

MEMOIRE DE STAGE DE FIN D'ETUDES

Pour l'obtention du

MASTERE PROFESSIONNEL

« Nouvelles Technologies des Télécommunications et Réseaux »

Présenté par :

Marwa KESRAOUI

**Conception et développement d'un bureau
Virtuel « WebTop »**

Soutenu le : 05 /02/2014

Devant le jury :

Mr : Khaled Jlassi : Président

Mme : Ahlem Ben Hsin : Membre

Mme : Houda Rekaya Houissa : Membre

Mr : Khaled Sammoud : Encadreur pédagogique

Dédicaces

*A mon cher père qui m'a toujours soutenue dans mes études et
à qui je souhaite une heureuse retraite.*

*A ma mère, à qui je dois tout et qui a énormément sacrifié pour
Mon avenir.*

*A mes deux frères Aymen et Mahdi qui m'ont longuement
soutenue et encouragée.*

*A mes amis et les enseignants de l'Université Virtuelle qui
m'ont aidé moralement et techniquement.*

Marwa

Remerciements

Je tiens à exprimer nos remerciements les plus vifs à ceux qui me font l'honneur de participer et de siéger dans ce jury.

Je voudrais exprimer ma sincère gratitude à Monsieur **Sliman Hemidi** mon encadreur professionnel à BFI

Mes remerciements vont également à Monsieur **Khaled Sammoud**, mon encadrant pédagogique à **UVT** pour ses directives et ses conseils enrichissants, pour ses critiques constructives et ses encouragements tout au long de la réalisation de ce projet.

Enfin, je remercie tous ceux qui nous m'ont aidés de près ou de loin à effectuer ce travail.

Table des matières

CHAPITRE 1 : CADRE DU PROJET	2
1. PRESENTATION DE L'ENTREPRISE D'ACCUEIL.....	2
2. CONTEXTE : TRAVAIL A DISTANCE.....	5
3. PROBLEMATIQUE	5
4. SOLUTION PROPOSEE.....	5
1. LES BUREAUX VIRTUELS.....	7
2. LA VIRTUALISATION.....	8
3. LE CLOUD COMPUTING.....	8
1. ULTEO OPEN SOURCE(2)	10
2. MYBOOO(3).....	11
3. MYNOX(4).....	12
4. ZOHDOCS (5)	13
• Zoho Sheet : Service pour créer les feuilles de calculs	14
• Zoho Show: Service pour les diapositives.	14
• Zoho Docs: Service pour la création et le stockage des documents texte.	14
1. LES METHODES AGILES (6).....	16
2. LE PROCESSUS UNIFIE (PU) (7).....	18
3. COMPARAISON ENTRE LES METHODES	19

4. METHODOLOGIE ADOPTEE : SCRUM	22
4.1. PRESENTATION DE SCRUM	22
CHAPITRE 3 : ANALYSE ET SPECIFICATION DES BESOINS.....	26
1. IDENTIFICATION DES ACTEURS.....	26
2. IDENTIFICATION DES BESOINS FONCTIONNELS.....	27
3. CAPTURE DES BESOINS NON FONCTIONNELS.....	28
1. DEFINITION DU BACKLOG DU PRODUIT.....	28
1. DIAGRAMME DE CAS D'UTILISATION GENERAL.....	30
1. DESCRIPTION DETAILLEE DU DIAGRAMME DU CAS D'UTILISATIONS.....	31
CHAPITRE 4 : CONCEPTION	39
1. DIAGRAMME DE CLASSES.....	43
2. DECOUPAGE EN PACKAGES	44
1. SCENARIO : «AUTHENTIFICATION ».....	52
2. SCENARIO : « AJOUT D'UN EMPLOYE ».....	53
3. SCENARIO 3 : MODIFICATION D'UN EMPLOYE	55
CHAPITRE 5 : MISE EN ŒUVRE.....	57
I. Environnement de travail et technologies utilisées	57
1. TECHNOLOGIES.....	57
2. ARCHITECTURE ET DEPLOIEMENT	60

1. AUTHENTIFICATION	61
2. INTERFACE PRINCIPALE	62
3. EDITION DES FICHIERS	63
4. CHARGEMENT DES FICHIERS	63
5. WEB SERVICE METEO	65
6. WEB SERVICE RSS FOURNI PAR BING	65
1. LES TESTS	67
IV chronogramme	69
CONCLUSION GENERALE ET PERSPECTIVES	A
NETOGRAPHIE	C
<i>BIBLIOGRAPHIE</i>	D

Table des figures

Figure 1-Organigramme de BFI	3
Figure 2 Schéma général d'un bureau virtuel	6
Figure 3-Bureau virtuel en mode Desktop Ulteo	11
Figure 4-Bureau virtuel en mode Desktop Payant Myboo.....	12
Figure 5-Bureau virtuel en mode web:Mynox	13
Figure 6-Services offerts par Zoho	14
Figure 7 Architecture générale de Scrum.....	23
Figure 8-Diagramme de cas d'utilisation global	30
Figure 9Prototype de l'interface d'accueil	36
Figure 10- Prototype de l'interface graphique du système de fichiers.....	37
Figure 11 Interface de gestion d'employé.....	37
Figure 12-Architecture 3 tiers	40
Figure 13-Model View Presenter (MVP) [Source: nielyde.org].....	42
Figure 14 Diagramme de classes	43
Figure 15Diagramme de packages	45
Figure 16package "Model"	46
Figure 17 Classes du package "View"	48
Figure 18- package "Presenter".....	50
Figure 19Classe Service du package Service.....	51
Figure 20-Diagramme de séquence objet: Authentification.....	52
Figure 21 Diagramme de séquences Ajout d'un employé	54
Figure 22 Scénario Modification d'un employé	55

Figure 23 Architecture de l'application	60
Figure 24-Interface de connexion	61
Figure 25-Interface d'échec de la connexion	62
Figure 26-Interface principale	62
Figure 27-Editer du texte.....	63
Figure 28 Chargement des fichiers.....	64
Figure 29-Service météo(Web Service Yahoo)	65
Figure 30-Flux RSS	66
Figure 31 Exemple de test 1	68
Figure 32 Exemple de test 2	68
Figure 33 Exemple de test 3	68

Liste des tableaux

Tableau 1-Tableau récapitulatif des bureaux virtuels.....	16
Tableau 2-Tableau comparatif entre les méthodologies de developpement.....	21
Tableau 3tableau illustrant les besoins fonctionnels.....	27
Tableau 4Définition du backlog du produit.....	29
Tableau 5Description du cas d'utilisation "Gérer les employés"	31
Tableau 6Description textuelle de cas d'utilisation "Gèrer les documents"	33
Tableau 7Description textuelle du cas d'utilisation "Importer/Exporter les documents"	34
Tableau 8 Description textuelle du cas d'utilisation "Partager les documents"	34
Tableau 9 Description textuelle du cas d'utilisation "Consulter historique"	35
Tableau 10chronogramme	71

Introduction générale

Dans un monde continuellement évolutif, et grâce à Internet qui a radicalement changé les habitudes de travail, la motivation d'avoir des moyens performants et efficaces de communication et d'échange d'informations devient de plus en plus fondamentale. Cette motivation donne naissance à une révolution favorisant le travail à distance et l'accès aux besoins en temps réduit et à partir de n'importe quel endroit.

Nous remarquons ces dernières années un développement exponentiel des espaces de stockages sur le web qui se sont répandus comme une traînée de poudre dans le monde.

Dans ce cadre, les espaces ou sites de stockages virtuels apparaissent pour rompre avec les disques de stockages physiques et donner une autre dimension à cette technologie. Tout en intégrant de nouvelles fonctionnalités, cette dernière attire la clientèle grâce à la disponibilité permanente de ses documents sur le web.

Grace à l'évolution de cette nouvelle technologie, les internautes n'aurons plus besoin désormais de support physique pour stocker leurs données, et n'aurons plus besoin également d'un système d'exploitation. Dans le futur proche, Chaque employé aura son propre espace de travail avec une simple procédure d'authentification.

Dans le cadre du projet de fin d'études pour les étudiants de la deuxième année master N2TR, nous sommes appelés à concevoir et développer un logiciel ou une application afin de couronner nos deux années d'études académiques.

C'est dans ce contexte que s'intègre notre projet de fin d'études intitulé : " **Conception et développement d'un bureau virtuel « WebTop »**. Ce projet, réalisé au sein de la société BFI a pour but de concevoir et de développer un bureau Ce présent rapport est organisé comme suit :

En premier lieu nous présenterons le cadre du projet incluant la présentation de l'organisme d'accueil et la problématique.

En second, nous définirons dans ce chapitre intitulé Etat de l'art les aspects théoriques liés à notre travail.

Nous aborderons ensuite dans le chapitre Analyse et Spécification des besoins la présentation du contexte de la solution proposée et ses besoins.

Ce quatrième chapitre nommé Conception sera consacré à la spécification générale et détaillée du système.

Enfin, le dernier chapitre s'intéressera à la Mise en œuvre de l'application. Finalement, nous conclurons en passant en revue les différentes phases de notre travail, ainsi que les perspectives de l'évolution du système.

Chapitre 1 : Cadre du projet

Introduction

Dans ce premier chapitre, nous commencerons par présenter l'organisme d'accueil ensuite par définir certains termes auxquels on aura recours dans cet ouvrage. Ensuite, nous allons définir la problématique et la solution proposée.

Nous allons finalement terminer ce chapitre par une conclusion.

1. Présentation de l'entreprise d'accueil

BFI est spécialisée depuis 1994 dans l'édition et l'intégration de solutions destinées aux banques et institutions financières. Bien implantée sur le marché africain, BFI a réussi à se forger une réputation de très haut niveau, en équipant plus de 180 clients dans 15 pays différents notamment avec des institutions de renommée mondiale.

Forte d'une expérience de plus de 15 ans d'installations réussies en Afrique et d'une politique de proximité auprès de ses clients, BFI offre un savoir-faire reconnu dans la mise en œuvre de projets couvrant: le suivi et le pilotage, l'intégration de systèmes, la méthodologie de test et de recette, la migration des données, la formation et le transfert de compétences, le conseil et l'accompagnement et enfin le support et la maintenance.

BFI a construit son offre autour de deux axes essentiels du domaine de la banque et de la finance les métiers de la banque et les systèmes de paiement.

Ainsi, BFI offre à ses clients une gamme de solutions couvrant la globalité des besoins d'une banque en matière de systèmes d'information front et back-office ainsi que tous les métiers de la banque : gestion des clients, gestion des risques, contrôle des engagements, des crédits, trésorerie, marché des capitaux, moyens de paiement, comptabilité... leur permettant ainsi de disposer d'un système d'information à la hauteur des enjeux stratégiques.

BFI possède des partenaires géants dont on cite :

-Microsoft.

- Oracle as a Gold Partner.

- IBM as a World Partner.

- Thales: leader mondial dans les marchés de haute technologie pour la défense et la sécurité notamment dans la sécurité des transactions financières.

Organigramme

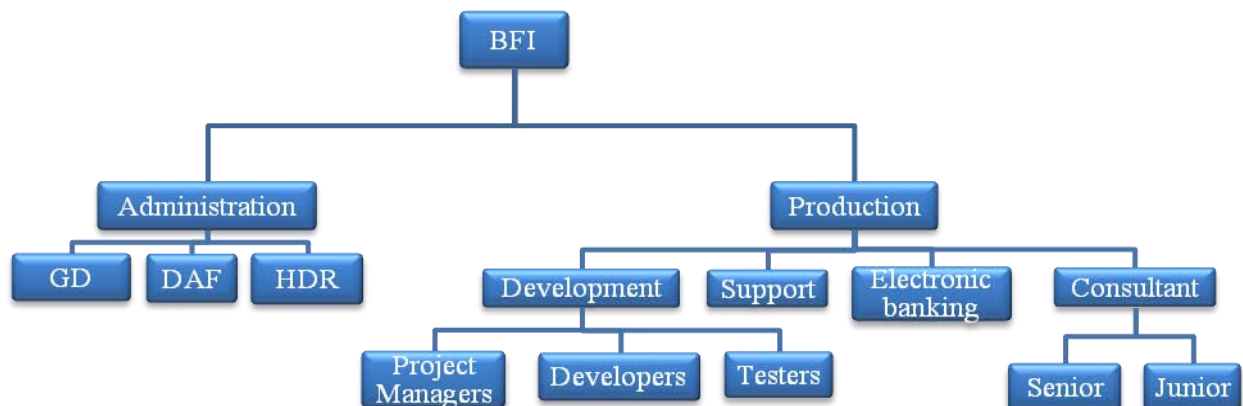


Figure 1-Organigramme de BFI

Le personnel de BFI est divisé en deux unités distinctes :

- Administration
- Production

L'unité d'administration se compose principalement de:

- La Direction générale: prend en charge toutes les décisions stratégiques de l'entreprise.
- Administration et finances (DAF): gère les activités courantes de tous les services et assure la coordination avec les autres directions. Elle se charge de toutes les transactions en espèces, l'entretien de la comptabilité, de budgétisation ...
- La Direction des Ressources Humaines (DRH): permet la gestion du personnel (recrutement, paiement, contrôle ...).

L'unité de production comprend:

- L'équipe de Consultants: représente les consultants seniors et juniors.
- L'équipe de développement: assure la production de produits de BFI.
- L'équipe du support technique: assure la maintenance et la surveillance des produits déjà livrés aux clients (SAV).
- L'équipe de services bancaires électroniques: permet la mise en œuvre de terminaux (TPE, GAB, DAB ...).

Solutions

BFI est engagée envers ses clients en tant que partenaire, en leur offrant la réussite de leurs projets dans les délais de mise en œuvre et le budget. Elle est constamment à leur écoute pour intégrer de nouvelles exigences fonctionnelles.

Ainsi BFI a gagné la confiance de ses clients et le monde de principal bailleur de fonds, comme la Banque mondiale (BM), la Banque africaine de développement (BAD) et la Millenium Challenge Corporation (MCC).

BFI propose des solutions modulaires, innovantes et personnalisables, abordant divers domaines tels que les systèmes bancaires de l'information et des marchés financiers.

2. Contexte : Travail à distance

Ce terme désigne l'aptitude à accéder à ses ressources logicielles ou matérielles à partir de n'importe quel autre terminal et à tout moment. Il existe plusieurs moyens de travailler à distance : Le mode console, le mode bureau ou desktop et le mode Web.

3. Problématique

Bien que le travail à distance facilite le travail collaboratif et apporte des gains en temps et ressources, il présente des problèmes .Par exemple, le travail à distance en mode console, assuré par un protocole d'accès à distance, n'offre aucune interface homme machine à l'utilisateur et ne supporte pas les clics de la souris et le « Drag-and-drop » devenus une coutume de l'utilisateur.

Le mode Bureau ou « Desktop » fonctionne en Gui- mode (Graphical User Interface), cependant, la communication entre hôtes est lente et limitée.

Le mode web, quasi nouveau, permet le travail à distance grâce au navigateur de n'importe quel terminal, cependant, la connexion doit être sécurisée et fiable.

4. Solution proposée

Il s'agit de développer une application qui permet aux employés d'accéder à un espace de travail en ligne personnel et de bénéficier des services des bureaux virtuels en ligne.

Les employés accèdent à cet espace à partir de chez eux, le siège de la société ou à partir de l'étranger et à partir de n'importe quel système d'exploitation.

Les services offerts par le bureau virtuel sont à priori les services de base des systèmes d'exploitation (Editeur de texte, classeur, Horloge, Calculatrice....)

La figure 2 présente la solution proposée aux problématiques citées ci –haut.

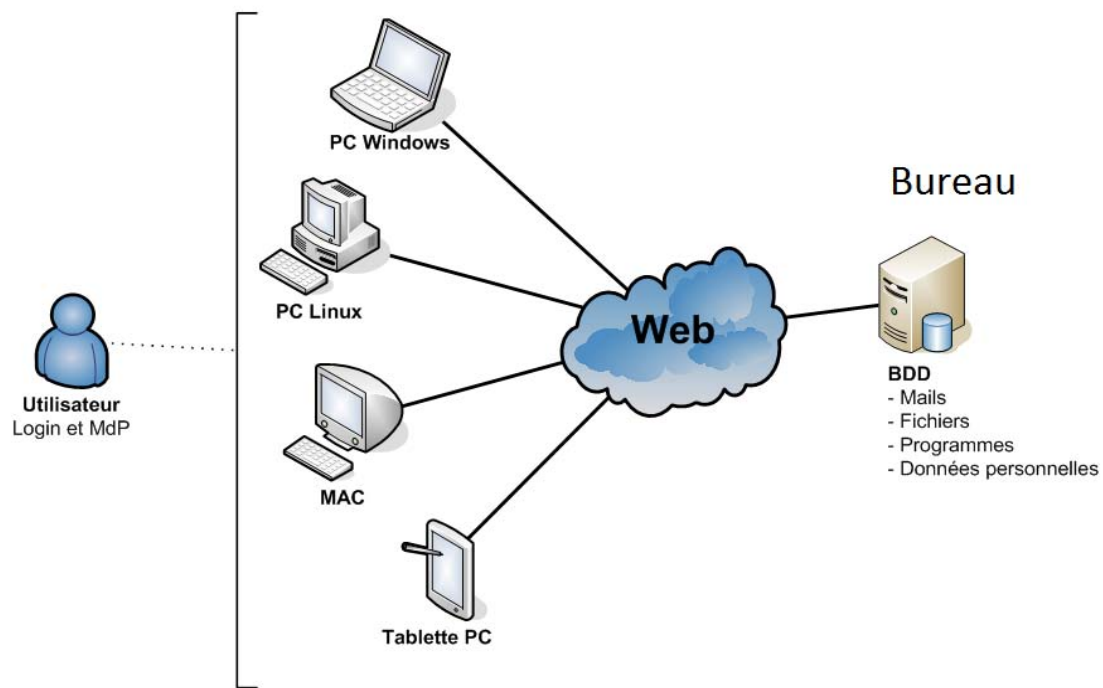


Figure 2 Schéma général d'un bureau virtuel

Conclusion

Nous avons offert une vue générale sur le projet en question, nous allons dans le prochain chapitre approfondir les recherches et les études.

Chapitre 2: Etat de l'art

Introduction

Dans cette partie, nous allons définir quelques concepts du domaine ensuite effectuer une étude générale sur les solutions existantes sur le marché et finir par une étude générale sur les méthodologies de développement.

I. Terminologies du domaine de travail à distance

1. Les bureaux virtuels

Un bureau virtuel est une suite d'outils interconnectés centralisant des données d'organisation, de communication et de collaboration, accessible à tout moment via différents appareils (ordinateur, Smartphone, tablette, etc.) et différentes interfaces (Web, IMAP, WebDAV, CalDAV, CardDAV, etc.)(1)

Concrètement, cette suite comporte des outils tels qu'un agenda, un carnet d'adresses, une messagerie, un stockage de documents, une liste de tâches, etc.

Les bureaux virtuels sont caractérisés par les points suivants :

- Accès universel : L'utilisateur peut accéder à ses données de la même manière au bureau, à domicile, sur la route, dans une salle de réunion, à l'hôtel, dans un cybercafé, etc.
- Ouverture : Alors que les logiciels traditionnels sont centrés sur une organisation, le bureau virtuel permet la collaboration inter-organisations.
- Modèle de location : Au lieu de payer une licence, l'utilisateur paie une redevance mensuelle ou annuelle. Certains éditeurs proposent une version de base gratuite.

Avec les logiciels gratuits en ligne, on peut aujourd'hui quasiment tout faire. Lire ses courriels, dialoguer avec ses amis, rédiger des documents, partager ses photos, ses vidéos... On peut même stocker ses fichiers, ses favoris Internet, ses contacts...

2. La virtualisation

La "virtualisation" représente globalement le fait de dématérialiser le comportement et les données d'un serveur ou d'une machine. En d'autres termes, cela revient à pouvoir regrouper plusieurs serveurs et/ou machines (Messagerie Windows, Comptabilité SAGE, Web sous Linux, Backup, BlackBerry...) sur un seul et même serveur physique.

Avant, le fait que des serveurs utilisent des OS différents, nécessitait des serveurs physique différents, d'autres nécessitaient deux serveurs physiques séparés par 'design'.

La 'Virtualisation' apportera donc :

- De grandes flexibilités d'administrations incomparables,
- Une meilleure utilisation des ressources (Puissance et Mémoire),
- Green IT,
- Une grande disponibilité,
- Des coûts en matériels diminués,

Le projet peut évoluer comme une ébauche sur le cloud computing ou l'informatique en nuage , comme perspective, pour cela, nous fournirons une définition de ce dernier.

3. Le Cloud computing

Le cloud computing représente un concept, apporté par la Virtualisation, et d'une certaine manière dérivé de celle-ci.

Il s'agit de pouvoir créer, supprimer, remplacer un service ou une application à la volée. Souvent des mécanismes de Virtualisation sont mis en œuvre par le tier pour délivrer le Cloud.

Le concept de cloud se décline donc à ce jour en 3 couches de services :

- L' IaaS : L'Infrastructure as a service (IaaS)

- PaaS : La Platform as a service (PaaS)

-SaaS : Le Software as a service (SaaS)

La figure ci-dessous présente une illustration du cloud computing

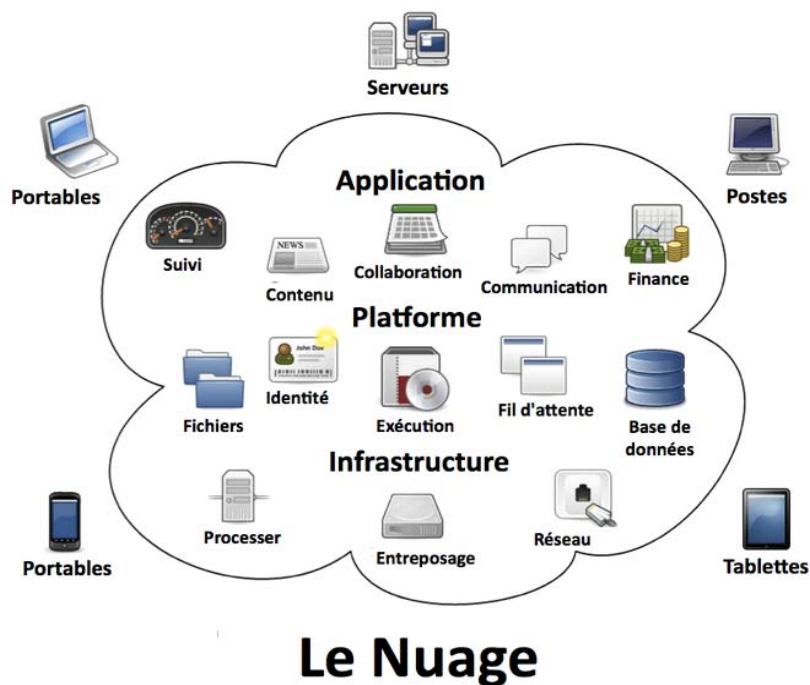


Figure 1-Illustration du cloud computing et ses services

II. Etude des solutions existantes

sur le marché, on trouve une panoplie de solutions qui offrent les fonctionnalités d'un bureau virtuel.

A ce niveau, nous allons présenter les bureaux virtuels rencontrés lors de nos recherches.

La recherche concerne le mode d'exécution, la gratuité et l'ouverture du code du logiciel.

1. Ulteo Open source(2)

« Ulteo Open Virtual Desktop » est une solution Open Source de bureau virtuel et de délivrance d'applications pour les entreprises. Elle permet aux départements informatique de fournir des applications et des bureaux utilisateur facilement et à un coût inférieur à d'autres solutions. La solution fonctionne pour des environnements Linux et Windows.

Les fonctionnalités de « Ulteo Open source » sont les suivantes :

Fonctionnalités :

- Un bureau Windows ou Linux avec un ensemble d'applications distantes Windows et/ou Linux
- Un portail applicatif, permettant d'exécuter des applications Windows et/ou Linux depuis des liens web
- Une publication d'Applications, permettant d'accéder aux applications distantes Windows et/ou Linux, de manière totalement intégrée au bureau local déjà existant, y compris au niveau du fenêtrage des applications.

Ce bureau virtuel est open source et gratuit.

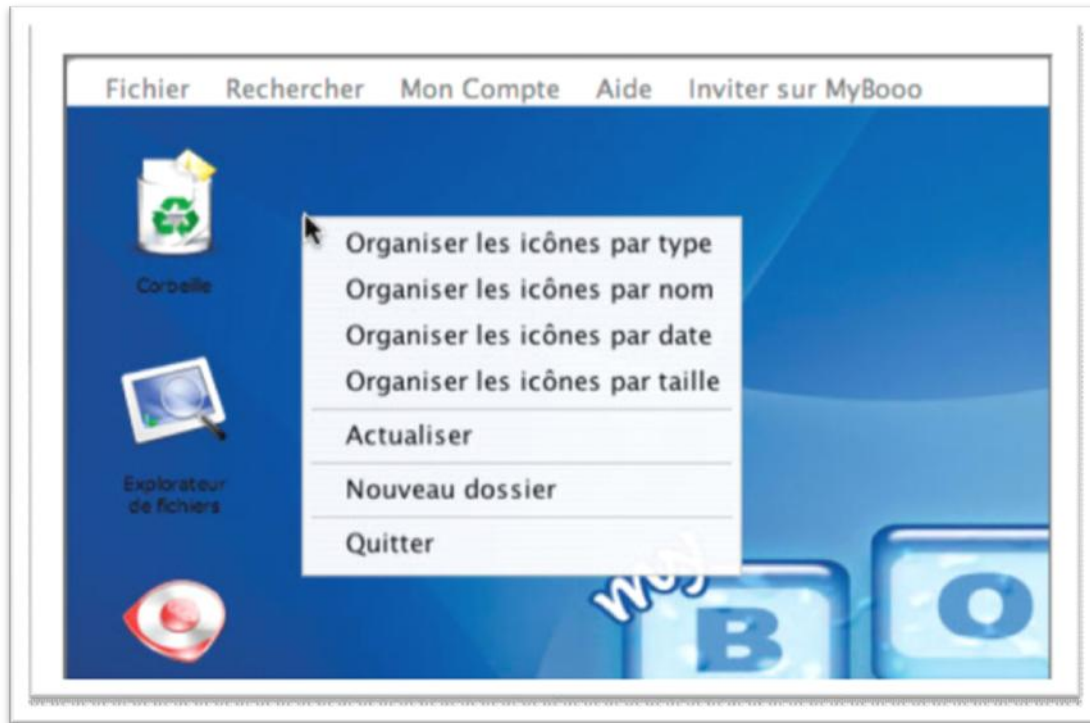


Figure 4-Bureau virtuel en mode Desktop Payant Myboo

3. Mynox(4)

MyNox est un bureau virtuel basé sur le web, et accessible avec un nom d'utilisateur et un mot de passe.

Les fonctionnalités de Mynox sont les suivantes :

- Stockage des documents (Espace de stockage 1 Gigaoctet).
- Partage des documents
- Ecoute de la musique
- Supporte du flux RSS
- Support des gadgets (Météo, Cours de devises, calendrier..)

- Support des éditeurs de documents (éditeurs de texte, classeurs, diaporamas)

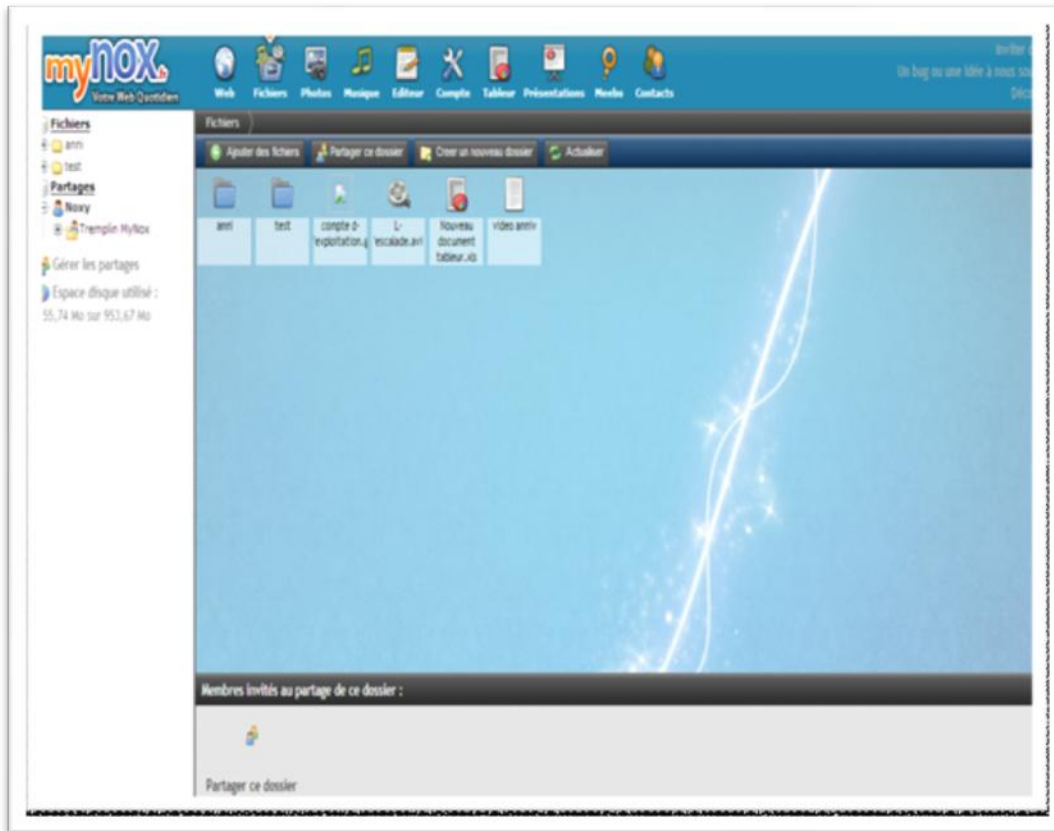


Figure 5-Bureau virtuel en mode web:Mynox

4. Zoho Docs (5)

Le Zoho Office Suite est une suite bureautique en ligne basé sur le Web qui contient un traitement de texte, tableurs, présentations, bases de données, la prise de notes, la gestion de la relation client (CRM), et d'autres applications développées par Zoho Corporation .

Zoho offre à ses abonnés les services suivants :

- Zoho Writer : Service pour la création des sites web.

- Zoho Sheet : Service pour créer les feuilles de calculs
- Zoho Show: Service pour les diapositives.
- Zoho Docs: Service pour la création et le stockage des documents texte.

La figure ci-dessous résume les différents services offerts par Zoho.

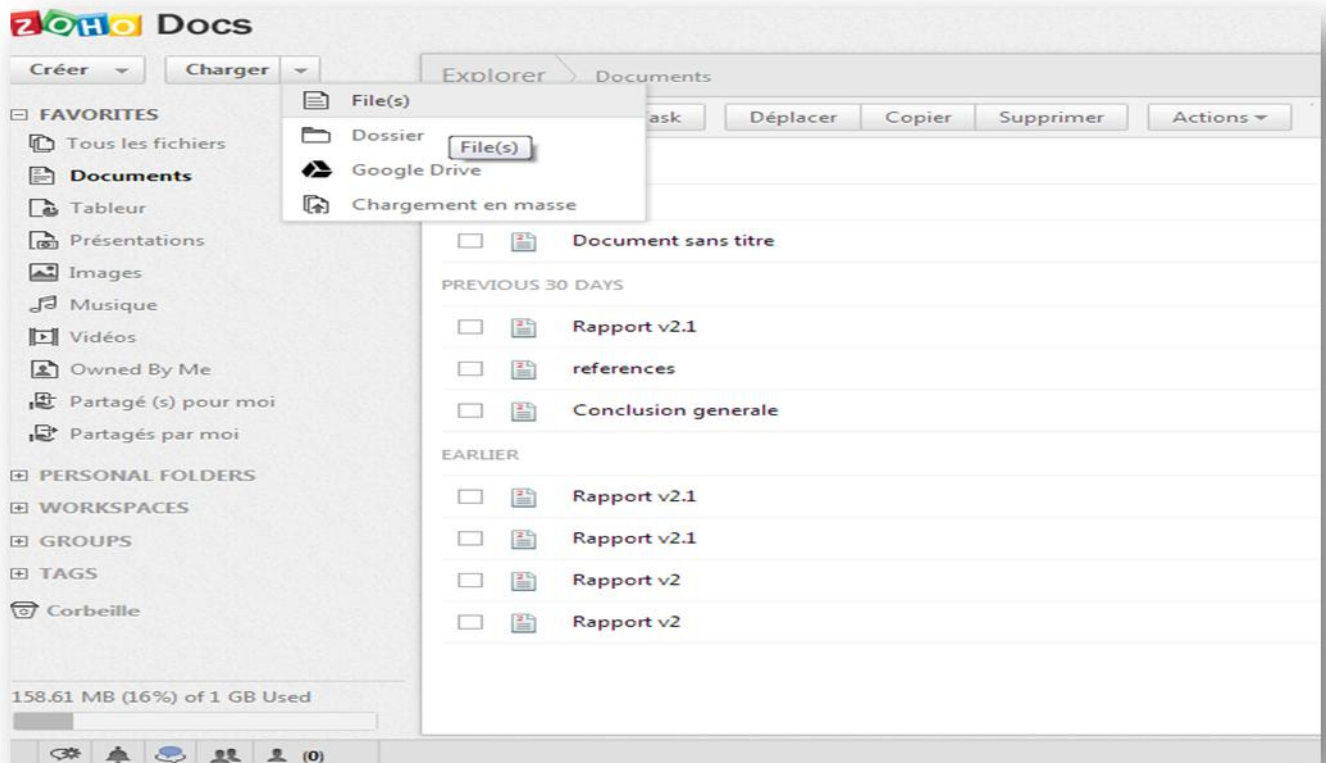


Figure 6-Services offerts par Zoho

Synthèse

Le tableau 1 récapitule les bureaux cités précédemment selon les critères suivants:

- Gratuité et/ou ouverture du code
- Facilité d'utilisation

- Usage (Personnel ou Professionnel)
- Le mode d'exécution

Solution	Critères			
	Gratuité et /ou ouverture du code	Facilité d'utilisation	Usage (Personnel/Professionnel)	Mode d'exécution
Ulteo	-Gratuit -Open source	-Utilisation difficile	-Professionnel	-Existe en version web -Existe en version Desktop -Existe en version mobile.
MyBooo	-Payant	-Utilisation facile	-Personnel	-Desktop
Mynox	-Gratuit	-Ergonomique -Utilisation facile	-Personnel	-web
Zoho Docs	-Existe en version gratuite et version payante	-Utilisation facile -Offre une suite bureautique en ligne complète	-Professionnel et personnel à la fois	-web

Tableau 1-Tableau récapitulatif des bureaux virtuels

Dans la partie ci-haut, nous avons effectué une étude de l'existant, maintenant, on consacrerà la seconde partie du chapitre à l'étude des méthodologies du développement.

III. méthodologies de développement

Une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client. Pour ce faire, nous partons d'un énoncé informel (le besoin tel qu'il est exprimé par le client, complété par des recherches d'informations auprès des experts du domaine fonctionnel, comme les futurs utilisateurs d'un logiciel), ainsi que de l'analyse de l'existant éventuel (c'est-à-dire la manière dont les processus à traiter par le système se déroulent actuellement chez le client).

On distingue deux familles de méthodes : Les méthodes agiles et les méthodes du processus unifié.

1. Les méthodes agiles (6)

Une méthode agile est une méthode de développement informatique permettant de concevoir des logiciels en impliquant au maximum le demandeur (client).

Les méthodes agiles reposent sur une structure commune (itérative, incrémentale et adaptative)

Les méthodes agiles utilisent un principe de développement itératif qui consiste à découper le projet en plusieurs étapes qu'on appelle "itérations".

Ces itérations sont en fait des mini-projets définis avec le client en détaillant les différentes fonctionnalités qui seront développées en fonction de leur priorité. Le chef de projet établit alors un macro planning correspondant aux tâches nécessaires pour le développement de ces fonctionnalités.

Parmi les méthodes agiles on cite :

- **Extrem programming(Xp)** (7)

Plus particulièrement orientée sur l'aspect réalisation d'une application. XP est adapté aux équipes réduites avec des besoins changeants.

L'Extreme Programming repose sur des cycles rapides de développement (des itérations de quelques semaines) dont les étapes sont les suivantes :

- Une phase d'exploration détermine les scénarios client qui seront fournis pendant cette itération
- L'équipe transforme les scénarios en tâches à réaliser et en tests fonctionnels
- Chaque développeur s'attribue des tâches et les réalise avec un binôme
- Lorsque tous les tests fonctionnels passent, le produit est livré

Le cycle se répète tant que le client peut fournir des scénarios à livrer. Généralement le cycle de la première livraison se caractérise par sa durée et le volume important de fonctionnalités embarquées. Après la première mise en production, les itérations peuvent devenir plus courtes (une semaine par exemple).

- **Scrum** (8)

Le principe de base de Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe d'une à quatre semaines, appelées sprints.

Chaque sprint possède un but à atteindre à partir duquel sont choisies les fonctionnalités à implémenter dans ce sprint. Un sprint aboutit toujours à la livraison d'un produit partiel fonctionnel.

L'utilisation de la méthodologie Scrum offre la possibilité de développer uniquement les fonctionnalités qui apportent une valeur ajoutée au produit, en toute transparence, avec le souci de la qualité et du respect des délais, avec un retour rapide de la part du client

2. Le processus unifié (PU) (7)

Le Processus unifié (PU ou UP pour Unified Process) est une méthode de prise en charge du cycle de vie d'un logiciel et du développement, pour les logiciels orientés objets. C'est une méthode générique, itérative et incrémentale.

Parmi les méthodes du processus unifié on cite :

RUP(10)

RUP est l'une des plus célèbres implémentations de la méthode PU permettant de donner un cadre au développement logiciel, répondant aux exigences fondamentales préconisées par les créateurs d'UML :

- Une méthode de développement doit être guidée par les besoins des utilisateurs
- Elle doit être centrée sur l'architecture logicielle
- Elle doit être itérative et incrémentale

2TUP(11)

2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens). Le processus s'articule ensuite autour de trois phases essentielles :

- **Une branche technique**
- **Une branche fonctionnelle**
- **Une phase de réalisation**

La branche fonctionnelle capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.

La branche technique capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système le sont indépendamment des fonctions à réaliser.

La phase de réalisation consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

3. Comparaison entre les méthodes

Il est très important de bien choisir la méthodologie qui répond parfaitement à ces besoins, s'adapte au mieux aux exigences du client et qui fournit l'ouvrage dans les plus brefs des délais.

Pour cela, il a fallu effectuer une étude comparative entre les méthodologies afin d'en sélectionner la meilleure qui s'adapte au projet.

Le tableau ci-dessous décrit méthodologies de développement ainsi que les points forts et les points faibles de chaque méthodologie.

Le critère de comparaison est l'aptitude de l'appliquer sur une équipe de développeurs à nombre réduit.

	Description	Points Forts	Points faibles
RUP (Rational Unified Process)	- Le RUP est à la fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans	-Itératif - Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes...	Coûteux à personnaliser - Très axé processus, au détriment du développement : peu de place pour le code et la technologie

	<p>un référentiel Web)</p> <ul style="list-style-type: none"> - adapté pour des projets qui comportent plus de 10 personnes 		
<p>2TUP (Two Track Unified Process)</p>	<ul style="list-style-type: none"> - S'articule autour de l'architecture - Propose un cycle de développement en Y - Cible des projets de toutes tailles 	<p>Itératif</p> <ul style="list-style-type: none"> - Fait une large place à la technologie et la gestion du risque - Définit les profils des intervenants, les livrables, les plannings, les prototypes. 	<p>Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement...</p>
<p>XP (eXtreme Programming)</p>	<p>méthode agile</p> <ul style="list-style-type: none"> - Ensemble de «Best Practices » de développement (Idéal pour le travail en groupe) - Cible des projets de moins de dix 	<p>Itératif</p> <ul style="list-style-type: none"> - Simple à mettre en œuvre - Fait une large place aux aspects techniques : 	<p>Une approche dénuée de modélisation</p> <ul style="list-style-type: none"> - Ne couvre pas les phases en amont et en aval au développement: capture des besoins,

	personnes	prototypes, règles de développement, tests...	support, maintenance, tests d'intégration... - Assez flou dans sa mise en œuvre
Scrum	<p>La méthode s'appuie sur le découpage d'un projet en « sprint », ainsi que l'auto-organisation de l'équipe de développement.</p> <p>-Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé, et contribue à augmenter la valeur d'affaires du produit.</p>		<p>-La méthode Scrum ne couvre aucune technique d'ingénierie du logiciel. Pour l'utiliser, il est nécessaire de la compléter avec des pratiques de qualité du logiciel. Par exemple, on pourra utiliser des pratiques issues de l'Extreme Programming,</p>

Tableau 2-Tableau comparatif entre les méthodologies de développement

4. Méthodologie adoptée : Scrum

Suite à l'étude comparative nous avons remarqué que les quatre méthodologies étudiées permettent le travail itératif pour un meilleur suivi de l'avancement global du projet, et chacune met l'accent sur une phase bien déterminée du projet.

Nous avons opté pour la méthodologie scrum car elle satisfait les conditions suivantes :

-Scrum convient aux équipes ayant un nombre de développeurs réduits. Ceci est le cas car le travail est individuel.

-Le client est impliqué dans le développement de l'application : La consultation du client est nécessaire dès l'achèvement d'une tâche.

-La progression des tâches s'effectue pendant une durée de développement courte.

4.1. Présentation de scrum

La méthodologie Scrum a été conçue pour améliorer grandement la productivité dans les équipes auparavant paralysés par des méthodologies plus lourdes. Le principe de base de Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe de une à quatre semaines, appelées sprints.

Chaque sprint possède un but à atteindre, défini par le directeur de produit, à partir duquel sont choisies les fonctionnalités à implémenter dans le sprint. Un sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel. Pendant ce temps, le Scrum Master a la charge de réduire au maximum les perturbations extérieurs et de résoudre les problèmes non techniques de l'équipe. Ce processus est illustré par la figure suivante

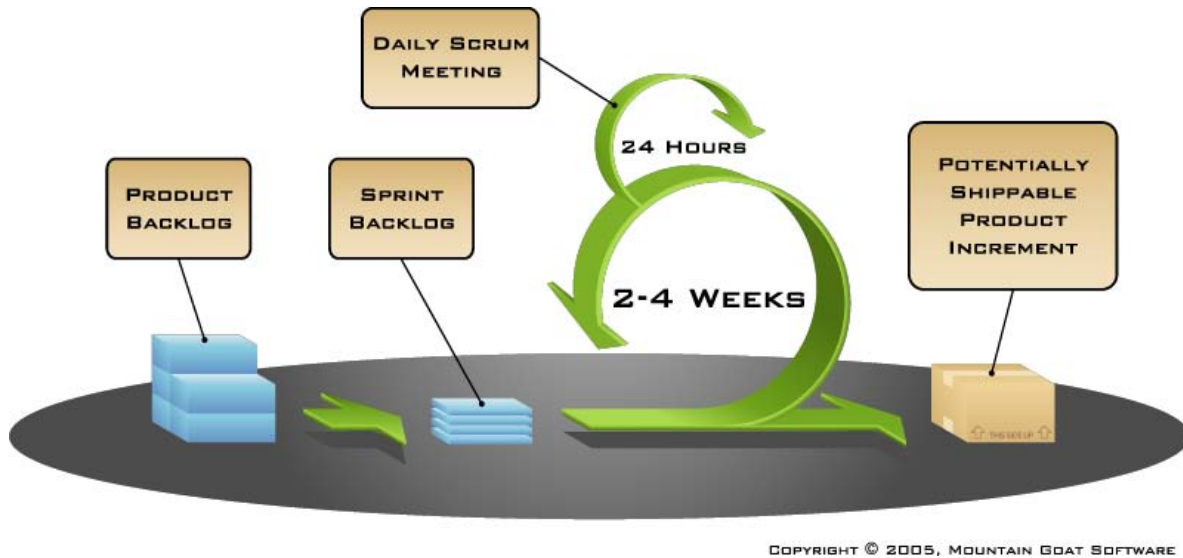


Figure 7 Architecture générale de Scrum

a) Les acteurs

On distingue plusieurs acteurs, dont :

Le directeur de produit (product owner) est le représentant des clients et des utilisateurs, et fait également parti de l'équipe.

Le ScrumMaster veille à l'application de la méthodologie Scrum au sein de l'équipe.

L'équipe qui contribue à la réalisation des fonctionnalités du projet (planification, développement, test et documentation). A noter que les membres de l'équipe travaillent tous ensemble : Chaque membre peut faire ainsi des propositions, exprimer idées et écoute les autres

b) Le processus

Tous les critères ou exigences du produit sont regroupées dans des journaux ou backlogs dont on distingue 2 types :

- Le backlog de produit ou < Product backlog > qui regroupe la liste des fonctionnalités du produit.

- Le backlog de sprint ou < sprint backlog > en fonction des fonctionnalités du produit, regroupe la liste des tâches qui devra être réalisées à l'itération en cours. Chaque tâche aurait fait l'objet d'une estimation préalable de charge par l'ensemble des membres de l'équipe afin d'estimer au mieux les tâches qui peuvent réaliser un sprint.

c) Planification

Le sprint : Dès le début d'un projet, la première planification permet de définir le périmètre de chaque itération appelé sprint. Chaque sprint dure quelques semaines et regroupe une liste de tâches (défini dans le backlog).

La mêlée quotidienne : De plus, elle est rythmée par ce qu'on appelle une mêlée quotidienne d'un quart d'heure qui consiste chaque jour avec les membres de l'équipe ainsi que le directeur de produit de se tenir au courant de l'avancement du projet, notamment en :

- Faisant le point sur le travail effectué la veille par chacun
- Définissant les tâches qui sont réalisées durant la journée
- Résolvant les éventuels problèmes qui avait ou qui pourrait être rencontré par chacun
- Le développement suit un processus itératif et incrémental : de nouvelles fonctionnalités sont rajoutées au produit.

d) La revue de sprint :

La fin d'un sprint aboutit à la réalisation d'un produit avec des fonctionnalités partielles avec la documentation associée. Dans la plupart des cas, cela conduit à une revue de sprint consistant à faire une démonstration de la réalisation du sprint devant le client afin de valider le travail réalisé et d'avoir un retour pour éventuellement ajuster le backlog de produit.

Conclusion :

Dans ce chapitre, nous avons défini les concepts du domaine, cité quelques exemples des solutions du marché choisi la méthodologie Scrum, que nous allons respecter durant les chapitres suivants.

En vue de suivre un avancement logique dans ce rapport, une analyse et spécification fera l'objet du prochain chapitre.

Chapitre 3 : Analyse et spécification des besoins

Introduction

Dans ce chapitre, nous allons d'abord identifier les besoins fonctionnels et non fonctionnels de l'application, ensuite, nous allons présenter le backlog du produit, suivi par une spécification des besoins et le prototype d'interfaces graphiques, et enfin terminer par une conclusion.

I. Identification des besoins

L'identification des besoins est une étape primordiale et importante afin de déterminer les besoins et les attentes exactes d'un client.

Ces besoins se déclinent en deux types de besoins : Fonctionnels et non fonctionnels

1. Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Les acteurs du système sont :

- L'employé:

L'utilisateur est un employé qui possède un compte et identifié par un nom d'utilisateur et un mot de passe qui lui permettent d'accéder à l'application et de bénéficier de ses services.

- L'administrateur

L'administrateur est la personne responsable de gérer la totalité du système.

2. Identification des besoins fonctionnels

Le système doit être capable de :

Fonctionnalité	Description
-Gérer les employés	Cette fonctionnalité permet à l'administrateur d'ajouter, modifier ou supprimer les employés du système.
-Gérer les documents	-Cette fonctionnalité permet à l'employé de gérer ses documents par la création , la modification ou la suppression.
-Importer et exporter les documents	
-Partager les documents	-Partager les documents avec un autre employé
-Personnaliser le bureau	-Cette fonctionnalité permet à l'employé de personnaliser le bureau dans la limite de changer la couleur de l'arrière-plan et la police ou la position des éléments
-Calendrier/Météo/Infos	

Tableau 3tableau illustrant les besoins fonctionnels

3. Capture des besoins non fonctionnels

Les besoins non fonctionnels représentent les exigences implicites auquel le système doit répondre. Parmi ces besoins on cite :

✓ **Sécurité**

La solution proposée permet à l'utilisateur une navigation sécurisée.
Elle n'est accessible qu'avec une authentification.

✓ **Haute disponibilité**

L'application doit être opérationnelle tous les jours 7j/7 24h/24.

✓ **Ergonomie**

L'application doit satisfaire les critères de l'ergonomie suivants : La lisibilité, le guidage et la facilité d'utilisation.

✓ **Portabilité**

L'application doit être accessible via n'importe quel navigateur

II. Backlog du produit

Dans cette section, nous allons définir le backlog du produit qui servira comme liste d'exigences.

1. Définition du backlog du produit

Le tableau 4 démontre le backlog de notre application décortiqué en sprints et User Stories. Chaque User Story est caractérisé par un ID, une priorité et une estimation de la durée de sa réalisation.

Story ID	Nom	Priorité	Estimation de la durée(jours)
1	Gestion des employés	1	21
2	Gestion des documents	2	21
3	Gestion de versions	3	7
4	Gestion de partages	4	5
5	Personnalisation du bureau	5	7

Tableau 4 Définition du backlog du produit

I. Spécification des besoins

Dans cette section, nous allons présenter le diagramme de cas d'utilisation global qui traduit d'une façon semi-formelle les fonctionnalités du système, ensuite, nous compléterons les diagrammes par une description textuelle.

1. Diagramme de cas d'utilisation général

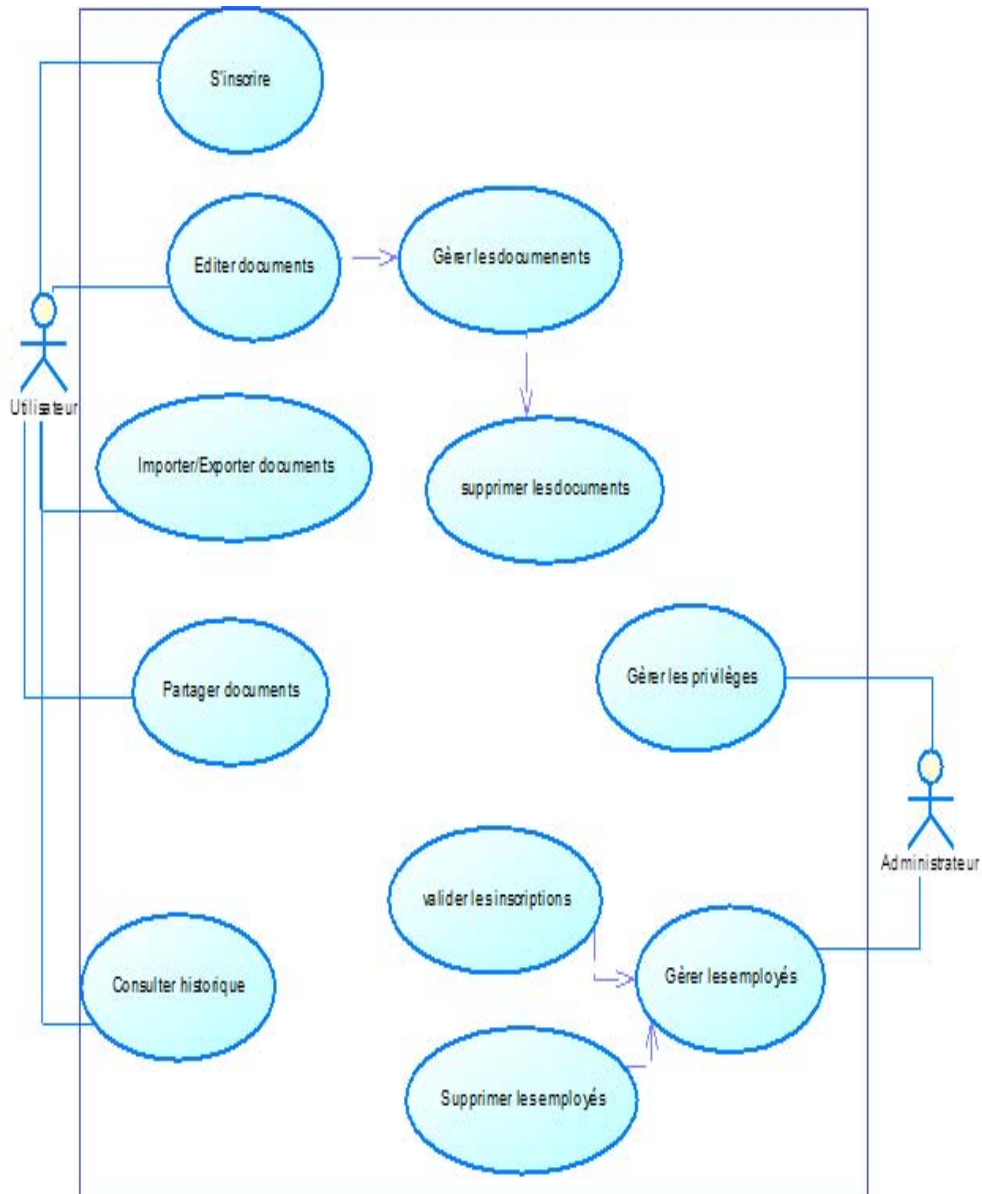


Figure 8-Diagramme de cas d'utilisation global

D'après le diagramme, l'employé en se connectant, peut importer des documents à partir de son poste de travail physique ou les exporter à partir de l'application vers sa machine. Il peut également éditer les documents, créer, renommer, modifier le contenu ou les supprimer. Il peut partager ses documents avec les autres employés et consulter son historique. L'administrateur, son rôle consiste gérer les employés, par exemple, valider les inscriptions de ces derniers ou les supprimer du système. Il peut aussi gérer les privilèges des employés.

1. Description détaillée du diagramme du cas d'utilisations

a) Cas d'utilisation : « Gérer les employés »

Résumé	Ce cas d'utilisation permet d'ajouter, modifier ou supprimer un employé.
Acteur	Administrateur
Précondition	Administrateur authentifié
Scénario nominal	
Ajout d'un employé	1-L'administrateur sélectionne l'employé à ajouter 2- Le système enregistre l'employé
Modification d'un employé	1-L'administrateur sélectionne l'employé à modifier 2- Le système met à jour les informations de l'employé
Suppression d'un employé	1-L'administrateur sélectionne l'employé à retirer 2- Le système annule l'adhésion de l'employé

Tableau 5 Description du cas d'utilisation "Gérer les employés"

b) Cas d'utilisation : « Gérer les documents »

Résumé	Ce cas d'utilisation permet de créer, modifier ou supprimer un document.
Acteur	Employé
Précondition	Employé authentifié
Scénario	
Ajout d'un document	<p>1-L'employé crée un document.</p> <p>2- Le système retourne l'interface appropriée</p> <p>3-L'employé accorde un nom au document</p> <p>4-Le système sauvegarde le document et affiche les détails</p>
Modification d'un document	<p>1-L'employé sélectionne le document à modifier</p> <p>2- Le système vérifie si le document est accessible en lecture et écriture</p> <p>3- Le système retourne le document à modifier</p> <p>4-L'employé effectue la modification</p> <p>5-Le système enregistre la modification</p>

Suppression d'un document	<p>1-L'employé sélectionne le document à supprimer</p> <p>2- Le système retire le document</p>
---------------------------	--

Tableau 6 Description textuelle de cas d'utilisation "Gérer les documents

c) Cas d'utilisation « Importer/Exporter les documents »

Résumé	Ce cas d'utilisation permet d'importer ou exporter des documents à l'espace de travail
Acteur	Employé
Précondition	Employé authentifié
Scénario	
Importer un document	<p>1-L'employé sélectionne un document à partir de sa machine physique</p> <p>2- Le système retourne l'interface appropriée</p> <p>3-L'employé valide l'ajout du document</p> <p>4-Le système sauvegarde le document et affiche les détails</p>
Exporter un document	<p>1-L'employé sélectionne le document à exporter sur sa machine physique</p> <p>2- Le système vérifie si le document est accessible en lecture et</p>

	<p>écriture</p> <p>3- Le système retourne le document à modifier</p> <p>4-L'employé valide l'export</p> <p>5-Le système exporte le document.</p>
--	--

Tableau 7 Description textuelle du cas d'utilisation "Importer/Exporter les documents"

d) Cas d'utilisation « Partager les documents »

Résumé	Ce cas d'utilisation permet de partager les documents avec les employés
Acteur	Employé
Précondition	Employé authentifié
Scénario	
	<p>1-L'employé sélectionne un document à partager</p> <p>2- Le système retourne l'interface appropriée</p> <p>3-L'employé sélectionne le destinataire</p> <p>4-Le système envoie le document au destinataire</p>

Tableau 8 Description textuelle du cas d'utilisation "Partager les documents"

e) Cas d'utilisation « Consulter historique »

Résumé	Ce cas d'utilisation permet d'afficher l'historique des opérations sur le système
Acteur	Employé
Précondition	Employé authentifié
Scénario	
	<p>1-L'employé sélectionne l'interface de la consultation historique</p> <p>2- Le système affiche l'historique utilisateur</p>

Tableau 9 Description textuelle du cas d'utilisation "Consulter historique"

III.Prototype de l'interface graphique

Dans cette section, nous allons définir le prototype ou la charte graphique de l'application.

Le prototypage est la clé de voûte du développement itératif. Les prototypes se différencient selon leur degré de réalisme. Un prototype présente la partie visible du logiciel, c'est à dire les fenêtres et les pages web de l'application. Le prototype est imposé par le client.(12)

Les principaux composants graphiques de l'application seront les suivants :

- Un menu de démarrage
- Un explorateur de fichier (Arborescence) : Contient la liste des fichiers et dossiers en possession de l'employé

- Des fenêtres modales : Retournent à l'employé des messages d'erreurs ou des messages d'informations.
- Des boutons : Servent à valider ou annuler une action, ou à appeler un autre composant
- Des boutons radios
- Des composants HTML comme les liens hypertextes
- Des validateurs de champs de saisies.

Les figure 10, 11 et 12 présentent respectivement le prototype de l'interface d'accueil , l'interface de l'explorateur de fichiers et l'interface de gestion d'employés.

a) Interface d'accueil

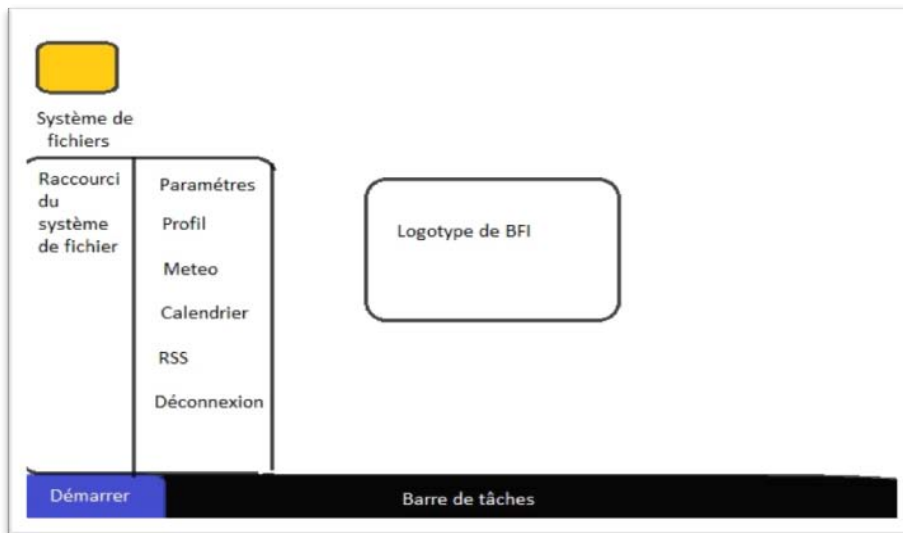


Figure 9 Prototype de l'interface d'accueil

Cette figure présente le prototype de l'interface d'accueil. Le milieu de la page est réservé à l'affichage des résultats de traitement, tandis que la partie gauche est réservée au menu démarrer et les menus qu'il héberge. Le bas de page est réservé à la barre de tâches.

b) Interface explorateur de fichiers

La figure suivante présente l'emplacement et la taille du système de fichiers ainsi que les éléments qui le

composent.

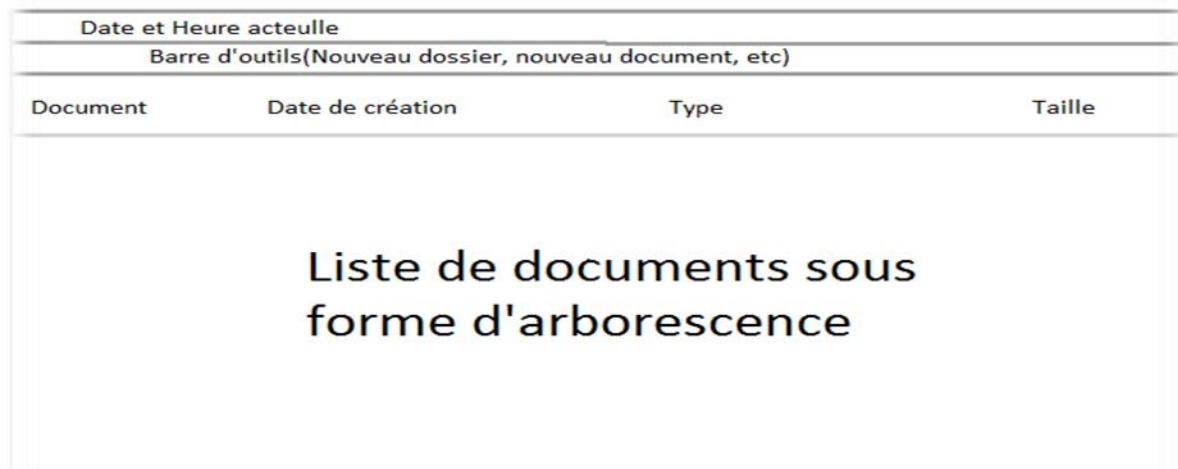


Figure 10- Prototype de l'interface graphique du système de fichiers

c) Interface de gestion d'utilisateurs

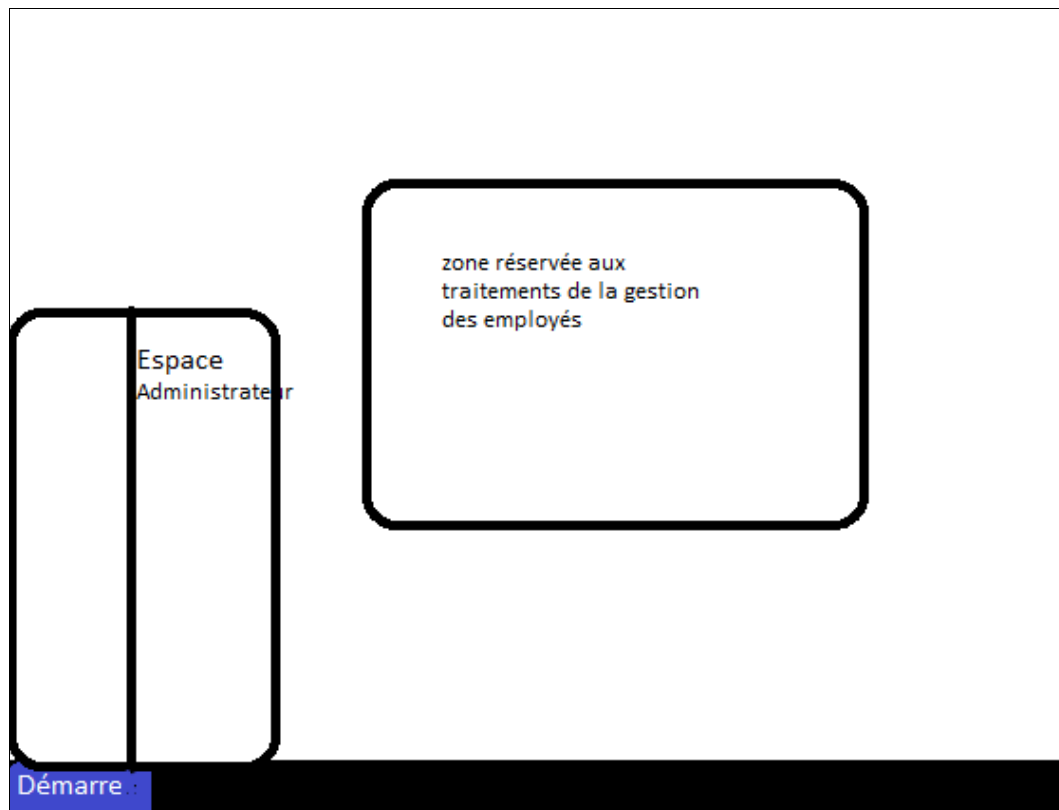


Figure 11 Interface de gestion d'employé

La figure 11 représente la zone réservée à la gestion des utilisateurs dans l'application lorsque l'administrateur accède à l'interface : « Gestion des employés ».

Conclusion

Ce chapitre a permis de détailler les spécifications du projet, d'identifier les acteurs et d'élaborer les diagrammes de cas d'utilisation. Par la suite, nous avons établi le prototype des interfaces graphiques. A présent nous sommes prêts à passer à l'étape de conception dans le chapitre suivant.

Chapitre 4 : Conception

Introduction

Après avoir spécifié les besoins fonctionnels et non fonctionnels de l'application, nous allons aborder la conception de différentes parties du projet moyennant des diagrammes proposés par le standard UML. Cette phase de conception permet de décrire de manière non ambiguë le fonctionnement futur du système, afin d'en faciliter la réalisation.

1. Architecture de l'application

Les choix architecturaux d'une application sont décisifs dès lors qu'ils interviennent sur les performances, l'évolutivité, le temps de développement, et bien sûr le coût. Aujourd'hui nous parlons d'une séparation des applications en différentes couches, et nous parlons alors d'applications multi niveaux.

L'architecture de notre système est une architecture multi niveaux à base de composants respectant le paradigme JEE. Elle est constituée des couches suivantes modélisées dans la figure ci-dessous. Il s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles dont le rôle est clairement défini :

La présentation des données : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;

Le traitement métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;

L'accès aux données persistantes : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Dans cette approche, les couches communiquent entre elles au travers d'un « modèle d'échange », et chacune d'entre elles propose un ensemble de services rendus. Les services d'une couche sont mis à disposition de la couche supérieure. On s'interdit par conséquent qu'une couche invoque les

services d'une couche plus basse que la couche immédiatement inférieure ou plus haute que la couche immédiatement supérieure (chaque couche ne communique qu'avec ses voisins immédiats).

Le rôle de chacune des couches et leur interface de communication étant bien définis, les fonctionnalités de chacune d'entre elles peuvent évoluer sans induire de changement dans les autres couches. Cependant, une nouvelle fonctionnalité de l'application peut avoir des répercussions dans plusieurs d'entre elles. Il est donc essentiel de définir un modèle d'échange assez souple, pour permettre une maintenance aisée de l'application.

Ce modèle d'architecture 3-tiers a pour objectif de répondre aux préoccupations suivantes :

- **Allègement du poste de travail client**
- **Prise en compte de l'hétérogénéité des plates-formes** (serveurs, clients, langages, etc.) ;
- **Amélioration de la sécurité des données**, en supprimant le lien entre le client et les données. Le serveur a pour tâche, en plus des traitements purement métiers, de vérifier l'intégrité et la validité des données avant de les envoyer dans la couche de données.

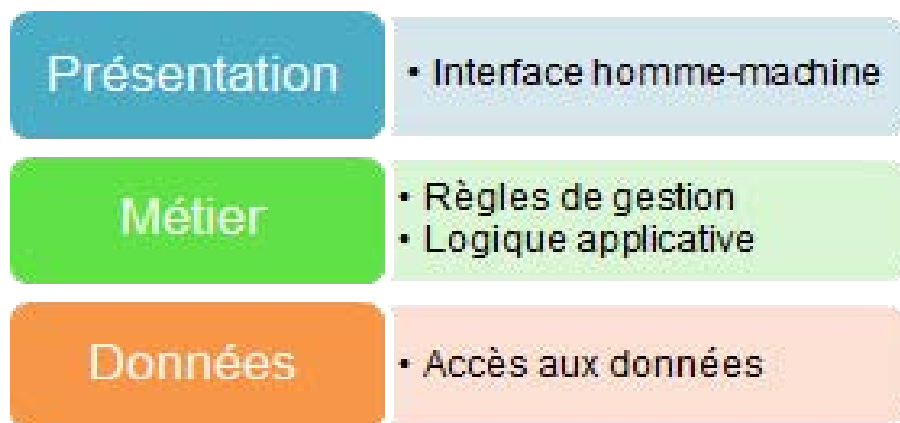


Figure 12-Architecture 3 tiers

II. Patron de conception MVP

MVP est un dérivé du modèle modèle-vue-contrôleur (MVC) (13), également utilisé principalement pour construire des interfaces utilisateur.

En MVP du présentateur suppose la fonctionnalité du "middle-man" (joué par le contrôleur dans MVC). En MVP, toute la logique de présentation est poussée à l'animateur. Finalement, le modèle de MVP est un modèle de conception d'interface utilisateur conçue pour faciliter les tests unitaires automatisés et améliorer la séparation des préoccupations dans la logique de présentation:

- Le modèle est une interface définissant les données à afficher ou autrement donné suite à l'interface utilisateur.
- La vue est une interface passive qui affiche des données (le modèle)
- Le présentateur agit sur le modèle et la vue. Il récupère les données de référentiels (le modèle) et des formats pour l'affichage dans la vue.

Le patron MVP est implémenté par le framework Google Web Toolkit. La figure 13 présente les composants GWT ainsi que leurs rôles respectifs selon le patron MVP.

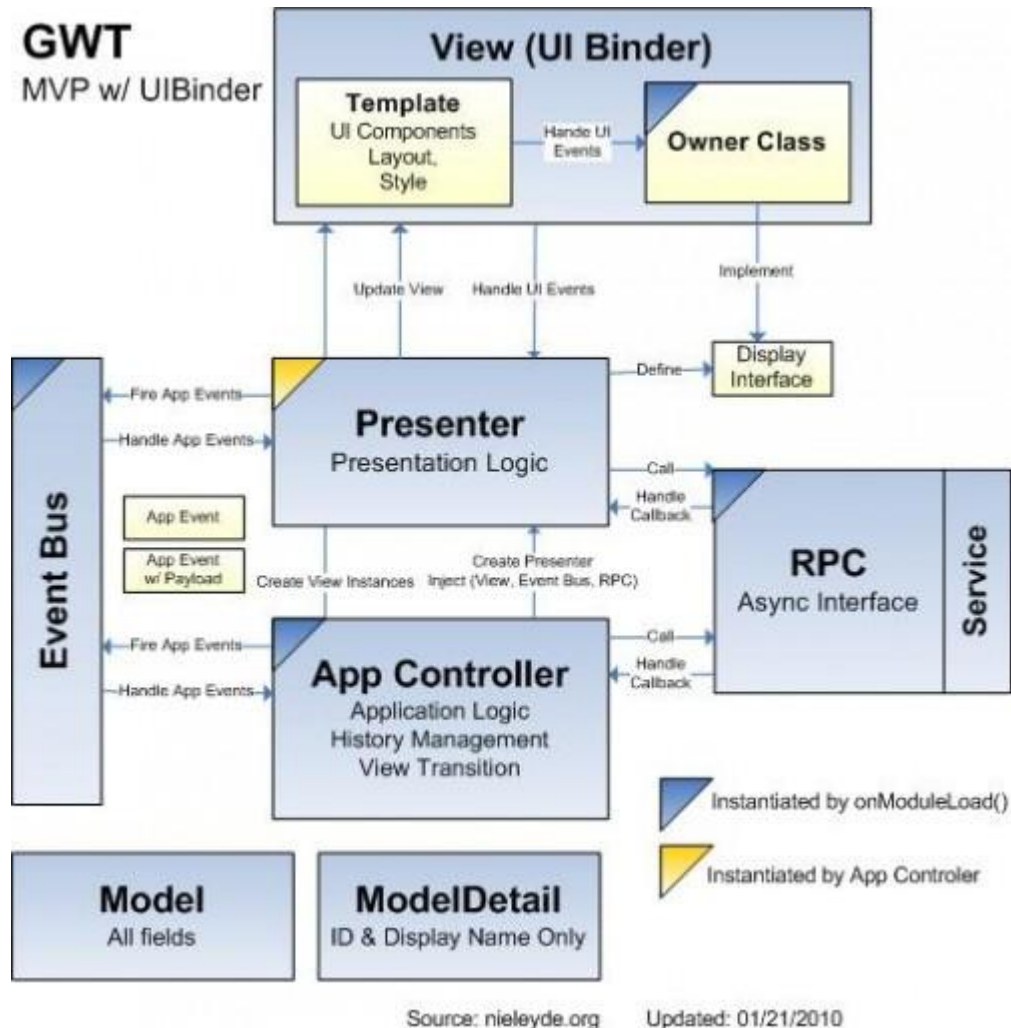


Figure 13-Model View Presenter (MVP) [Source: nielyde.org]

III. ETUDE STATIQUE

Dans cette partie, nous allons représenter une perspective de l'application à travers quelques diagrammes UML qui nous serviront par la suite de support pour la phase de réalisation.

Pour la modélisation statique de notre système nous allons présenter le diagramme de classes , ensuite, utiliser le découpage en packages ensuite les décortiquer ces derniers en détaillant chaque classe des packages.

1. Diagramme de classes

La figure 14 présente le diagramme de classes correspondant à l'application.

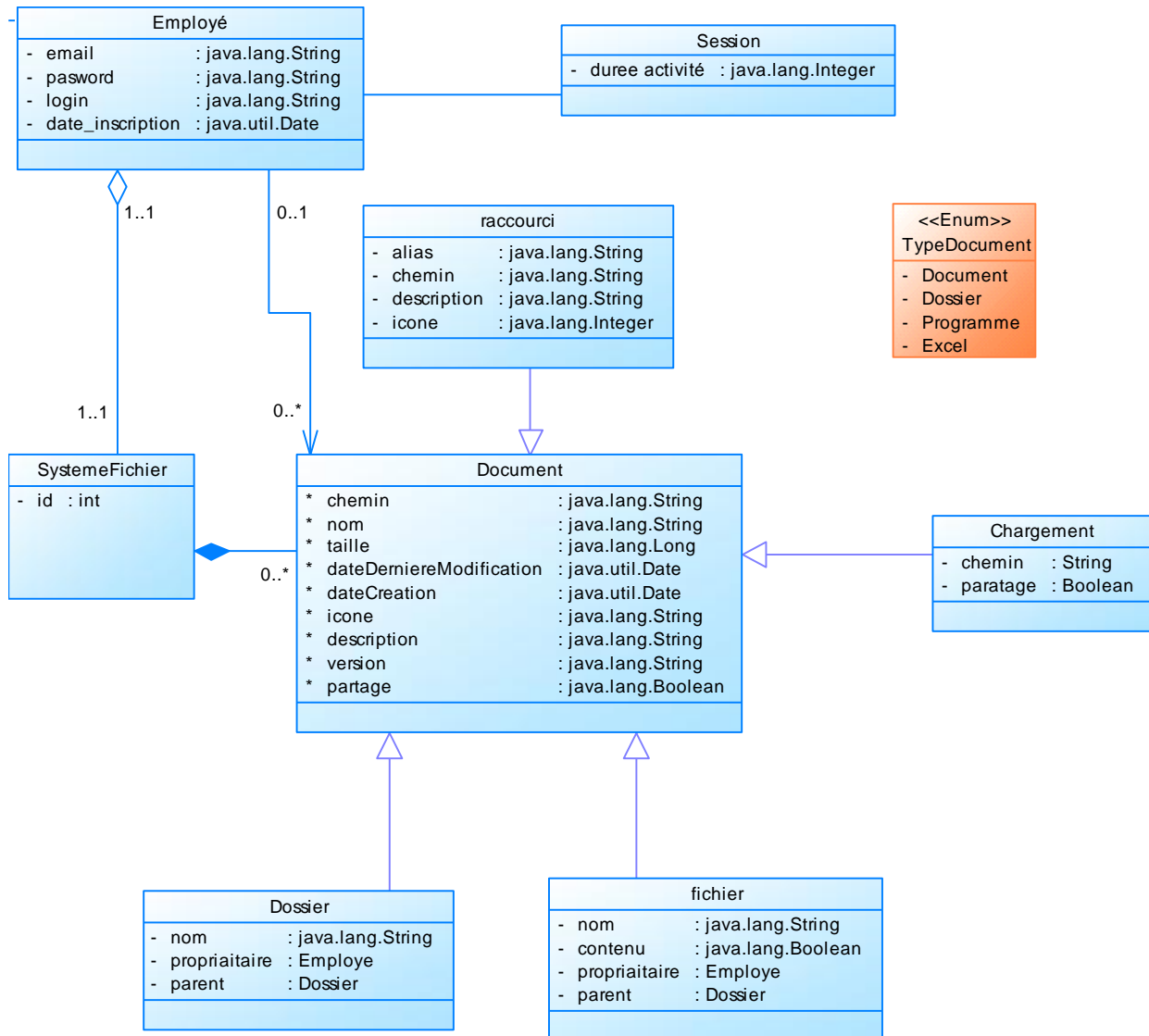


Figure 14 Diagramme de classes

Les différentes classes du diagramme de classes sont les suivantes :

- « Employé » : Cette classe contient les informations d'un employé utilisateur de l'application.
- « SystemeFichier » : Cette classe désigne tout l'espace de travail qui appartient à l'utilisateur. « SystemeFichier » est composé de zéro ou plusieurs « documents ».

- « Document » : Cette classe contient les attributs nécessaires d'un document. « Document » possède quatre classes filles « Dossier », « Fichier », « Chargement » et « Raccourci ».
- « Dossier » : Cette classe contient les attributs d'un dossier tels que : nom, propriétaire et parent.
- « Fichier » : Cette classe est caractérisée par l'attribut contenu.
- « Chargement » : Cette classe désigne les fichiers chargés à partir de l'extérieur vers l'application.
- « raccourci » : Cette classe désigne un raccourci d'un dossier ou d'un fichier.
- « Session » : Cette classe contient les informations de la connexion de l'employé.
- « TypeDocument » : Est une énumération qui représente le type d'un document.

2. Découpage en packages

Dans cette partie, nous allons découper les packages selon les exigences du pattern MVP.

La figure 15 présente les packages qui constituent notre application ainsi que les relations entre eux.

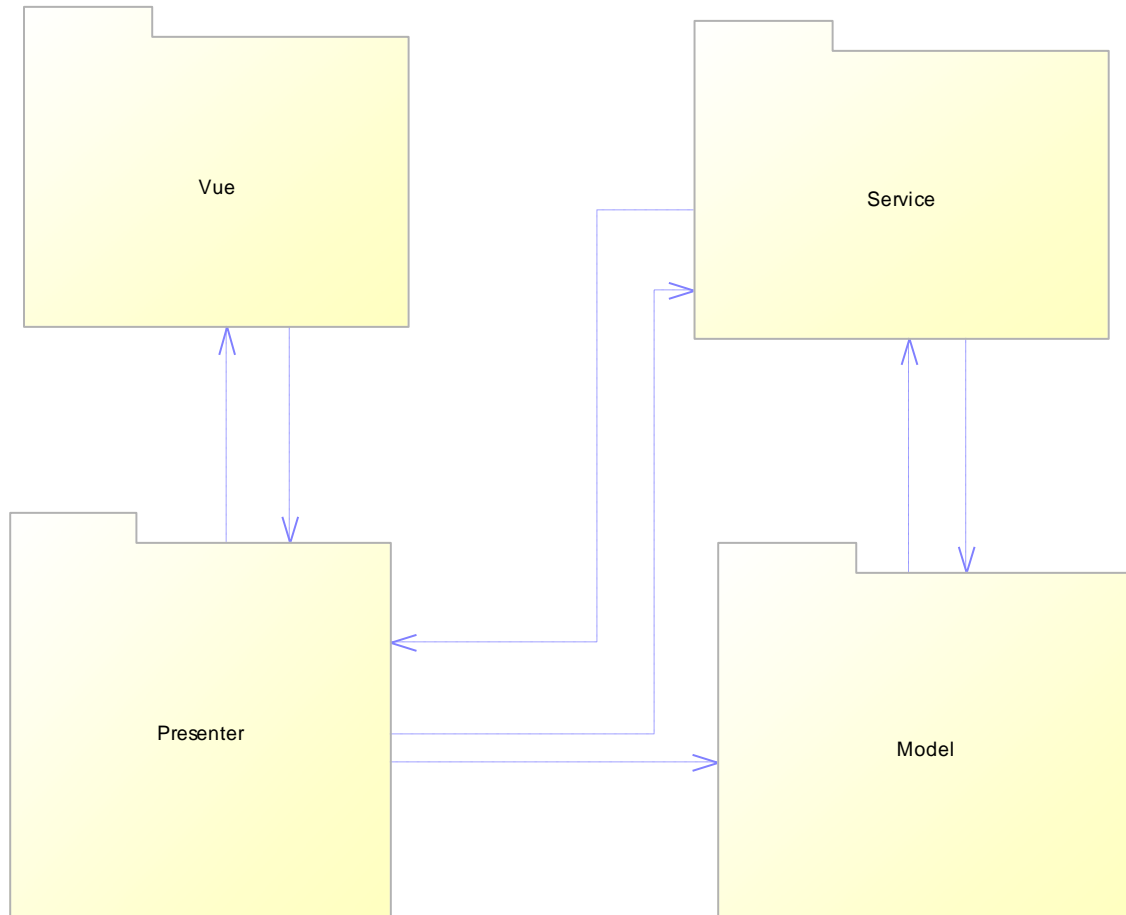


Figure 15 Diagramme de packages

- Le package « **Vue** » regroupe l'ensemble des classes qui constituent les interfaces graphiques. Ces classes sont toutes écrites en Java.
- Le package « **Presenter** » contient les classes chargées de la communication entre les classes des packages « **vue** » et « **Model** ».
- Le package « **Service** » contient les opérations « **CRUD** » qui agissent sur les classes du package « **Model** ».

Maintenant, nous allons nous focaliser sur chacun des packages et citer les classes qui le composent.

a) Package « Model »

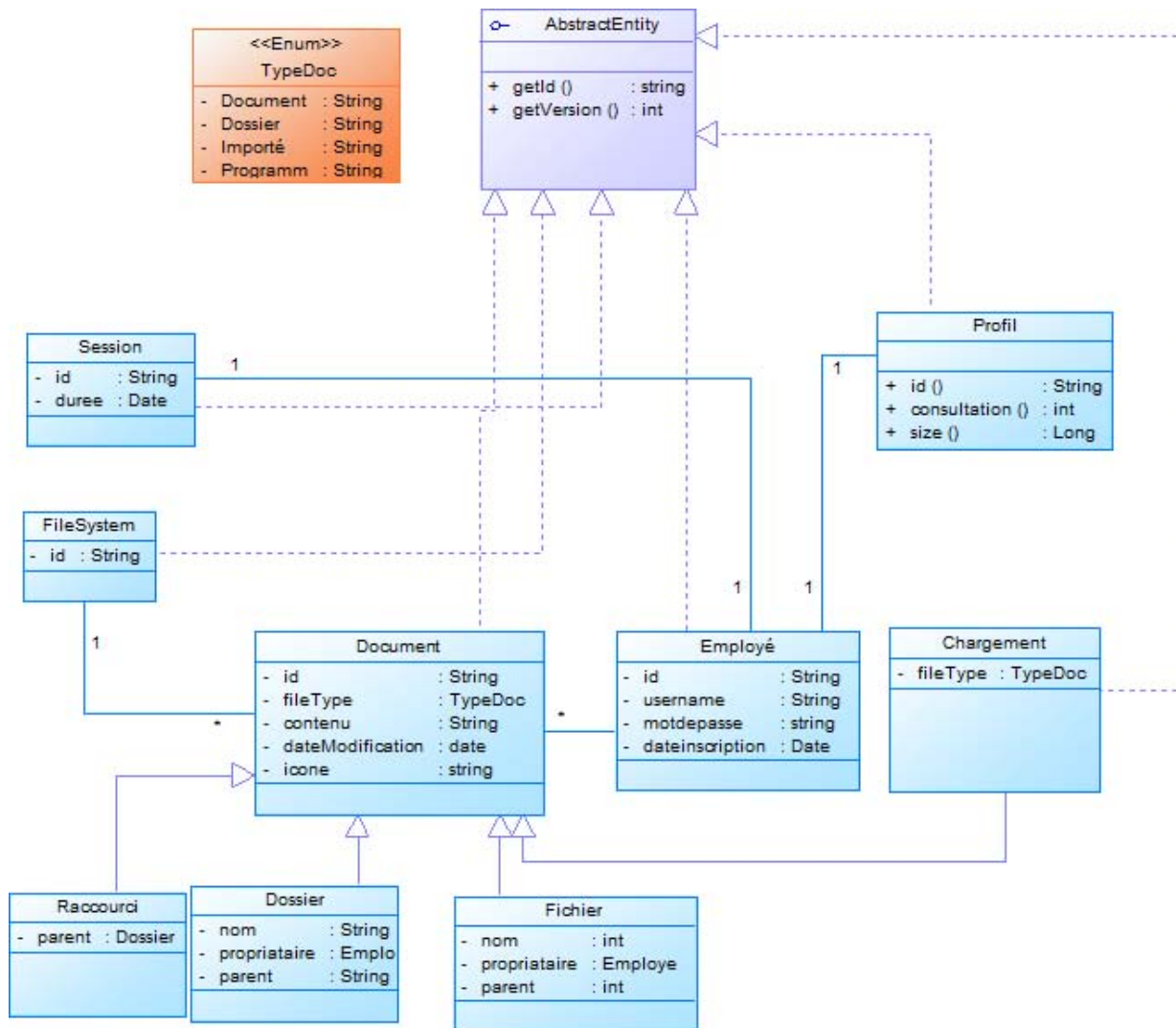


Figure 16package "Model"

La figure 16 représente les classes du package « Model ». Chaque classe représente une donnée gérée par l'application. Ces classes sont les suivantes :

- « AbstractEntity » : Il s'agit d'une interface qui contient deux méthodes getId() et getVersion(). L'utilisation de cette interface est importante lors de la création des entités car, une entité est caractérisée par un id et une version. « AbstractEntity » permet aux entités de ne pas changer d'état lors de leur sérialisation.

En effet, les classes « Employé », « Document », « FileSystem », « Profil », « Session » et « chargement » implémentent « AbstractEntity ».

- « Employé » :
- « FileSystem »
- « Document »
- « Session »
- « Profil »
- « Session »
- « Chargement »
- « Fichier »
- « Dossier »
- « Raccourci »
- « TypeDoc »

b) Package « View »

Les classes du package view sont données par la figure 16.

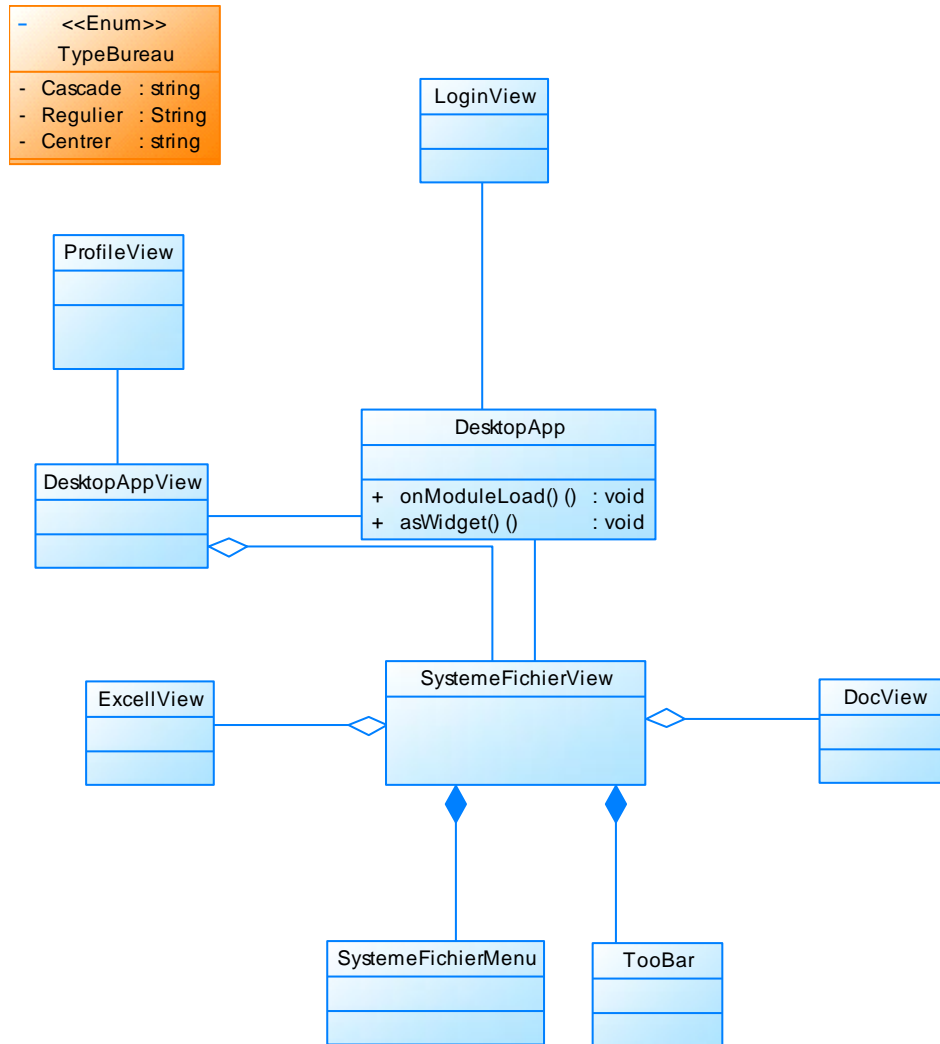


Figure 17 Classes du package "View"

Les classes ci-dessus sont responsables de l’affichage des vues. Il s’agit des interfaces graphiques chargées de retourner le résultat du traitement.

- La classe « DesktopApp » est l’interface principale. Grâce à la méthode « onModuleLoad() », toutes les interfaces reliées sont appelées. Quant à la méthode asWidget(), elle retourne les composants. Avec l’aspect qu’il a été défini par l’utilisateur.

- La classe « DesktopAppView » dépend de la classe « SystemeFichierView » relative à l’affichage du système de fichier. Cette dernière dépend de « ExcelView », « DocView » et est composée des deux classes « SystemeFichierMenu » et « ToolBar ».
- « SystemeFichierMenu » : Cette classe contient un menu contextuel.
- « ToolBar » : Cette classe contient une barre d’outils avec des boutons et des icônes.
- « ProfileView » : Cette classe se charge de l’affichage des informations du profil de l’employé.
- « ExcelView » : Cette classe offre une fenêtre d’édition de classeurs.
- « DocView » : Cette classe offre une fenêtre d’édition de fichiers texte.
- « LoginView » : Cette classe se charge de l’affichage du formulaire réservé à l’authentification.

c) package « Presenter »

La composition du package « Presenter » est donnée par la figure 18.

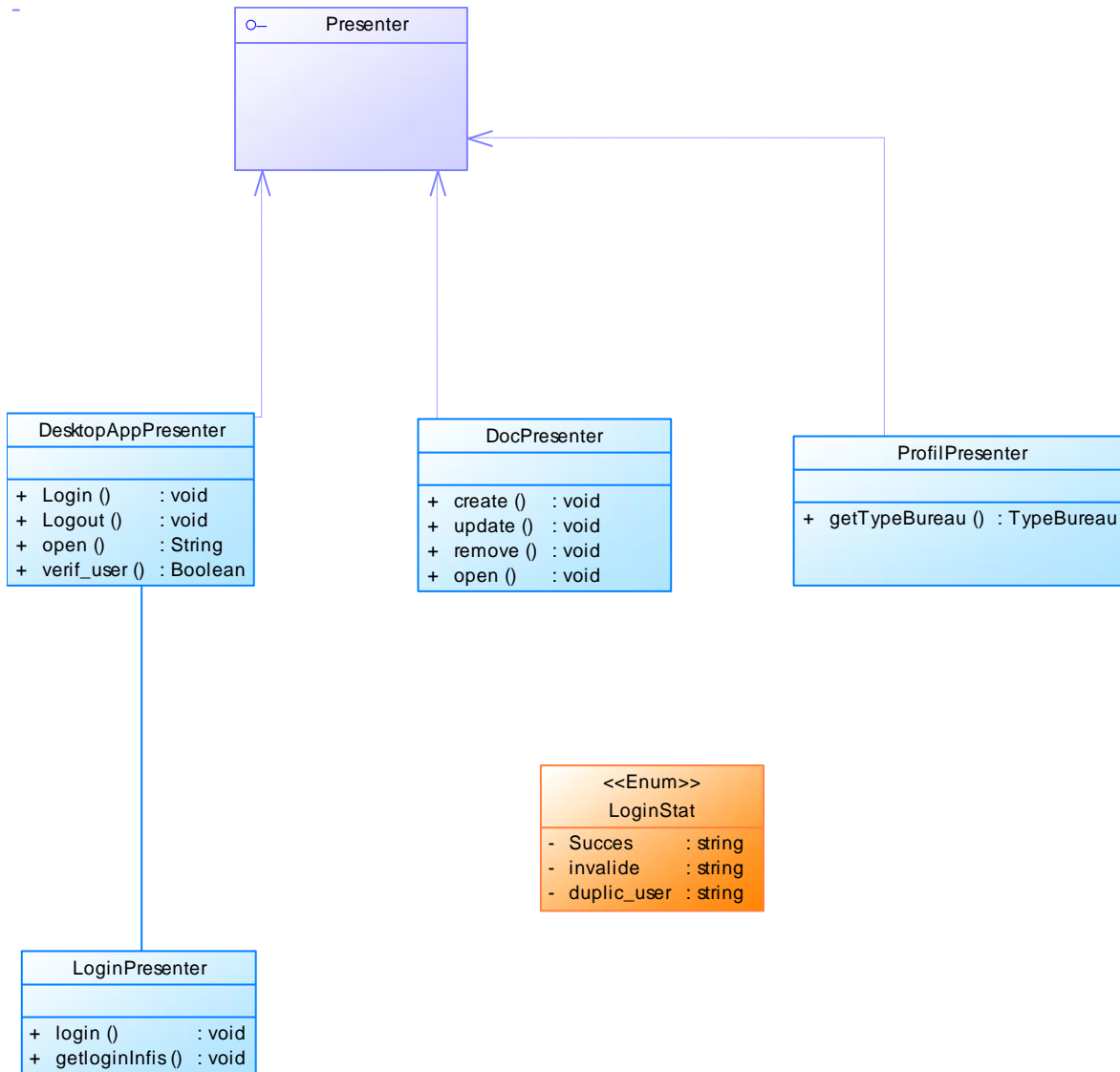


Figure 18- package "Presenter"

Les classes ci-dessus, sont les classes qui récupèrent les données du modèle pour l'affichage dans la vue.

Pour chaque vue, nous associons un presenter pour les faire communiquer ensemble.

D'autre part, nous définissons :

- « Presenter » : Cette interface implémentée par tous les autres presenters du package.
- « LoginStat » : Il s'agit d'une énumération qui définit les états que peut prendre un login lors de la tentative d'une connexion.

d) Package « Service »

Les classes de ce package sont données par la figure 19.

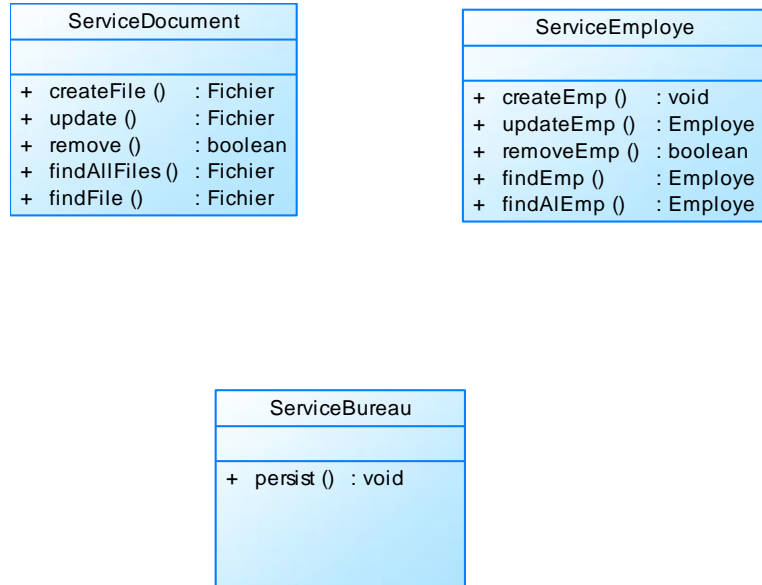


Figure 19 Classe Service du package Service

Le package « Service » est composée de trois classes « ServiceDocument », « ServiceEmploye » et « ServiceBureau » .

- « ServiceDocument » se charge des opérations « CRUD » sur l'entité « Document ». Ces opérations sont l'ajout d'un document(CreateFile), la modification d'un employé(Update), la suppression(Remove) et la recherche(findAllFiles et findFile).
- « ServiceEmp » se charge des opérations « CRUD » sur l'entité « Employé ».
- « ServiceBureau » se charge de sauvegarder les changements et la personnalisation du bureau.

III. ETUDE DYNAMIQUE

Dans cette partie, nous présentons les principaux scénarios de l'application via des diagrammes de séquences objets.

Pour se faire, nous nous sommes appuyés sur les stéréotypes de Jacobson(15). En effet, ceux-ci permettent de mieux décrire le rôle de chacune des classes de l'application.

Les entités de Jacobson sont définies comme suit :

- Les vues « **Boundary** » : Permettent les interactions entre l'application et l'application.
- Les control « **Controller** » peuvent interagir avec les boundary, les entités, ou d'autres contrôleurs.
- Les entités « **Entity** » ne peuvent interagir qu'entre elles

1. Scénario : «Authentification»

La figure 20 présente le diagramme de séquence objet relatif au cas d'utilisation « Authentification ».

Nous allons ensuite présenter les diagrammes de chaque sprint.

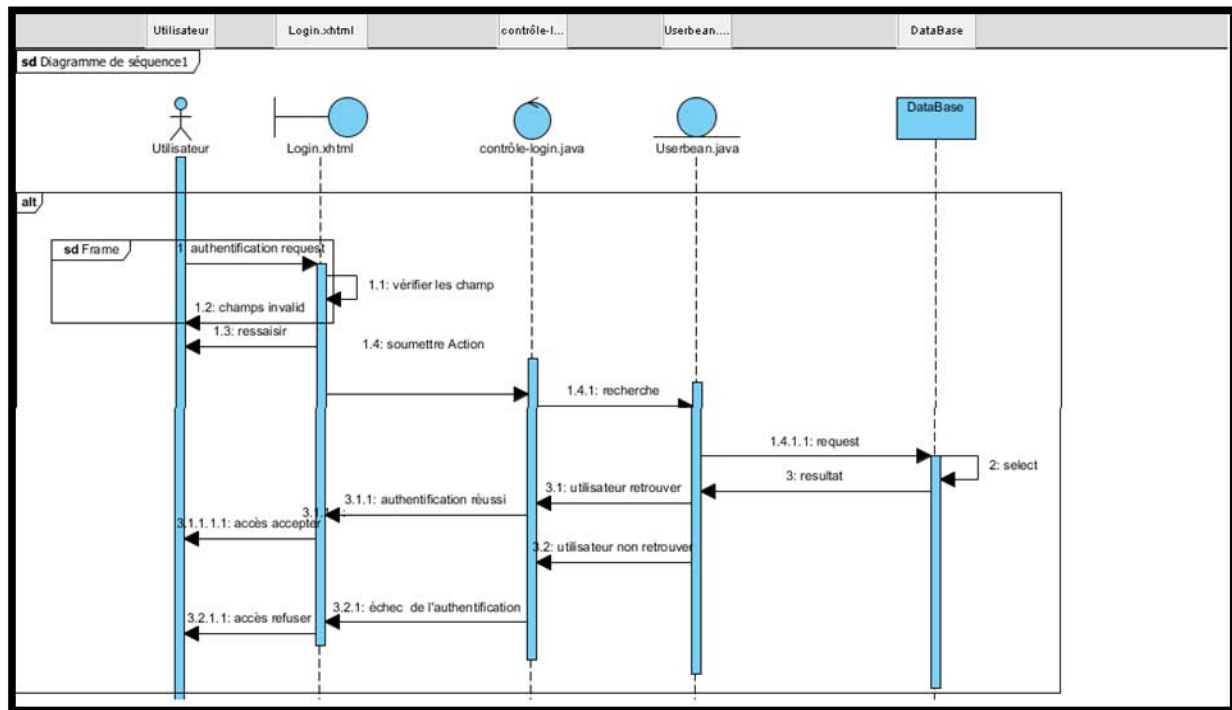


Figure 20-Diagramme de séquence objet: Authentification

Le diagramme de séquence décrit la communication et la propagation des messages entre les composants de l'application. L'acteur interagit d'une façon directe avec la vue, en lui envoyant une requête de connexion. La vue effectue une vérification des champs vide et soumet l'action au contrôleur qui, interroge la bean « UserBean ». Cette dernière retourne le résultat de la requête au UserBean qui, la retournera au contrôleur, et ce dernier la transmettra à la vue où elle sera affichée et lisible par l'utilisateur.

2. Scénario : « Ajout d'un employé »

La scénario « Ajout d'un employé » est donné par la figure 21.

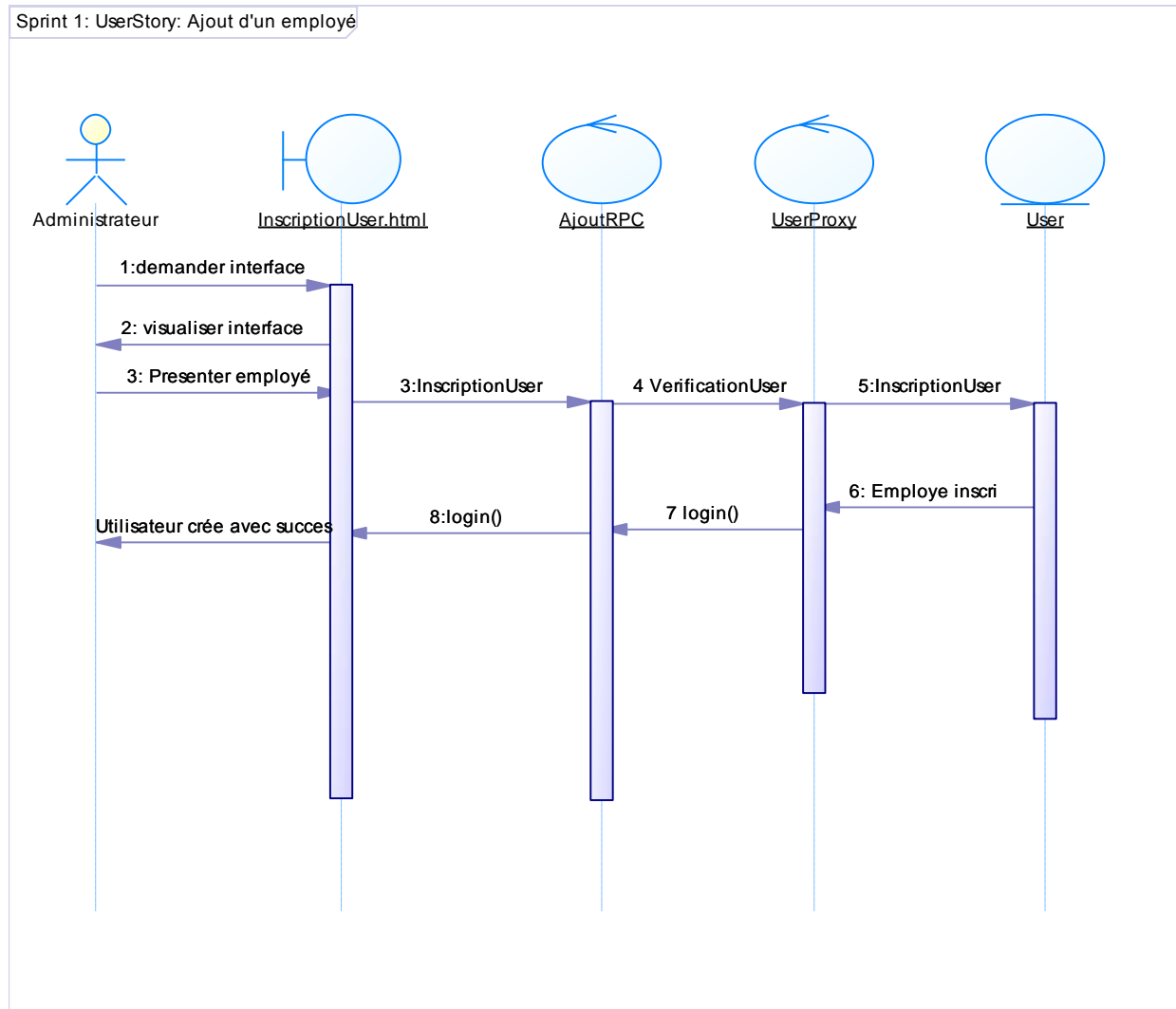


Figure 21 Diagramme de séquences Ajout d'un employé

L'administrateur demande l'interface « InscriptionUser.html » pour ajouter un utilisateur. La vue demande alors le presenter « AjoutRPC » qui lui, de sa part demande « UserProxy » pour vérifier la cohérence de la requête et transmet l'opération a l'entité « user » qui ajoute l'utilisateur.

3. Scénario 3 : Modification d'un employé

Le scénario « Modification d'un employé » est donné par la figure 22.

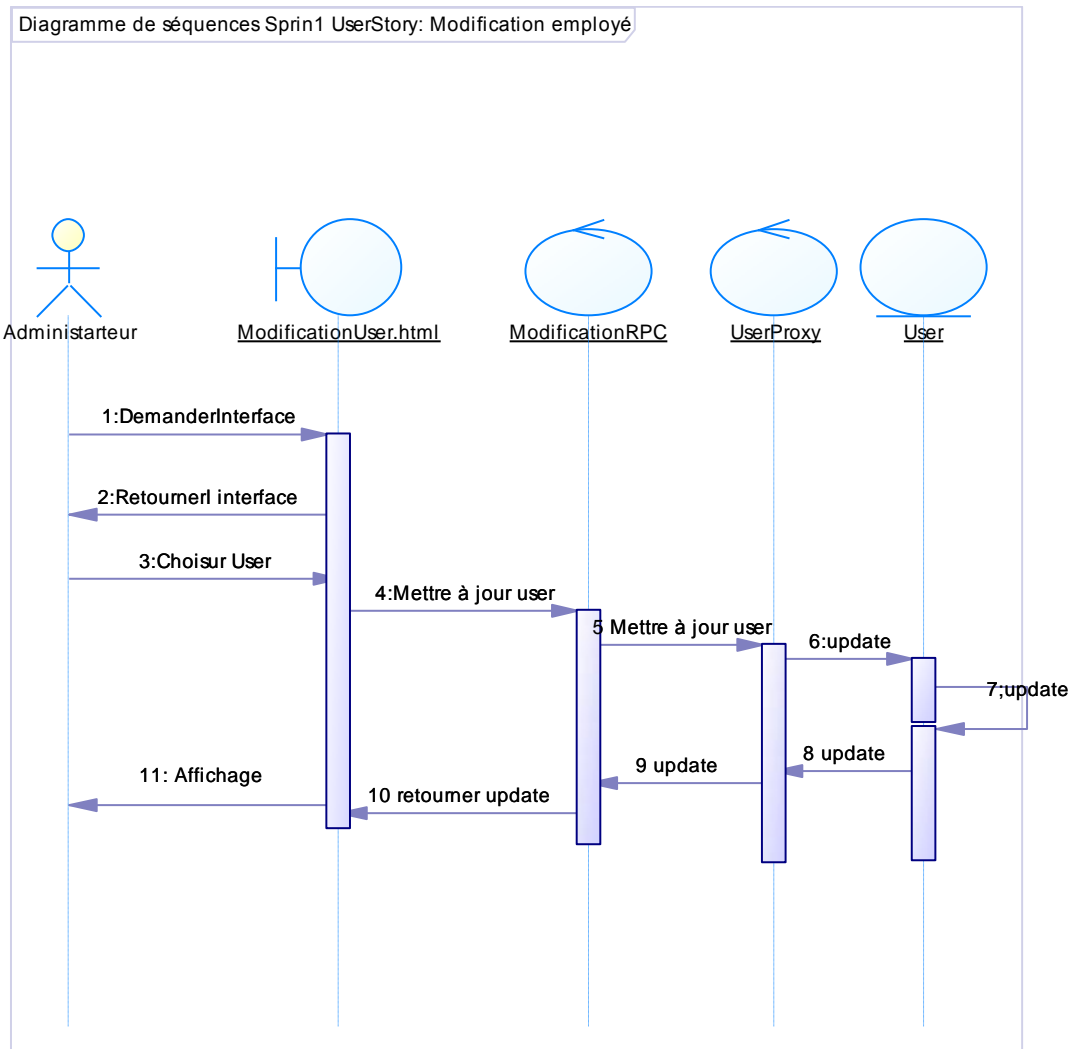


Figure 22 Scénario Modification d'un employé

Pour modifier un employé, l'administrateur accède à l'interface modificationUser.html. Cette dernière, consulte le presenter « ModificationRPC » qui lui, de sa part demande « UserProxy » pour vérifier la cohérence de la requête et transmet l'opération a l'entité « user » qui ajoute l'utilisateur.

Conclusion :

Au cours de ce chapitre, nous avons présenté l'architecture de l'application, ensuite nous avons découpé les packages et cité les classes qui compose chaque package. Nous avons également établi les diagrammes de classe et de séquences objets pour avoir une vue statique et une autre dynamique de l'application.

Le chapitre suivant intitulé « Mise en œuvre » sera consacré à la réalisation de l'application.

Chapitre 5 : Mise en œuvre

Introduction

Après avoir achevé la phase d'analyse et de conception, nous entamons dans ce chapitre la partie réalisation. Nous commençons tout d'abord par présenter l'architecture de l'application, l'environnement matériel et les outils de développement utilisés pour mener à terme ce travail. Par la suite, nous exposerons le travail réalisé à l'aide des captures d'écran. Enfin, nous illustrons par un chronogramme les différentes étapes par lesquelles nous sommes passés.

I. Environnement de travail et technologies utilisées

1. Technologies

Les technologies, logiciels, éditeurs et frameworks mis en œuvre pour élaborer l'application sont les suivants :

Java /J2EE (Java 5)

J2EE (Java 2 Enterprise Edition) est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

On parle généralement de «plate-forme J2EE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution.

Google Web Toolkit (SDK :2.5)

GWT est un ensemble d'outils logiciels développé par Google, permettant de créer et maintenir des applications web dynamiques mettant en œuvre JavaScript, en utilisant le langage et les outils Java. C'est un Framework libre distribué selon les termes de la licence Apache 2.0.

GWT est une technologie permettant de réaliser une application RIA (Rich Internet Application) qui cumule les qualités du web (utilisable via un navigateur, multi plates-formes, déployabilité,...) et celles des applications client-serveur (ergonomie, séparation données/IHM, performance,...).

Smart GWT Showcase

Smart GWT est un framework basé sur GWT qui permet de développer des RIA et d'utiliser les bibliothèques complètes de GWT qui sont les widgets et les composants primitifs et complexes. Il inclut :

- Haute performance et un parfait design
- Encapsulation du code JavaScript dans les classes Java
- Prise en charge complète de la procédure distante en utilisant GWT RPC, JSON et XML
- Prise en charge de Java 1.6

Sencha GXT (3.1)

GXT est une bibliothèque écrite en Java et implémente GWT. Elle inclut :

- Haute performance
- Bien conçu, le code source cohérent et parfaitement documenté
- Solution native GWT sans JavaScript externe ou bibliothèques 3ème partie
- Prise en charge complète de la procédure distante en utilisant GWT RPC, JSON et XML
- Prise en charge de Java 1.6

MySQL 5

Système de gestion de base de données (SGBD)

C'est un produit gratuit et Open Source constitué d'un serveur SQL qui supporte différents systèmes de stockage, plusieurs logiciels clients et librairies, outils d'administration, ainsi que de nombreuses interfaces de programmation

JBoss 7.1

JBoss est un serveur d'applications J2EE open source, libre entièrement écrit en Java, publié sous licence GNU LGPL. Parce que le logiciel est écrit en Java, JBoss Application Server

Eclipse IDE (Juno)

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java, et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions

JSON(JavaScript Object Notation)

JSON (JavaScript Object Notation – Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable par des machines. JSON est un format texte complètement indépendant de tout langage. Ces propriétés font de JSON un langage d'échange de données idéal.

REpresentational State Transfert

REST (REpresentational State Transfer) est un style d'architecture pour concevoir un service web, utilisant au maximum les possibilités de HTTP. Ses principales caractéristiques sont les suivantes :

- **application client/serveur** : le transport sur le réseau est assuré par HTTP (essence même d'un web service)
- **interface uniforme** : tout élément offert à la manipulation par l'application est nommé ressource et est identifié de manière unique par les Identifiants de Ressource Uniforme

SYBASE Power AMC

Power AMC est un logiciel de conception, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

2. Architecture et déploiement

En se basant sur les technologies citées dans la section précédente, nous élaborons ce schéma (figure 22) pour représenter l'architecture de l'application.

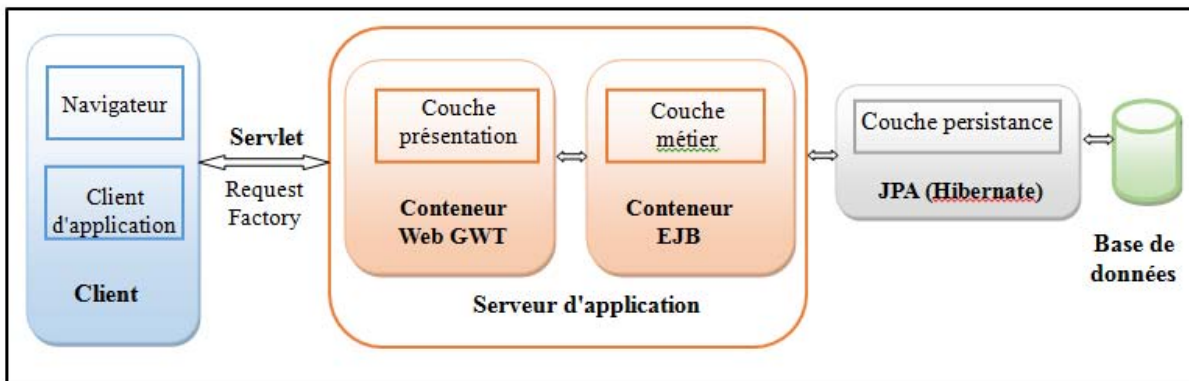


Figure 23 Architecture de l'application

La couche client : représente le navigateur composé par des pages HTML. Cette couche communique avec le serveur d'application à travers une servlet.

La couche Présentation : est articulée autour d'un serveur Web qui génère l'interface graphique de l'application. Elle correspond à la partie de l'application visible et interactive avec les utilisateurs.

La couche Métier (couche applicative) : correspond à la partie fonctionnelle de l'application et contient les traitements représentant les règles métier.

La couche persistance (couche d'accès aux données) : offre le service transactionnel avec le SGBDR.

Tous les accès à la base de données doivent se faire à travers des connexions gérées par le serveur d'application. Ainsi, on doit être capable de vérifier l'absence de toute autre utilisation de SQL dans les autres couches.

II. Captures d'écran

Nous présentons dans cette section les plus importantes interfaces qui illustrent les différents cas d'utilisation déjà vus dans le chapitre étude préliminaire.

1. Authentification

Au lancement de notre application, une interface d'authentification démarre comme le montre la figure 24 :

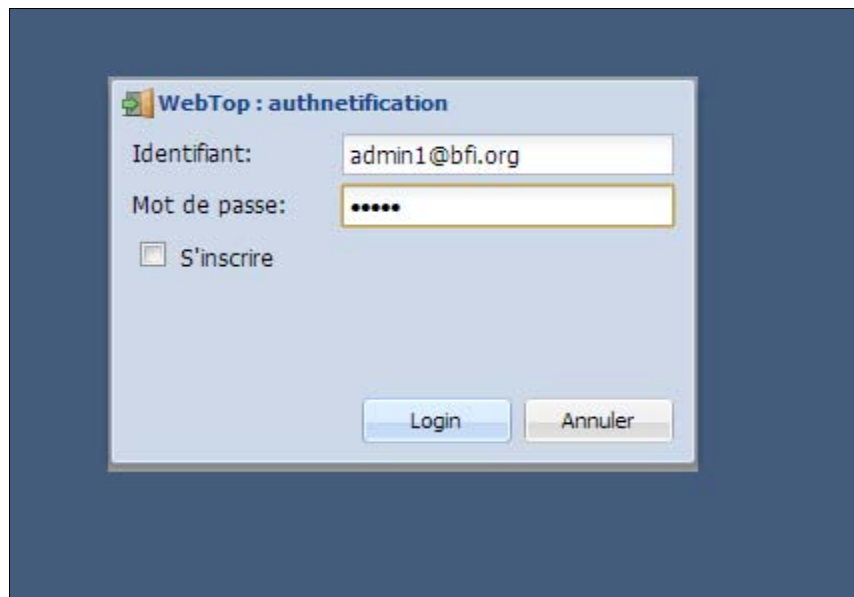


Figure 24-Interface de connexion

Ce module constitue le point d'accès unique à l'application, tous les utilisateurs doivent s'inscrire pour pouvoir bénéficier de la totalité des services de Webtop.

En cas d'erreur de mot de passe ou de login, le système prévient l'utilisateur comme ainsi

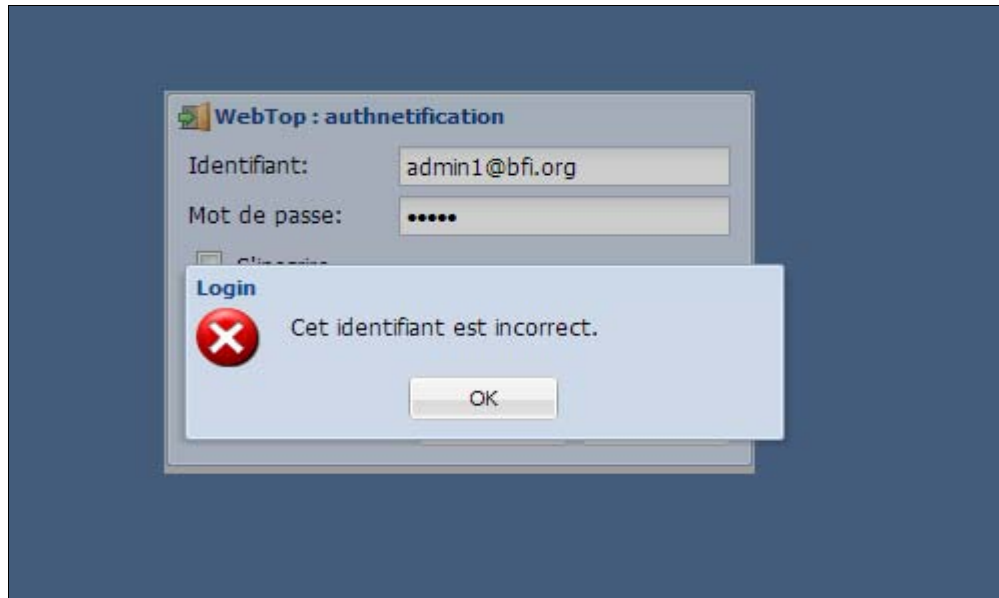


Figure 25-Interface d'échec de la connexion

2. Interface principale

Une fois authentifié, l'utilisateur accédera à l'ensemble des fonctionnalités de l'application.

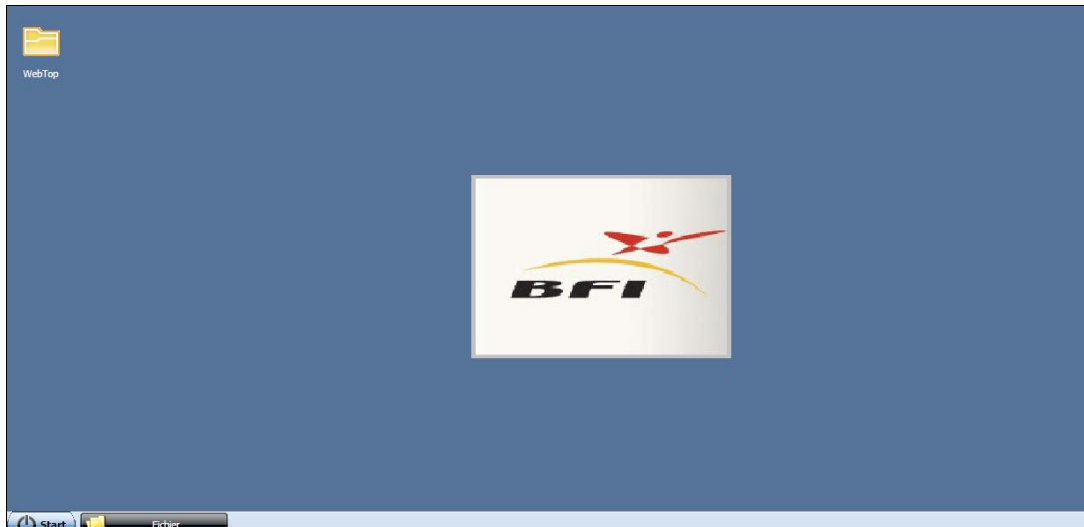


Figure 26-Interface principale

Cette interface est l'interface principale de l'application, l'utilisateur peut accéder au menu « démarrer » ou bien cliquer sur le dossier webtop qui désigne le système de fichiers.

3. Edition des fichiers

L'Édition des fichiers est donnée par la figure suivante

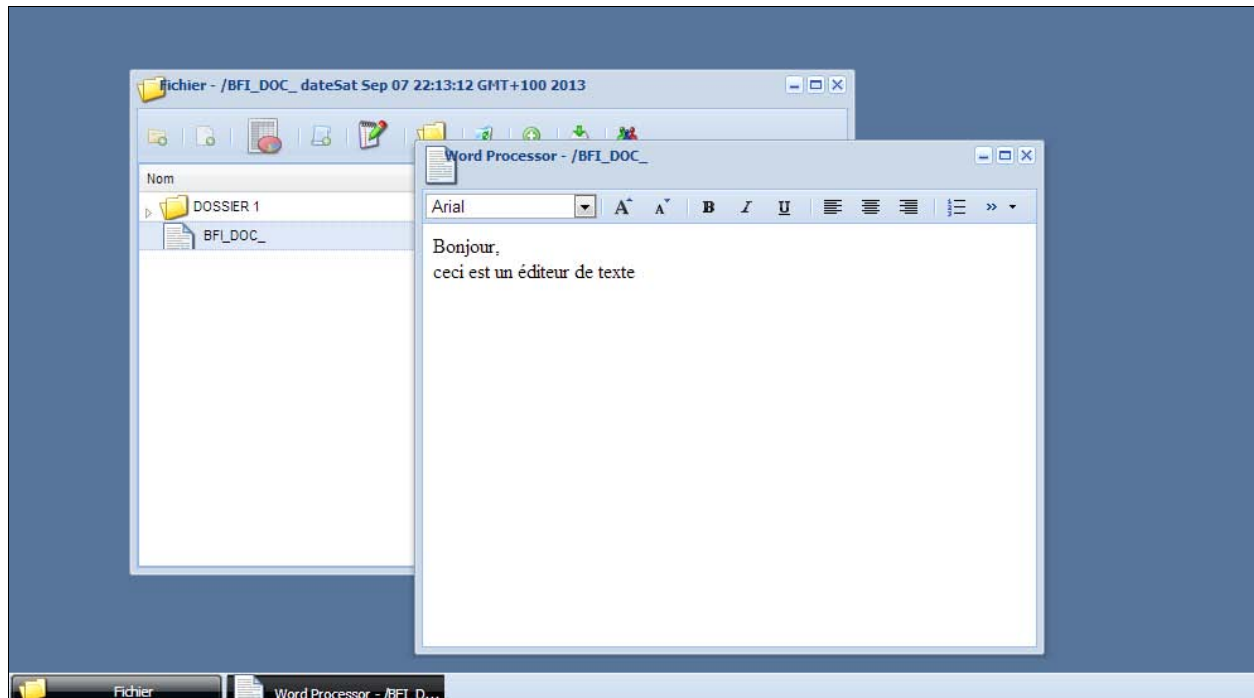


Figure 27-Editer du texte

Cette interface offre à l'utilisateur le service édition des fichiers .Dès la création d'un fichier texte, l'employé peut ouvrir le fichier et écrire dedans. Ce contenu est désormais enregistré et accessible. Le cas est identique pour l'édition des feuilles de claculs et de programmes.

4. Chargement des fichiers

La figure suivante présente le chargement des fichiers

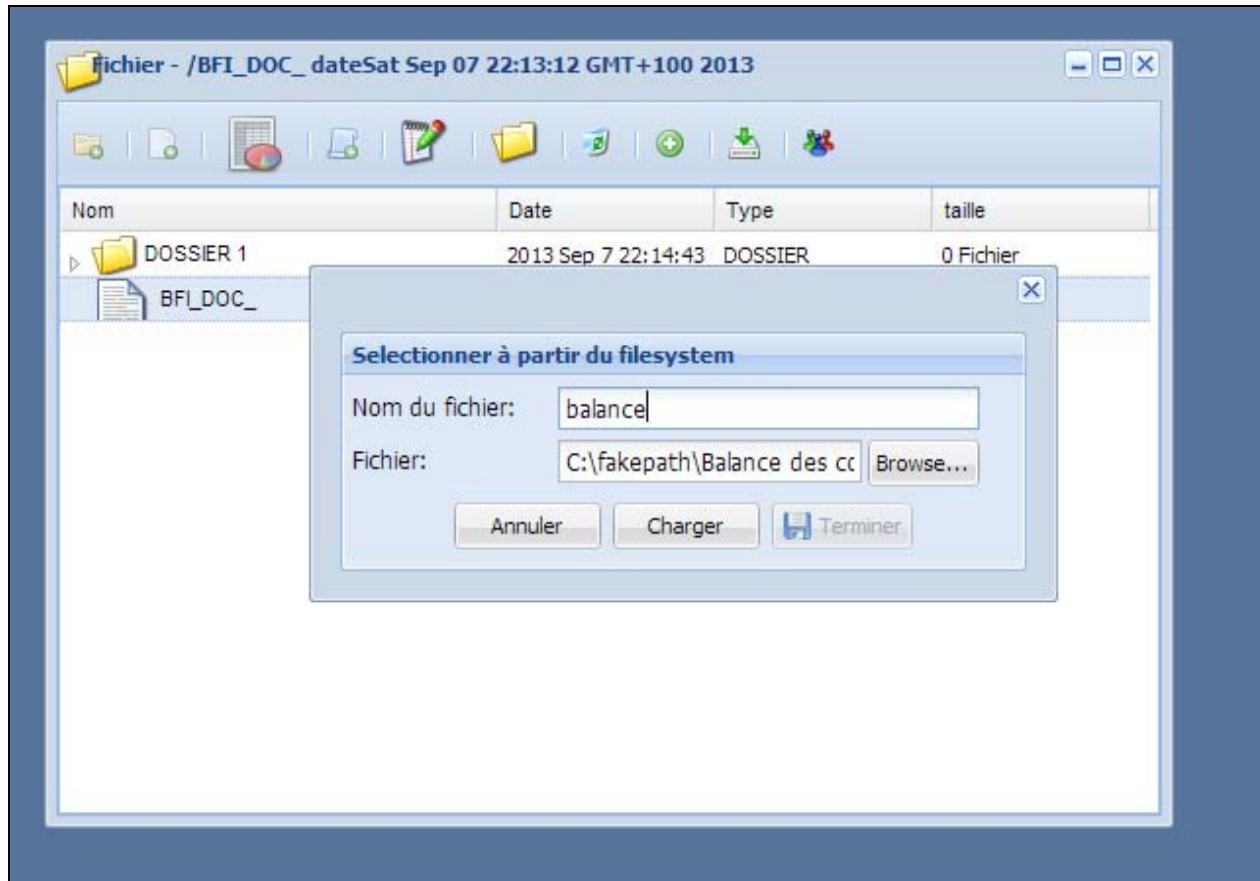


Figure 28 Chargement des fichiers

Le chargement des fichiers à partir de l'extérieur se passe avec quatre clics de souris : L'employé clique sur l'icône « charger fichier » ou bien sur le menu charger du menu contextuel. Le système lui retourne un champs pour charger le fichier désiré. L'employé clique sur Charger pour commencer le chargement, alors le système affiche une barre d'avancement et attend à sa fin la confirmation de l'employé. Quand il clique sur terminer, le chargement est effectué avec succès et le dossier importé est affiché.

5. Web service Météo

La figure 29 présente le retour du service Météo.

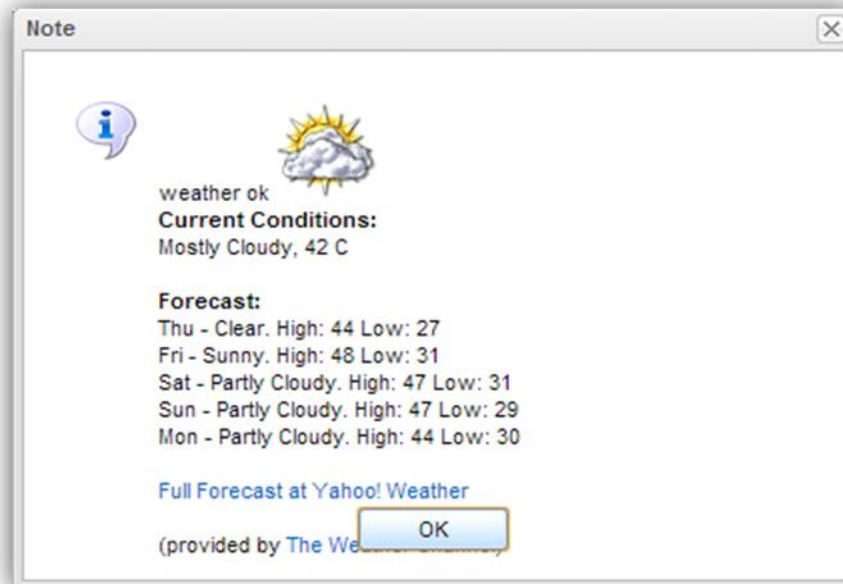


Figure 29-Service météo(Web Service Yahoo)

Lorsque l'employé clique sur le menu Météo et entre sa localisation actuelle, une fenêtre affiche le météo .Ce service est fourni par Yahoo.

6. Web Service RSS fourni par bing

La figure 30 affiche le flux RSS

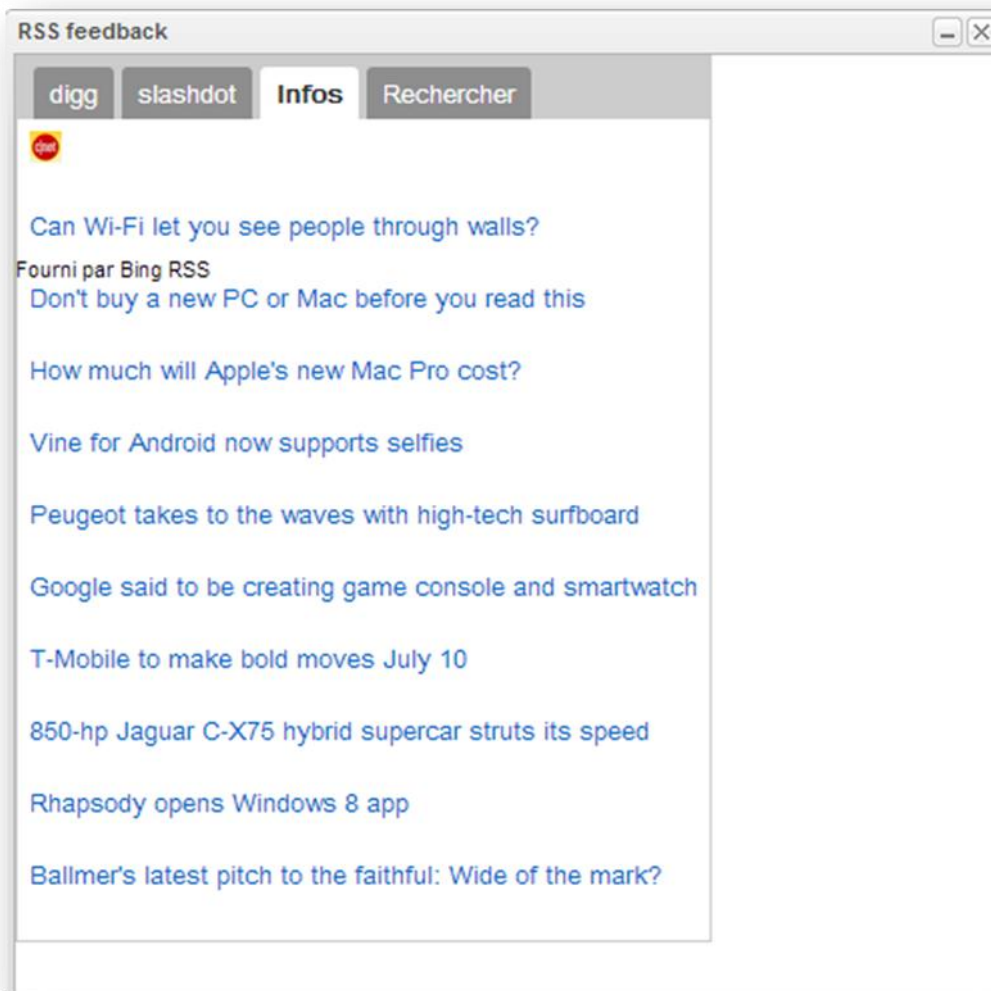


Figure 30-Flux RSS

La figure ci-dessus représente le flux RSS qui est aussi un service offert par Bing appelé lorsque l'utilisateur clique sur le bouton RSS. Cette fenêtre est composée de quatre onglets « Infos », « Slashdot », « dig » et « rechercher »

III. III. TEST

Cette étape de cycle de vie du logiciel consiste à tester le logiciel à plusieurs itérations afin de vérifier si le produit final est conforme au cahier des charges.

L'étape du test comprend :

1. Les tests

a) Les tests unitaires :

Il s'agit pour le programmeur de tester un module, indépendamment du reste du programme, ceci afin de s'assurer qu'il répond aux spécifications fonctionnelles et qu'il fonctionne correctement en toutes circonstances. Cette vérification est considérée comme essentielle, en particulier dans les applications critiques. Elle s'accompagne couramment d'une vérification de la couverture de code.

Pour ceci, et afin de renforcer la solidité du code, nous avons utilisé le framework de test JUnit(15) avec la version 4.

JUnit est un framework de test unitaire pour le langage de programmation Java. Créée par Kent Beck et Erich Gamma, il définit deux types de fichiers de tests. Les TestCase sont des classes contenant un certain nombre de méthodes de tests. Un TestCase sert généralement à tester le bon fonctionnement d'une classe. Une « TestSuite » permet d'exécuter un certain nombre de TestCase déjà définis.

Les figures 30,31 et 32 présentent un exemple de test unitaire effectué avec succès sur les méthodes des classes « YahooWeateherServiceImpl », « SendEmailImpl » et « UploadImpl ».

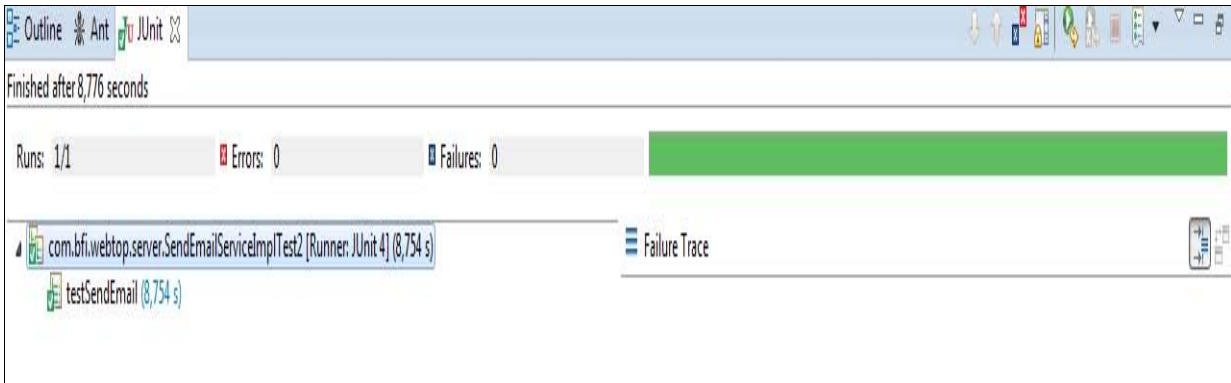


Figure 31 Exemple de test 1

Le résultat de ce test est succès et le résultat est retourné après 8 secondes.

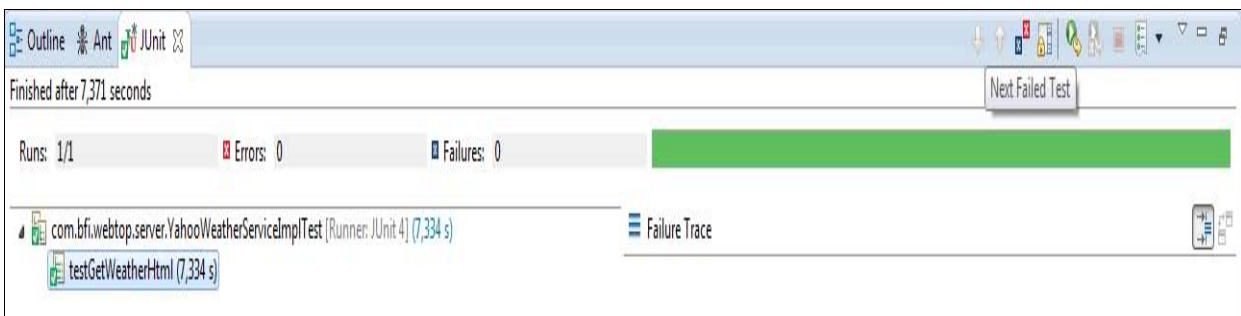


Figure 32 Exemple de test 2

Le résultat de ce test est succès et le résultat est retourné après 7 secondes.

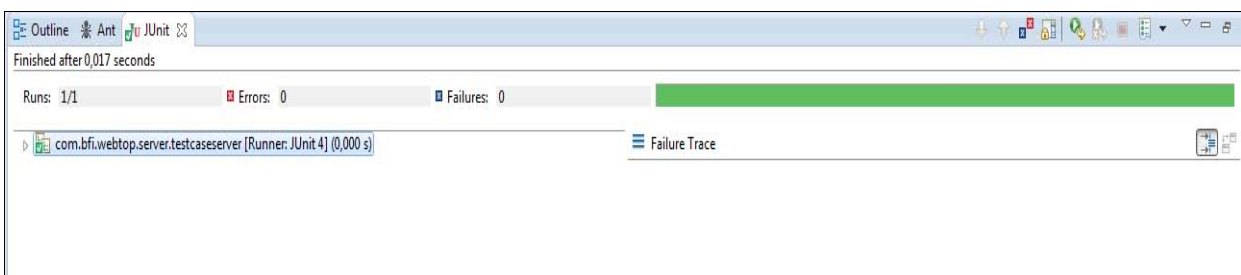


Figure 33 Exemple de test 3

Le résultat de ce test est succès et le résultat est retourné après 8 secondes.

b) Les tests d'intégration :

Généralement, cette étape consiste à, une fois que les développeurs ont chacun validé leurs développements ou leurs correctifs, à regrouper leurs modifications ensemble dans le cadre d'une livraison, mais vu que ce projet est réalisé en monôme, ce test a été utiliser à intégrer les modules ensemble et tester leur cohérence.

c) Autres tests

- Nous avons testé l'application en **mode de développement** .
- Le mode de développement est un mode d'exécution spécifique au plugin Google pour Eclipse, et qui permet de communiquer le navigateur et le code compilé avec TCP/IP.
- Nous avons testé la partie présentation sur le serveur **Jetty** (16) :
Jetty est un serveur HTTP et un moteur de servlet entièrement fondé sur la technologie Java. Jetty est un logiciel libre distribué selon les termes de la licence Apache 2.0.
- Nous avons utilisé **FireBug** pour contrôler les fichiers CSS et HTML :
- FireBug est un outil de développement web sous forme d'une extension pour Mozilla Firefox qui permet de déboguer, modifier et contrôler le HTML, le CSS, le DOM, le XHR et le JavaScript d'une page web.
- Finalement nous avons testé l'application sur les trois navigateurs Google Chrome et Mozilla Firefox pour éviter les problèmes de l'incompatibilité entre navigateurs.

IV chronogramme

Le chronogramme suivant présente l'enchaînement des tâches effectuées tout le long du projet de fin d'étude :

Sprint ou Tâche	Date de début	Date de fin	Détails
------------------------	----------------------	--------------------	----------------

Documentation et recherche de l'environnement de travail	17/03/2013	28/03/2013	Documentation sur l'organisme d'accueil Documentation sur les bureaux virtuels Documentation sur GWT
Etude préalable	02/04/2013	12 /04/2013	Etude sur les bureaux virtuels existants
Elaboration du backlog du produit	01/05/2013	03/05/2013	Plannification
Sprint 1 « Gestion des employés »	04/05/2013	28/05/2013	Ajout des employés Modification des employés Suppression des employés
Sprint 2 « Gestion des documents »	03/06/2013	27/06/2013	Création de documents Ajout de documents Modification des documents Suppression des documents
Sprint 3 « Gestion des versions »	02/07/2013	24/07/2013	Gérer les versions
Sprint 4 « Gestion des partages »	26/07/2013	05/08/2013	Partager les documents
Sprint 5 « Personnalisation du bureau »	13/08/2013	20/08/2013	Personnalisation du bureau

Test	21/08/2013	23/09/2013	Tests unitaires
			Tests d'intégrité
			Autres tests

Tableau 10 chronogramme

Conclusion

Dans ce chapitre, qui vient de clôturer ce présent rapport, nous avons décrit l'architecture de l'application, l'environnement du travail, nous avons exposé la réalisation de notre application à travers quelques aperçus d'écran et nous avons enfin présenté le chronogramme de notre projet.

Conclusion générale et perspectives

Le présent travail, nous a permis de nous familiariser avec le travail sur la plateforme JEE.

Non seulement, les bénéfices ont été réalisés sur le plan technique mais aussi sur le plan social. L'intégration d'une équipe de travail a été une expérience qui marquera la période de réalisation du projet au sein de BFI.

Le stage quotidien au sein de la société a aussi été pour moi une occasion unique pour épanouir mes capacités de communication dans un environnement professionnel et aussi l'échange de l'information avec mes camarades stagiaires. C'était une expérience très enrichissante sur tous les domaines.

Ce projet a eu pour but la conception, le développement et l'intégration d'une application nommée WebTop au sein de la société BFI afin d'apporter une valeur ajoutée et un meilleur service aux clients. Pour ce faire, nous avons suivi la méthodologie Scrum.

Nous sommes parvenus à développer la plupart des fonctionnalités du système dans les temps accordés.

Ce stage nous a permis également d'approfondir nos connaissances acquises tout le long de notre formation, par la pratique des nouvelles technologies, avec lesquels, le développement n'a pas été une tâche facile, cependant, nous n'avons pas hésité à les adopter et ce malgré le manque de supports en terme de documentation, d'encadrement et de support.

Bien que les principaux objectifs de mon projet soient atteints, l'application que j'ai développée pourrait être enrichie par d'autres fonctionnalités avancées. Des améliorations peuvent être envisagées pour l'enrichir, tel que le renforcement du module de l'authentification avec la

technologie LDAP par exemple et pourquoi ne pas migrer tout le concept sur le cloud computing.

On pourra aussi ajouter une multitude de web services et de gadgets et aussi offrir au propriétaire du compte une adresse email avec le nom de domaine « @webtop.com ».

J'envisage également développer une version de l'application fonctionnant sur le SE Android.

Enfin, je souhaite que ce modeste travail apporte satisfaction aux membres du jury et à toute personne intéressée, de près ou de loin.

Références

Netographie

[1]<http://www.contactoffice.com/index.html>[consulté le 23 Fevrier 2013.]

[2]. HYPERLINK "http://www.ulteo.com/" <http://www.ulteo.com> /accueil.html [consulté le 23 Fevrier 2013]

[3].<http://www.presse-citron.net/mybooo-un-bureau-virtuel-assez-convaincant>. [Consulté le : 25 Fevrier 2013.]

[4]. <http://www.mynox.fr/index.html>. [Consulté le 24 02 2013]

[5]. zoho.html. fr.wikipedia.org. [Consulté le 24 02 2013].

[6]. <http://www.access-dev.com/access-dev/la-gestion-de-projet-methodes-classiques-vs-methodes-agiles/>. [Consulté le : 28 fevrier 2013.]

[7]. http://fr.wikipedia.org/wiki/Unified_Process. [Consulté le: 02 Mars 2013.]

[8]. <http://www.techno-science.net/?onglet=glossaire&definition=743> [Consulté le : 12 Mars 2013].

9. [En ligne]

[10] <http://fr.wikipedia.org/wiki/Prototypage>.

[11] <http://www.techno-science.net/?onglet=glossaire&definition=5266>. [consulté le : 11 Mai 2013.]

[16]. developpers.google.com/Jettywebserver. [Consulté le30 04 2013].

Bibliographie

[12]. UML in action. [Franck Vallée ,Pascal Roques]. *De l'analyse des besoins à la conception J2EE*. s.l. : EyRolles.

[13]. Diagramme de séquences. [Simone Nelson] . *Analyse, Conception Objet*. 2012.

[14]. GWT in action[Marty Hall]2010.

[15]. Junit Test Case white book[Adrian Walker] 2009.

[16]. developers.google.com. *Jettywebserver*. [En ligne] 30 04 2013.

