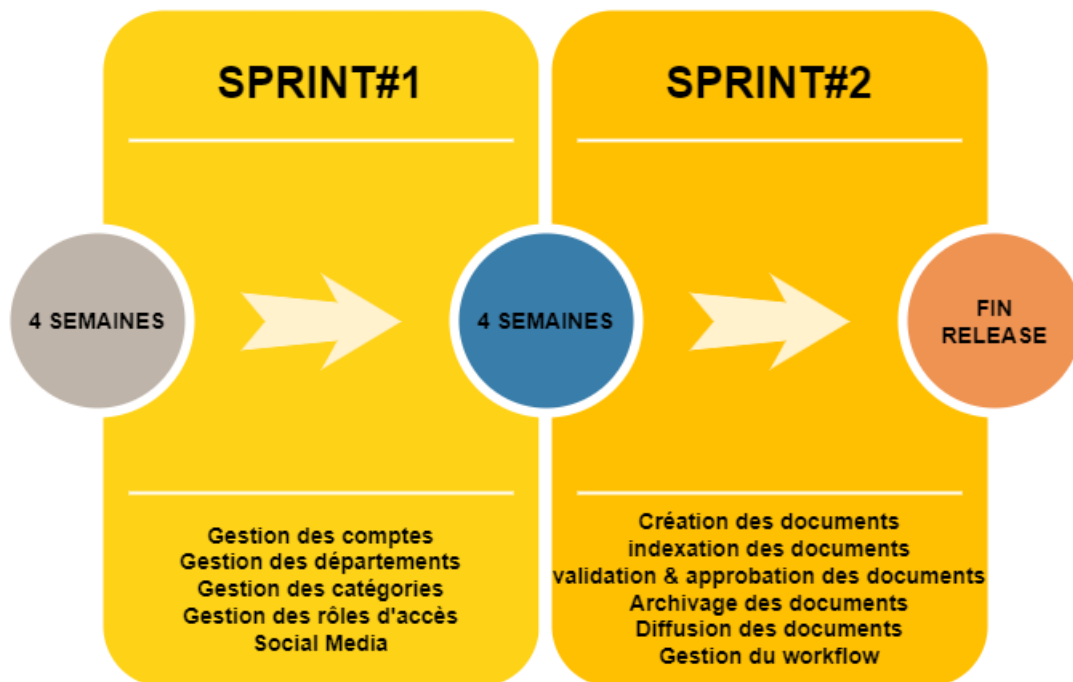


FIGURE 2.7 – Planification de la release

Conclusion

Dans ce chapitre, nous avons pu analyser et énoncer les besoins du système. Nous avons identifié les différents acteurs du système d'une part, les tâches de chacun d'autre part et éventuellement leurs associés des priorités.

Par ailleurs, nous avons projeté quelques prototypes permettant d'avoir une vue approximative autour des interfaces que nous désirons réaliser.

Nous passons maintenant à présenter l'environnement logiciel et matériel, ainsi que l'architecture adoptée pour la réalisation de la solution.

CHAPITRE 3

ÉTUDE ET RÉALISATION DE LA RELEASE

Introduction	35
1 Étude et réalisation du Sprint1	35
1.1 Backlog Sprint1	35
1.2 Analyse et spécification fonctionnelle	36
1.3 Conception	43
1.4 Réalisation	47
1.5 Revue du sprint1	49
2 Étude et réalisation du Sprint2	50
2.1 Backlog Sprint2	50
2.2 Analyse et spécification fonctionnelle	50
2.3 Conception	55
2.4 Diagramme d'état transition	56
2.5 Réalisation	56
2.6 Revue du sprint2	58
Conclusion	58

Introduction

Ce chapitre est consacré à l'étude et la réalisation du premier sprint.

1 Étude et réalisation du Sprint1

1.1 Backlog Sprint1

Nous projetons l'ensemble des fonctionnalités (user-stories) écrites par le Product Owner et pris en charge par l'équipe de développement pendant le sprint en cours.

Nous estimons un travail quotidien de six heures pendant quatre semaines pour la conception et la réalisation de ce premier sprint. Nous dressons la liste des fonctionnalités en tenant compte de la quantité de travail (en heures) estimée pour chaque user-story.

TABLE 3.1 – Backlog Sprint1

ID	Thème	En tant que	Je voudrais (User story)	Estimation (/H)
1	Authentification	Administrateur	M'authentifier	10
2	Gérer les comptes	Administrateur	Créer un compte	10
3			Modifier un compte	10
4			Archiver un compte	10
5			Consulter un compte	10
6	Modification des rôles	Administrateur	Modifier les rôles d'accès	10
7	Gérer les catégories	Administrateur	Créer une catégorie	10
8			Modifier une catégorie	10
9			Supprimer une catégorie	10
10			consulter une catégorie	5
11	Gérer les départements	Administrateur	Créer un département	10
12			Modifier un département	10
13			Supprimer un département	10

Page suivante

TABLE 3.1 – Backlog Sprint1

ID	Thème	En tant que	Je voudrais (User story)	Estimation (En Heures)
14			consulter un département	10
15	Gérer les processus	Administrateur	Créer un processus	10
16			Modifier un processus	10
17			Supprimer un processus	10
18			consulter un processus	5

1.2 Analyse et spécification fonctionnelle

1.2.1 Diagramme de cas d'utilisation du sprint1

Nous présentons les fonctions de ce premier sprint du point de vue utilisateur, à travers le diagramme de cas d'utilisation modélisé ci-dessous.

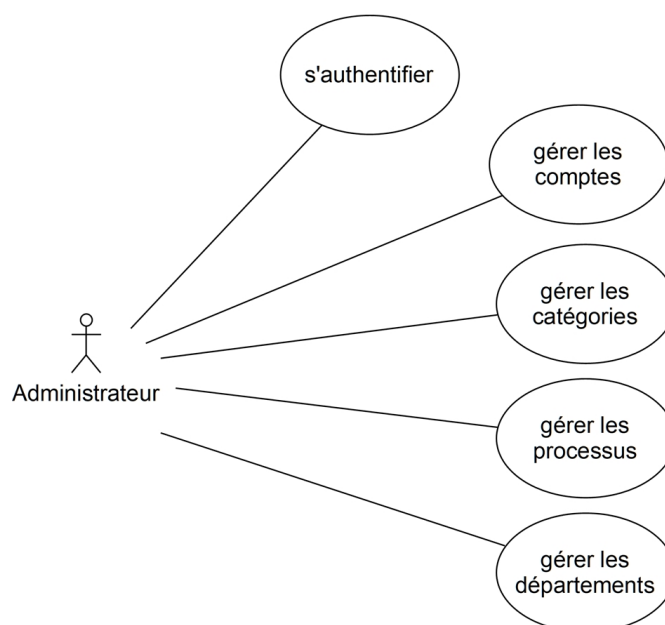


FIGURE 3.1 – Diagramme de cas d'utilisation du sprint1

1.2.2 Raffinement des cas d'utilisation du sprint1

Le raffinement des cas d'utilisation produit plus de détails sur les caractéristiques fournis par le système et les contraintes qui leurs sont associées.

* Raffinement du CU « gérer les comptes »

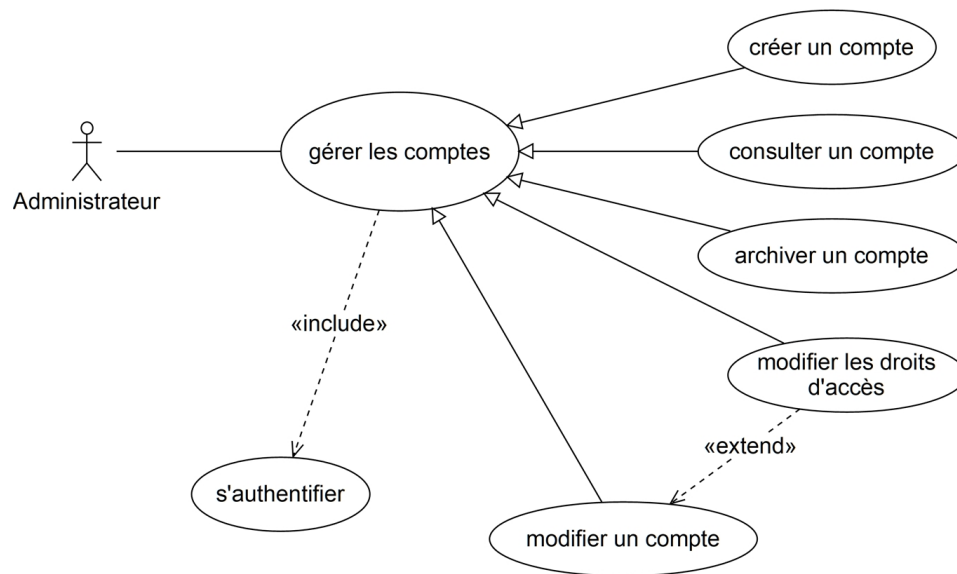


FIGURE 3.2 – gérer les comptes

* Description textuelle du CU « S'authentifier »

TABLE 3.2 – Description textuelle du CU « S'authentifier »

Acteur principal	Administrateur
Objectifs	Lors de l'accès au site, l'administrateur doit se connecter.
Pré-conditions	L'administrateur doit être figuré dans la base de données et connaître ses codes d'accès.
Postconditions	Redirection vers l'espace d'administration.
Scénario nominal	<ol style="list-style-type: none"> ❶ L'administrateur demande le formulaire d'identification; ❷ Le système communique le formulaire demandé; ❸ L'administrateur remplit le formulaire avec l'ensemble des informations nécessaires à son identification;; ❹ Le système vérifie les informations et signale le succès de l'opération; ❺ Le système redirige l'administrateur vers son espace approprié.

Alternatives	<i>A1 - Code(s) d'accès invalide(s)</i> L'enchaînement A1 démarre au point 4 du scénario nominal. ❶ Le système renvoie un message d'erreur et signale à l'administrateur de recommencer . Retour au point 3 du scénario nominal.
Contraintes	La réception d'une réponse à l'identification (Point 5) ne doit pas dépasser 3 secondes.

*** Description textuelle du CU « Créer un compte »**

TABLE 3.3 – Description textuelle du CU « Créer un compte »

Acteur principal	Administrateur
Objectifs	L'administrateur doit pouvoir créer un nouveau utilisateur (validateur, approbateur ou rédacteur)
Pré-conditions	→ L'administrateur doit être identifié; → La liste des utilisateurs est disponible.
Postconditions	Nouvel utilisateur ajouté à la base de données.
Scénario nominal	❶ L'administrateur demande le formulaire de création d'un nouvel utilisateur; ❷ Le système communique un formulaire demandé; ❸ L'administrateur renseigne le formulaire et le soumet au système; ❹ Le système enregistre les informations et signale le succès de l'opération.
Alternatives	<i>A1-a) - champs erronés</i> L'enchaînement A1 démarre au point 4 du scénario nominal. ❶ Aller au point 3 du scénario nominal. <i>A1-b) - champs déjà existant</i> ❶ Aller au point 3 du scénario nominal.

*** Description textuelle du CU « Consulter un compte »**

TABLE 3.4 – Description textuelle du CU « consulter un agent »

Acteur principal	Administrateur
Objectifs	L'administrateur veut trouver le plus rapidement possible un utilisateur précis.
Pré-conditions	<p>→ L'administrateur doit être identifié;</p> <p>→ La liste des utilisateur est disponible.</p>
Postconditions	L'administrateur a trouvé l'utilisateur précis qu'il cherchait, voire plusieurs.
Scénario nominal	<p>❶ L'administrateur lance une recherche rapide à partir de mots-clés;</p> <p>❷ Le système affiche une page de résultat. Les utilisateurs sont classés par défaut par date de création, le plus récent en premier;</p> <p>❸ L'administrateur sélectionne un utilisateur;</p> <p>❹ Le Système lui présente une fiche détaillée pour l'agent sélectionné.</p>
Alternatives	<p><i>A1-a Le système n'a pas trouvé d'utilisateur correspondant à la recherche</i></p> <p>L'enchaînement A1 démarre au point 2 du scénario nominal.</p> <p>❸ Le système signale l'échec à l'administrateur et lui propose d'effectuer une nouvelle recherche. Le cas d'utilisation redémarre à l'étape 1 du scénario nominal.</p> <p><i>A1-b Le système a trouvé de très nombreux utilisateurs</i></p> <p>❸ Le système signale le nombre d'utilisateurs à l'administrateur et lui affiche une première page de résultats. Les autres pages sont accessibles directement ou par des symboles Suivante et Précédente.</p> <p>❹ L'administrateur navigue dans ces pages et enchaîne éventuellement sur l'étape 3 du scénario nominal.</p> <p><i>A1-c L'administrateur n'est pas intéressé par les résultats</i></p> <p>❸ L'administrateur revient à l'étape 1 du du scénario nominal pour lancer une nouvelle recherche.</p> <p>❹ L'administrateur abandonne la recherche. Le cas d'utilisation se termine en échec.</p>

Contraintes	<p>→ La recherche doit être la plus rapide possible;</p> <p>→ Les résultats de la recherche doivent être pertinents, c'est-à-dire correspondre à la requête;</p> <p>→ Le formulaire de recherche rapide doit être toujours visible et donc se situer dans la partie supérieure de toutes les pages, quelle que soit la résolution d'écran du responsable.</p>
--------------------	---

*** Raffinement du CU « gérer les catégories »**

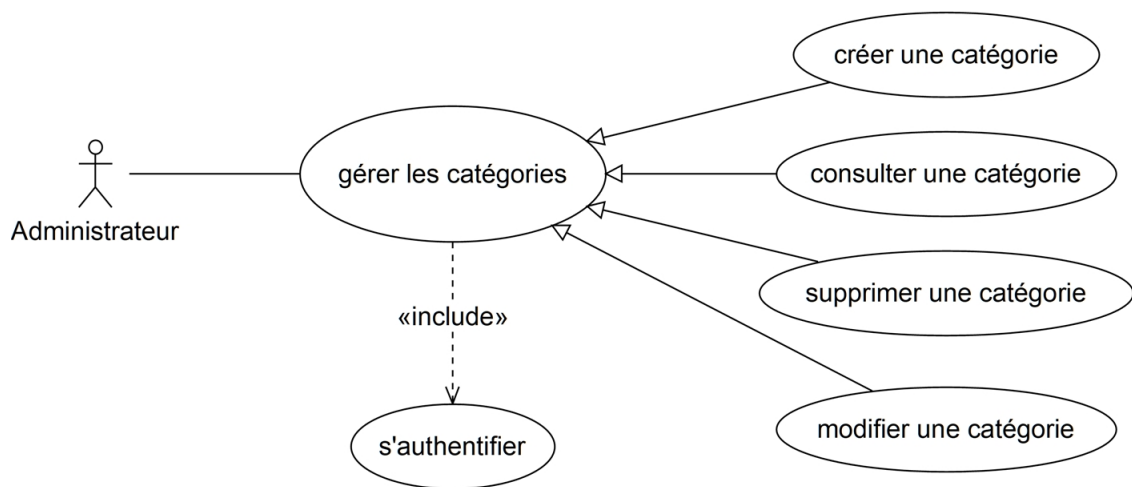


FIGURE 3.3 – Raffinement du CU « gérer les catégories »

*** Description textuelle du CU « Créer une catégorie »**

TABLE 3.5 – Description textuelle du CU « Créer une catégorie »

Acteur principal	Administrateur
Objectifs	L'administrateur doit pouvoir créer une nouvelle catégorie de documents
Pré-conditions	<p>→ L'administrateur doit être identifié;</p> <p>→ La liste des catégories est disponible.</p>
Postconditions	Nouvelle catégorie de document ajoutée à la base de données.

Scénario nominal	<p>❶ L'administrateur demande le formulaire de création d'une nouvelle catégorie;</p> <p>❷ Le système communique le formulaire demandé;</p> <p>❸ L'administrateur renseigne le formulaire et le soumet au système;</p> <p>❹ Le système enregistre les informations et signale le succès de l'opération.</p>
Alternatives	<p><i>A1-a) - champs erronés</i></p> <p>L'enchaînement A1 démarre au point 4 du scénario nominal.</p> <p>❺ Aller au point 3 du scénario nominal.</p> <p><i>A1-b) - champs déjà existant</i></p> <p>❺ Aller au point 3 du scénario nominal.</p>

*** Description textuelle du CU « Supprimer une catégorie »**

TABLE 3.6 – Description textuelle du CU « Supprimer une catégorie »

Acteur principal	Responsable
Objectifs	L'administrateur souhaite supprimer définitivement une catégorie
Pré-conditions	<p>→ L'administrateur doit être identifié;</p> <p>→ La liste des catégories est disponible.</p>
Postconditions	Catégorie supprimée.
Scénario nominal	<p>❶ L'administrateur sélectionne la catégorie et clique sur le bouton <i>supprimer</i>;</p> <p>❷ Le système affiche un message de confirmation;</p> <p>❸ L'administrateur confirme l'opération de suppression;</p> <p>❹ Le système confirme le succès de l'opération.</p>
Alternatives	<p><i>A1 : L'administrateur peut annuler l'opération de suppression</i></p> <p>L'enchaînement A1 démarre au point 3 du scénario nominal.</p> <p>Le cas se termine en échec.</p>

*** Raffinement du CU « gérer les départements »**

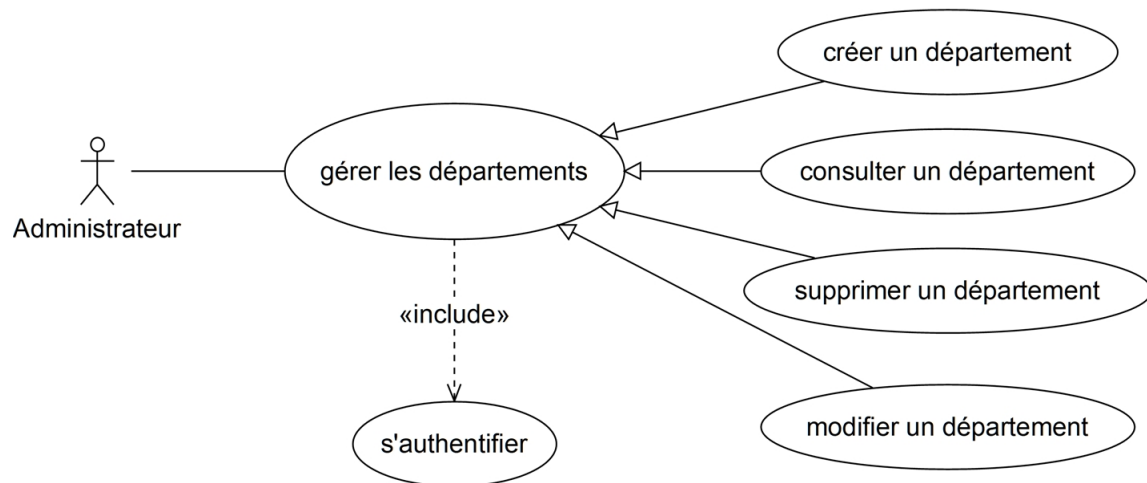


FIGURE 3.4 – Raffinement du CU « gérer les départements »

* Description textuelle du CU « Créer un département »

TABLE 3.7 – Description textuelle du CU « Créer un département »

Acteur principal	Administrateur
Objectifs	L'administrateur doit pouvoir créer un nouveau département
Pré-conditions	<p>→ L'administrateur doit être identifié;</p> <p>→ La liste des départements est disponible.</p>
Postconditions	nouveau département ajouté à la base de données.
Scénario nominal	<p>❶ L'administrateur demande le formulaire de création d'un nouveau département;</p> <p>❷ Le système communique le formulaire demandé;</p> <p>❸ L'administrateur renseigne le formulaire et le soumet au système;</p> <p>❹ Le système enregistre les informations et signale le succès de l'opération.</p>
Alternatives	<p><i>A1-a) - champs erronés</i></p> <p>L'enchaînement A1 démarre au point 4 du scénario nominal.</p> <p>❺ Aller au point 3 du scénario nominal.</p> <p><i>A1-b) - champs déjà existant</i></p> <p>❺ Aller au point 3 du scénario nominal.</p>

1.3 Conception

1.3.1 Diagrammes de séquences de conception

Texte

* Diagramme de séquences CU « S'authentifier »

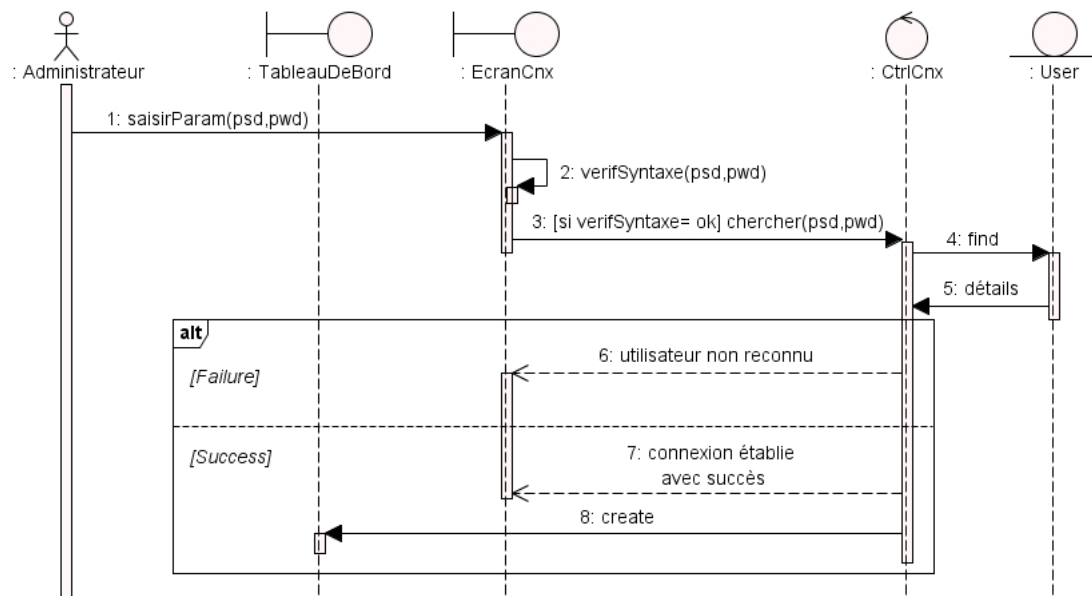


FIGURE 3.5 – Diagramme de séquences CU « S'authentifier »

L'administrateur veut accéder à son tableau de bord, il saisit ses paramètres d'identification, la vérification des paramètres est de la responsabilité de la classe du dialogue EcranCnx. Le contrôle CtrlCnx délègue ensuite à une entité User une opération de recherche. L'entité renvoie au contrôle le(s) résultat(s) de la recherche. Celui-ci initialise le dialogue Tableau-Bord (en cas d'une authentification réussite) , sinon il renvoie un message d'erreur à l'interface EcranCnx.

* Diagramme de séquences CU « Créer un compte »

Texte....

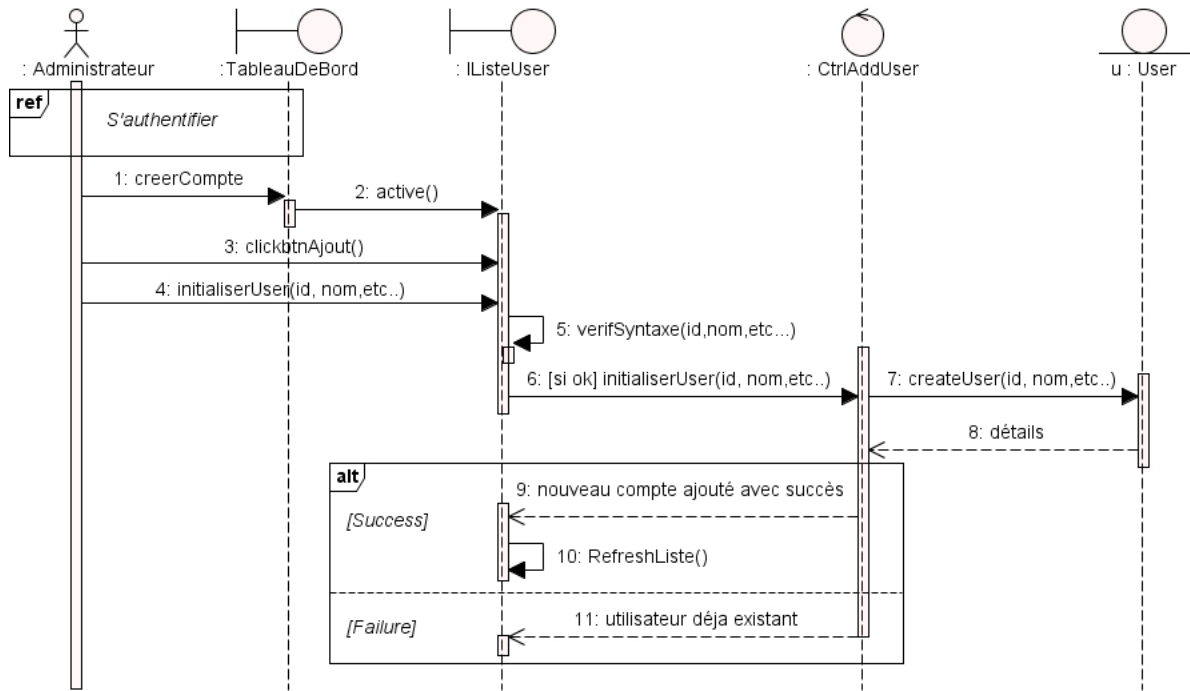


FIGURE 3.6 – Diagramme de séquences CU « Créer un compte »

* Diagramme de séquences CU « Consulter un compte »

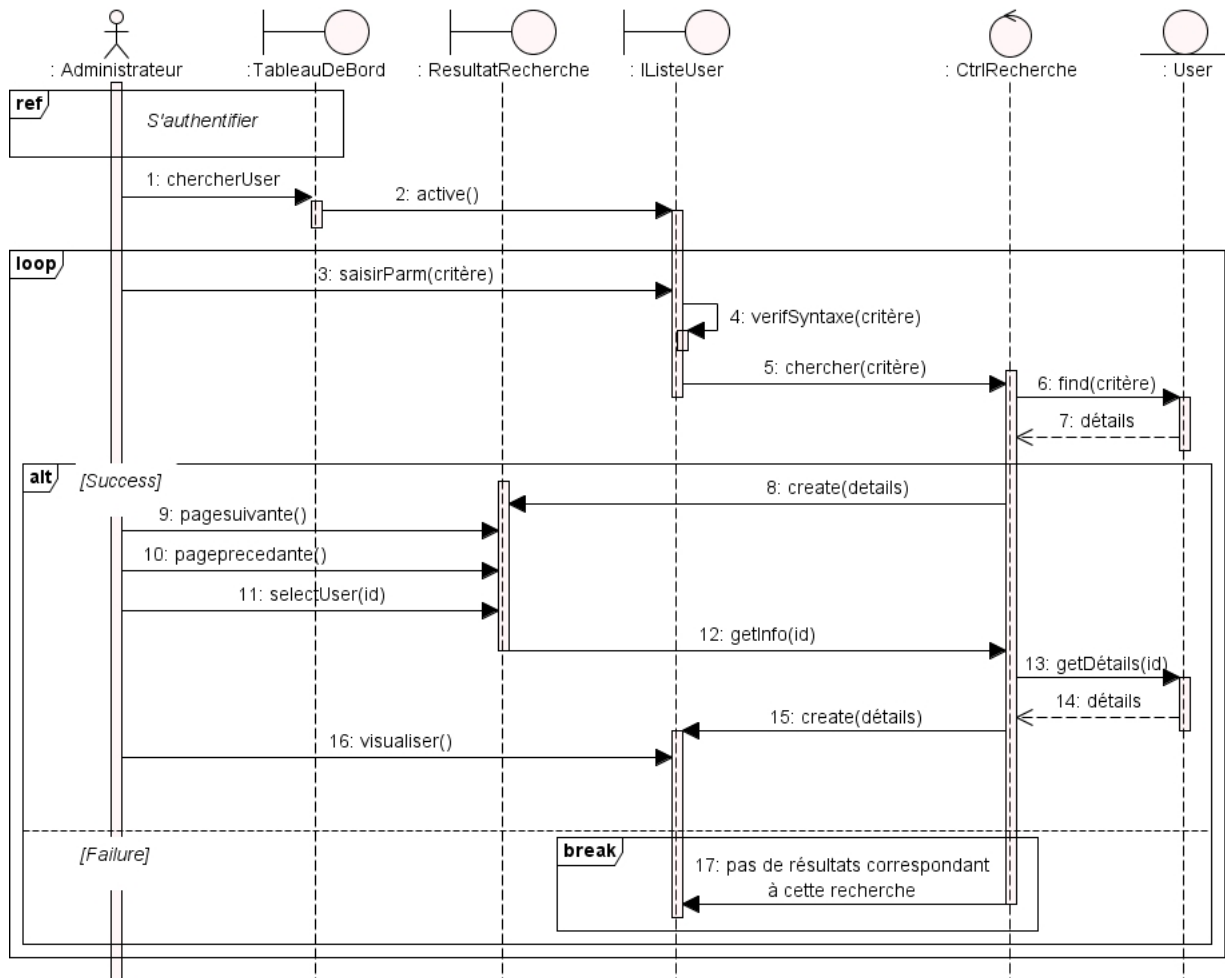


FIGURE 3.7 – Diagramme de séquences CU « chercher un compte »

L'administrateur veut trouver le plus rapidement possible un utilisateur. Suite a son identification, il choisit le lien vers la page de recherche. Il saisit une phrase de recherche (id, nom, prénom etc.).

Nous notons que la vérification syntaxique de la phrase de recherche est de la responsabilité de la classe dialogue IListeUser elle-même (et pas du contrôle associé qui n'est invoqué que dans le cas favorable où il n'y a pas d'erreur de syntaxe).

Le contrôle CtrlRecherche délègue ensuite à une entité User l'opération de la recherche. L'entité User construit alors les résultats correspondants à la recherche, qu'elle renvoie au contrôle CtrlRecherche. Celui-ci initialise le dialogue chargé du résultat, en lui passant les détails de recherche.

L'administrateur peut ensuite naviguer dans les différentes pages du résultat.

* Diagramme de séquences CU « Modifier les rôles d'accès un droit d'accès »

Texte....

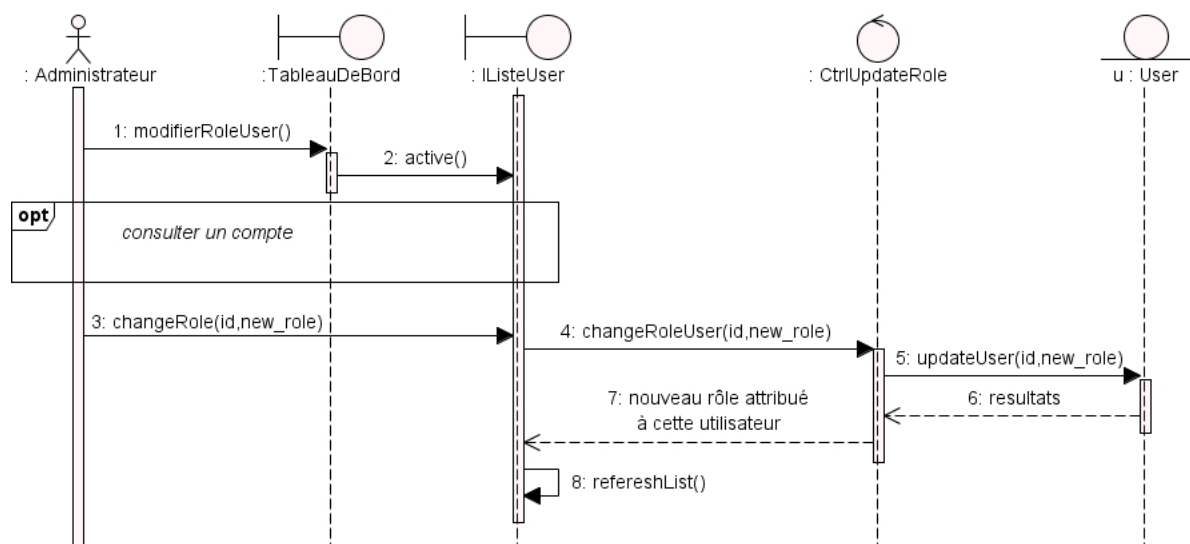


FIGURE 3.8 – Diagramme de séquences CU « Modifier les rôles d'accès un droit d'accès »

* Diagramme de séquences CU « Créer une catégorie »

Texte....

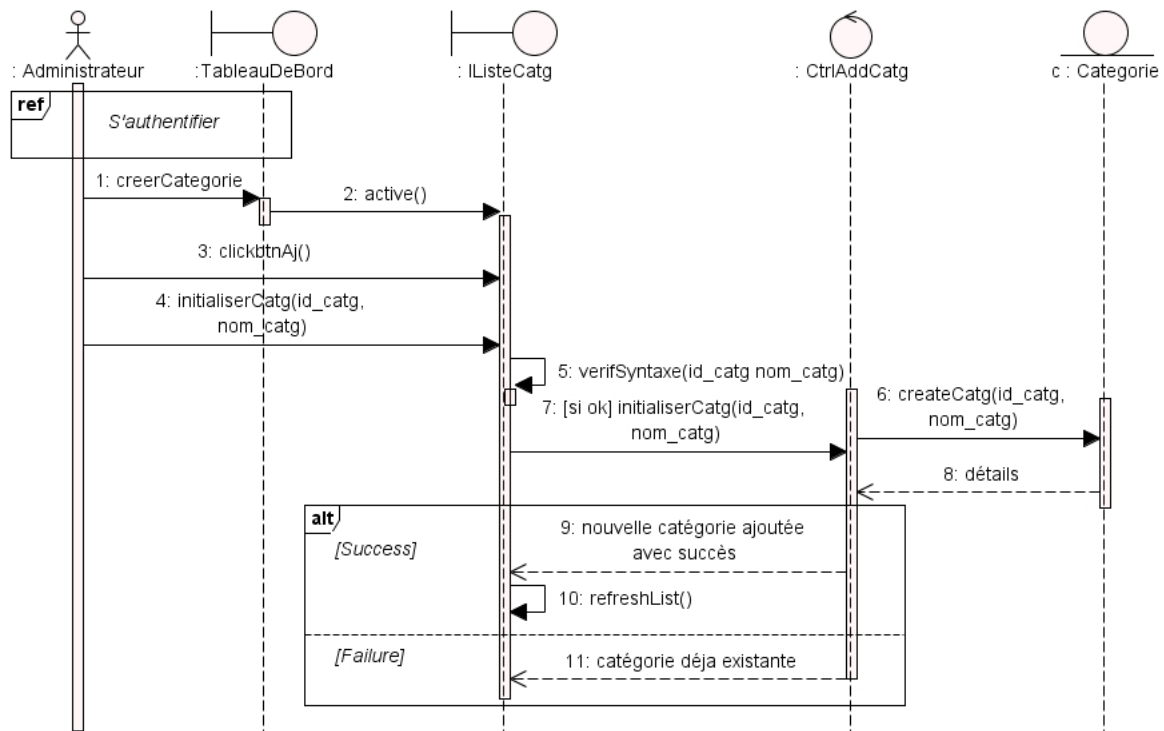


FIGURE 3.9 – Diagramme de séquences CU « Créer une catégorie »

* Diagramme de séquences CU « Supprimer une catégorie »

Texte....

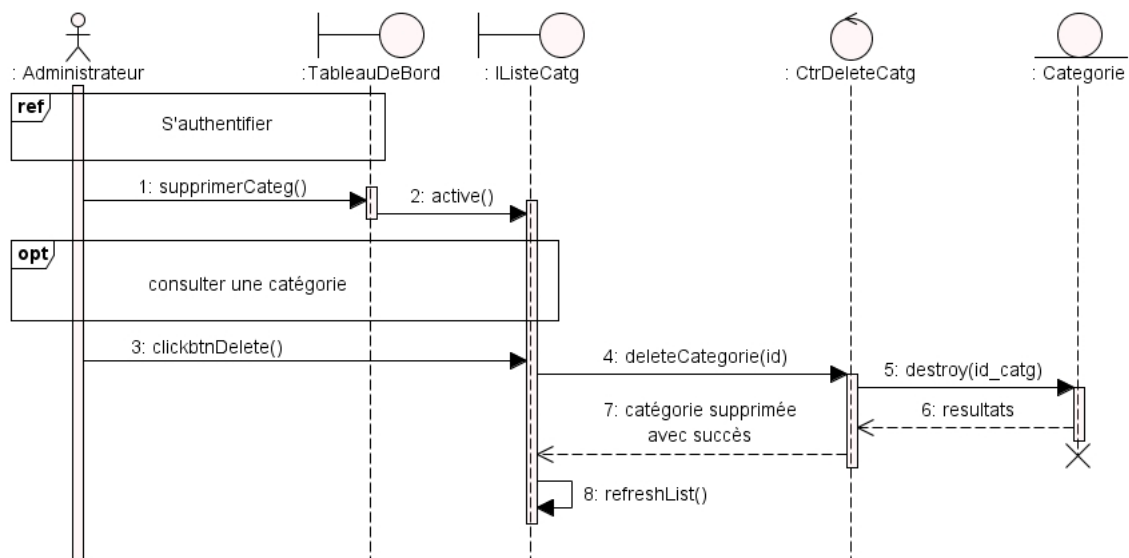


FIGURE 3.10 – Diagramme de séquences CU « Supprimer une catégorie »

* Diagramme de séquences CU « Créer un département »

Texte....

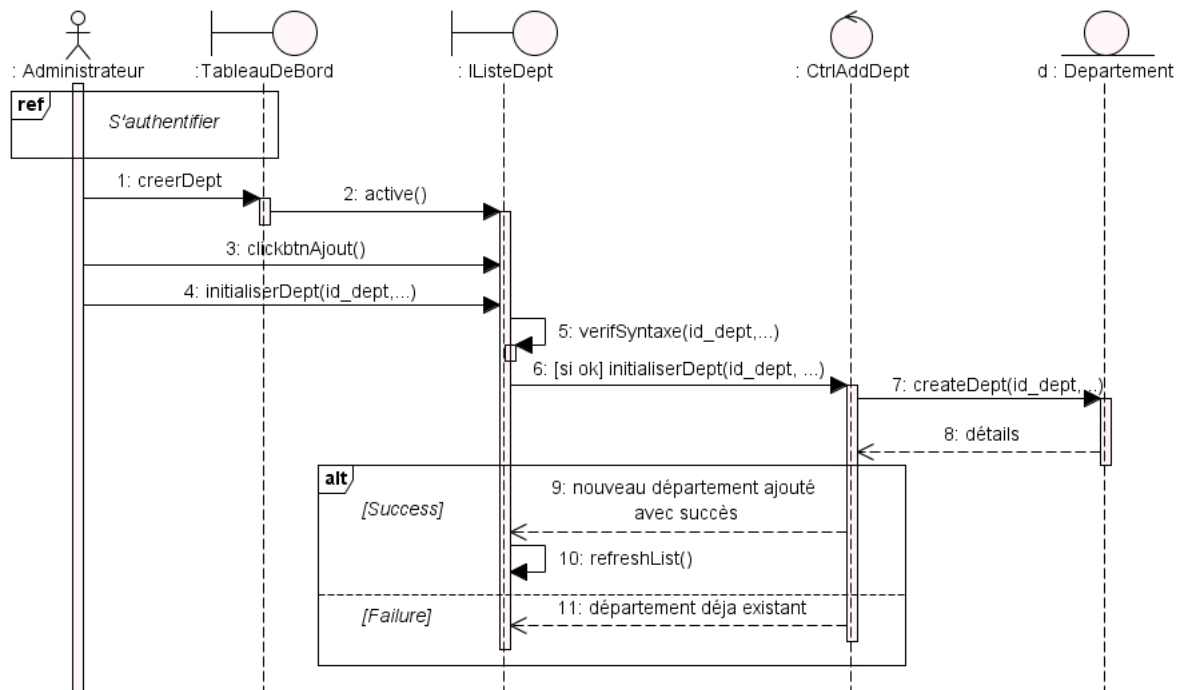


FIGURE 3.11 – Diagramme de séquences CU « Créer un département »

1.3.2 Diagramme de classes du sprint1

Texte ici

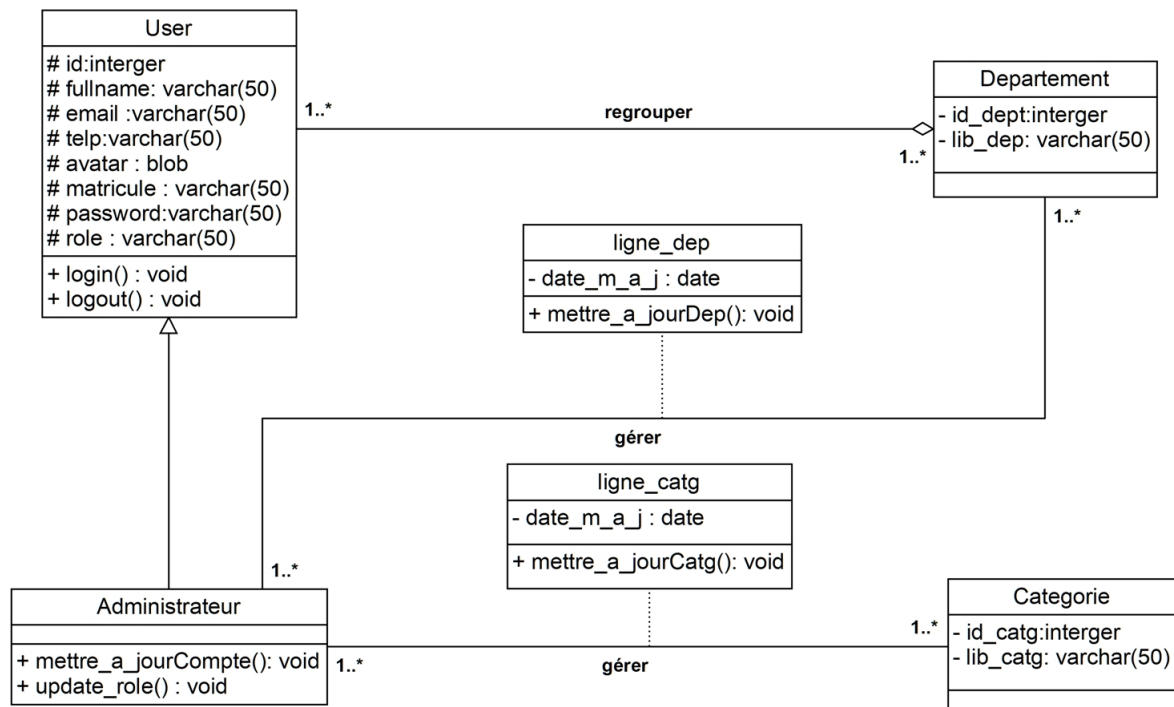


FIGURE 3.12 – Diagramme de classes du sprint1

1.4 Réalisation

Texte

*



FIGURE 3.13 – Capture

*



FIGURE 3.14 – Capture

*



FIGURE 3.15 – Capture

*



FIGURE 3.16 – Capture

*



FIGURE 3.17 – Capture

*



FIGURE 3.18 – Capture

1.5 Revue du sprint1

La figure 3.19 permet de représenter sous forme graphique l'évolution de la quantité de travail restante pour une période donnée.

L'unité de temps est généralement le jour, la quantité de travail étant quant à elle exprimée en nombre d'heures. La courbe réelle (marquée en bleu) reste au-dessus de la droite idéale (marquée en orange). C'est assez normal puisque durant cette période, nous prenons connaissance des nouvelles tâches embarquées dans le sprint courant. C'est également à ce moment-là que nous nous apercevons que certaines tâches sont finalement plus complexes à développer que prévu.



FIGURE 3.19 – Revue du sprint1

2 Étude et réalisation du Sprint2

2.1 Backlog Sprint2

TABLE 3.8 – Backlog Sprint2

ID	Thème	En tant que	Je voudrais (User story)	Estimation (En Heures)
1	Gérer un document	Rédacteur	Créer un document	-
2		Valideur	Modifier un document	-
3		Approbateur	Supprimer un document	-
4		Administrateur	Consulter un document	-
5	Indexation	Rédacteur Valideur Approbateur Administrateur	Indexer un document	-
6	Validation	Valideur Approbateur Administrateur	Valider un document	-
7	Aviser un document	Approbateur Administrateur	Approuver un document Réfuter un document	-
8	Diffusion	Approbateur Administrateur	Diffuser un document	-
9	Archivage	Approbateur Administrateur	Archiver un document	-

2.2 Analyse et spécification fonctionnelle

2.2.1 Diagramme de cas d'utilisation du sprint2

Texte ici

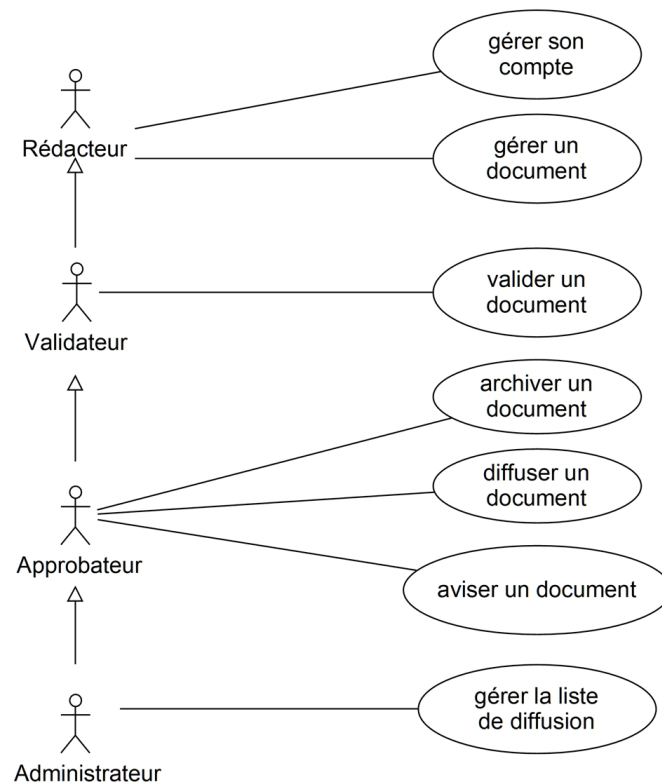


FIGURE 3.20 – Diagramme de cas d'utilisation du sprint2

2.2.2 Raffinement des cas d'utilisation du sprint2

* Raffinement du CU « gérer un document »

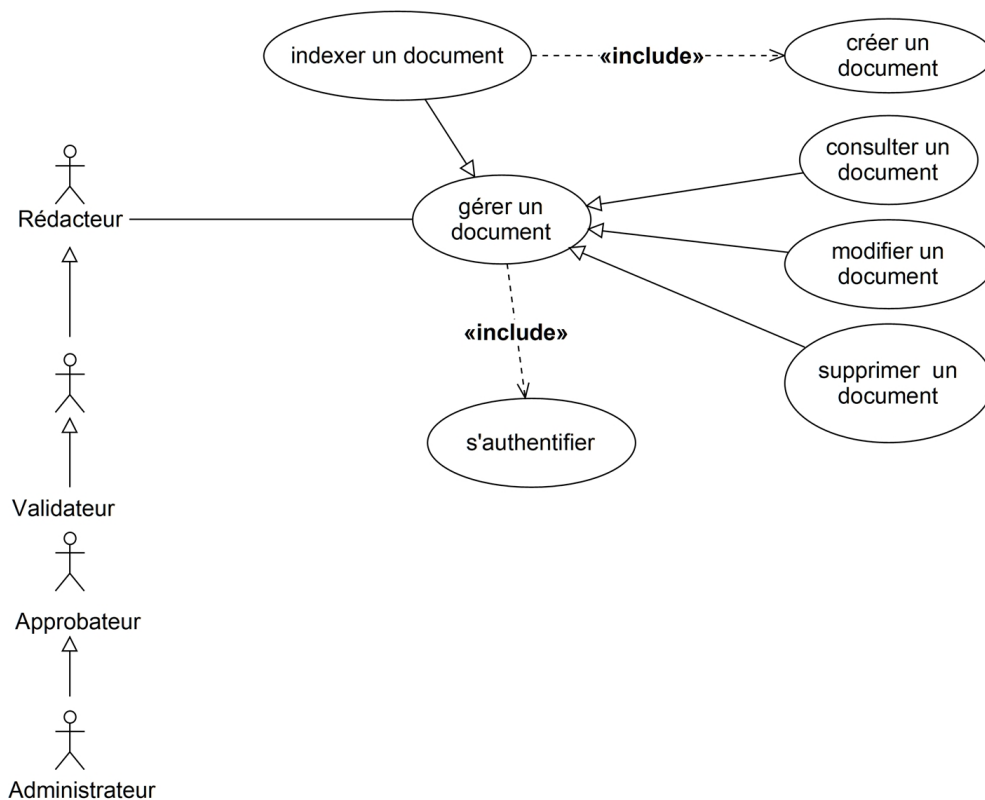


FIGURE 3.21 – Raffinement du CU « gérer un document

* Description textuelle du CU « créer un document »

TABLE 3.9 – Description textuelle du CU « créer un document »

Acteur principal	Rédacteur, Validateur, Approbateur et Administrateur.
Objectifs	Création d'un nouveau document
Pré-conditions	<p>→ L'acteur doit être identifié;</p> <p>→ La liste des documents est disponible.</p>
Postconditions	Nouveau document ajouté à la base de données.
Scénario nominal	<p>❶ L'acteur demande le formulaire de création d'un nouveau document;</p> <p>❷ Le système communique le formulaire demandé;</p> <p>❸ L'acteur renseigne le formulaire (nom, type, description etc..) et le soumet au système;</p> <p>❹ Le système enregistre les informations et signale le succès de l'opération.</p>
Alternatives	<p><i>A1-a) Champ (s) erroné(s)</i></p> <p>L'enchaînement A1 démarre au point 4 du scénario nominal.</p> <p>❺ Aller au point 3 du scénario nominal;</p> <p><i>A1-b) Champ (s) déjà existant (s)</i></p> <p>❺ Aller au point 3 du scénario nominal;</p>

* Description textuelle du CU « Indexer un document »

TABLE 3.10 – Description textuelle du CU « indexer un document »

Acteur principal	Rédacteur, Validateur, Approbateur et Administrateur.
Objectifs	Octroyer des références uniques aux documents.
Pré-conditions	<p>→ L'acteur doit être identifié;</p> <p>→ Le document à indexer doit être disponible</p>
Postconditions	Document référencé

Scénario nominal	❶ Le système génère une référence unique liée à chaque document ; ❷ enregistre les informations et signale le succès de l'opération.
Alternatives	Néant.

*** Raffinement du CU « valider un document »**

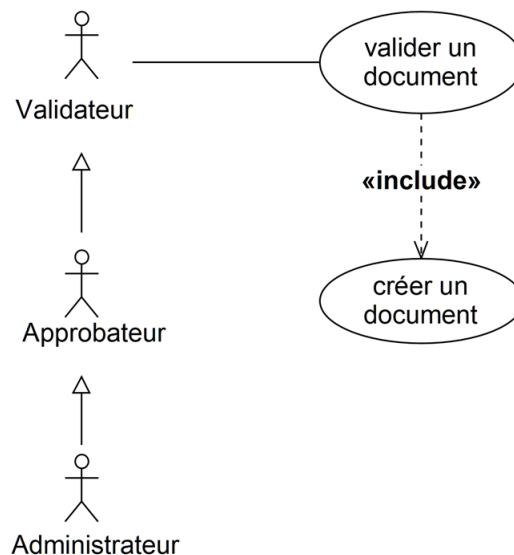


FIGURE 3.22 – Raffinement du CU « valider un document

*** Description textuelle du CU « valider un document »**

TABLE 3.11 – Description textuelle du CU « valider un document »

Acteur principal	Validateur, Approbateur et Administrateur.
Objectifs	Validation des documents.
Pré-conditions	→ L'acteur doit être identifié; → Le document est créé et y est référencé.
Postconditions	Document traité.
Scénario nominal	❶ L'acteur choisit entre deux options <i>validation-accept</i> ou <i>validation-refus</i> ; ❷ Le système enregistre les changements et signale le succès de l'opération.

Alternatives	<p><i>A1 - Option non cochée</i></p> <p>L'enchaînement A1 démarre au point 1 du scénario nominal.</p> <p>❷ Le système demande à l'acteur de choisir entre deux option pour traiter le document;</p> <p>Retour au point 1 du scénario nominal.</p>
---------------------	---

*** Raffinement du CU « aviser un document »**

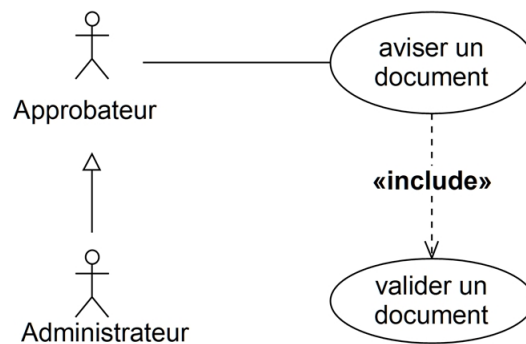


FIGURE 3.23 – Raffinement du CU « valider un document

*** Description textuelle du CU « aviser un document »**

TABLE 3.12 – Description textuelle du CU « aviser un document »

Acteur principal	Approbateur et Administrateur.
Objectifs	Ré-validation des documents traités auprès des validateurs.
Pré-conditions	<p>→ L'acteur doit être identifié;</p> <p>→ Le document est déjà traité par le validateur.</p>
Postconditions	Approbation ou Rejet du document
Scénario nominal	<p>❶ L'acteur choisit entre deux options <i>approuver</i> ou <i>rejeter</i>;</p> <p>❷ Le système enregistre les changements et signale le succès de l'opération.</p>
Alternatives	<p><i>A1 - Option non cochée</i></p> <p>L'enchaînement A1 démarre au point 1 du scénario nominal.</p> <p>❷ Le système demande à l'acteur de choisir entre deux option pour ré-valider le document;</p> <p>Retour au point 1 du scénario nominal.</p>

2.3 Conception

2.3.1 Diagrammes de séquences

Texte

* Diagramme de séquences CU « créer un document »

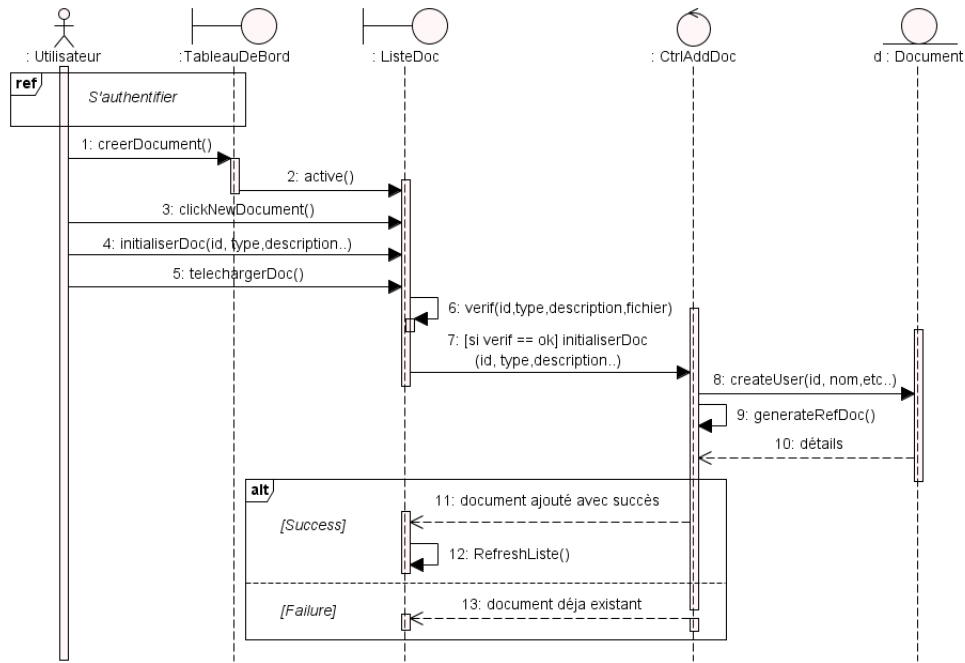


FIGURE 3.24 – Diagramme de séquences CU « créer un document »

* Diagramme de séquences CU « valider un document »

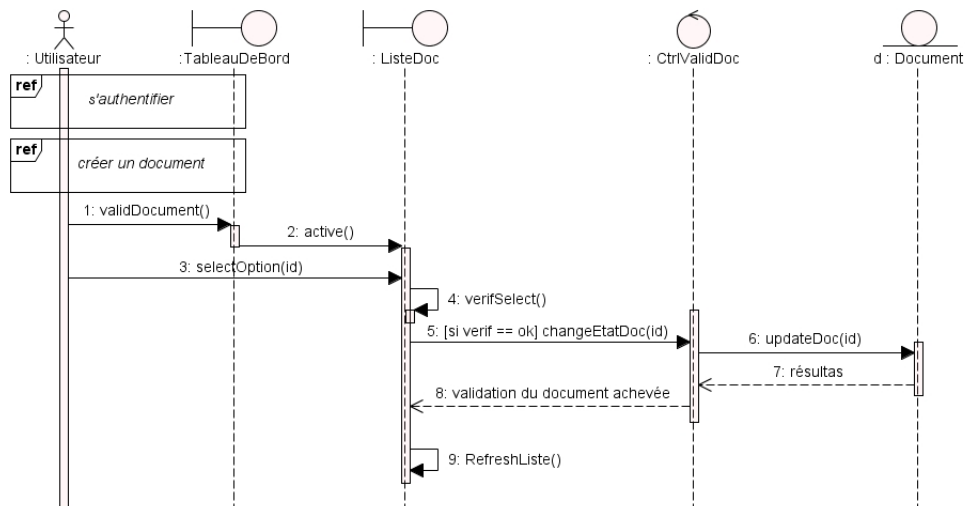


FIGURE 3.25 – Diagramme de séquences CU « valider un document »

* Diagramme de séquences CU « aviser un document »



FIGURE 3.26 – Diagramme de séquences CU « aviser un document »

2.3.2 Diagramme de classes du sprint2

Texte ici



FIGURE 3.27 – Diagramme de classes du sprint2

2.4 Diagramme d'état transition



FIGURE 3.28 – Diagramme d'état transition d'un document

2.5 Réalisation

Texte

*



FIGURE 3.29 – Capture

*



FIGURE 3.30 – Capture

*



FIGURE 3.31 – Capture

*



FIGURE 3.32 – Capture

*



FIGURE 3.33 – Capture

*



FIGURE 3.34 – Capture

2.6 Revue du sprint2

Texte ici

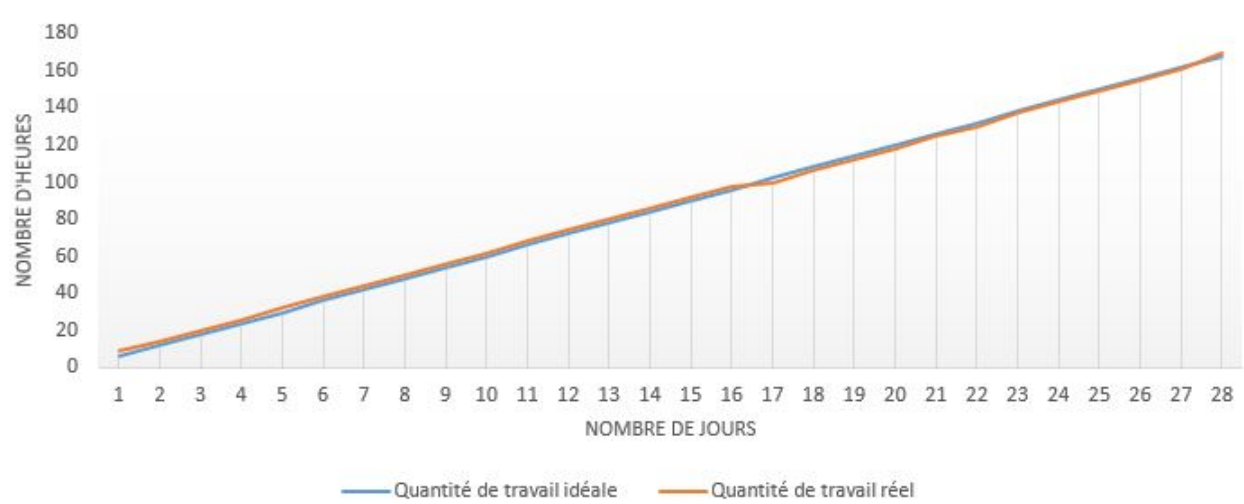


FIGURE 3.35 – Revue du sprint2

Conclusion

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Nom Prénom. *Titre du livre*. Eyrolles, 2022.
- [2] Nom Prénom. Titre du article. <<https://myscol.com/>>, 2022. [En ligne; Accès le 17-01-2021].