

**Université Sidi Mohammed Ben Abdellah
FACULTE DES SCIENCES ET TECHNIQUES FES
DEPARTEMENT D'INFORMATIQUE**



**UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES DE FES**

**Projet de Fin d'Etudes
Licence Sciences et Techniques Génie Informatique**

Application web de la gestion des chèques et des factures



Lieu de stage : Société MySoft FES.

Réalisé par :

BENJELLOUN Zineb

Encadré par :

Pr. BEN ABBOU Rachid

Soutenu le 06/06/2018 devant le jury composé de :

Pr. K. ABBAD

Pr. A. BENABBOU

Pr. R. BEN ABBOU

Année Universitaire 2017-2018

Dédicace

Je dédie ce modeste travail

*A ma mère la lanterne qui éclair mon chemin et m'illumine de douceur et
d'amour.*

*A mon père, en signe d'amour, de reconnaissance et de gratitude pour tous
les soutiens et les sacrifices dont il a fait preuve à mon regard.*

*A toute ma famille, aucun mot ne pourra décrire vos dévouements et vos
sacrifices.*

*A tous mes amis, en témoignage de l'amitié sincère qui nous a liées, leurs
soutiens et des bons moments passés ensemble.*

*A tous les gens qui ont cru en moi et qui me donnent l'envie d'aller en
avant, je vous remercie tout, votre soutien et vos encouragements me
donnent la force de continuer. Vous êtes pour moi un exemple de courage et
de sacrifice continu.*

Remerciements

Au terme de ce travail, je tiens à exprimer ma profonde gratitude et mes sincères remerciements à M. Adil AKASBI, qui m'a donné la permission de passer ce stage au sein de la société MySoft.

Je remercie particulièrement Mr. BEN ABOU Rachid pour son encadrement, son soutien, ainsi que pour ses conseils instructifs durant toute la période de ce travail.

Mes plus vifs remerciements s'adressent aussi à tout le cadre professionnel et administratif de la formation LST génie informatique, pour leur patience et savoir qui nous a illuminés durant cette année de formation.

Mes sincères remerciements à M. Mustapha IJJaali, Doyen de la FST, à tout le travail qu'il effectue pour nous offrir le cadre et les moyens d'effectuer et de poursuivre nos études dans les meilleures conditions.

Enfin, Je tiens à remercier tous les membres du jury qui m'ont fait l'honneur d'accepter de juger mon travail. Pour tous ceux que j'ai oubliés, je vous dis Merci.

Résumé

Le projet réalisé dans ce stage vise à développer une application web de gestion financière pour l'entreprise MySoft.

Cette application permettra de l'automatisation de tous les processus de facturation à savoir :(devis, bons de commande, factures...).

Avant de passer à l'implémentation, une étude conceptuelle a été réalisée en appliquant les acquis du cours Modélisation en UML pour faciliter la réalisation de l'application, et en implémentant JQuery3.3.3 et Bootstrap4, qui sont des Frameworks permettant de créer efficacement des sites web complexes et flexibles.

Abstract

This project aims to develop an application to manage the financial documentation for a private business called MySoft, this app will not only help manipulate all the vouchers that are part of the financial process of this business but also manage the users and their profiles.

Before stepping into the actual code a system design was made, to organize ideas and create path for the workflow so that the process is clarified using some of UML's diagrams, and by implementing JQuery3.3.3 and Bootstrap4, which are Frameworks for efficiently creating complex and flexible websites.

Table de Matières

Liste des Figures	8
Introduction.....	9
Chapitre I :	10
Contexte général du projet	10
1. Présentation du lieu de stage	11
1.1. La société MySoft	11
1.2. Organigramme	11
2. Présentation du projet	12
2.1. Cas existant et problématique	12
2.2. Solution proposée	12
2.3. Diagramme de Gantt	13
Chapitre II :	14
Analyse et conception.....	14
1. Le choix de la méthode	15
2. Capture des besoins fonctionnels et techniques	16
2.1. Besoins fonctionnels	16
2.1.1. Diagramme de packages.....	17
2.1.2. Diagramme de cas d'utilisation.....	19
2.1.3. Diagramme de séquence :	23
2.2. Besoins techniques	23
2.3. Diagramme de classes	25
2.4. Modélisation logique des données	26
3. Architecture applicative.....	29
4. Outils et langages utilisés	29
Chapitre III :	32
Présentation de l'application	32

5. Présentation de l'application	33
5.1. Fenêtre d'Authentification.....	33
5.2. Page d'accueil : Directeur.....	33
5.3. Page d'accueil : secrétaire / observateur	34
5.4. Fenêtre des factures sur l'espace achats.....	34
5.5. Aperçu sur la facture.....	35
5.6. Régler le paiement d'une facture en attente ou payé mais sans justificatif...	35
5.7. Listes des bons de réception	36
5.8. Création d'une facture à partir d'un BL existant	37
5.9. Fenêtre d'ajout d'un BR.....	37
5.10. Liste des produits en stock	38
5.11. Fenêtre des opérations sur les comptes bancaires de la société	38
5.12. Ajouter une opération.....	39
5.13. Les utilisateurs	39
5.14. Fenêtre d'ajout d'un utilisateur.....	40
5.15. Liste des clients	40
Annexe :	41
Définition.....	42
5.16. Caractéristiques	42
5.17. Le processus 2TUP.....	42
5.18. Etude Comparative des processus de développement	45
Conclusion et perspectives.....	46
Webographie	47

Liste des Figures

Figure 1 : Diagramme de Gantt	13
Figure 2 : Les diagrammes d'UML	15
Figure 3 : L'approche 2TUP	Error! Bookmark not defined.
Figure 4 : Processus de vente	16
Figure 5 : Processus d'achat.....	16
Figure 6 : Diagramme de packages	Error! Bookmark not defined.
Figure 7 : Diagramme de cas d'utilisation (Directeur).....	Error! Bookmark not defined.
Figure 8 : Diagramme de cas d'utilisation (Secrétaire).....	Error! Bookmark not defined.
Figure 9 : Diagramme de cas d'utilisation (Observateur)	Error! Bookmark not defined.
Figure 10 : Diagramme de classes	26
Figure 11 : Schéma de la base de données	28
Figure 12 : Fenêtre d'authentification	33
Figure 13 : Page d'accueil : Directeur	34
Figure 14 : Page d'accueil : Directeur	34
Figure 15 : Fenêtre des factures sur l'espace achats	35
Figure 16 : Aperçu sur la facture	35
Figure 17 : Réglage du paiement d'une facture en attente ou payé mais sans justificatif	36
Figure 18 : Listes des bons de réception	36
Figure 19 : Création d'une facture à partir d'un BL existant	37
Figure 20 : Fenêtre d'ajout d'un BR.....	37
Figure 21 : Liste des produits en stock.....	38
Figure 22 : Fenêtre des opérations sur les comptes bancaires de la société	38
Figure 23 : Ajouter une opération.....	39
Figure 24 : Interface des utilisateurs	39
Figure 25 : Fenêtre d'ajout d'un utilisateur	40
Figure 26 : Liste des clients	40

Introduction

La technologie des applications Web a évolué très rapidement, de nos jours, où plusieurs logiciels Web réservés à cet usage sont sortis sur le marché un après l'autre, les entreprises de leur part utilisent cette plateforme pour accomplir l'automatisation et la simplification des tâches et pour offrir une communication fluide et disponible en permanence entre ses agents commerciaux, de ses administrateurs et les clients.

Dans cette optique, j'ai effectué mon stage au sein d'une entreprise privée dont ma mission était de développer une application web qui permet de faciliter le processus de la facturation et du paiement.

Le système va offrir la possibilité de créer tous les documents (devis, Bon de commande, Bon de livraison, Bon de réception, factures), leur gestion ainsi que la gestion des produits, des clients/fournisseurs et les utilisateurs.

Ce rapport représente le travail que j'ai effectué durant mon stage. Il évoque les différentes phases de développements de l'application organisé comme suite:

- Le premier chapitre est consacré à la description du contexte de notre projet. Il présente en premier lieu l'organisme d'accueil, ensuite une description générale du projet et du cahier des charges.
- Dans le deuxième, je vais présenter l'analyse des besoins de notre application et la conception du système.
- Le dernier chapitre est consacré à la présentation des outils de développements utilisés ainsi que l'application réalisée.

Enfin, je terminerai ce mémoire par une conclusion dans laquelle nous présentons le travail réalisé et quelques perspectives futures.

Chapitre I :

Contexte général du projet

1. Présentation du lieu de stage

Dans cette partie, je présenterai le contexte général du projet : je commence par une présentation de l'organisme d'accueil « MySoft SARL », ses clients ainsi que sa structure et ses domaines d'activités, ensuite je donne une brève présentation du projet de ce stage, à savoir l'application réalisée durant cette période.

1.1. La société MySoft

MySoft S.A.R.L. se trouve à Fès, Maroc. Créée en 2002, la société travaille sur la commercialisation des matériaux informatiques, et propose des solutions informatiques, maintenance sécurité réseau informatique, installation logiciel, réseaux intranet.

Actuellement son Capital Social est de 1.2Mdh et son chiffre d'affaire avoisine les 9Mdh H.T par an.

La société emploie directement une dizaine d'employés et indirectement une vingtaine d'autres,

1.2. Organigramme

L'organigramme de la société se présente ainsi :

La société est gérée par son chief executive officer (CEO) Adil AKASBI qui chapeaute trois départements :

- **Le département administratif** : Il se compose d'un responsable administratif. Son rôle est de superviser la tenue d'une comptabilité saine, le contrôle des opérations financière ainsi que la gestion administrative du personnel.
- **Le département des marchés** : responsable de la logistique des ventes et des achats.
- **Le département technique** : est tenu par un responsable qui supervise les techniciens responsables des services rendus par la société.

2. Présentation du projet

2.1. Cas existant et problématique

Pour gérer les factures et les opérations bancaires, la société s'appuie sur le tableur Excel.

Cette technique entraîne des problèmes sur plusieurs niveaux :

- La cohérence de l'état financier : une mise à jour n'est pas automatiquement distribuée ce qui peut causer des documents erronés, en plus, sans parcours des listes élément par élément les utilisateurs ne peuvent pas savoir une mise à jour effectuée par un autre utilisateur.
- L'accès et modification: pour accéder à une information il faut parcourir tous le(s) tableau(x) aussi si une ligne n'est pas ajoutée dans l'ordre on doit refaire toutes les lignes qui suivent ce qui consomme du temps.
- Sécurité : la main de manipulation est donnée à n'importe qui et n'est pas contrôlée en plus que ces fichiers sont faciles à perdre.
- Confidentialité : n'importe qui peut savoir les détails des activités de la société.

A partir de toutes ces observations, nous constatons qu'il y a un manque dans la structuration du travail au sein de l'entreprise. En effet, Il n'y a pas une bonne gestion et une attribution claire des rôles et des tâches pour chaque intervenant dans l'entreprise

2.2. Solution proposée

On souhaite créer une application web qui gère l'acquisition et la vente des matériels et le processus de facturation correspondante, cette gestion comprend la gestion des chèques, paiements, refus, création... la gestion de la facturation en plus du suivi des opérations bancaires de l'entreprise.

Cette application web sera l'intermédiaire entre les employés qui font partie de cette procédure en offrant les services suivants :

- Assurer la confidentialité et la sécurité : la nécessité d'une authentification et la limitation d'accès par rannort à chaque utilisateur.

- Assurer le partage immédiat des mises à jour.
- Faciliter la gestion des chèques et des documents.
- Mettre en place un système de notifications si nécessaires.
- Simplifier l'accès aux informations grâce aux filtrages.
- Garder trace de toutes les activités financières.
- Assurer la validité des données par le contrôle des entrées de chaque utilisateur.

2.3. Diagramme de Gantt

Le diagramme de Gantt est un outil utilisé en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet. Il s'agit d'une représentation d'un graphe connexe, valué et orienté, qui permet de représenter graphiquement l'avancement du projet.

Mon projet a été réalisé en suivant le diagramme de Gant ci-dessous:

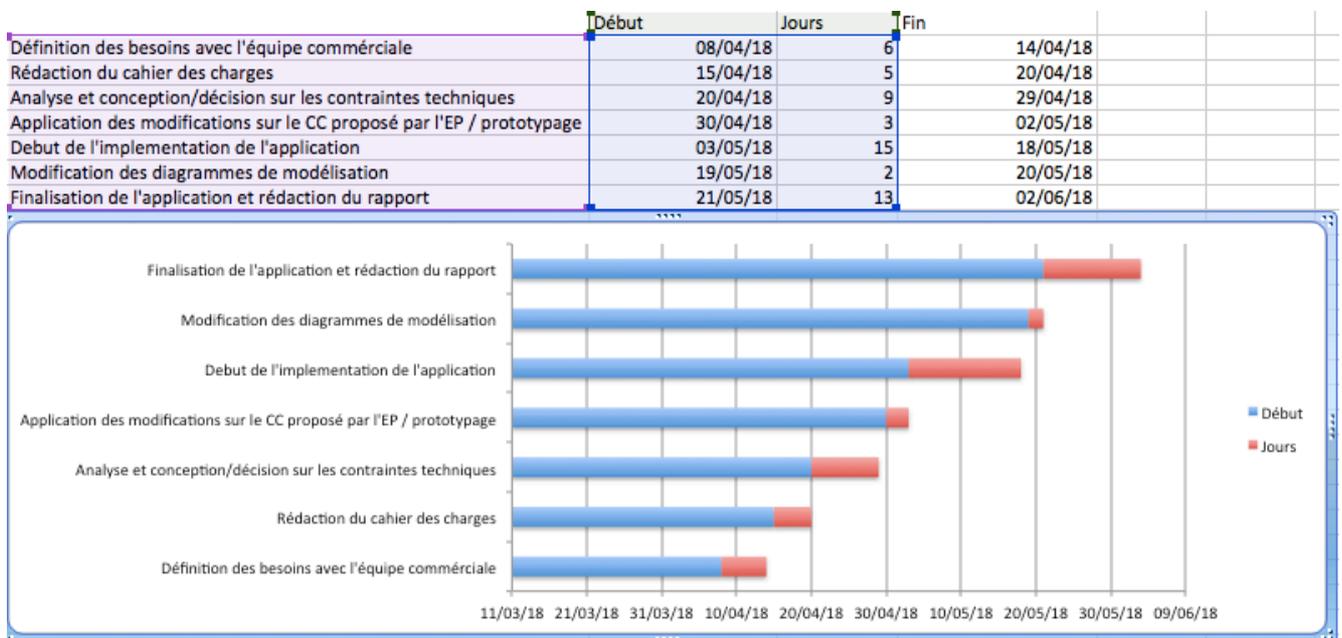


Figure 1 : Diagramme de Gantt

Chapitre II :

Analyse et conception

1. Le choix de la méthode

Pour modéliser d'une manière claire et précise la structure et le comportement de notre système indépendamment de tout langage de programmation, je vais adopter la démarche 2TUP et le langage de modélisation UML (Unified Modeling Language).

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 représente le standard de modélisation objet le plus répandu et le plus utilisé aujourd'hui. Il s'articule autour de treize types de diagrammes (Les diagrammes d'UML comme le montre la Figure, chacun d'eux étant dédié), la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes ; des Diagrammes structurels et comportementaux.

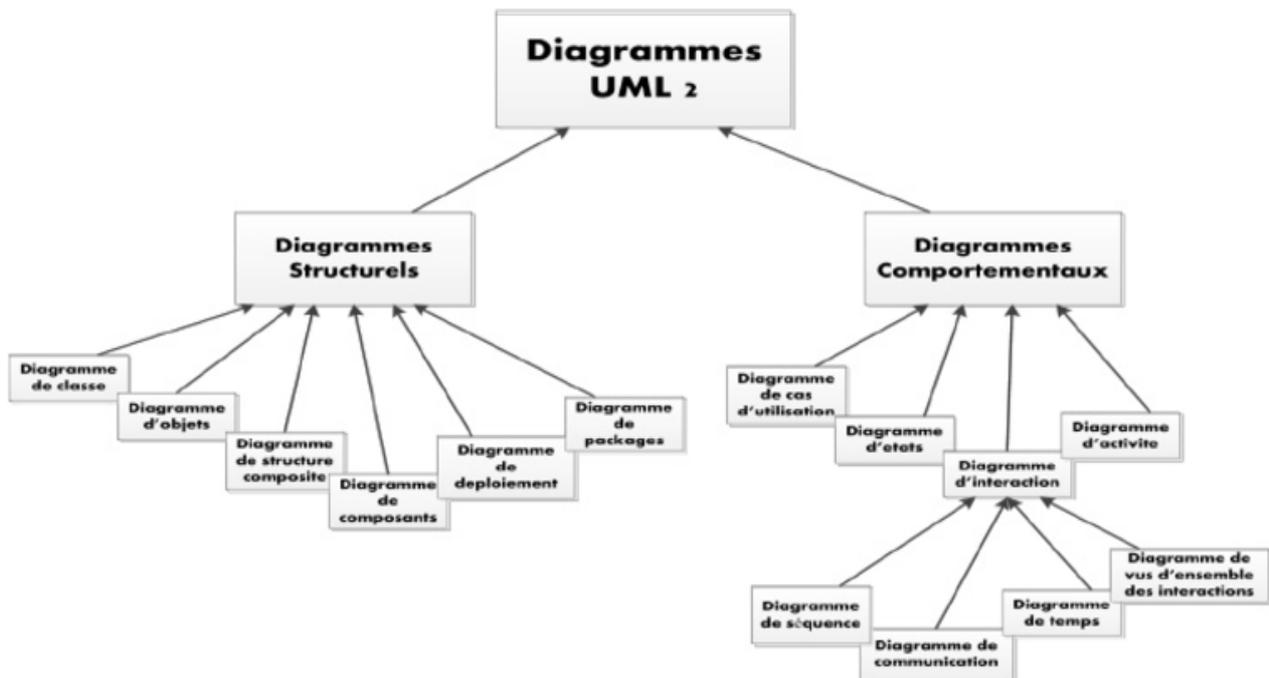


Figure 2 : Les diagrammes d'UML

(Voir annexe)

2. Capture des besoins fonctionnels et techniques

2.1. Besoins fonctionnels

Le service demandé consiste à automatiser les tâches de facturation et de paiement ce qui concerne la gestion des devis, des bons de commandes, des bons de livraison/réceptions et des factures.

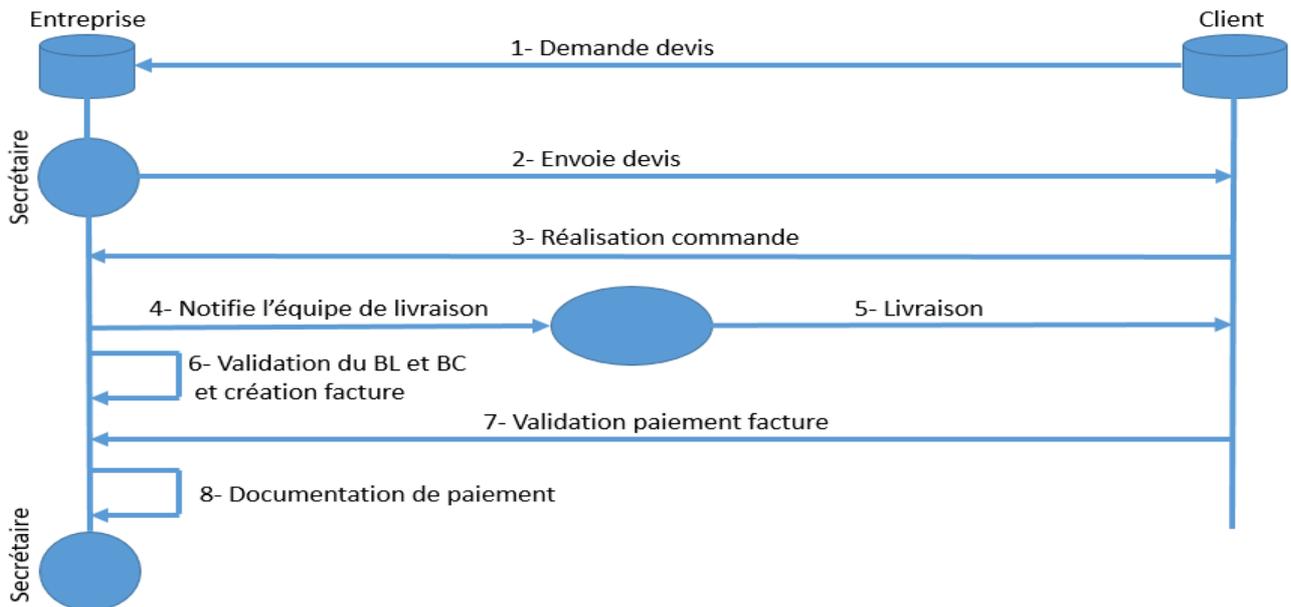


Figure 3 : Processus de vente

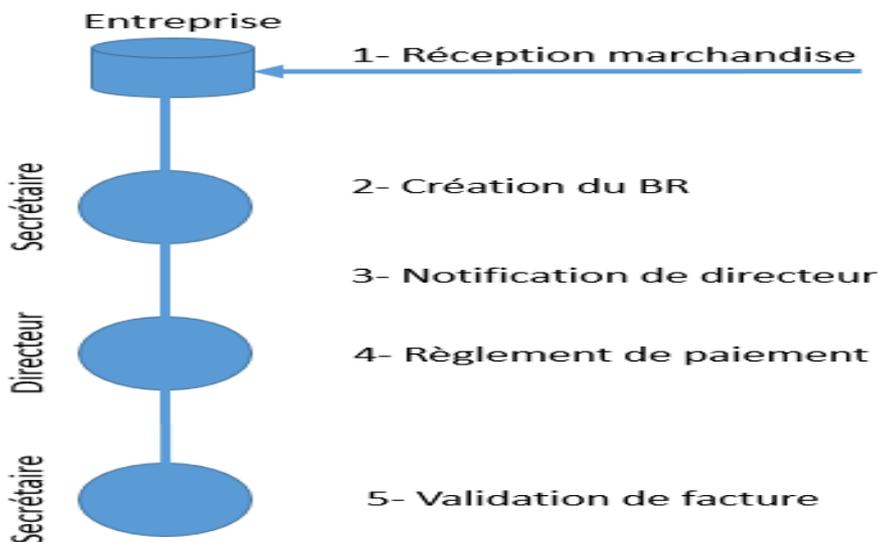


Figure 4 : Processus d'achat

Les principales fonctionnalités demandées sont :

- Etablir la liaison entre chaque facture et son mode de paiement et chaque opération bancaire avec l'ensemble des factures en question.
- Maintenir les données des clients/fournisseurs.
- Tenir à jour les informations sur les articles en stock;
- Faciliter de manipulation et la recherche sur les listes des documents avec possibilité d'impression.
- Sécuriser les informations de la société.
- Rappeler sur les dates importantes (échéance, livraison...).
- Identifier la responsabilité de chaque intervenant sur le système

2.1.1. Diagramme de packages

Un diagramme de packages est la représentation des dépendances entre les paquets (un paquet étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML). Les paquetages peuvent avoir des dépendances spéciales de type package import (importation de paquetage) et package merge (fusion de paquetages).

Un package import est « une relation entre un paquetage important un espace de nom et un paquetage, indiquant que l'espace de nom qui importe ajoute les noms des membres du paquetage à son propre espace de nom ». Par défaut, une dépendance entre deux paquetages est interprétée comme une relation de type package import.

Un package merge est « une relation dirigée entre deux paquetages, indiquant que les contenus des deux paquetages doivent être combinés. Elle est très similaire à la relation de généralisation dans le sens où l'élément source ajoute conceptuellement les caractéristiques de l'élément cible à ses propres caractéristiques; résultant en un élément combinant les caractéristiques des deux ».

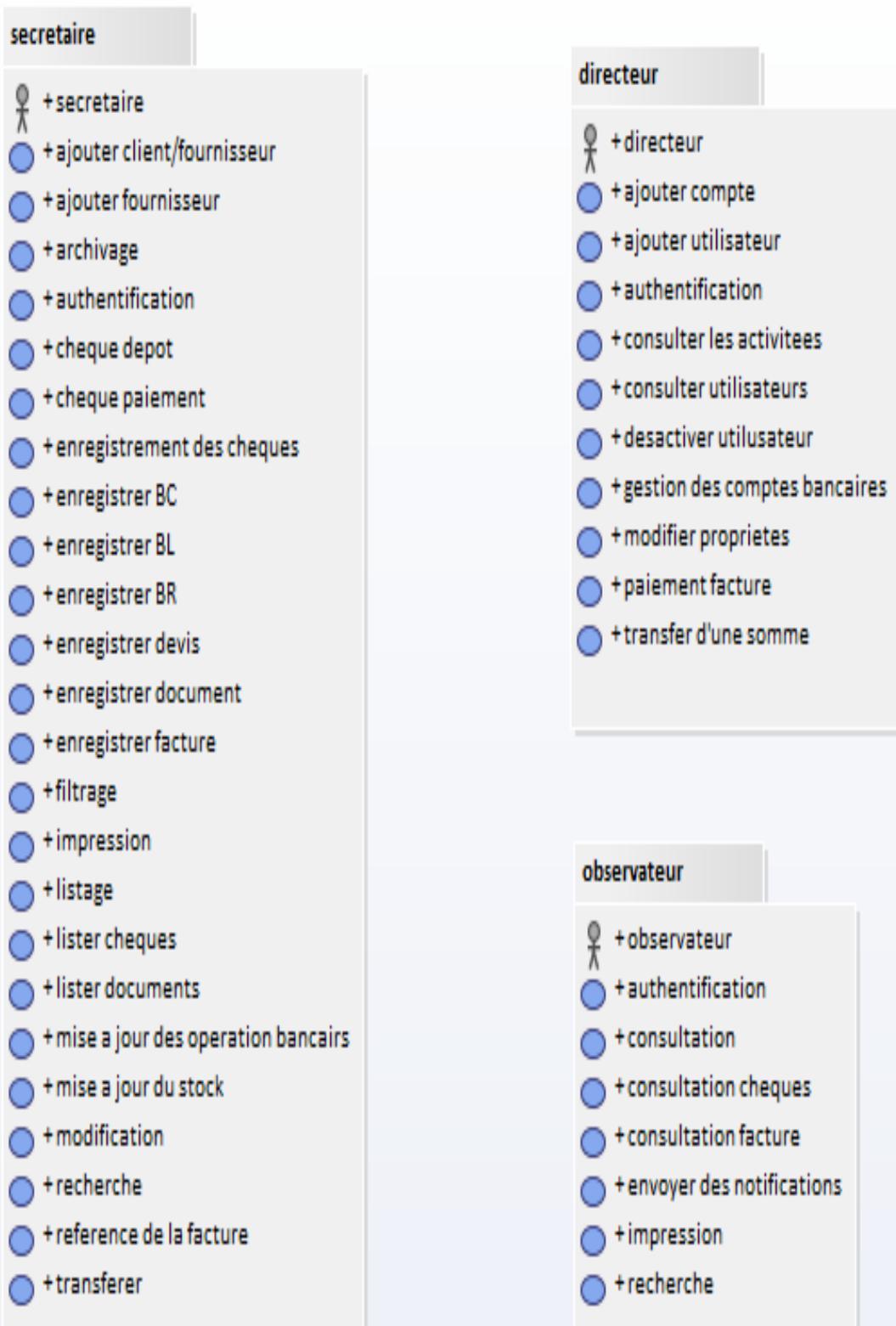


Figure 5 : Diagramme de packages

2.1.2. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

Un acteur est un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Les acteurs qui interagissent avec notre application sont : Le directeur, la secrétaire et l'observateur.

Acteur 1 : Directeur

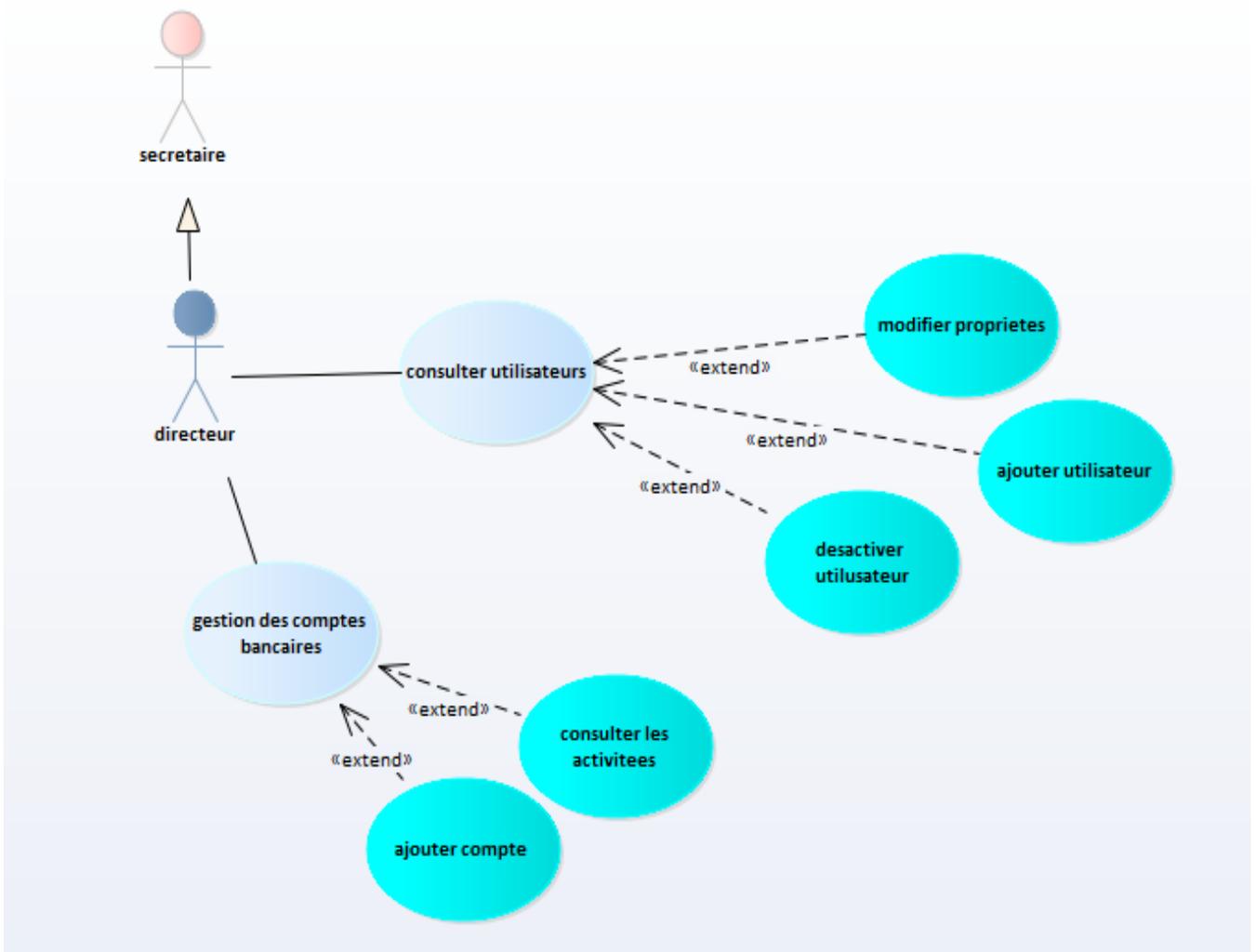


Figure 6 : Diagramme de cas d'utilisation (Directeur)

Directeur (ou admin) : L'administrateur hérite la totalité des fonctionnalités de la secrétaire. Il possède d'autres fonctionnalités supplémentaires :

- **Consulter utilisateurs :** il peut gérer l'ensemble d'intervenant sur le système, il peut ajouter un compte désactiver ou activer un compte changer ses informations.
- **Gestion des comptes bancaires :** ou il peut consulter tous les opérations bancaires et contrôler la validité des justificatifs ainsi que mettre à jours les informations des comptes bancaires de la société.

Acteur 2 : Secrétaire

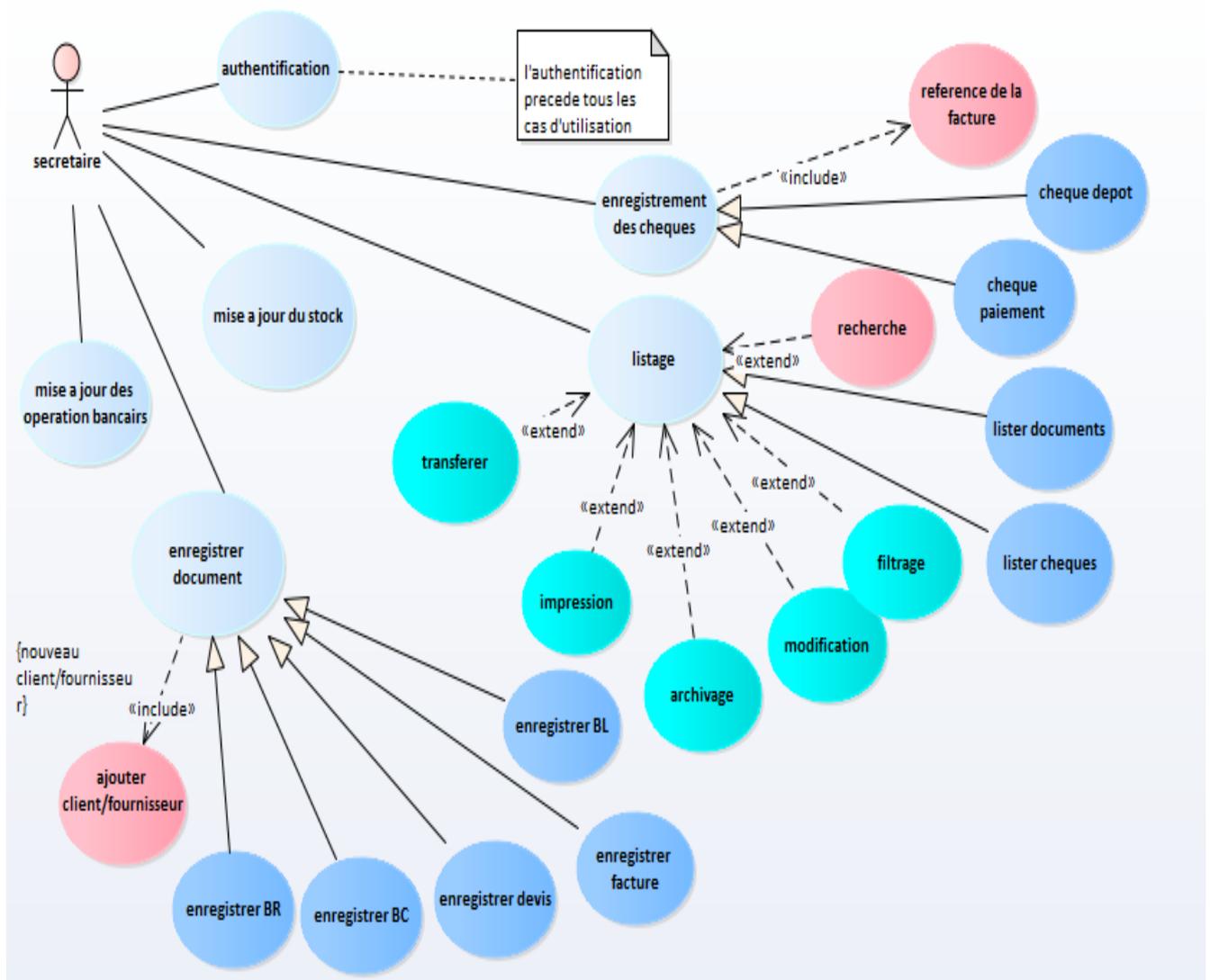


Figure 7 : Diagramme de cas d'utilisation (Secrétaire)

La secrétaire :

- **Listage :** cette fonction permet de consulter les différents documents (les devis les bons de commande les bons de livraison les bons de réception les factures en plus des chèques)

L'utilisateur demande la liste des éléments à lister puis le système affiche la liste, l'utilisateur peut ensuite choisir le mode de filtrage et le système rafraîchit la liste. On peut consulter chaque élément et éventuellement effectuer des modifications, archiver ou imprimer, le système après validation fait les mises à jour nécessaires.

- **Transférer :** en cas de listage des documents sauf les chèques, le système offre l'option de créer à partir d'un document le document qui suit, pour créer un BC d'un devis par exemple l'utilisateur choisi le devis, le système envoie le formulaire de création d'un bon de commande en remplissant le préalablement les informations déjà disponibles dans le devis, à savoir : La référence du client et la liste des produits.

Après la validation, le système enregistre le bon de commande sous l'état. « En attente » et le devis en état transférée cas des factures elle sont tout simplement validé et enregistrer sous l'état payé avec spécification du modèle paiement

- **Enregistrement des chèques :** l'utilisateur peut ajouter deux types de chèques un chèque de paiement : cheque payé par la société, ou un chèque de dépôt : cheque paye par un client
- **L'ajout d'un chèque :** permet de l'associer à une facture comme mode de paiement
- **Mise à jour du stock :** normalement après avoir effectué d'un achat ou bien une vente le système met à jour automatiquement le stock mais en cas exceptionnel l'utilisateur peut faire des modifications sur la liste des articles en stock ou ajouter / supprimer un article.

- **Enregistrer document** : cette fonction permet d'ajouter au système l'un des documents : devis, BC, BL, BR, facture. Le cas est initié par un lancement utilisateur puis le système envoie le formulaire à remplir avec la possibilité d'ajouter un nouveau client s'il n'existe pas déjà sur la liste des clients (un fournisseur en cas d'achat), l'utilisateur en suite ajoute la liste des articles un par un qu'on peut supprimer, le système ajoute le document automatiquement en cas d'erreur on va devoir annuler le formulaire.
- **Mise à jour des opérations bancaire** : l'utilisateur peut ajouter des justificatifs sur chaque opération bancaire à partir des listes des factures et chèques disponibles sur le système.

Acteur 3 : Observateur

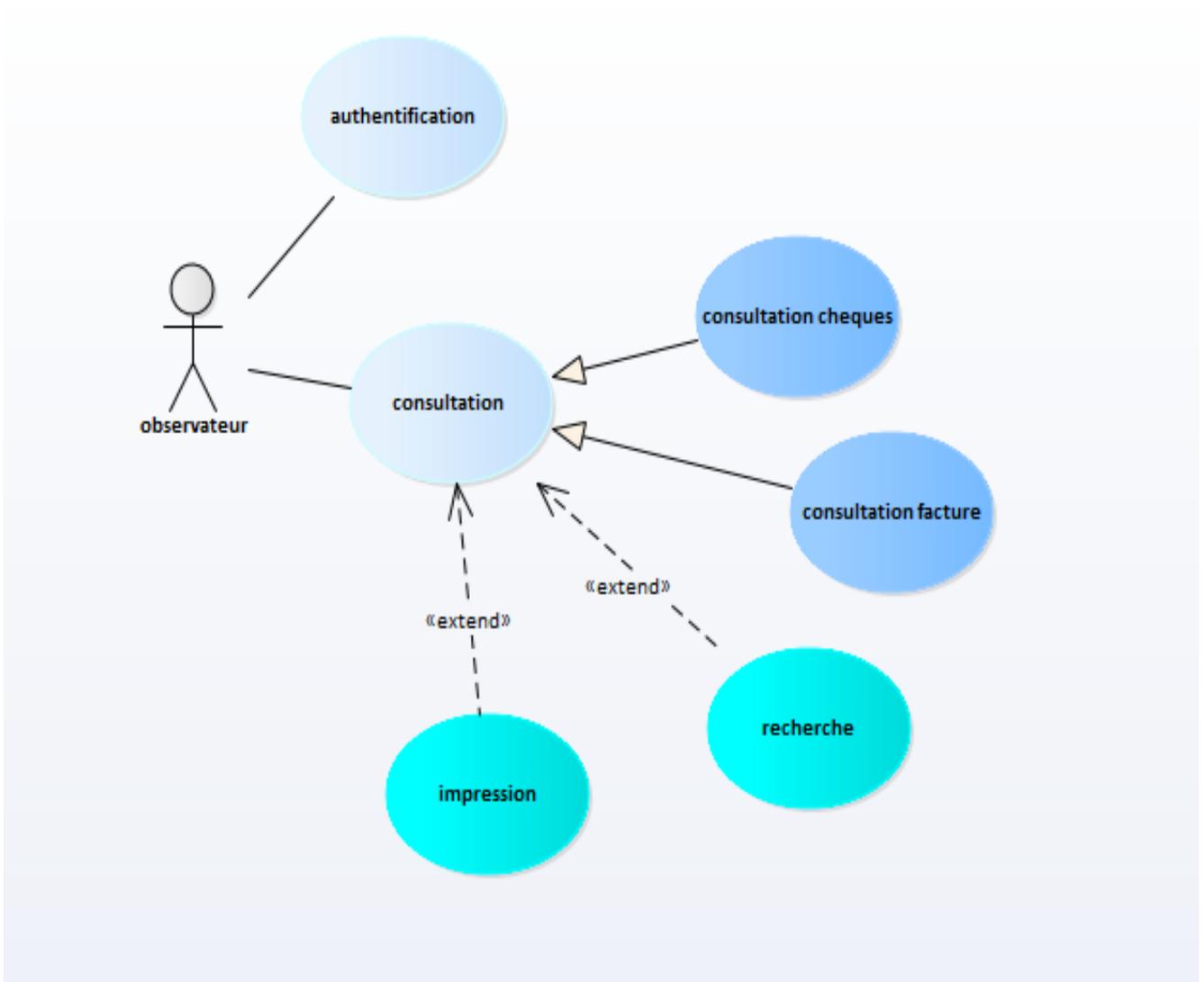


Figure 8 : Diagramme de cas d'utilisation (Observateur)

Observateur : un observateur peut être un stagiaire ou un membre du personnel qui veut accéder à une information.

Les fonctionnalités et l'accès pour cet acteur se limite à :

- **Consulter les factures et les chèques :** il ne peut que lister les factures et les chèques avec option de recherche/filtrage et impression il peut aussi envoyer des notifications si nécessaires.

2.1.3. Diagramme de séquence :

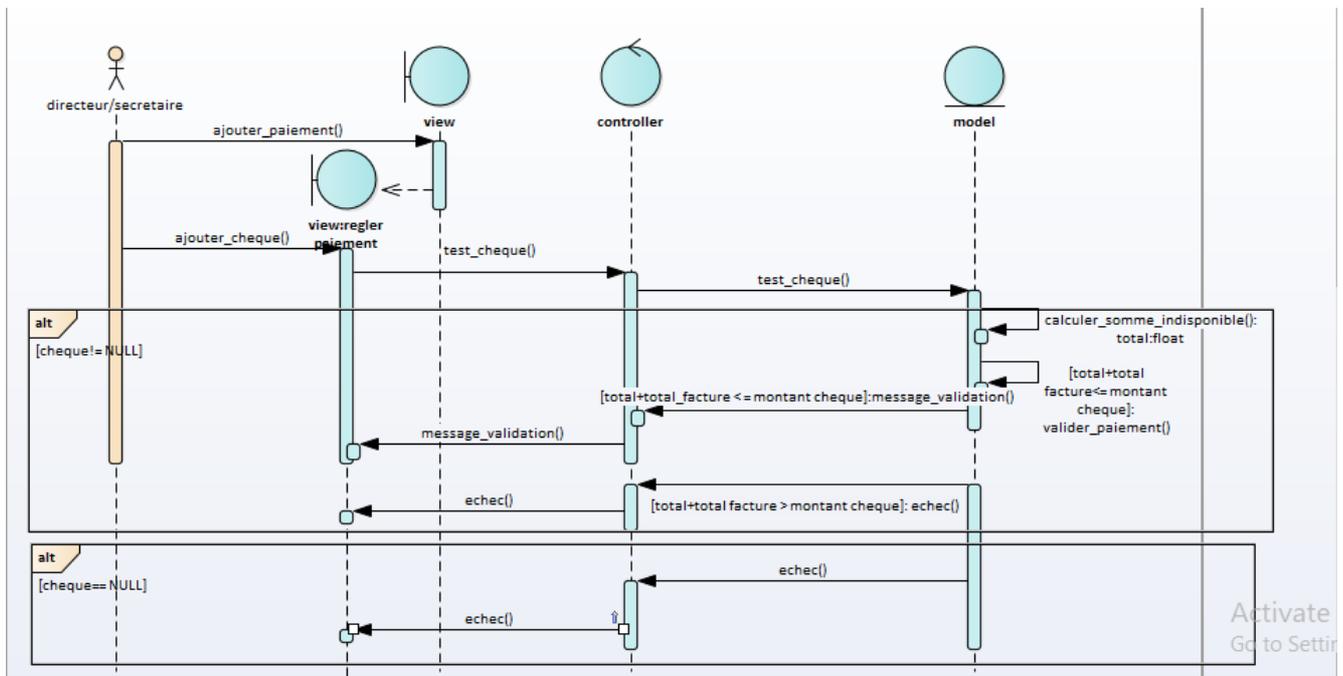


Figure 27 : ajouter paiement à une facture

2.2. Besoins techniques

Le service rendu par cette application se divise en cinq rubriques qu'on peut y accéder après authentification :

A. Vente :

- L'ajout d'un devis est fait par la secrétaire ou le directeur, un devis se compose de la liste des articles avec les quantités, informations sur le client et la date de création et le prix.

On peut lister les devis existants et chercher dans cette liste par le nom du client ou bien par date.

- On peut créer un BC à partir d'un devis en ajoutant la date de création du BC ou le créer à nouveau selon le besoin du client.

Après la période de validité un devis est automatiquement supprimé, on peut le supprimer avant sauf si on a déjà créé un BC.

- Dans le cas normal les articles sont livrés après la commande ce qui entraîne la création d'un BL à partir du BC.

En cas de commande annulé on enregistre le BC avec l'état annulé.

Le 3eme cas c'est avant livraison si le client veut modifier la commande (ou cas d'erreur) création d'un nouveau BC.

- Quand le client reçoit sa commande, on peut générer la facture qu'on peut imprimer (on peut imprimer tous les éléments précédents), on doit aussi ajouter le mode de paiement de la facture.

En cas de livraison incomplète on enregistre le BL original avec les justificatifs et on crée un nouveau BL avec les éléments non livrés qui sera transformé en facture de son côté.

- Le paiement en espèce est valable juste pour les factures moins de 10 000dh

Le paiement par chèque nécessite l'ajout d'un chèque.

- On ajoute la date de paiement si un chèque est payé si non on doit rétablir un autre chèque pour la facture, après que le paiement soit fait avec succès.

B. Achat :

Un bon de réception est ajouté au système quand une commande est livrée est mise en attente on peut le transformer en facture ensuite puis ajouter les justificatifs du paiement de la facture (le chèque associé).

C. Mise à jour du stock :

Le stock est automatiquement mis à jour quand on effectue un achat ou une vente (ajouter article, modifier quantité/prix).

D. Gestion des utilisateurs :

Cette notion est réservée pour le directeur. Dans ce service, le directeur peut ajouter un compte avec les droits du secrétaire ou bien un observateur il peut modifier les droits d'accès de chaque utilisateur comme il peut désactiver/activer son compte.

E. Gestion financière :

Réservé pour le directeur où il peut :

- Consulter les activités des comptes bancaires, comme il est constamment notifié par les dates d'échéance des chèques.
- La secrétaire peut ajouter les justificatifs des opérations bancaires.

2.3. Diagramme de classes

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classes est de modéliser les entités du système d'information. Ces informations sont regroupées ensuite dans des classes.

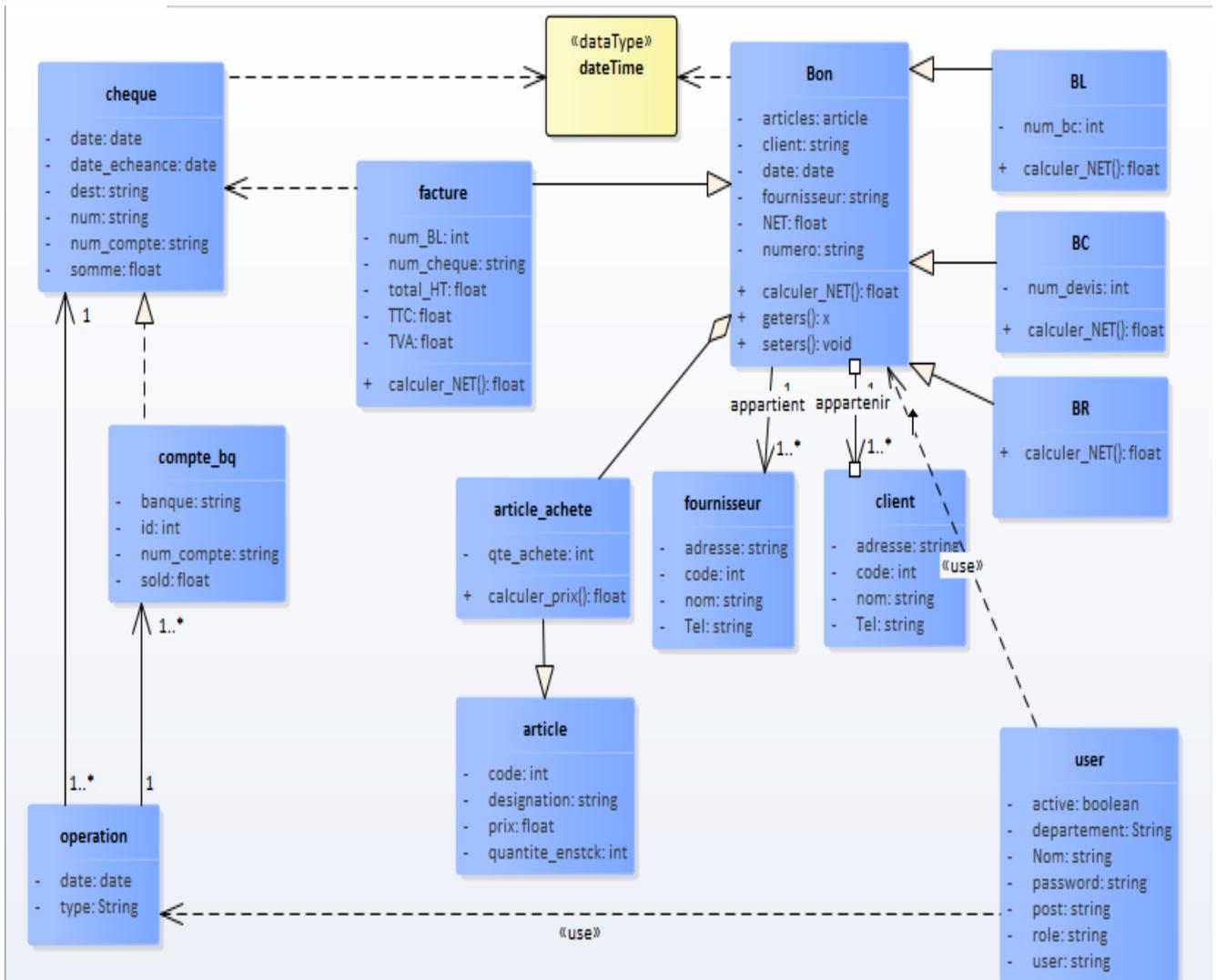


Figure 9 : Diagramme de classes

2.4. Modélisation logique des données

Cette partie permet de donner une vue générale sur les données qui peuvent circuler dans la société et la manière la plus propice pour les organiser sous forme d'une base de données. Cette dernière est un ensemble organisé de données.

Modèle logique des données : Le modèle relationnel représente la base de données comme un ensemble de tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Les tables constituent donc la structure logique du modèle

relationnel où il est possible de relier ces structures à des tables au niveau logique. Les tables ne représentent donc qu'une abstraction de l'enregistrement physique des données en mémoire.

Les règles de passage sont :

- Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.
- Transformation des associations : Nous distinguons trois familles d'associations
- Association 1.. : il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.
- Association .. et n-aire et classes-association : la classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.
- Association 1.. 1 : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.

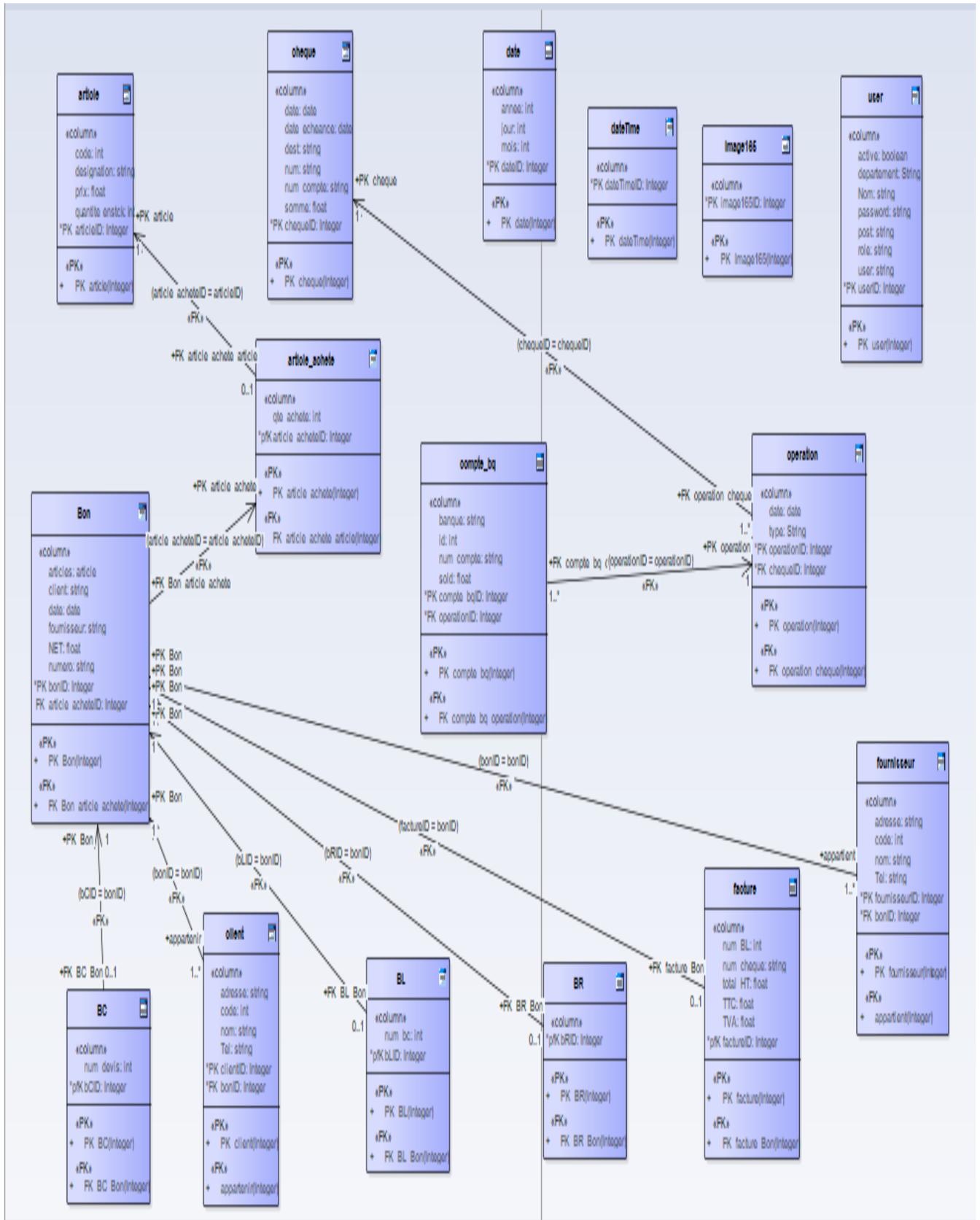


Figure 10 : Schéma de la base de données

3. Architecture applicative

L'architecture applicative qui est adopté est une architecture client-serveur de données, le poste client se contente de déléguer la gestion des données à un service spécialisé. Le cas typique de cette architecture est une application de gestion fonctionnant sous Windows ou Linux et exploitant un SGBD centralisé, ce qui est le cas pour notre application.

La gestion des données est prise en charge par un **SGBD** centralisé, s'exécutant sur un serveur **MAMP**. Ce dernier est interrogé en utilisant un langage de requête qui est **SQL**. Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes.

Cette architecture présente de nombreux avantages qui lui permettent de présenter un bilan globalement positif :

- Elle permet l'utilisation d'une interface utilisateur riche,
- Elle a permis l'appropriation des applications par l'utilisateur,
- Elle a introduit la notion d'interopérabilité.

4. Outils et langages utilisés



➤ **Bootstrap**

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.



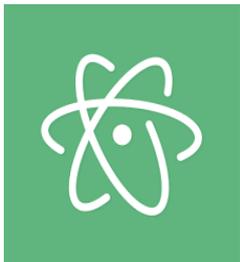
➤ **CSS3**

Il permet un gain de temps lors d'une modification à porter sur le style, le CSS défini dans une feuille de style externe, évite une redondance de classes identiques, tout en clarifiant le code.



➤ HTML5

Issu d'une des trois inventions à la base du World Wide Web, le HTML a été créé dans le but de pouvoir concevoir des pages web.



➤ Atom

Atom est un éditeur de texte libre pour OS X, GNU/Linux, et Windows développé par GitHub. Il supporte des plug-ins écrits en Node.js et implémente Git Control. La plupart des extensions sont sous licence libre et sont maintenues par la communauté.



➤ PHP5

Ressemblant syntaxiquement et structurellement au JAVA, proche du C, le PHP se distingue de ces langages par le fait qu'il puisse intégrer du code HTML. Il permet de rendre un site internet dynamique et reste très convoité par son utilité à communiquer avec des bases de données.



➤ JavaScript

JavaScript permet de programmer des scripts, il est à ce jour utilisé majoritairement sur Internet, conjointement aux pages Web (HTML ou XHTML).



➤ **jQuery**

jQuery est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. La première version est lancée en janvier 2006 par John Resig.



➤ **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde³, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.



➤ **Enterprise Architect**

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus.



➤ **Balsamiq**

Balsamiq est un outil de wireframing rapide qui aide à travailler plus vite et plus intelligemment. Il reproduit l'expérience de l'esquisse sur un tableau blanc, mais en utilisant un ordinateur.

Chapitre III :

Présentation de

l'application

5. Présentation de l'application

5.1. Fenêtre d'Authentification

Retourne un message d'erreur si l'utilisateur est désactivé ou mot de passe/nom d'utilisateur incorrect

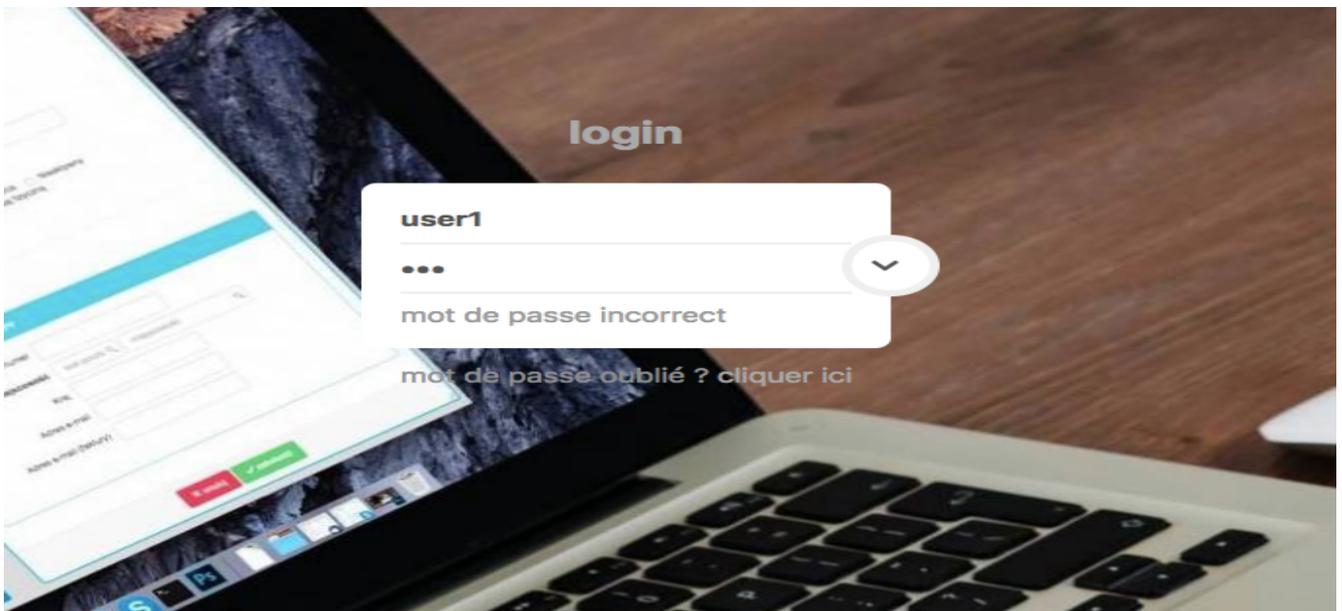


Figure 11 : Fenêtre d'authentification

5.2. Page d'accueil : Directeur

Cette page permet un accès visuel des espaces disponible pour le directeur en plus d'un bref alerte de l'état critique du système

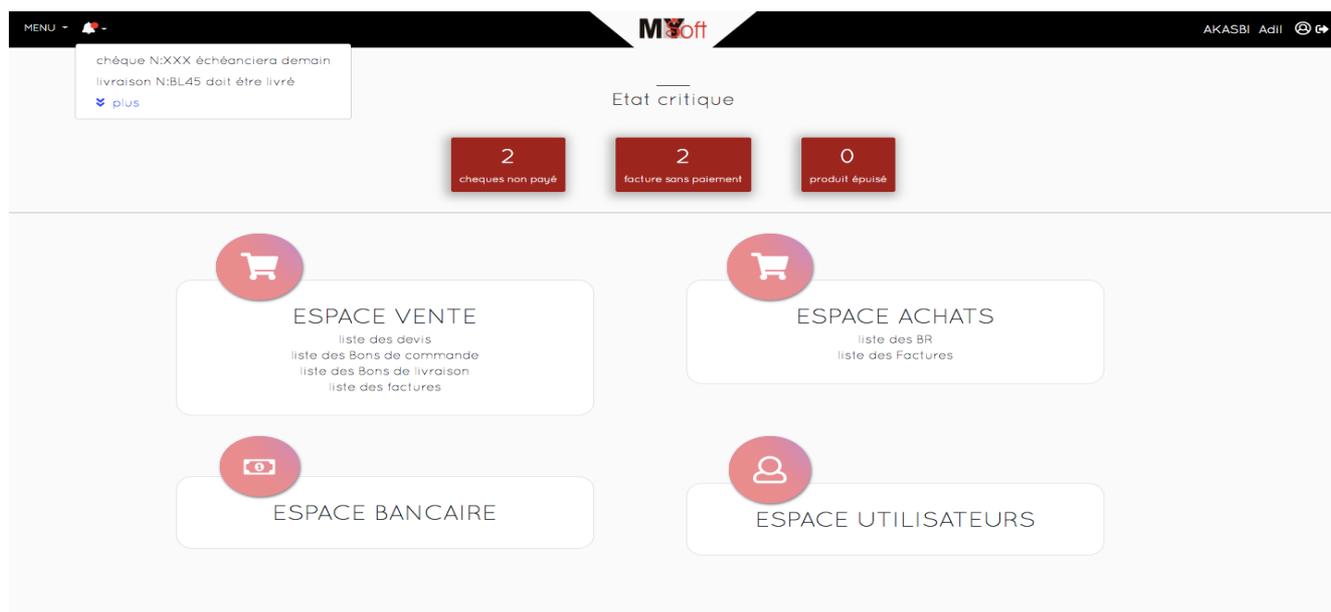


Figure 12 : Page d'accueil : Directeur

5.3. Page d'accueil : secrétaire / observateur

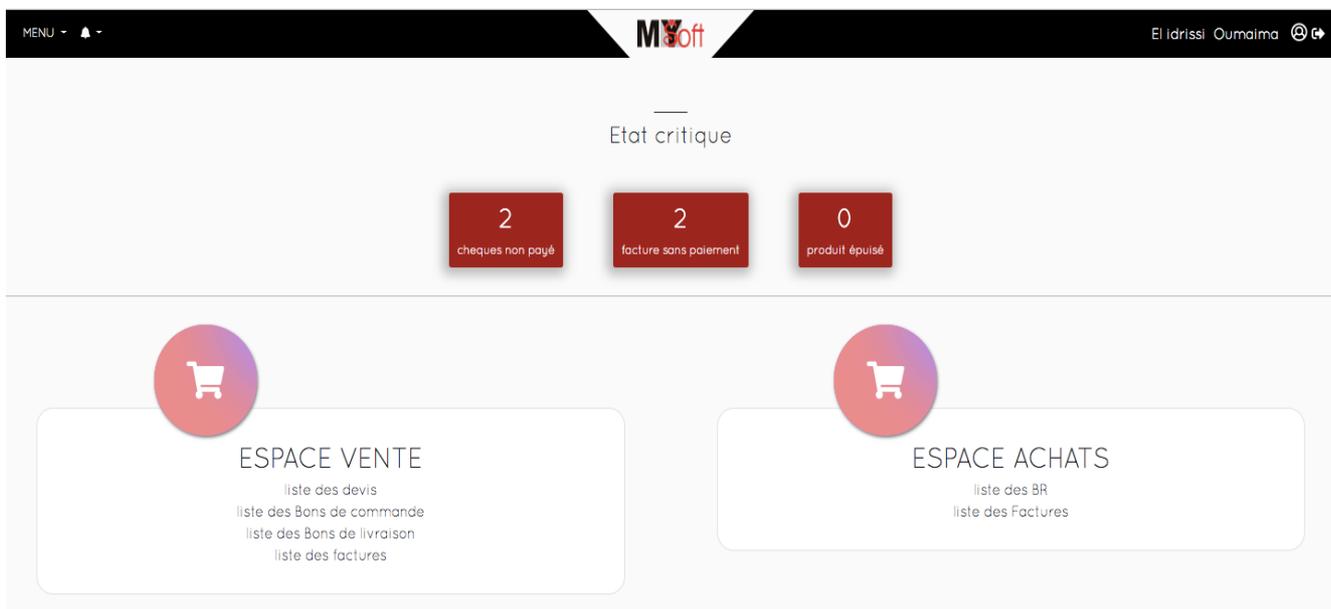


Figure 13 : Page d'accueil : Directeur

5.4. Fenêtre des factures sur l'espace achats

Permet de lister les factures avec un aperçu sur les justificatifs si une facture est payée (num cheque et num opération)

Une facture payé est indisponible en modification

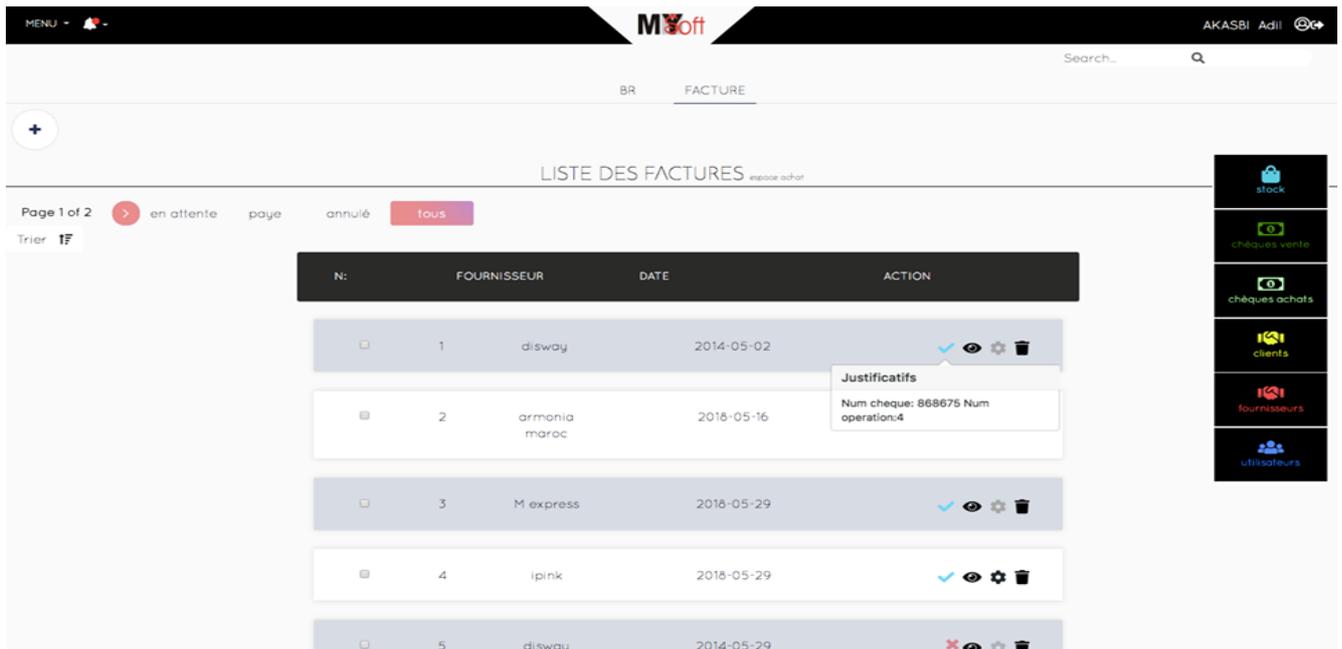


Figure 14 : Fenêtre des factures sur l'espace achats

5.5. Aperçu sur la facture



Figure 15 : Aperçu sur la facture

5.6. Régler le paiement d'une facture en attente ou payé mais sans justificatif

La fenêtre retourne un message d'erreur si : le chèque n'existe pas ou la somme du chèque restante est inférieure à la somme de la facture (car un chèque peut payer plusieurs factures)



Figure 16 : Réglage du paiement d'une facture en attente ou payé mais sans justificatif

5.7. Listes des bons de réception

The screenshot displays a web application interface for "LISTE DES BONS DE RECEPTION" (Purchase Receipts List). The header includes a navigation bar with "MENU", "MSoft", and "AKASBI Adil". Below the header, there are tabs for "BR" (selected) and "FACTURE". The main content area shows a table with the following data:

N:	FOURNISSEUR	DATE	ACTION
<input type="checkbox"/>	1 v	0000-00-00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	2 cd	0000-00-00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	3	0000-00-00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	4 w	0000-00-00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

On the right side of the interface, there is a vertical sidebar with icons for "stock", "chèques vente", "chèques achats", "clients", "fournisseurs", and "utilisateurs".

Figure 17 : Listes des bons de réception

5.8. Création d'une facture à partir d'un BL existant



nouvelle facture

BL N: 18

Fournisseur: ds

Liste des articles:

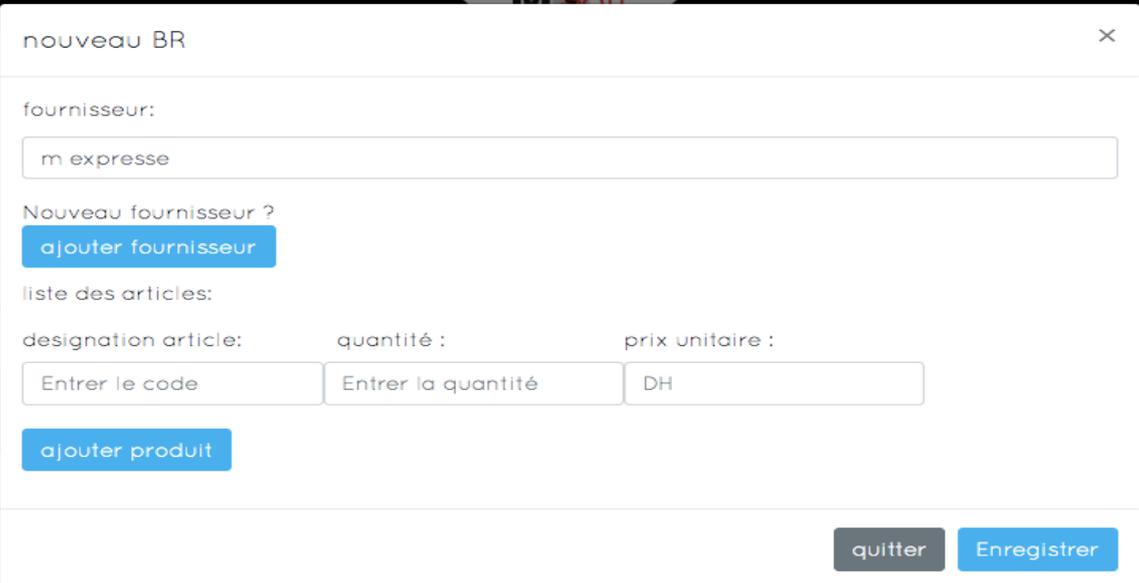
code: 7	qte: 23	
---------	---------	--

valider

Figure 18 : Création d'une facture à partir d'un BL existant

5.9. Fenêtre d'ajout d'un BR

Pour ajouter un BR il faut ajouter les informations des articles achetés si un article existe déjà une mise à jour de la quantité est effectué



nouveau BR

fournisseur:
m expresse

Nouveau fournisseur ?
ajouter fournisseur

liste des articles:

designation article: Entrer le code	quantité : Entrer la quantité	prix unitaire : DH
--	----------------------------------	-----------------------

ajouter produit

quitter Enregistrer

5.10. Liste des produits en stock

On peut ajouter un devis par sélectionner les articles

code	designation	prix	qnalite	ajouter
1	imprimante	1434	342	
2	produit1	322	443	
3	produit2	23	3	

Figure 20 : Liste des produits en stock

5.11. Fenêtre des opérations sur les comptes bancaires de la société

La listes des opérations avec le type de le l'opération (Crédit ou débit), le mode de paiement (virement /versement ou par chèque).

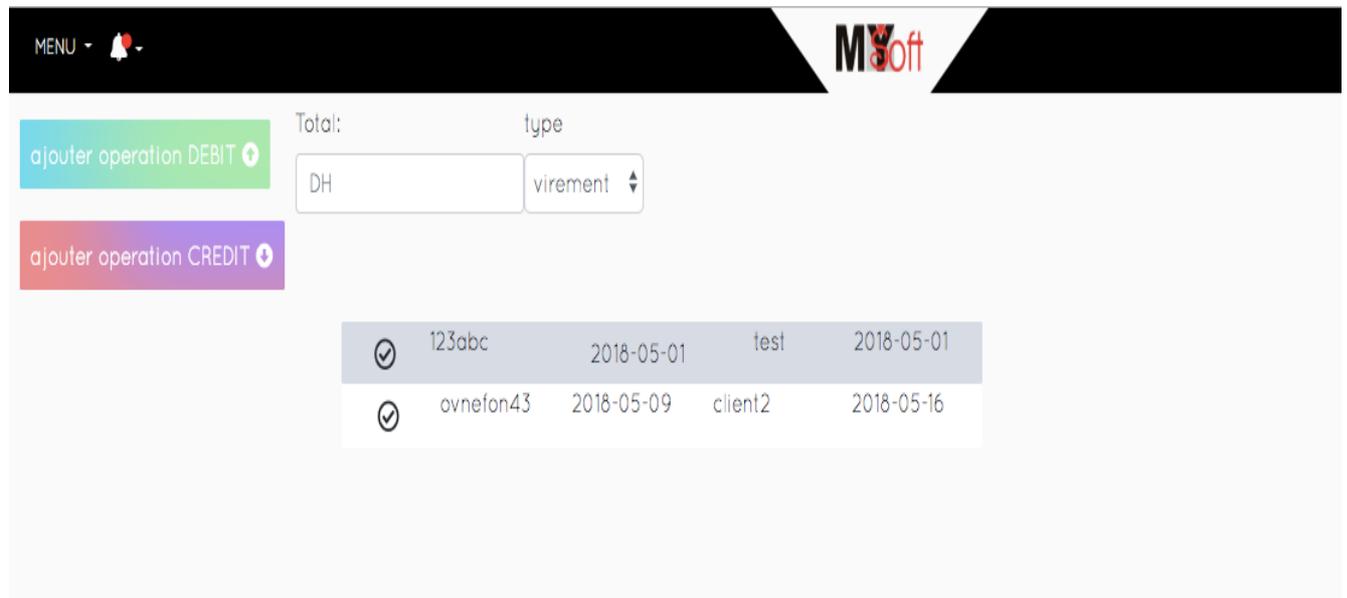
En cas de paiement par chèque les numéros des chèques en question et les factures associé sont cité.

Opération N:	le :	Total :
1	2018-05-16	:42
2	2018-05-08	:344
12	2018-06-06	:646

Figure 21 : Fenêtre des opérations sur les comptes bancaires de la société

5.12. Ajouter une opération

Une opération payée par chèque nécessite l'ajout des chèques dans la liste, le total de l'opération et la somme des chèques sélectionnés doivent être égaux.



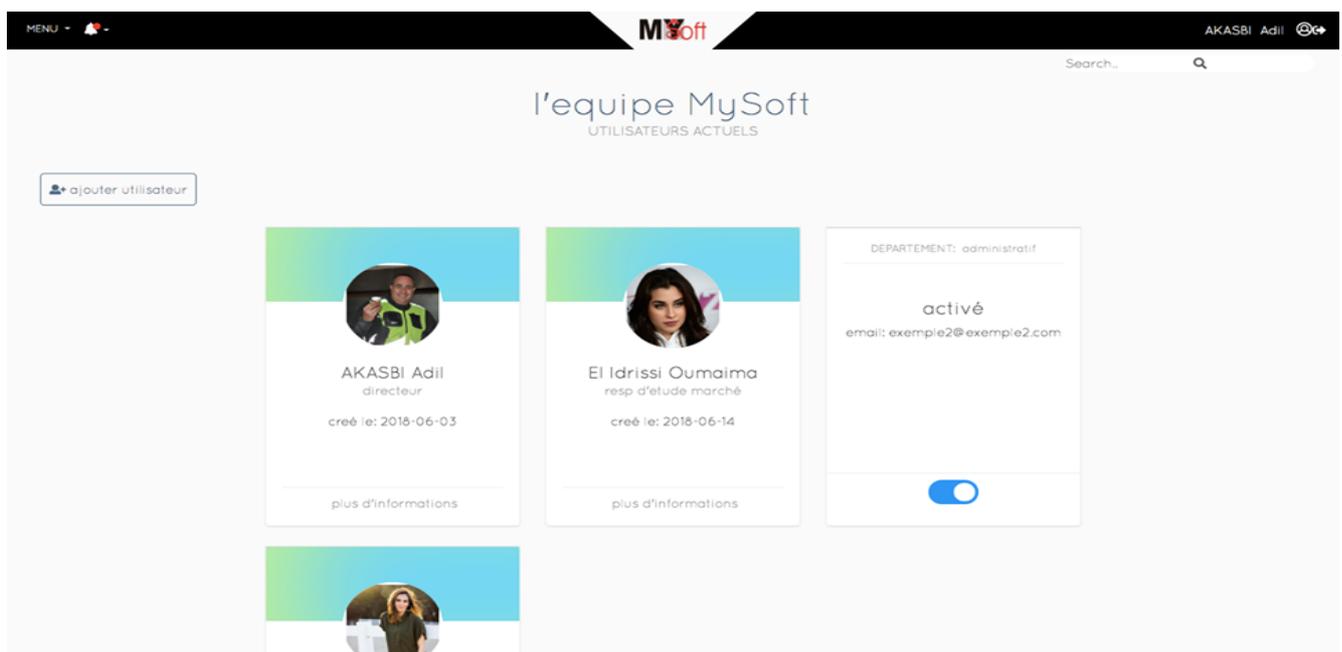
	Total:	type
	DH	virement

<input checked="" type="checkbox"/>	123abc	2018-05-01	test	2018-05-01
<input checked="" type="checkbox"/>	ovnefon43	2018-05-09	client2	2018-05-16

Figure 22 : Ajouter une opération

5.13. Les utilisateurs

Ou le directeur peut consulter les utilisateurs du système et activer/désactiver un utilisateur.



AKASBI Adil
directeur
créé le: 2018-06-03
plus d'informations

El Idrissi Oumaima
resp d'etude marché
créé le: 2018-06-14
plus d'informations

DEPARTEMENT: administratif
activé
email: exemple2@exemple2.com
[toggle switch]

Figure 23 : Interface des utilisateurs

5.14. Fenêtre d'ajout d'un utilisateur

nouveau utilisateur

enregistrement effectué

Nom: Prenom:

photo de profil

Username:

mot de passe:

confirmer le mot de passe:

profession: Departement:

Email address:

Role:

- secretariat
- observation

Figure 24 : Fenêtre d'ajout d'un utilisateur

5.15. Liste des clients

Un ajout d'un client ou un fournisseur n'est pas indépendant si un client est nouveau, on ajout ces information en parallèle avec le document créé.

MENU MySoft

Search...

LISTE DES CLIENTS

Page 1 of 8 >

	société	adresse	e-mail	
0623557698	ipink	adresse1	something@something	
0693635798	clienttest	adresse2	something@something	
0693542398	vrsa	adresse3	something@something	

stock
chèques vente
chèques achats
clients
fournisseurs
utilisateurs

Figure 25 : Liste des clients

Annexe :

Définition

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Plus exactement ce sont les meilleures pratiques du développement objet suivies pour la réalisation d'un système.

5.16. Caractéristiques

Un processus unifié se distingue par les caractéristiques suivantes :

- Itératif : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs ce qui permet de développer une solution effective de façon incrémentale par des itérations multiples ;
- Piloté par les risques : les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité ;
- Centré sur l'architecture : le choix de l'architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du système est basée sur ce choix ;
- Conduit par les cas d'utilisation : le processus est orienté par les besoins utilisateurs présentés par des cas d'utilisation.

Dans la communauté objet et nouvelles technologies ils existent plusieurs processus unifiés en vogue comme eXtreme Programming (XP) et Rational Unified Process (RUP). Dans notre étude, on a choisi de travailler avec le processus 2TUP parce qu'il cible des projets de toute taille, et il a pu faire une large place dans le domaine de la technologie et les risques des projets.

5.17. Le processus 2TUP

Le processus 2TUP (Two Track Unified Process) est un processus unifié. Il gère la complexité technologique en donnant part à la technologie dans son processus de développement.

Le 2TUP propose un cycle de développement qui sépare les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches :

fonctionnelle (étude de l'application) et la technique (étude de l'implémentation). Illustré dans (Figure ci-dessous), le processus 2TUP s'articule autour de trois branches :

- Une branche technique
- Une branche fonctionnelle
- Une branche de conception et réalisation.

La figure suivante détaille les étapes de développement des trois branches du processus 2TUP.

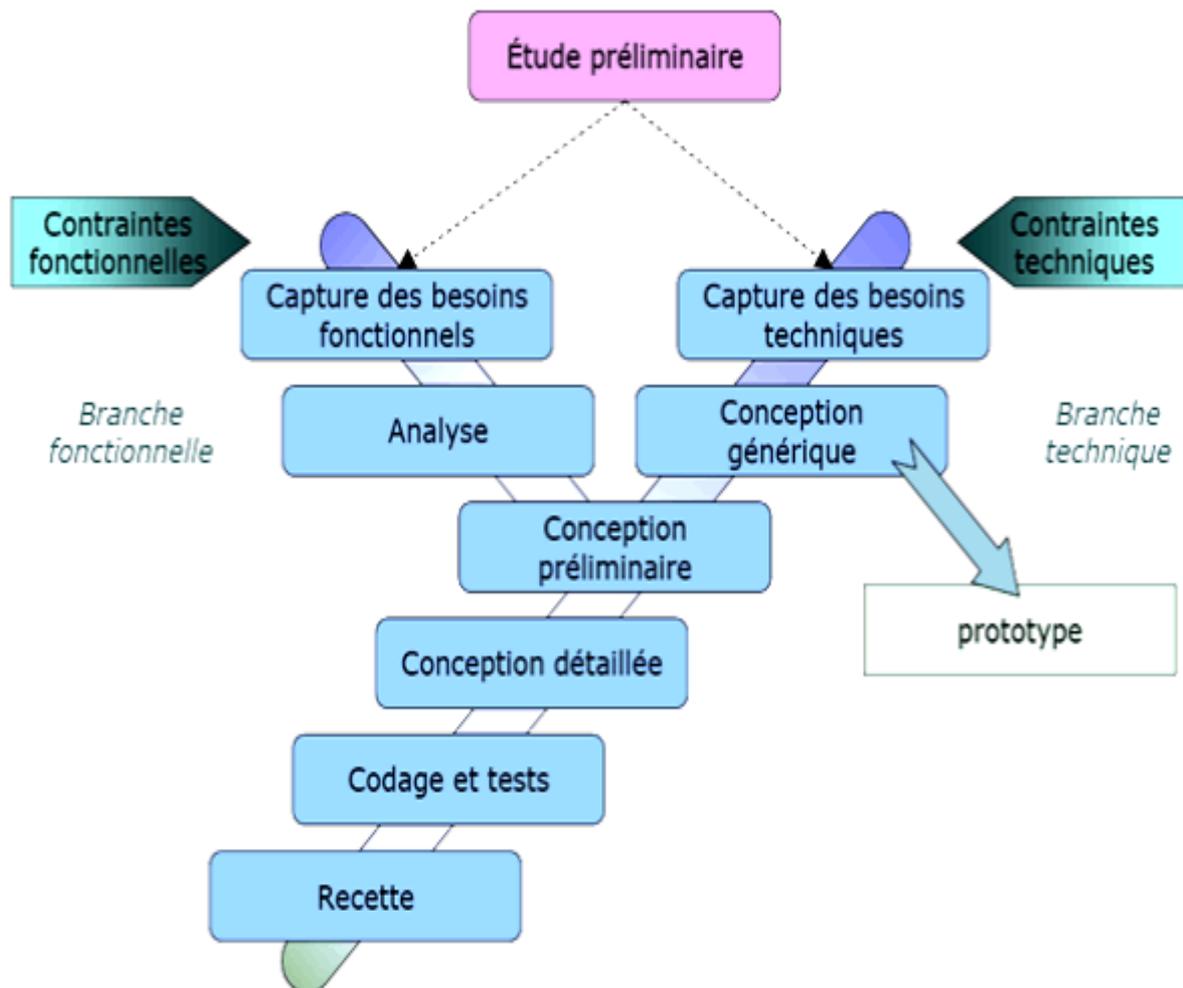


Figure 26 : L'approche 2TUP

5.17.1. Branche fonctionnelle

Les principales étapes de la branche fonctionnelle se présentent comme suit :

L'étape capture des besoins fonctionnels : Cette phase a pour objectif de définir :

- La frontière fonctionnelle entre le système et son environnement.
- Les activités attendues des différents utilisateurs par rapport au système.

L'étape d'analyse : consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

5.17.2. Branche technique

Les principales étapes de la branche technique se présentent comme suit :

L'étape capture des besoins techniques : Cette étape recense toutes les contraintes sur les choix de technologies pour la conception du système. Les outils et le matériel sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique).

L'étape conception générique : Définit les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle permet de générer le modèle de conception technique qui définit les Frameworks.

5.17.3. Branche conception - réalisation

Les principales étapes de cette branche se présentent comme suit :

L'étape conception préliminaire : Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels, Ce qui induit le regroupement des informations des branches technique et fonctionnelle.

L'étape conception détaillée : permet d'étudier comment réaliser chaque composant. Le résultat fournit l'image prête fabriqué du système complet.

L'étape de codage : permet d'effectuer la production des composants et les tests des unités de code au fur et à mesure de leur réalisation.

L'étape de recette : consiste à valider les fonctionnalités du système développé.

5.18. Etude Comparative des processus de développement

Le tableau suivant représente une comparaison entre les différentes méthodes du processus unifié.

	Description	Points fortes	Points faibles
RUP : Rational Unified Process	<ul style="list-style-type: none"> - Promu par Rational - RUP est à la fois une méthodologie et un outil prêt à l'emploi. - Cible des projets de plus de 10 personnes. 	<ul style="list-style-type: none"> - Incrémental itératif - Spécifie le dialogue entre les différents intervenants du projet : les livrables, les prototypes... - Propose des modèles de documents pour des projets type 	<ul style="list-style-type: none"> - Couteux à personnaliser - Très axé processus, au détriment du développement : peu de place pour le code et la technologie.
XP : eXtreme Programming	<ul style="list-style-type: none"> - Ensemble de "Best Practices" de développement en équipe, transfert de compétences... 	<ul style="list-style-type: none"> - Itératif - Fait une large place aux aspects techniques (prototypes, règles de développement, tests...) 	<ul style="list-style-type: none"> - Ne couvre pas les phases en amont et en aval au développement. - Flou dans sa mise en œuvre : quels intervenants ? quel livrables ?
2TUP : Two track Unified process	<ul style="list-style-type: none"> - S'articule autour de l'architecture. - Propose un cycle de développement en Y. - Cible les projets de toute taille. 	<ul style="list-style-type: none"> - Fait une large place à la gestion technologie et à la gestion des risques. - Définit des profils des intervenants, les livrables, les prototypes... 	<ul style="list-style-type: none"> - Difficulté de fusion et d'intégration des deux branches - Ne propose pas de documents types.

Conclusion et perspectives

Le présent projet réalisé au cours de mon stage de fin de formation a été très bénéfique tant au niveau personnel qu'au niveau académique. Ainsi, il m'a permis de consolider mes connaissances en termes d'outils de programmation et de conception, ainsi que la gestion de projets. Il a été une opportunité pour développer de nouvelles compétences dans la programmation avec PHP.

Ce stage m'a aussi offert la chance de découvrir et travailler dans un contexte réel et affronter de vraies problématiques. Ces dernières m'ont aidé à acquérir une expérience très riche et diversifiée. Ainsi, j'ai travaillé avec une base de données réelle et volumineuse en termes de tables et relations avec tout ce que cela implique en termes de complexité du projet.

La société MYSOFT SARL qui m'a accueilli pendant ce stage est en train de mettre en place une application de gestion. Partant je suis chanceuse et très fière d'avoir pu contribuer au développement de cette solution de gestion de facturation et de paiement. Fort de cette expérience et en réponse à la modernisation de la gestion au sein de la société d'accueil, il est prévu comme perspectives :

- Ajouter un système de gestion des factures payé en espèce ;
- Ajouter un système de notification interne ;
- Utiliser PHP mailer pour envoyer les listes des opérations et leurs justificatifs au comptable de la société ;
- Utiliser firebase cloud storage pour le stockage des documents PDF programmer en orienté objet ;
- Ajouter un module de gestion des ressources humaines et la gestion du stock.

Webographie

Documentation PHP :

<http://www.php.net/manual>

Documentation jQuery :

<https://api.jquery.com/>

<http://www.stackoverflow.com/tags/jquery/info>

Documentation Bootstrap :

<http://www.w3schools.com>

<http://www.getbootstrap.com/css/>

Forum & Astuces :

<http://stackoverflow.com>

Cours en ligne :

<https://www.udemy.com/cours-javascript>

<https://openclassrooms.com/courses/simplifiez-vos-developpements-javascript-avec-jquery>

Vidéos tutoriels:

<https://www.youtube.com/user/DevTipsForDesigners>