

Université Assane Seck de Ziguinchor
UFR Sciences et Technologies
Département informatique



Mémoire de fin d'études

Mention : Informatique

Spécialité : Génie Logiciel

Sujet :

Automatisation de la gestion des biens matériels de
l'UFR des Sciences et Technologies

Présenté par :

M. El Hadji Mamadou Korka DIALLO

Soutenance le 29/03/2018

Membres du jury

- Pr. Salomon SAMBOU (**Président du jury**)
- M. Gorgoumack SAMB (**Rapporteur**)
- Dr. Youssou DIENG (**Examineur**)
- Dr. Ibrahima DIOP (**Encadrant**)
- Dr. Khadim DRAME (**Encadrant**)
- M. Alain Charles GOMIS (**DRH**)
- M. Alioune Badara SAKHO (**Responsable des services généraux**)
- M. Al Ousseynou FALL (**CRI**)

Sous la direction de :

- Dr. Ibrahima DIOP
- Dr. Khadim DRAME

Sous la supervision de :

- Pr. Salomon SAMBOU

Résumé

L'Université Assane Seck de Ziguinchor (UASZ) est une institution qui ne cesse de grandir. En effet, elle continue à recruter du personnel et les infrastructures poussent en permanence. Ce qui implique donc une forte demande de matériels pour pouvoir satisfaire les besoins. Ainsi la gestion devient de plus en plus fastidieuse, car l'exécution de certaines tâches reste toujours difficile ou manuelle. De plus, il est noté que deux logiciels sont utilisés, mais malheureusement ils ne règlent pas tous les problèmes.

Inscrit dans ce cadre, notre stage a pour objectif de concevoir et d'implémenter une application portant sur l'automatisation de la gestion des biens matériels. La problématique est par conséquent la suivante : comment apporter une solution, qui satisfait le besoin du comptable de matières dans une époque où il existe des outils pour identifier les matières et réduire la saisie manuelle dans la gestion des biens matériels. Notre cadre d'application est la gestion des biens matériels à l'Unité de Formation et de Recherche (UFR) Sciences et Technologies (ST).

Ainsi, notre solution, en plus d'automatiser la gestion des biens matériels de l'UFR ST s'appuie sur l'utilisation des codes à barres pour l'identification de chaque matériel, d'une imprimante thermique pour imprimer ces codes-barres et d'une douchette pour la lecture de ces codes afin de diminuer la saisie manuelle.

Remerciements

En préambule à ce mémoire, nous rendons grâce à ALLAH, qui nous aide et nous donne la patience et le courage durant ces longues années d'étude.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenons à remercier Dr. Ibrahima DIOP et Dr. Khadim DRAME, qui, en tant qu'encadrants, se sont toujours montrés disponible tout au long de la réalisation de ce mémoire, ainsi que pour l'inspiration, l'aide et le temps qu'ils ont bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Nous tenons à remercier Dr. Youssou Dieng d'avoir mis en place ce cadre de partenariat entre le Département d'Informatique et la DRH, c'est qui nous a permis d'avoir ce stage.

Ces remerciements vont également à l'endroit du personnel enseignant et administratif de l'UASZ notamment ceux de l'UFR des Sciences et technologiques (ST) particulièrement au département informatique, pour la richesse et la qualité de leurs enseignements et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Nous tenons également à exprimer notre gratitude à Mme MANE et à M. Alioune Badara SAKHO, pour leur confiance manifeste, leurs précieux conseils et leur foi dans l'action.

Nous n'oublions pas nos camarades de formation et les remercions chaleureusement pour tous ces agréables moments passés ensemble.

Je tiens enfin à remercier tout particulièrement ma famille qui m'a accordé la liberté d'action et la patience nécessaire pour réaliser ce travail ainsi que toutes les personnes qui m'ont soutenu.

Dédicace

Je dédie ce mémoire à

Mes parents

Ma mère Dieinabou DIALLO et mon Père Thierno Sadou DIALLO pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études

Mes chers frères et sœur

Mamadou Saliou DIALLO, Idrissa DIALLO et de Rouguiyatou DIALLO pour leurs encouragements, et leur soutien moral tout au long de mon parcours universitaire.

Table des matières

Résumé.....	i
Remerciements.....	ii
Dédicace.....	iii
Table des matières.....	iv
Liste des figures	vii
Liste des Tableaux	ix
Liste des Abréviations.....	x
Introduction générale	1
Chapitre I: Description du sujet de mémoire	2
I. Cadre du projet.....	2
1. Présentation de l'Université Assane Seck de Ziguinchor	2
1.1 Le Rectorat.....	2
1.2 Les UFR	3
2. La gestion des matières	4
2.1 La commission de réception	4
2.2 Le processus de gestion du matériel	4
2.2.1 La réception ou l'entrée	5
2.2.2 Attribution ou affectation	5
2.2.3 Les sorties	5
2.3 L'existant pour la gestion des biens de l'UASZ	6
2.4 Les problèmes	6
3. Problématique.....	7
3.1 Objectif du sujet.....	7
3.2 Scénario d'utilisation de notre solution	7
Chapitre II: Méthodologie.....	9

I. Les méthodes classiques	9
II. Les méthodes agiles	9
1. La méthode Scrum.....	10
Chapitre III: Spécification et analyse des besoins fonctionnels.....	13
I. Spécification des besoins fonctionnels	13
1. Identification des acteurs.....	13
2. Identification des fonctionnalités	13
3. Diagramme de cas d'utilisation.....	14
3.1 Diagramme de cas d'utilisation pour la gestion des informations.....	14
3.2 Diagramme de cas d'utilisation pour la gestion des entrées.....	15
II. Analyse des besoins fonctionnels	16
1. Analyse de l'authentification.....	16
1.1 Description du cas d'utilisation « authentification »	16
1.2 Activités de l'authentification.....	16
1.3 Le scénario normal de l'authentification	17
2. Analyse du cas « ajouter groupe » du package « gestion des informations ».....	17
2.1 Description du cas d'utilisation « ajouter groupe »	18
2.2 Les activités du cas « ajouter groupe »	18
2.3 Le scénario normal du cas « ajouter groupe »	19
3. Analyse du cas « gérer entrée ».....	20
3.1 Description du cas d'utilisation « gérer entrée ».....	20
3.2 Les activités de « gérer entrée ».....	20
Chapitre IV : Conception du système	22
I. Architecture du système de notre application	22
1. Architecture générale	22
1.1 L'architecture 3-tiers.....	23
1.2 L'architecture MVC	23

2.	Architecture du client	24
II.	Conception générale.....	25
1.	Diagramme de composant.....	25
2.	Diagramme de déploiement.....	26
3.	Diagramme de paquetage	27
III.	Conception détaillé.....	27
1.	le Diagramme de Classe	28
2.	Le dictionnaire des données	30
Chapitre V : Implémentation et présentation de l’application		33
I.	Implémentation de notre base de données dans le SGBDR MySQL.....	33
II.	La plateforme JavaEE	33
1.	Spring Boot.....	33
2.	Maven.....	34
3.	Java Persistence API (JPA)	34
4.	Thymeleaf.....	35
5.	Bootstrap	35
III.	Outils de travail utilisé	35
1.	Intellij IDE.....	35
2.	StarUML.....	35
3.	Serveur Tomcat.	36
4.	POS driver setup.....	36
5.	Zxing	36
IV.	Présentation de l’application	36
1.	L’interface d’authentification.....	37
2.	L’interface de la page d’accueil	37
3.	L’interface « d’ajout d’une réception »	38
4.	Attribution	44

Conclusion générale.....	46
Bibliographie et Webographie	47

Liste des figures

Figure 1 : Organigramme de l'UASZ.....	3
Figure 2 : Cycle de Scrum [1], [8], [9] et [10]	12
Figure 3:Diagramme de cas d'utilisation pour la gestion des informations	15
Figure 4: Diagramme de cas d'utilisation pour la gestion des informations	15
Figure 5: Diagramme d'activité de l'authentification	17
Figure 6: Diagramme de séquence pour l'authentification.....	17
Figure 7: Diagramme d'activités du cas ajouter groupe.....	19
Figure 8 : Diagramme de Séquence de cas ajouter un Groupe	19
Figure 9 : Activité de gérer les entrées	21
Figure 10: Architecture trois tiers	23
Figure 11 : Architecture générale.....	24
Figure 12 : Architecture générale de notre application.....	25
Figure 13 : Diagramme de composant	26
Figure 14 : Diagramme de déploiement.....	26
Figure 15 : Diagramme de Paquetage	27
Figure 16 : Diagramme de Classes de l'authentification.....	28
Figure 17 : Diagramme classe participante à la fonctionnalité Organisation	29
Figure 18 : Diagramme de classe participante à la fonctionnalité catalogue.....	29
Figure 19 : Diagramme classe participante à la gestion des bons.....	30
Figure 20 : Entité avec annotation JPA.....	34
Figure 21 : Arborescence des menus de la page d'accueil	37
Figure 22 : Page d'accueil	38
Figure 23 : liste de toutes les réceptions	38
Figure 24 : Formulaire d'ajout d'une réception.....	39
Figure 25 : Le choix du montant de la facture	39
Figure 26 : Formulaire d'ajout d'une réception de matière	40

Figure 30 : Bon d'entrée avec les différents produits	42
Figure 31 : Codes-barres du premier produit ajouté	42
Figure 35 : Codes-barres du deuxième produit ajouté	42
Figure 32 : Commission de réception	43
Figure 33 : Le PV de réception du nouveau Bon d'entrée.....	43
Figure 34 : Formulaire du Procès-Verbal	44
Figure 35 : Procès-Verbal complété.....	44
Figure 40 : Configuration de la base de donnée avec Intelij.....	49
Figure 42: Configuration de MySQL connecteur	50
Figure 43 : Génération de la base de données.....	51
Figure 44: Code Source de l'impression	52
Figure 45 : Code source de la classe BarCode.....	53
Figure 46 : Code source de Génération de code-barre.....	54

Liste des Tableaux

Tableau 1 : Les acteurs.....	13
Tableau 2: Le Backlog du produit.....	13
Tableau 3 Description du cas Authentifier	16
Tableau 4 : Description du cas ajouter Groupe.....	18
Tableau 5 : Dictionnaire de donnée	30

Liste des Abréviations

IDE :	Environnement de Développement Intégré
Java EE :	Java Entreprise Edition
LASH :	Lettres, Arts Sciences humaines
MPD :	Model Physique de Données
PATS :	Personnel Administratif, technique et de Service
PER :	Personnel Enseignant et Recherche
RUP:	Rational Unified Process
RAD:	Rapid Application Development
SES :	Sciences Economiques et Sociales
SGBDR :	Système de Gestion de Base de Données Relationnel
ST :	Sciences et Technologies
UASZ :	Université Assane Seck de Ziguinchor
UFR :	Unité de Formation et de Recherche
XP :	Extreme Programming

Introduction générale

Une petite structure peut au début de son existence assurer sa gestion administrative, financière ou matérielle de façon manuelle. Une fois que la structure commence à grandir, l'automatisation de sa gestion par un système devient nécessaire. Ainsi, à un certain moment, la structure atteindra un niveau où la gestion, plus précisément la gestion matérielle, deviendra plus complexe comme c'est le cas avec notre institution qui est l'Université Assane Seck de Ziguinchor (UASZ).

Des solutions ont pu être proposées dont l'une d'entre elles est utilisée dans notre Unité de formation et de recherche (UFR) Sciences et Technologies (ST). Mais malheureusement, ces types de solutions ne règlent pas tous les problèmes liés à la gestion des biens. Ces solutions sont généralistes, c'est-à-dire ils sont conçus pour la gestion des biens matériels de façon générale et non pas pour le cas particulier de la gestion des biens matériels de l'UFR ST ou de l'UASZ. Ce qui fait que ces logiciels ont des fonctionnalités non utilisées au moment où des fonctionnalités essentielles pour le contexte spécifique de son utilisation sont absentes.

C'est dans ce cadre que s'inscrit notre mémoire qui consiste à concevoir et implémenter une application optimale pour la gestion des biens matériels. Cette solution, en plus d'automatiser la gestion des biens matériels de l'UFR ST, s'appuie sur l'utilisation des codes à barres pour l'identification de chaque matériel, d'une imprimante thermique pour imprimer ces codes-barres et d'une douchette pour la lecture de ces codes afin de diminuer la saisie manuelle.

Pour vous présenter les objectifs et les résultats de ce travail, nous avons rédigé ce mémoire qui s'articule autour de cinq chapitres. Le premier chapitre est consacré à la présentation du contexte de notre sujet de mémoire. D'abord, nous allons présenter notre institution l'UASZ et son organisation structurelle. Ensuite, nous allons parler de la comptabilité des matières au sein des UFR en exposant les problèmes rencontrés. Enfin la problématique de notre sujet de mémoire. Dans le deuxième chapitre, nous présenterons la méthodologie adoptée pour la réalisation de notre travail. Dans le troisième chapitre, il sera question de faire ressortir tout ce dont l'application aura besoin en termes de fonctionnalités et d'entités en interaction avec le système, ainsi que les différents scénarios entre le système et les entités externes. Le quatrième chapitre présentera l'architecture du système en montrant les différents éléments qui interagissent dans et avec le système ainsi que les différents diagrammes qui explicitent la conception du système. Le cinquième chapitre va présenter les outils que nous avons utilisés pour la réalisation du présent projet et l'application mise en œuvre.

Chapitre I: Description du sujet de mémoire

Comprendre notre sujet nécessite de connaître notre environnement ainsi que de faire une description sur la gestion des biens matériels à l'UASZ. C'est ce qui fait que dans ce chapitre, nous allons d'abord aborder le cadre de notre travail où nous allons présenter notre institution, la problématique, l'existant et la solution proposée.

I. Cadre du projet

1. Présentation de l'Université Assane Seck de Ziguinchor

L'Université Assane SECK de Ziguinchor (UASZ) est créée par le décret 208-537 du 22 mai 2008. Elle commença ses services en février 2007 avec seulement trois (3) Unités de Formation et de Recherche (UFR) que sont l'UFR des Sciences et Technologies (ST), l'UFR des Sciences économiques et sociales (SES) et l'UFR des Lettres, Arts et Sciences humaines (LASHU). Avec 257 étudiants comme effectifs de départ, pour l'année 2016/2016 elle compte 3535 étudiants répartis dans quatre (4) ; L'UFR des Sciences de la Santé (SS) a été créée pendant l'année académique 2011.

En plus des UFR, l'Université a aussi d'autres entités comme le Rectorat, qui gère l'administration de l'UASZ.

À l'UASZ, la gestion des biens matériels est assurée séparément par le Rectorat et par les UFR.

1.1 Le Rectorat

Il est la structure chargée de gérer toutes les tâches administratives. Il est constitué de plusieurs services et de directions dans lesquelles nous retrouvons des agents qui constituent le **Personnel administratif**, technique et de **Service (PATS)**. Parmi eux, il y'a le *Comptable des matières central* chargé de gérer les biens matériels des différents services et directions du Rectorat.

Parmi ces différents services et structures, nous pouvons citer:

- le **vice-rectorat**, qui est subdivisé en deux sous-structures qui sont le **vice-rectorat chargé des Études et vies universitaires** et le **vice-rectorat chargé de la recherche, de la Coopération et des Relations avec le Monde professionnel** ;
- la **direction des études et des réformes universitaires** et la **direction de la vie de l'Université**, qui est rattachées au vice-rectorat chargé des études et de la vie universitaire ;

Sujet : Automatisation de la gestion des biens matériels de l'UFR des Sciences et Technologies

- le **Centre des Ressources informatiques (CRI)**, qui a pour rôle d'assurer le développement des moyens informatique, réseaux, télécommunication et de veiller à la disponibilité des ressources matérielles et logicielles ;
- le **secrétariat général**, qui gère plusieurs directions, à savoir la direction de la scolarité centrale, la direction des finances et de la comptabilité, la direction des ressources humaines, la direction de la gestion des patrimoines et de la maintenance où est rattaché le **comptable des matières central** chargé de gérer les biens matériels des différents services et directions du Rectorat.

1.2 Les UFR

Une Unité de Formation et de Recherche (UFR) est la structure chargée d'assurer la bonne gestion des questions d'ordre pédagogique. Chaque UFR est composée d'un ou plusieurs départements auxquelles sont rattachés des enseignants. Ces derniers sont des agents et constituants le **Personnel d'Enseignement et de Recherche (PER)**. Dans chaque UFR, nous avons un **comptable des matières** qui est chargé de gérer tous les biens matériels de cette UFR.

La figure 1 ci-dessous décrit l'organigramme de l'UASZ.

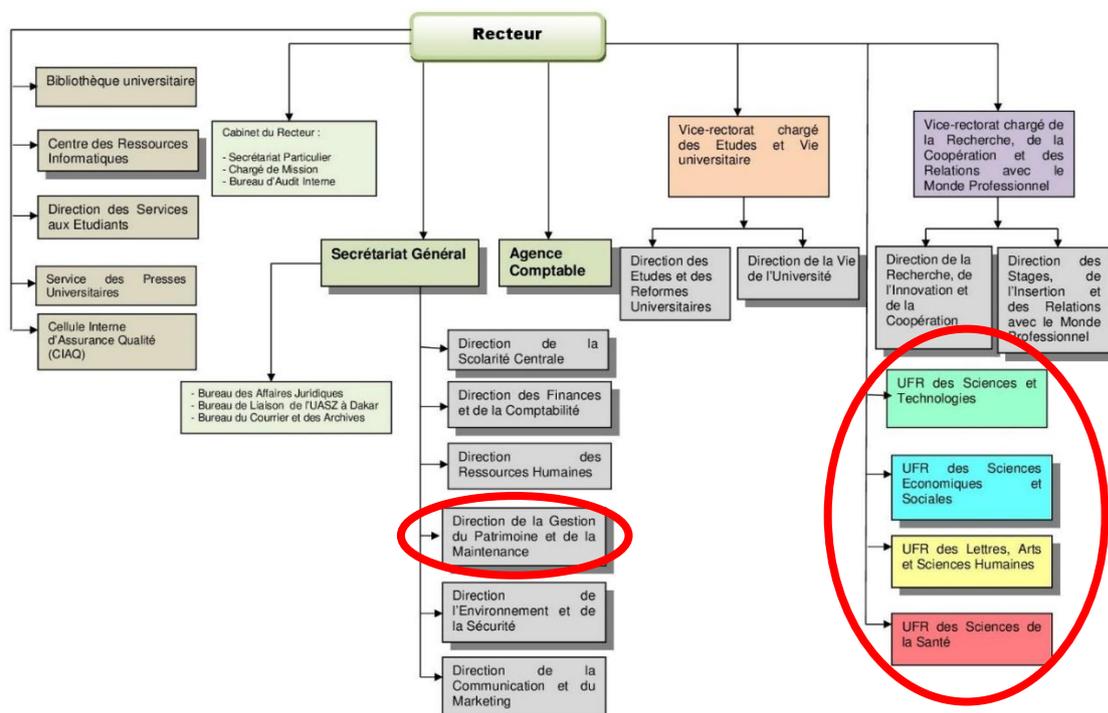


Figure 1 : Organigramme de l'UASZ

2. La gestion des matières

Par définition, la comptabilité est l'ensemble des techniques d'enregistrement des entrées et sorties financières d'une entreprise, d'une administration, ou d'une association.

Selon le journal officiel n° 4860 du 10 octobre 1981, la comptabilité des matières est la conservation dans le temps d'un patrimoine, qui en dernier ressort, appartient à la collectivité nationale et qui doit être sauvegardé, qu'il y ait ou non personnalité morale distincte et autonomie financière.

Pour mettre en harmonie la gestion des biens matériels, l'Etat a beaucoup contribué en la réglementant par des chartes juridiques.

En 1982, le ministre d'État, le ministre de l'Économie et des Finances de l'époque avait signé les instructions *N°90/MEF/DGF/DMTA du 27 août 1982 et 6680 du 21 novembre 1985* relatives à la comptabilité des matières. En effet, ces instructions avaient pour objet de préciser les modalités d'application du décret *n° 81-844 du 20 août 1981* relative à la comptabilité des matières appartenant à l'État, aux collectivités locales et aux établissements publics (*journal officiel n° 4860 du 10 octobre 1981, page 923*).

De nos jours, fait à Dakar le 23 mars 2007 par le Premier ministre de l'époque, le nouveau décret *n°2007-434 du 23 mars 2007* met hors usage et remplace les articles 5, 7, 13, 9 alinéas 3 et 4 du décret *81-844 août 1981* relatifs à la comptabilité des matières appartenant à l'État, aux collectivités locales et aux établissements publics.

En effet, la mise en pratique du décret *81-844 août 1981* a fait apparaître des insuffisances qui ont entraîné la nomination de comptables principaux des matières qui ont pour rôles de traiter et finaliser les informations venant du comptable secondaire des matières.

Pour mieux appréhender la gestion des matières, nous allons aborder les points suivants :

2.1 La commission de réception

Elle est chargée d'assister aux opérations de réception des matières dont la valeur est supérieure à 300 000 F CFA. Elle est nommée pour une période d'un an.

Pour l'UFR des sciences et technologies, sa commission de réception est mise en place par le directeur de l'UFR. Elle comprend au moins trois membres dont l'un est le comptable des matières de l'UFR.

2.2 Le processus de gestion du matériel

C'est tout un long processus qui se déroule pendant la gestion du matériel qui commence de sa réception jusqu'à sa sortie définitive dans l'université.

Dans ce processus, nous notons les entrées (réceptions), les affectations et les sorties.

2.2.1 La réception ou l'entrée

La réception de biens à l'UASZ peut se faire de différentes manières selon la nature de l'entrée des biens, qui peut être une commande, un don ou un prêt.

❖ Par commande

L'agent de l'Université fait part au comptable des matières de ses besoins en biens matériels. En cas de validation de la demande, le comptable des matières vérifie si le matériel n'est pas dans le stock. Il fait alors une commande.

Lors de la livraison, il est chargé de réceptionner les matériels de la commande. Cependant, deux (2) éventualités peuvent se présenter suivant le montant total de la facture :

- si le montant est inférieur ou égal à 300 000 F CFA, il établit seulement un bon d'entrée ;
- par contre si le montant est supérieur à 300 000 F CFA, la réception se fera avec la présence d'une commission de réception, qui va vérifier la validité de la commande. Un procès-verbal de réception est obligatoirement dressé.

❖ Par Don

L'UASZ peut aussi bénéficier de biens venant d'autres entités qui peuvent être externes (entreprise, association ...) ou internes (entre différentes structures).

❖ Par Prêt

En interne, les différentes structures peuvent se prêter des biens en cas d'incapacité d'effectuer une commande ou jugeant trop lent le processus d'approvisionnement en de nouveaux matériels.

2.2.2 Attribution ou affectation

Tout bien matériel a une finalité qui est d'être utilisé. Cependant, pour qu'il soit utilisé, il doit être attribué à un bénéficiaire (personne, service, direction....)

2.2.3 Les sorties

Les biens matériels à l'UASZ peuvent effectuer des mouvements de sorties. La sortie peut être provisoire ou définitive.

❖ Sortie provisoire

Une sortie de bien matériel est provisoire si le bien est défectueux et est transféré pour être réparé. C'est à l'image des produits du 1^{er} groupe.

❖ Sortie définitive

Dans ce cas, le produit ne fait plus partie du patrimoine de l'Université. Soit il n'est plus utilisable, soit l'UASZ le donne en don à une entité externe (école par exemple).

2.3 L'existant pour la gestion des biens de l'UASZ

À l'UASZ, même si on dispose déjà des applications pour la gestion des biens, certaines tâches se font manuellement.

Concernant les applications, à l'UASZ les comptables des matières utilisent deux (2) logiciels différents :

L'un est un logiciel développé par un étudiant qui a fait tout son cursus universitaire à l'UASZ. Ce logiciel est actuellement utilisé à la direction de la gestion du patrimoine et de la maintenance. L'autre logiciel présente beaucoup plus de fonctionnalités dont certains ne sont d'aucune utilité au comptable des matières. Celui-ci est utilisé par le responsable des services généraux de l'UFR ST, comptable des matières de l'UFR des Sciences et Technologies.

2.4 Les problèmes

La gestion des matières est un travail fastidieux du fait de certains problèmes que rencontre le comptable des matières parmi lesquels :

- le référencement des matériels : à l'UASZ, nous notons une absence de référencements des matières. Les biens ne possèdent pas une identification unique qui représente une instance ;
- L'inexistence de catalogue des biens : nous avons constaté aussi qu'une matière donnée n'a pas de catalogue permettant ainsi d'avoir un aperçu de l'élément avec toutes ces caractéristiques ;
- la sécurité : une matière déjà attribuée peut se retrouver à un endroit où il n'est pas censé être comme le fait qu'un fauteuil soit déplacé dans une autre salle ou bureau ;
- le non-respect de l'article 8 qui stipule que les cessions gratuites de matières sont interdites ;
- la traçabilité des matières : pour un élément donné, nous ne gardons pas ses mouvements à travers les différentes structures ainsi que ses différents possesseurs au fil du temps ;
- le plus alarmant c'est qu'aujourd'hui l'Université ne cesse de croître surtout avec l'avènement du CDP (Contrat de Performance) qui a fait qu'aujourd'hui l'UASZ soit un grand chantier: de nouveaux amphithéâtres, un nouveau restaurant, des salles pédagogiques ... sont en construction. Ce qui implique une augmentation des biens matériels pour répondre à ces besoins. À ce rythme, les comptables des matières seront débordés de travail.

3. Problématique

3.1 Objectif du sujet

Notre stage a pour but d'automatiser la gestion des biens matériels de l'UASZ en mettant en place une application qui va permettre de :

- Faciliter la saisie par :
 - la mise en place d'un système de référencements des matières en générant des codes-barres ;
 - l'impression des codes-barres grâce à une mini imprimante ;
 - l'utilisation de lecteur pour lire les codes-barres générés ;
- gérer l'entrée ou la réception de biens ;
- gérer l'attribution ou affectation de biens aux personnels et les sorties de biens de l'UASZ ;
- localiser une matière au sein de l'UASZ ;
- déclarer un problème sur un bien matériel ;
- faire un suivi et une maintenance des biens matériels ;
- effectuer des études statiques sur les biens matériels.

3.2 Scénario d'utilisation de notre solution

Pour gérer les biens matériels à travers notre application, nous allons procéder comme suit :

a) Entrée

À la réception des biens matériels dans le cas d'un bon d'entrée, le comptable des matières va enregistrer ce bon à travers l'application et rédiger un PV de réception dans le cas d'un bon dont le montant total de la facture est supérieur à 300 000 F CFA.

Après cette étape, le système va générer les codes-barres associés à chaque matière du bon enregistré pour les matières du premier groupe. C'est là que va intervenir la mini imprimante. En effet, le comptable des matières va via l'application imprimer le code-barre relatif à chaque bien matériel. Ainsi ces petites impressions vont constituer les étiquettes qui seront collées sur les biens correspondants. Maintenant le comptable des matières peut stocker les biens dans les magasins ou dans les entrepôts.

b) Gestion des biens

C'est à cette étape qu'intervient le lecteur de codes-barres. En outre, dans le cas d'une attribution, le comptable des matières va prendre la matière puis scanner son

Sujet : Automatisation de la gestion des biens matériels de l'UFR des Sciences et Technologies

code-barre grâce au lecteur afin d'effectuer l'opération qui peut être une attribution, ou une réparation.

c) Sortie

Pour effectuer un bon de sortie, le comptable des matières scanne les codes-barres des matières puis effectue le bon de sortie qui peut être provisoire ou définitive.

Dans cette partie, nous avons fait le tour sur la gestion des biens matériels à l'UASZ. Cela nous a permis de mieux comprendre le fonctionnement ainsi que les problèmes rencontrés pour une gestion efficace des matières de l'Université. Dans la suite, nous allons parler de la méthodologie de gestion de projet utilisé.

Chapitre II: Méthodologie

Compte tenu de tous les problèmes listés ainsi que les objectifs fixés, notre sujet sera traité comme un projet informatique. Il devient alors nécessaire d'adopter une méthodologie de gestion de projet informatique pour respecter certains paramètres essentiels comme la durée, la qualité...

Nous avons étudié deux types de méthodologies : les méthodes classiques et les méthodes agiles.

I. Les méthodes classiques

Ce sont des méthodes de gestion calquées sur la gestion de projet dans l'industrie, utilisées dans les années 50-60. Ces méthodes avaient la particularité de s'effectuer d'une manière séquentielle ou il faut valider l'étape précédente pour passer à la suivante. Dans ce cas, le chef de projet doit alors faire un planning précis de réalisation du projet en prévoyant des jalons de début et de fin de phases ainsi que les tâches à effectuer. Les risques sont détectés tardivement puisqu'il faut attendre la fin du développement pour faire un test. Plus le projet avance, plus l'impact des risques augmente ; ainsi il sera toujours plus difficile et coûteux de revenir en arrière lorsque nous découvrons une anomalie tardivement.

II. Les méthodes agiles

Les méthodes agiles sont de nouveaux modèles plus légers en termes de gestion, car ils prônent moins de documentation et moins de contrôle sur le procédé. Ces modèles s'adressent à des projets de petite ou moyenne taille avec une équipe réduite. Ces modèles permettent de s'ajuster rapidement aux changements des spécifications tout en garantissant des livraisons fréquentes. Ils ont vu le jour en 2001 à la suite d'une rencontre de dix-sept experts en développement logiciel dans le but de mettre en place une solution suite à un taux d'échec des projets observé dans les années 90.

Cette rencontre a donné naissance à un manifeste définissant quatre principes :

- individus et itérations au lieu de processus et outils ;
- logiciel fonctionnel au lieu de documentation massive ;
- collaboration du client au lieu de négociation de contrats ;
- réagir au changement au lieu de suivre le plan.

Ces méthodes utilisent un principe de développement itératif qui consiste à découper le projet en plusieurs étapes appelé *itérations*.

Vu la taille de notre projet, les méthodes agiles sont les mieux adaptées. Cependant il existe plusieurs méthodes agiles :

- Scrum : signifiant mêlé au rugby, est aujourd'hui la méthode agile la plus populaire. Elle est caractérisée par une itération appelée sprint assez court soit 1 mois au maximum ;
- EXtreme Programming (XP) : son objectif est de réduire les coûts du changement. Elle met l'accent sur la revue de code, les tests, la conception continue, la simplicité, la traduction des besoins en métaphores ;
- Rational Unified Process (RUP) : elle peut être considérée comme la moins agile. Elle est un mélange des méthodes classiques et de méthodes agiles. Le principe est de parcourir un cycle de vie durant une itération. Son approche assez lourde et le coût d'investissement de cette méthode la réservent à des projets de grande ou moyenne taille ;
- Rapid Application Development (RAD) : c'est la méthode agile la plus ancienne et celle qui a été la première à être en rupture avec les méthodes classiques. Elle a introduit les notions d'itérations et d'incrément. Elle vise à adopter la solution la plus stratégique, la moins risquée, la plus fiable et la moins coûteuse.

Parmi ces méthodes agiles, nous avons choisi la méthode **Scrum**.

1. La méthode Scrum

Scrum est la méthode la plus répandue du fait de sa simplicité. Elle peut être combinée avec d'autres méthodes et admet de courtes itérations appelées sprints. Ces sprints ont une durée de 2 à 4 semaines.

Quant à l'équipe, elle réalise les fonctionnalités en toute liberté sans aucune influence externe. Elle décide sur la manière d'atteindre les objectifs. La seule obligation qu'elle a est de livrer une solution répondant aux besoins du client tous les 30 jours.

a) Rôle

Cette équipe est constituée d'un responsable officiel du produit (**Product owner**), d'un expert de l'application de Scrum (**Scrum Master**), d'une équipe de projet (**Scrum Team**) et d'un décideur (**users and stackholders**).

- **Product owner** : personne représentant le client et l'utilisateur du scrum master et de l'équipe de développement. Il définit la liste des fonctionnalités du produit puis les classifie par ordre de priorité.
- **Scrum Master** : la personne intermédiaire entre le responsable de l'équipe et l'équipe de projet. Son rôle est de préserver l'équipe de projet contre toute perturbation extérieure, car pouvant les empêcher d'avancer. Il veille au bon

fonctionnement du projet en prêtant attention sur le respect des normes à suivre pour chaque sprint.

- Scrum Team : elle est responsable de la réalisation opérationnelle des tâches. De plus, elle est plurifonctionnelle et se gère elle-même afin de délivrer un produit à chaque sprint.
- Users and stakeholders : personne dont le rôle est d'observer et donner des conseils à l'équipe. Ils peuvent être des utilisateurs, le client, etc.
- Fonctionnement
- User story est un scénario utilisable par l'utilisateur. Le back log contient toutes les user story que l'équipe aimerait pouvoir faire.
- Le Product Backlog : également appelé carnet de produit. Il est exprimé sous forme de besoins. C'est l'ensemble des fonctionnalités qui constituent le produit à réaliser. Ces fonctionnalités sont répertoriées sur une liste et rangées par ordre de priorité.
- Sprint Backlog : il consiste à décomposer le Product Backlog en un sous ensemble. En d'autres termes, c'est une liste définissant le travail d'une équipe durant un sprint.
- Sprint Planning Meeting : c'est une réunion durant laquelle le Product owner explique les éléments du Product Backlog les plus prioritaires et fixe les objectifs du prochain sprint. Il dure environs 4 heures
- pour un sprint de 2 à 3 semaines Daily Scrum meeting : réunion de courte durée (environ 15 minutes) tenue tous les jours et pendant laquelle chaque membre fait le point sur ce qui a été fait puis sur ce qui sera fait aujourd'hui et enfin signale les obstacles afin de les pallier.
- ***Sprint Review Meeting*** : il a une durée de 2 heures pour un sprint de 2 semaines. Il a lieu à la fin de chaque sprint durant laquelle l'équipe de développement montre les tâches accomplies.

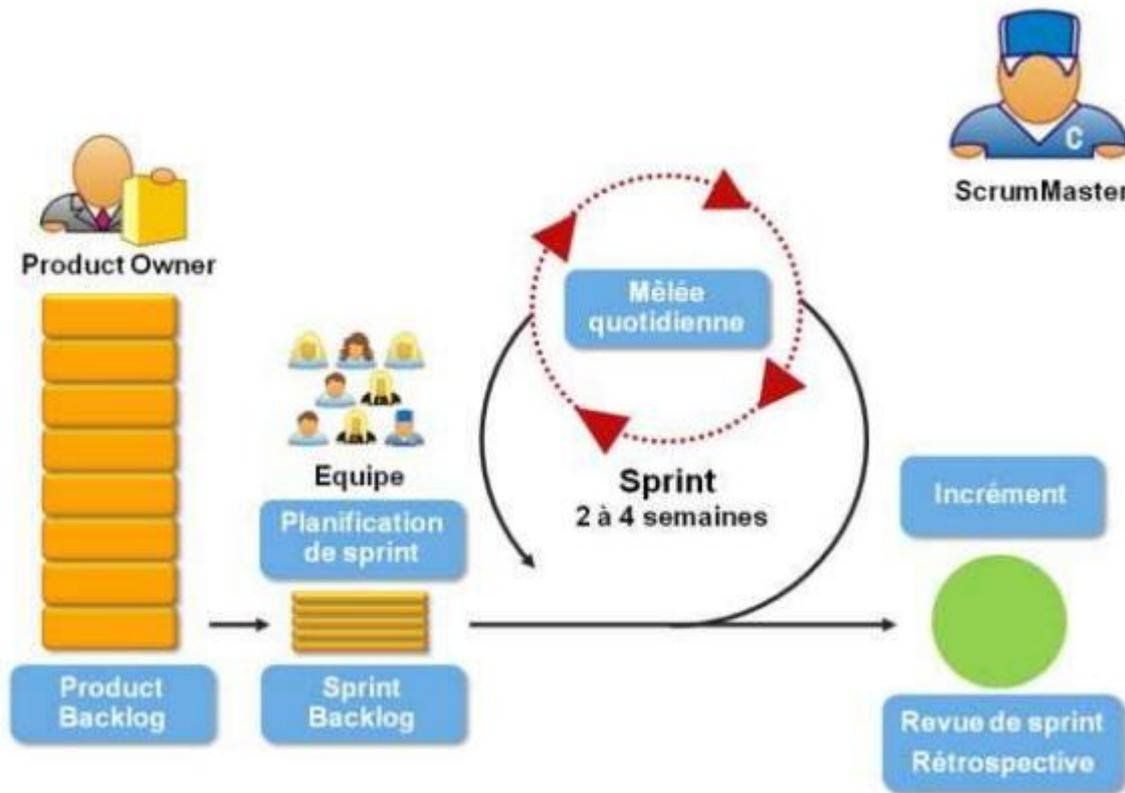


Figure 2 : Cycle de Scrum [1], [8], [9] et [10]

b) Application du Scrum à notre projet

Dans notre projet, l'application des rôles de la méthode scrum nous donne l'équipe de scrum suivant :

Product Owner (Responsable officiel du produit) : M Ibrahima SAKHO ;

Scrum Master (expert de l'application de Scrum) : Dr Khadim DRAME et Dr Ibrahima DIOP ;

Scrum Team (équipe de développement) : El hadji Mamadou Korka DIALLO.

Conclusion

Dans cette partie, nous avons parlé de la méthodologie de gestion de projet ou nous avons listé les différentes méthodes qui existent puis expliqué notre choix sur une méthodologie. Dans la suite, nous allons faire la spécification et l'analyse des besoins fonctionnels.

Chapitre III: Spécification et analyse des besoins fonctionnels

Dans ce chapitre, nous allons en premier lieu faire la spécification des besoins fonctionnels. Après cela, nous allons aborder la partie analyse des fonctionnalités de notre système.

I. Spécification des besoins fonctionnels

Dans cette partie, nous allons identifier les différents acteurs du système et les différentes fonctionnalités que doit avoir notre application.

1. Identification des acteurs

Un acteur, c'est une entité externe qui interagit avec le système. Il peut être un utilisateur physique, un périphérique externe ou un système externe. Dans le cas de notre sujet, nous avons identifié deux acteurs, répertoriés dans le tableau suivant.

Tableau 1 : Les acteurs

Acteurs	Description
Comptable des matières	peut effectuer tous les tâches en relation avec la gestion des biens matériels en relation avec le système
Administrateur	maintenir le système

2. Identification des fonctionnalités

Par fonctionnalité, on peut entendre une action réalisée par le système en réponse à une action d'un acteur. Pour la gestion des biens matériels, le backlog du produit (liste de fonctionnalités) a été identifié et illustré dans le tableau suivant.

Tableau 2: Le Backlog du produit

Fonctionnalité	Acteur
S'authentifier	comptable des matières
Gérer un compte utilisateur	comptable des matières
Gérer un groupe (ajouter, modifier, supprimer)	comptable des matières
Gérer une catégorie (ajouter, modifier, supprimer)	comptable des matières
Gérer un type (ajouter, modifier, supprimer)	comptable des matières
Gérer un catalogue (ajouter, modifier, supprimer)	comptable des matières

Gérer un fournisseur (ajouter, modifier, supprimer)	comptable des matières
Gérer une commission de réception (ajouter, modifier, supprimer)	comptable des matières
Gérer un bon d'entrée (ajouter, modifier, supprimer)	comptable des matières
Gérer les matières d'un bon d'entrée	comptable des matières
Génération de code-barre	comptable des matières
Impression de code-barres	comptable des matières
Gérer un bon de sortie (ajouter, modifier, supprimer)	comptable des matières
Gérer les matières d'un bon de Sortie	comptable des matières
Attrition de matière	comptable des matières
Ajouter un bénéficiaire (ajouter, modifier, supprimer)	comptable des matières
Déclarer un problème	comptable des matières
Localiser un matériel	comptable des matières
Effectuer un suivi et une maintenance d'une matière	comptable des matières
Faire des études statistiques et des tendances	comptable des matières

3. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système.

Il donne une vue du système dans son environnement extérieur. Il définit la relation qui existe entre l'utilisateur et les éléments que le système met en œuvre.

Les éléments qui composent un diagramme de cas d'utilisation sont :

- un acteur qui est l'archétype de l'utilisateur qui interagit avec le système ;
- un cas d'utilisation modélisant le service rendu par le système sans imposer le mode de réalisation.

Dans la suite, nous allons donner quelques cas pratiques de diagrammes de cas d'utilisation de notre projet.

3.1 Diagramme de cas d'utilisation pour la gestion des informations

Le comptable de matières gère les informations relatives aux produits en ajoutant, modifiant ou supprimant des informations qui peuvent être un groupe, une catégorie, un type ou un catalogue.

La figure suivante en est une illustration.

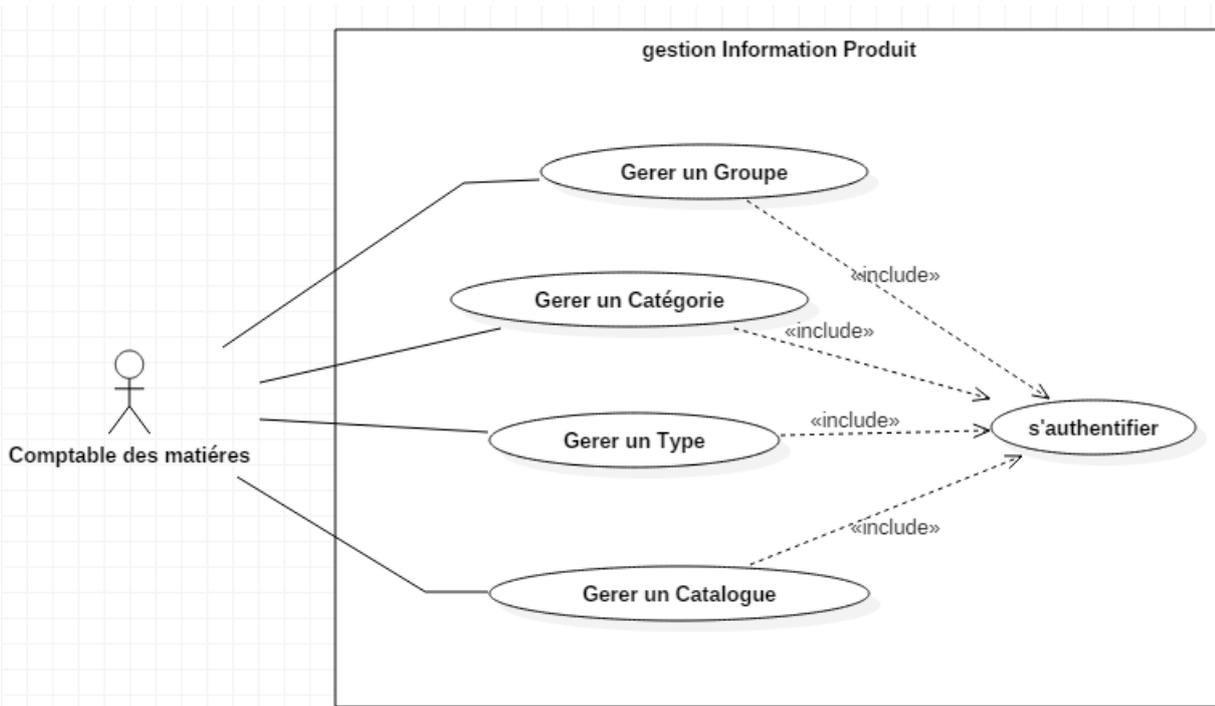


Figure 3: Diagramme de cas d'utilisation pour la gestion des informations

3.2 Diagramme de cas d'utilisation pour la gestion des entrées

Dans ce cas d'utilisation, le comptable des matières a déjà toutes les informations nécessaires pour gérer les bons. Alors à partir de son interface, il peut réceptionner des produits en ajoutant des bons d'entrée et en rédigeant le procès-verbal de réception dans le cas où le montant de la facture est supérieur à 300 000 F CFA.

La figure suivante le montre de façon explicite.

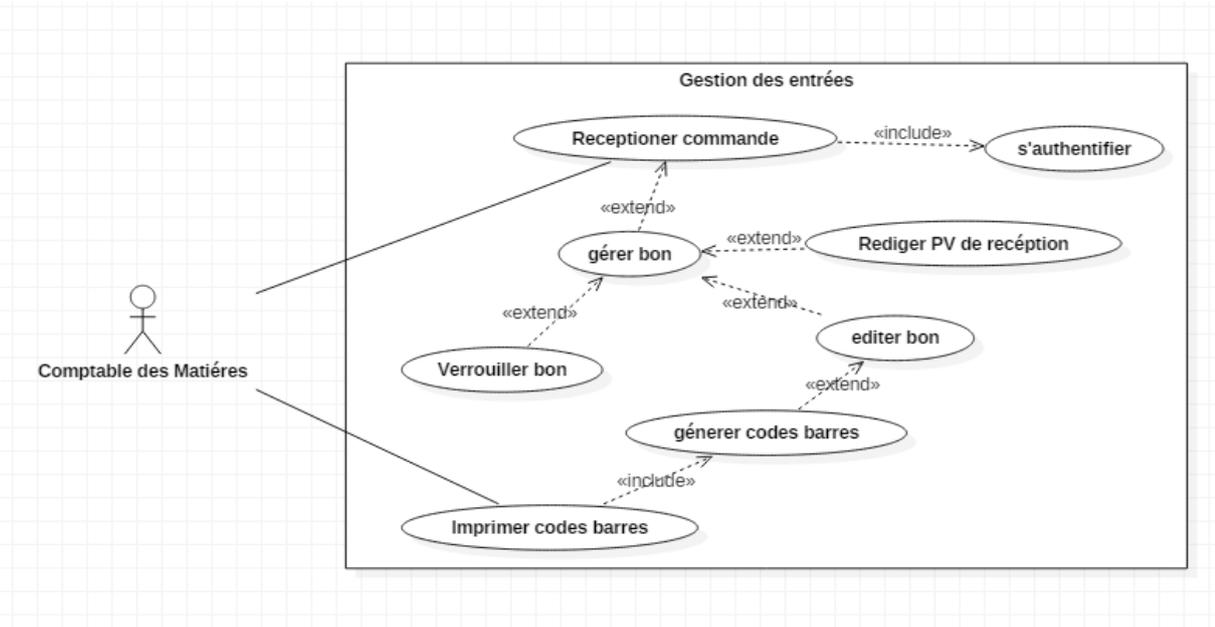


Figure 4: Diagramme de cas d'utilisation pour la gestion des informations

II. Analyse des besoins fonctionnels

Dans cette partie, nous allons faire l'analyse de quelques fonctionnalités du système en faisant une description détaillée et un diagramme de séquences pour montrer les différents scénarios. Pour le reste des descriptions, elles seront mises dans l'annexe de ce présent document. Pour ce faire, nous allons d'abord analyser l'authentification, puis la gestion des informations et la gestion des bons d'entrée.

1. Analyse de l'authentification

Pour mieux analyser ce cas, nous allons d'abord faire sa description, ensuite donner son diagramme d'activité pour enfin donner son diagramme de séquences.

1.1 Description du cas d'utilisation « authentification »

Le tableau ci-dessous donne une description du cas d'utilisation de l'authentification.

Nom	Authentifier
Résumé	C'est une fonctionnalité indispensable qui permet d'accéder au système
Acteurs	Comptable des matières
Précondition	Accéder à l'interface de connexion
Description	<ul style="list-style-type: none">• le comptable des matières demande à s'authentifier ;• le système fournit le formulaire de connexion ;• le comptable des matières saisit son login et son mot de passe puis valide ;• le système vérifie les informations ;• le comptable des matières accède à la page d'accueil.
Exception	Connexion avec des paramètres pas conformes.
Post-condition	Le comptable des matières est connecté au système

Tableau 3 Description du cas Authentifier

1.2 Activités de l'authentification

Pour s'authentifier, l'utilisateur doit saisir son login et son mot de passe. Ensuite, le système effectue une vérification. Si les valeurs saisies ne sont pas correctes, le système affiche un message d'erreur et nous sommes renvoyés à l'étape de saisie des paramètres de connexion. Par contre, si les paramètres sont corrects alors nous accédons à la page d'accueil.

La figure suivante illustre les propos évoqués ci-dessus.

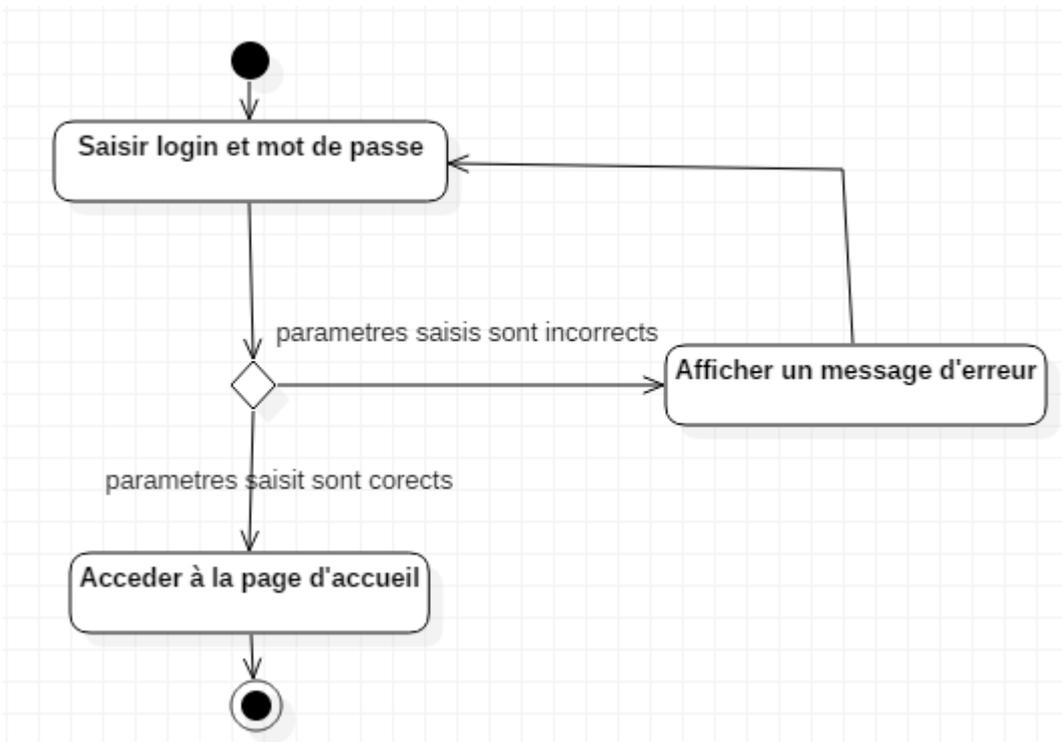


Figure 5: Diagramme d'activité de l'authentification

1.3 Le scénario normal de l'authentification

Dans ce diagramme, nous allons décrire l'authentification en nous basant sur un scénario. Alors sur la figure suivante qui le montre de façons explicites.

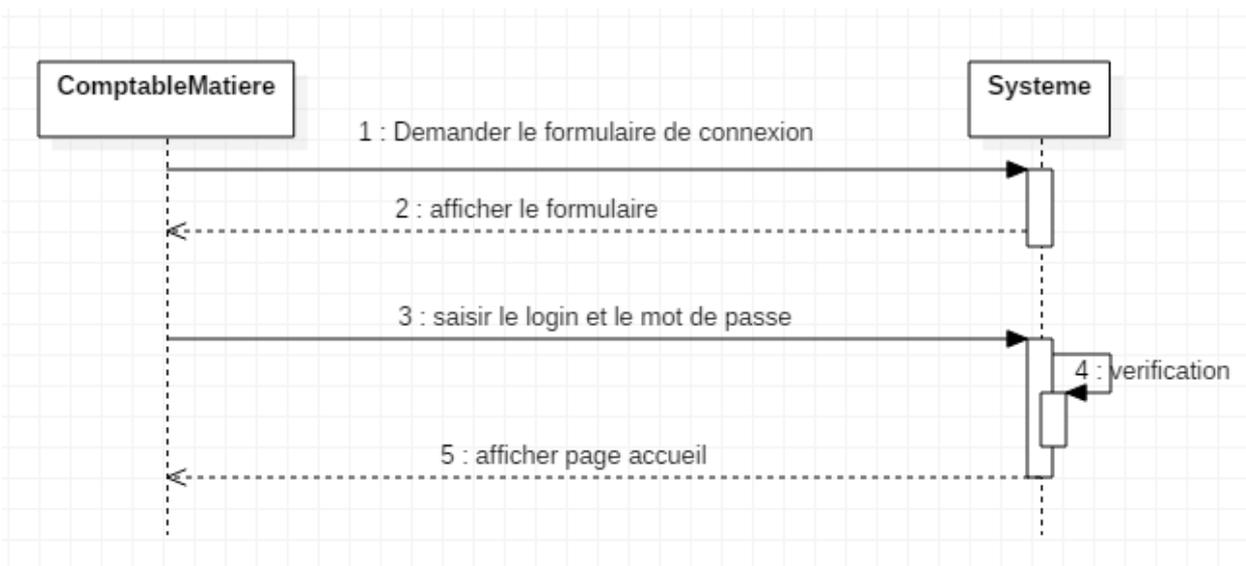


Figure 6: Diagramme de séquence pour l'authentification

2. Analyse du cas « ajouter groupe » du package « gestion des informations »

Dans le but d'analyser ce cas, d'utilisation nous allons d'abord faire sa description, ensuite donner son diagramme d'activités pour enfin donner son diagramme de séquence du cas d'utilisation.

2.1 Description du cas d'utilisation « ajouter groupe »

Le tableau suivant montre la description du cas « ajouter groupe ».

Tableau 4 : Description du cas ajouter Groupe

Nom	Ajouter un groupe
Résumé	Ce cas d'utilisation permet au comptable des matières d'ajouter un nouveau groupe
Acteurs	Comptable des matières
Précondition	Authentification
Description	<ul style="list-style-type: none">• le comptable des matières clique sur le bouton ajouter ;• le Système affiche un pop-up contenant le formulaire d'ajout de groupe ;• le Comptable des matières remplit les champs (code et nom du groupe) ;• il clique sur le bouton Ajouter ;• le Système vérifie les données saisies ;• le Système enregistre les données.
Exception	Le Comptable des matières saisit des informations incorrectes
Post-condition	Un nouveau groupe est ajouté

2.2 Les activités du cas « ajouter groupe »

Pour ajouter un groupe, l'utilisateur clique sur le bouton « Ajouter », le formulaire s'affiche puis il remplit les différents champs. Une fois saisie terminée, il essaye d'enregistrer. Pendant ce temps, le système effectue une vérification des données. Dans le cas où les données sont valides, l'enregistrement est effectué avec succès. Sinon il sera demandé de remplir à nouveau les champs. La figure suivante en est une illustration.

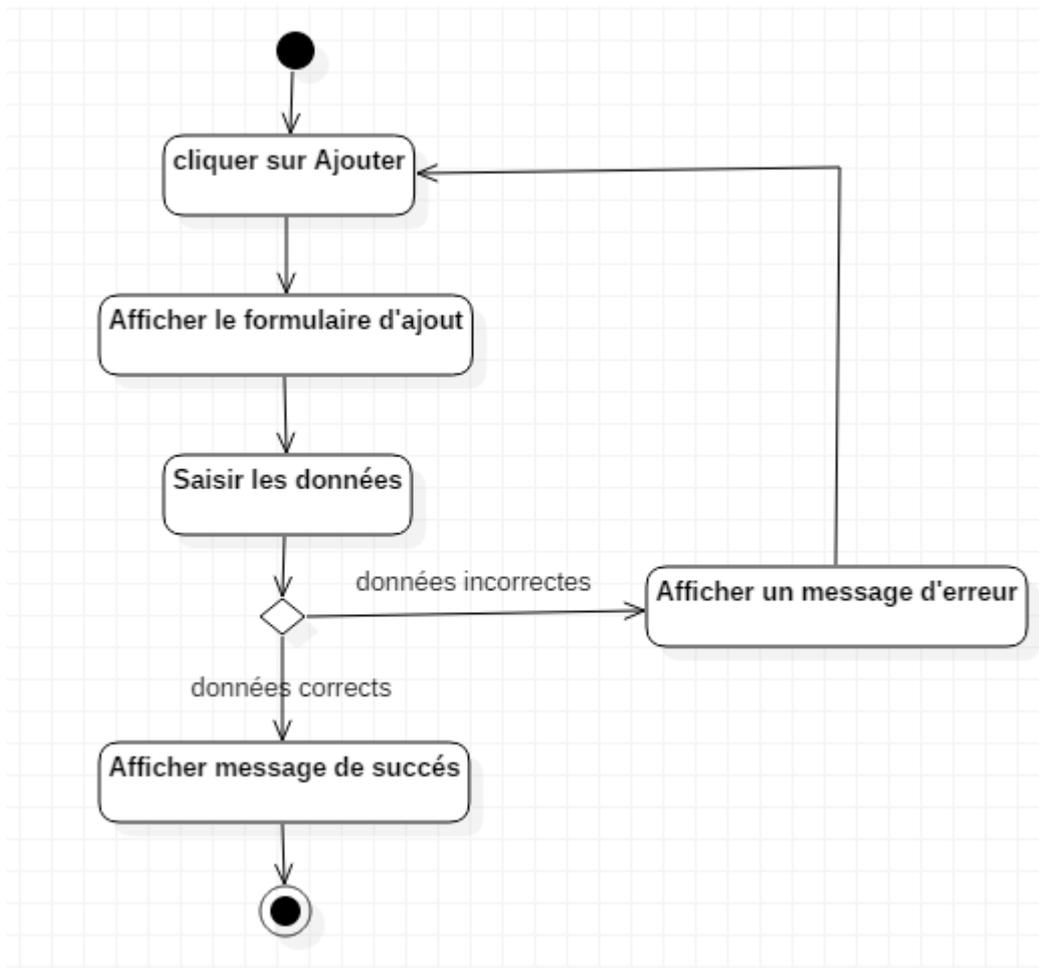


Figure 7: Diagramme d'activités du cas ajouter groupe

2.3 Le scénario normal du cas « ajouter groupe »

Le scénario que nous voulons expliquer ici est identique à ce qui est dans la section précédente (Diagramme d'activité). Nous allons montrer ce scénario à travers le diagramme de séquences ci-dessous.

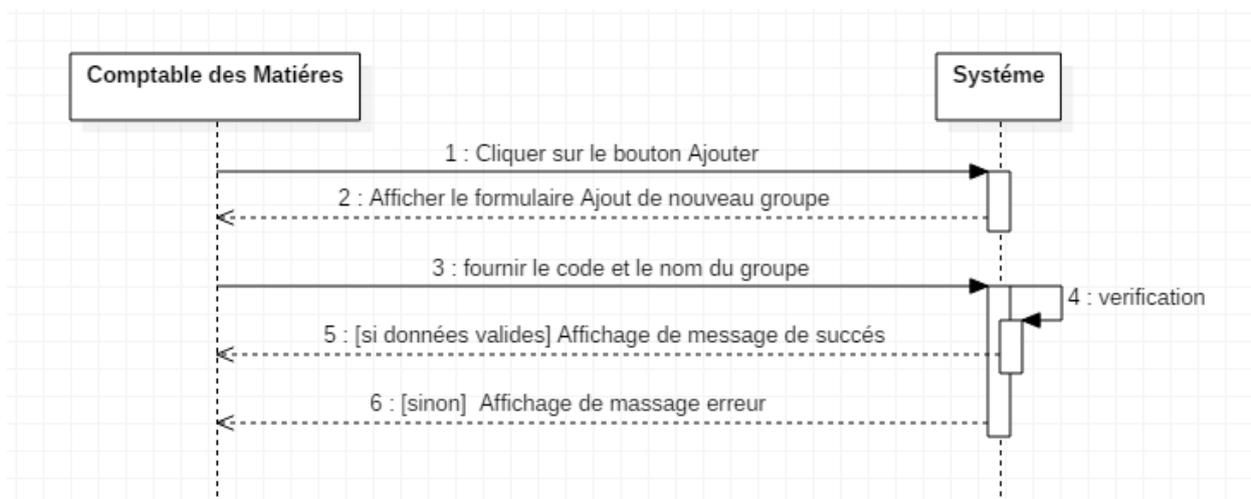


Figure 8 : Diagramme de Séquence de cas ajouter un Groupe

3. Analyse du cas « gérer entrée »

3.1 Description du cas d'utilisation « gérer entrée »

Le tableau suivant montre la description du cas géré entrée.

Nom	Gérer Entrée
Résumé	Ce cas d'utilisation permet au comptable des matières de gérer les entrées
Acteurs	Comptable des matières
Précondition	Authentification, gérer information, gérer commission
Description	<ul style="list-style-type: none">• le Comptable des matières clique sur le bouton ajouter ;• le Système affiche un pop-up contenant le formulaire d'ajout ;• il choisit le type de bon puis valide ; dans le cas d'une commande, il choisit parmi les options suivantes : montant supérieur ou inférieur à 300 000 F CFA ;• il sélectionne le nouveau bon pour l'édition ;• le Système enregistre les données.
Exception	Le comptable des matières saisit des informations incorrectes
Post-condition	Un nouveau bon est enregistré

3.2 Les activités de « gérer entrée »

La gestion des entrées peut être effectuée comme suit en supposant que les informations existent déjà: le comptable des matières ajoute un nouveau bon tout en précisant le type du bon (prêt, don, commande), ajoute les produits relatifs à chaque bon, de là il peut générer les codes-barres, les imprimer et verrouiller le bon.

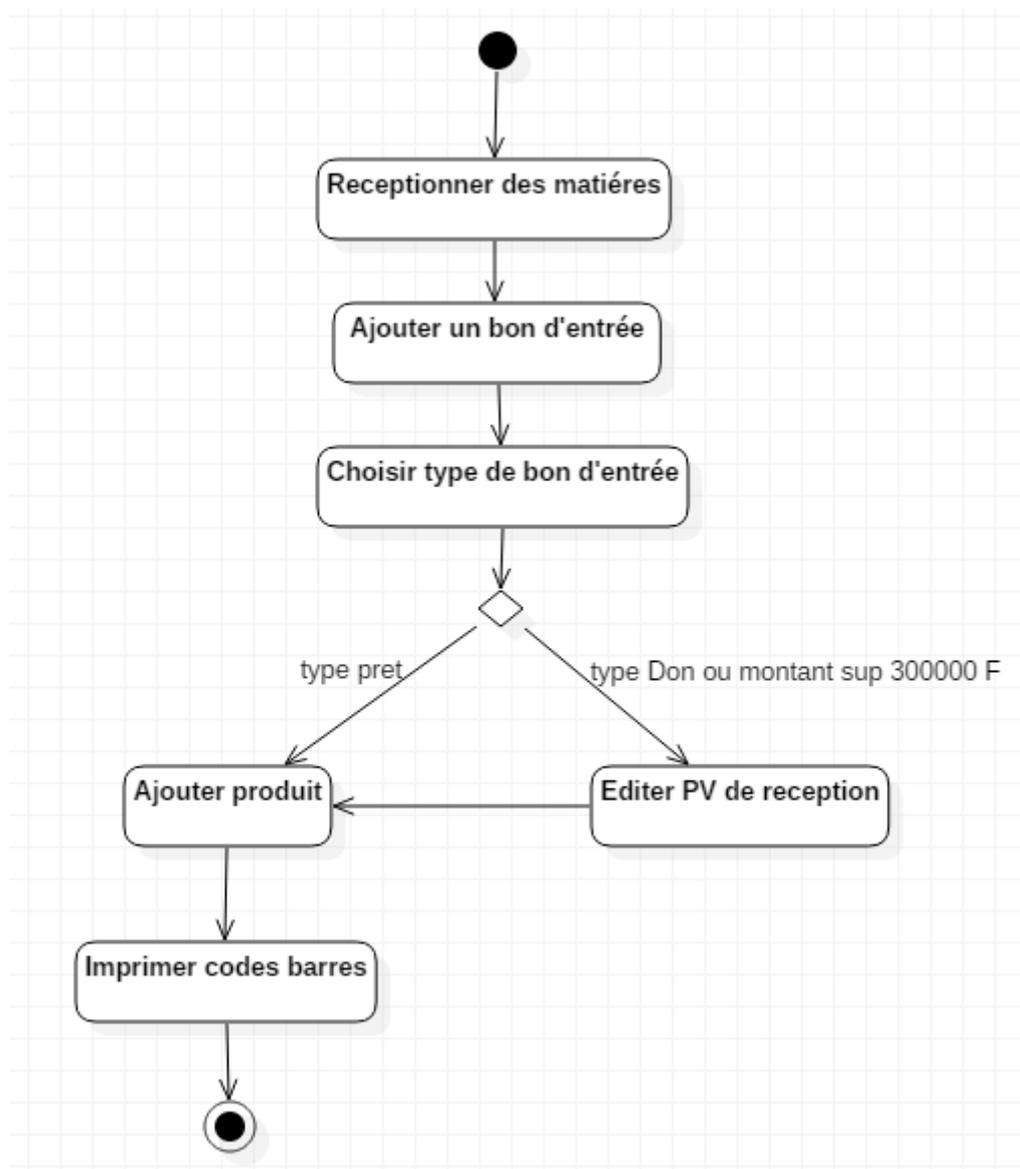


Figure 9 : Activité de gérer les entrées

Dans cette partie nous avons parlé de la spécification des besoins fonctionnels en montrant les différents acteurs et les fonctionnalités de notre système, pour ensuite faire l'analyse de quelques fonctionnalités. Dans le prochain chapitre, nous allons faire la conception de notre système.

Chapitre IV : Conception du système

Dans le chapitre précédant, nous avons exhibé le backlog de notre projet. Ainsi dans ce chapitre, nous allons parler de la conception du de notre application en nous attaquant d'abord à l'architecture du système, ensuite faire la conception générique pour enfin terminer avec la conception détaillée.

I. Architecture du système de notre application

L'architecture logicielle décrit, d'une manière symbolique et schématique, les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions. Son rôle est contrairement à la spécification fonctionnelle qui décrit ce que doit réaliser un système informatique, mais plutôt comment il doit être conçu de manière à répondre aux spécifications c'est-à-dire le « comment le faire ».

Pour mieux comprendre notre architecture, nous allons donner les différents éléments qui constituent le système.

Notre système est constitué des éléments suivants :

1. un système de gestion de code barre ;
2. le client web qui va interagir avec le système ;
3. le serveur d'application ;
4. la base de données où nous allons stocker toutes les données ;
5. l'imprimante thermique qui va permettre d'imprimer les codes-barres ;
6. la douchette qui est un lecteur de code-barre chargé de scanner et de lire les codes-barres imprimés à travers l'imprimante thermique.

Dans la logique de notre démarche, nous allons répartir nos différents éléments à savoir la base de données, le serveur d'application et le client

D'après les éléments listés ci-dessus, nous avons pu proposer le schéma suivant qui montre l'architecture du système selon deux approches.

1. Architecture générale

En ce qui concerne notre projet, notre architecture est composée de deux architectures particulières à savoir l'architecture 3-tier et l'architecture MVC.

1.1 L'architecture 3-tiers

L'architecture trois tiers, aussi appelée architecture à trois niveaux ou architecture à trois couches, est l'application du modèle plus général qui est le multi-tiers. L'architecture logique du système est en trois niveaux :

- la couche de présentations des données ou la couche client : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- la couche de traitement métier (serveur d'application) : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- l'accès aux données persistantes (serveur de base de données): correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Voici une figure qui le montre :

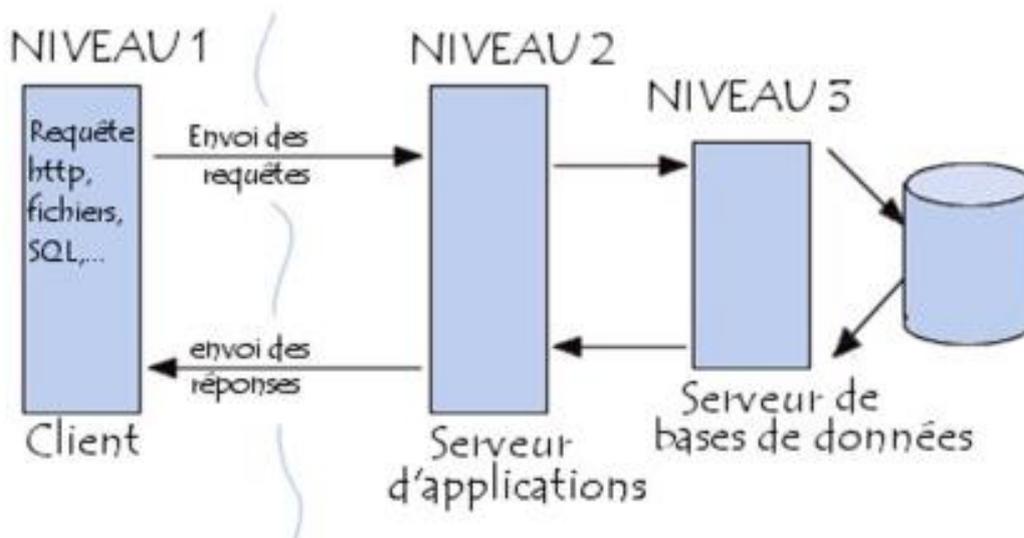


Figure 10: Architecture trois tiers

Notre choix sur cette architecture est dû au fait que nous voulons répartir nos 3 composants afin de garantir la disponibilité des services.

1.2 L'architecture MVC

C'est l'un des plus célèbres *design patterns* qui signifie **Model Vue Contrôleur**. Il permet de bien organiser son code source, mais aussi de savoir quels fichiers créer et de définir leur rôle. Son but est de séparer la logique du code source en trois parties que sont :

- le **modèle** : il gère les données de notre application. Son rôle est d'aller récupérer les informations brutes dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur ;
- la **vue** : c'est tout ce qui concerne l'interaction avec l'utilisateur ainsi que la mise en forme et l'affichage ;

- le **contrôleur** : il gère la logique du code qui prend des décisions. Il est comme un intermédiaire entre le modèle et la vue. Il va demander au modèle les données, les analyses, prendre les décisions et renvoyer le texte à afficher à la vue.

La figure suivante montre l'architecture générale de notre système.

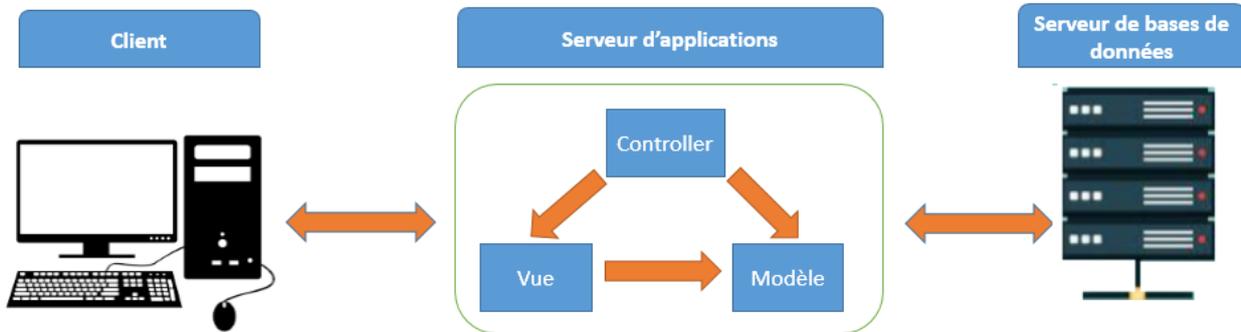


Figure 11 : Architecture générale

L'architecture MVC permet aussi d'avoir un code plus facile à maintenir et à faire évoluer. C'est ce qui explique notre choix.

2. Architecture du client

Nous allons montrer les différents équipements qui interagissent avec notre application en partant de la réception à l'attribution. Après avoir réceptionné les matières, le comptable des matières rédige un bon d'entrée. Les codes-barres automatiquement générés (étape 1 de la figure 12) vont être imprimés par le biais de l'imprimante thermique (étape 2). Ces impressions seront ensuite collées sur les matières constituant ainsi leurs identifiants (étape 3). À partir de là, le comptable des matières peut procéder à l'attribution qui constitue à prendre une matière et à l'affecter à une entité (personne, service). Pour se faire, il devra prendre la douchette et scanner le code-barres de la matière à attribuer (étape 4). L'application va recevoir l'identifiant de la matière (étape 5) et le comptable des matières pourra ainsi choisir le bénéficiaire avant de valider.

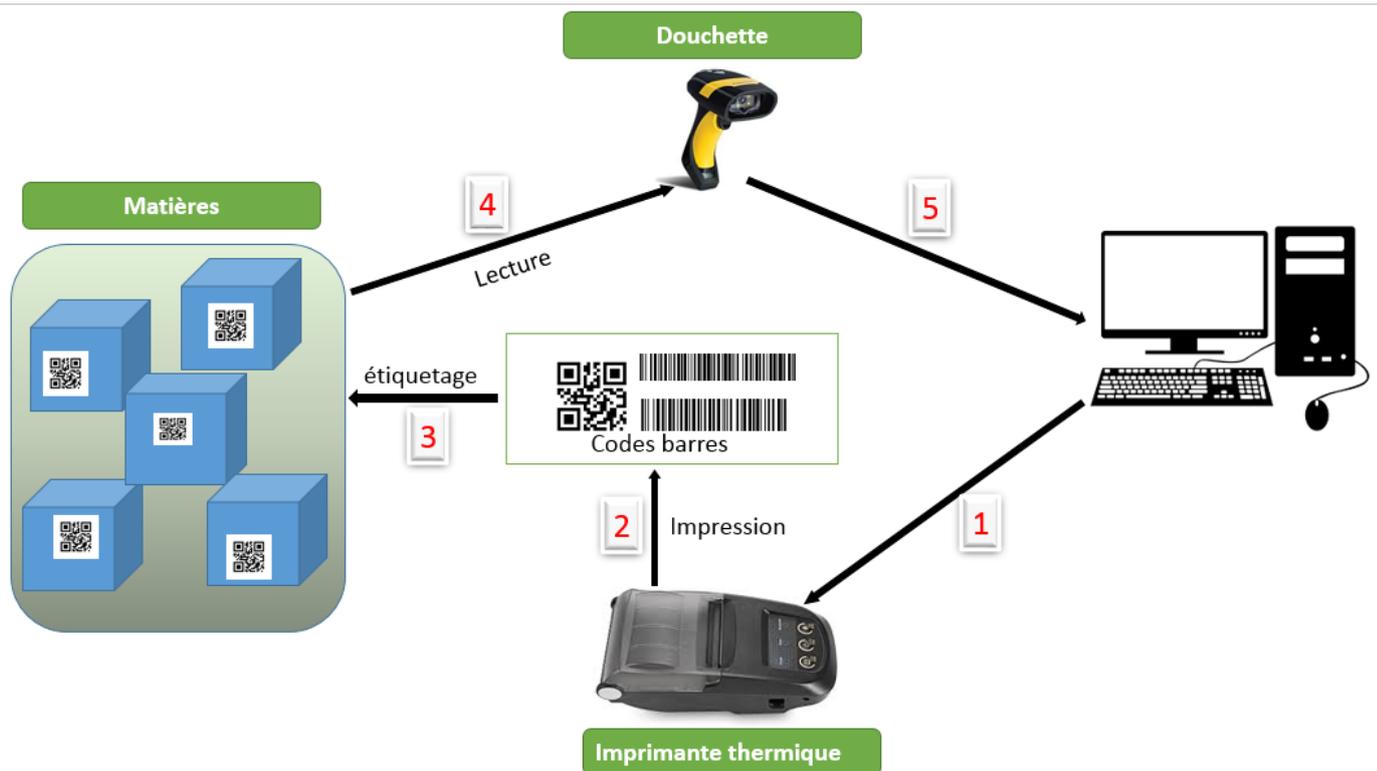


Figure 12 : Architecture générale de notre application

II. Conception générale

Nous sommes maintenant en mesure de montrer la manière dont notre futur système sera conçu. Pour ce faire nous allons d'abord aborder la conception générale en donnant le diagramme de composant, le diagramme de package et le diagramme de déploiement. Ensuite, nous allons aborder la conception détaillée où nous va donner les diagrammes de classe par fonctionnalité ainsi que le dictionnaire de données de notre base de données.

1. Diagramme de composant

Le diagramme de composant représente l'architecture des composants physique d'une application en décrivant leurs relations dans l'environnement de réalisation sous forme de modules (fichiers sources, librairie, exécutables, etc.).

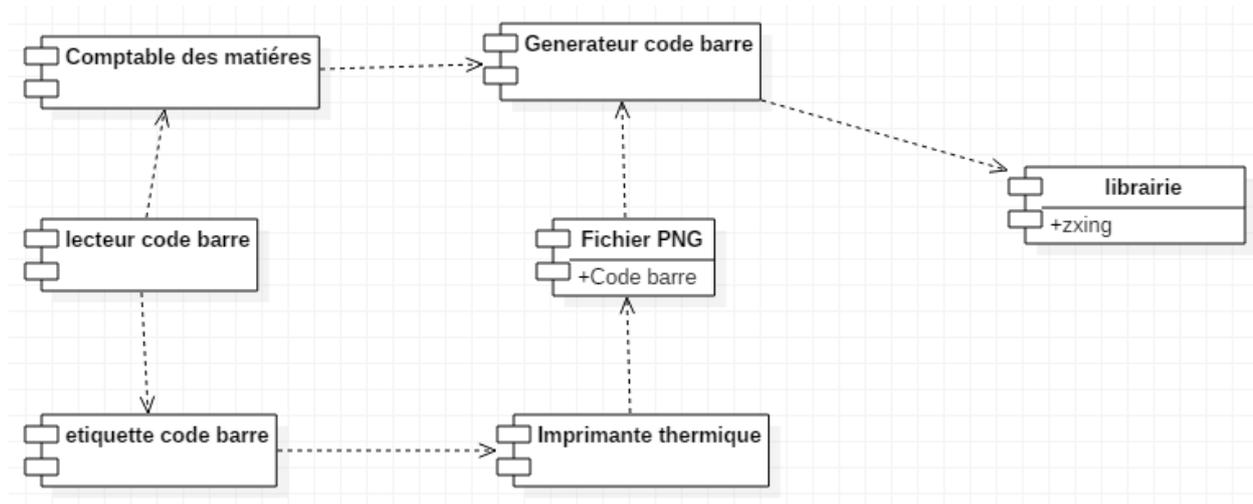


Figure 13 : Diagramme de composant

2. Diagramme de déploiement

Le diagramme de déploiement fait une représentation de l'architecture matérielle et logicielle du système. Il a les caractéristiques suivantes :

- les différents matériels sont matérialisés par des nœuds ;
- un lien de dépendance représente un lien de communication ;
- il donne la structure d'une plateforme physique ;
- il permet de préciser où se trouvent les processus ;
- il montre comment les objets se créent et se déplacent dans une architecture distribuée.

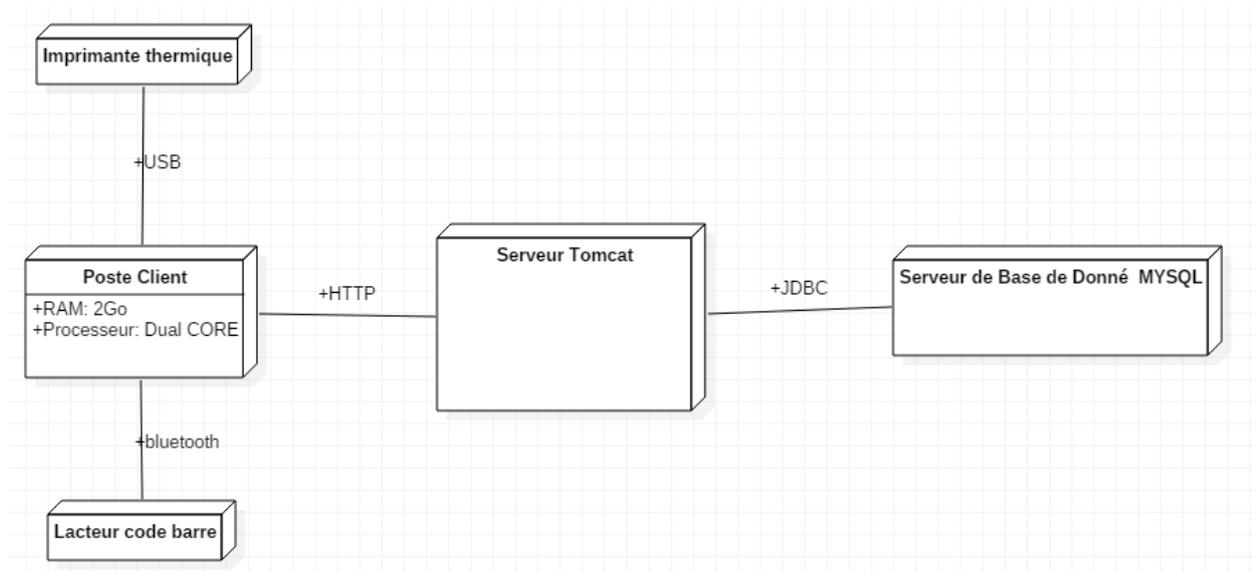


Figure 14 : Diagramme de d ploiement

3. Diagramme de paquetage

Il permet de représenter la structure hiérarchique des modules d'un projet sous forme de catégories.

Voici quelques modules de notre système :

- **Authentification** : il renferme toutes les classes nécessaires pour accéder au système suivant le profil des utilisateurs qui s'y connectent.
- **Gestion des catalogues** : c'est dans ce sous-module que nous allons gérer les groupes, les catégories, les types et les catalogues en les ajoutant, les modifiant ou en les supprimant;
- **Gestion des bons** : ce sous-module est composé de deux sous-modules qui sont gestion des bons d'entrée et gestion des bons de sortie. Ce module permet de gérer les bons, de rédiger les procès-verbaux (PV) de réception et de générer des codes-barres.

La figure suivante illustre le diagramme de paquetage de notre système.

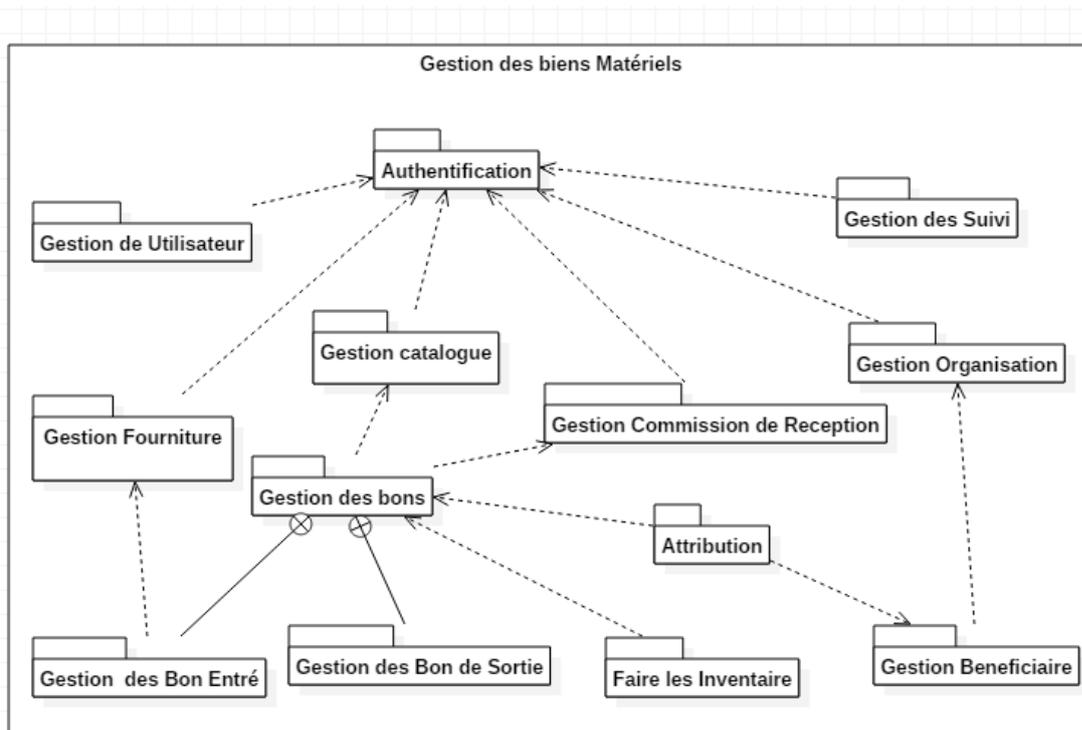


Figure 15 : Diagramme de Paquetage

III. Conception détaillé

La conception détaillée est la partie où nous allons parler des diagrammes de classe et du dictionnaire de donnée.

1. le Diagramme de Classe

Une classe est une description abstraite d'un ensemble d'objets du domaine de l'application ayant :

- des propriétés similaires appelées attributs ;
- un comportement commun appelé opérations ou méthodes ;
- des relations communes avec d'autres objets ;
- des sémantiques communes.

Dans la suite nous allons donner le diagramme de classe de quelques fonctionnalités.

a) Classe participante à la fonctionnalité d'authentification

Dans l'authentification, nous avons les classes suivantes qui y participent : la classe *User* et la classe *ComptableMatière*.

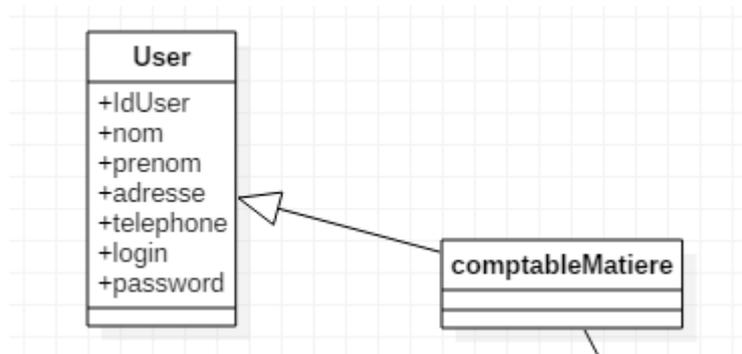


Figure 16 : Diagramme de Classes de l'authentification

b) Classe participante à la fonctionnalité Organisation

C'est dans cette fonctionnalité que nous allons gérer les différentes structures de l'UASZ ainsi que leurs sous-structures. Dans cette organisation nous avons la Classe *Structure* qui est la généralisation des classes *UFR* et *Rectorat*. Le diagramme suivant montre en détail les différentes classes.

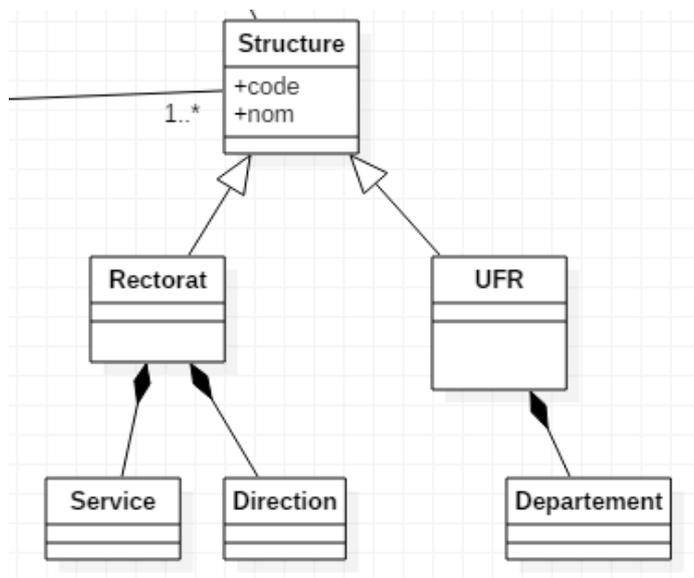


Figure 17 : Diagramme classe participant à la fonctionnalité Organisation

c) Les classes participant aux fonctionnalités gestion des Catalogues

Avant de créer le catalogue d'une matière, nous devons d'abord créer le groupe, ensuite la catégorie puis le type du produit. C'est à partir de là que nous pouvons enregistrer une matière. Voir la figure la suivante pour les classes participantes.

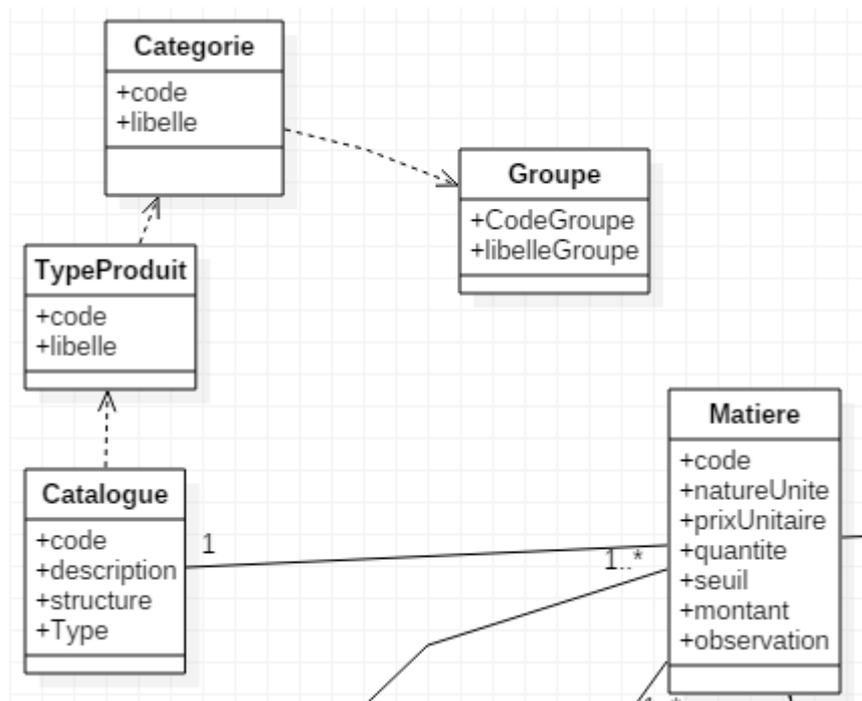


Figure 18 : Diagramme de classe participant à la fonctionnalité catalogue

d) Les classes participant aux fonctionnalités gestion des bons

Pour les bons, nous avons les classes *BonEntreExistant*, *Element*, *BonSortie*, *Fournisseur*, *PvReception*, *CommissionReception* et la classe Produit également appelée *Matiere*.

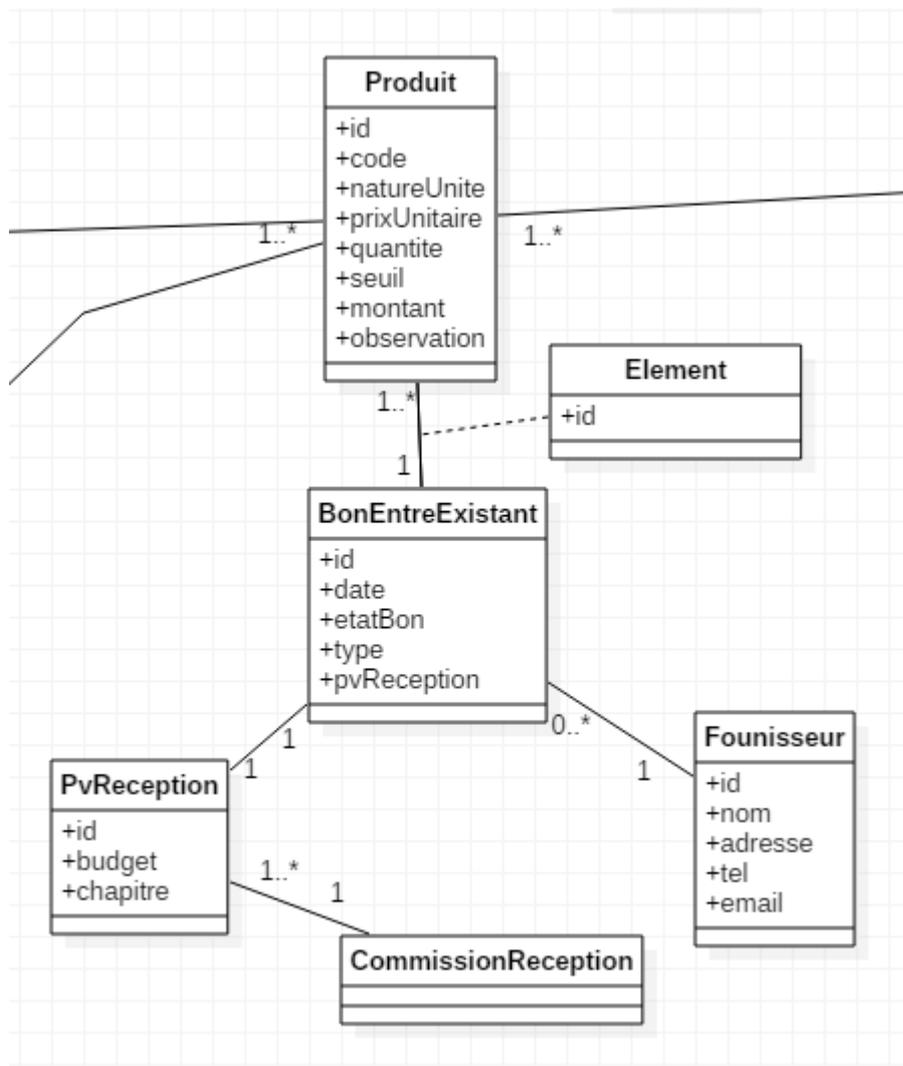


Figure 19 : Diagramme classe participante à la gestion des bons

2. Le dictionnaire des données

Le tableau suivant est le dictionnaire des données de notre application.

Tableau 5 : Dictionnaire de donnée

Nom de la table	Attribut	Type	Null	Défaut
Groupe	code	bigint(20)	Non	
	nom	varchar (255)	Oui	NULL
Catégorie	code	bigint(20)	Non	
	libelle	varchar (255)	Oui	NULL
Type	code	varchar (255)	Non	
	libelle	varchar (255)		NULL
Catalogue	code	varchar(255)	Non	

	description	varchar(255)	Oui	NULL
	seuil	bigint(20)	Oui	NULL
Structure	type	varchar(10)	Non	
	code	varchar(255)	Non	
	nom	varchar (255)	Oui	NULL
PV réception	id	bigint(20)	Non	
	budget	varchar (255)	Oui	NULL
	chapitre	varchar (255)	Oui	NULL
Produit	id	bigint(20)	Non	
	code	varchar (255)	Oui	NULL
	montant	double	Oui	NULL
	nature_unite	varchar (255)	Oui	NULL
	observation	varchar (255)	Oui	NULL
	prix_unitaire	double	Oui	NULL
	quantite	bigint(20)	Oui	NULL
membre	id	bigint(20)	Non	
	fonction	varchar (255)	Oui	NULL
	nom	varchar (255)	Oui	NULL
	prenom	varchar (255)	Oui	NULL
Fournisseur	id	bigint(20)	Non	
	adresse	varchar (255)	Oui	NULL
	email	varchar(255)	Oui	NULL
	nom	varchar (255)	Oui	NULL
	telephone	varchar(255)	Oui	NULL
element	id	bigint(20)	Non	
	produit_id	bigint(20)	Oui	NULL
	bon_sortie_id	bigint(20)	Oui	NULL
departement	code_dep	varchar (255)	Non	
	nom_dep	varchar (255)	Oui	NULL
	ufr_code	varchar(255)	Oui	NULL
Commissio_reception	id	bigint(20)	Non	
	date	date	Oui	NULL
Bon_sortie	id	bigint(20)	Non	
	date	varchar(255)	Oui	NULL
Bon_entree_existant	id	bigint(20)	Non	
	code	varchar(255)	Oui	NULL
	date	varchar(20)	Oui	NULL
	type	varchar(255)	Oui	NULL
	etat_bon	bit(1)	Oui	NULL
	pv_reception	bit(1)	Oui	NULL

	fournisseur_id	bigint(20)	Oui	<i>NULL</i>
User	id	bigint(20)	Non	
	login	varchar(255)	Oui	<i>NULL</i>
	password	varchar(255)	Oui	<i>NULL</i>
	nom	varchar (255)	Oui	<i>NULL</i>
	prenom	varchar (255)	Oui	<i>NULL</i>
	profil	varchar (30)	Non	
	telephone	varchar(255)	Oui	<i>NULL</i>
	email	varchar (255)	Oui	<i>NULL</i>
	adresse	varchar (255)	Oui	<i>NULL</i>

En somme, à partir du backlog du produit, nous avons pu donner une architecture de notre système. Nous avons aussi décrit les diagrammes qui montrent en détail les différents composants de notre système. Dans la dernière partie, nous avons donné le dictionnaire des données. Dans le prochain chapitre, nous allons décrire l'implémentation et la présentation de notre application.

Chapitre V : Implémentation et présentation de l'application

Nous entamons dans ce chapitre l'implémentation de notre application en présentant les outils de développement qui nous ont permis d'effectuer notre travail ainsi que les technologies utilisées. Par la suite, nous donnerons un aperçu de notre travail grâce à des captures d'écrans.

I. Implémentation de notre base de données dans le SGBDR MySQL

Un SGBDR est un système qui joue le rôle d'interface entre les utilisateurs et la base de données. Il permet de décrire, manipuler et interroger les données d'une base de données. Il est chargé de tous les problèmes liés aux accès concurrents, à la sauvegarde et la restauration des données. Il doit, de plus, veiller au contrôle, à l'intégrité et la sécurité des données.

Notre choix a porté sur MySQL, car c'est un SGBDR avec une licence libre et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public que par des professionnels, au même titre que Oracle et Microsoft SQL Server.).

Une base de données est un outil qui stocke les données de manière organisée et permet de les retrouver facilement par la suite. Le langage qui permet de communiquer avec la base de données est le SQL

II. La plateforme JavaEE

Le terme Java EE signifie *Java entreprise edition* et fait référence à une extension de la plateforme standard **Java SE** (*Java standard Edition*). Il est entièrement écrit en langage java et son objectif est de faciliter le développement d'application web robuste et distribuée, déployée et exécutée sur un serveur d'application [2].

1. Spring Boot

Spring est un Framework bien connu des développeurs Java pour les nombreuses fonctionnalités qu'il apporte sur les aspects web, sécurité, ou encore accès aux données dans le cadre du développement d'une application. Cependant il est aussi réputé pour une configuration complexe et fastidieuse. C'est ce qui a poussé les équipes de spring de travailler sur un projet qui va permettre aux développeurs de produire des applications spring plus facilement et le plus rapidement possible. C'est dans ce cadre qu'est né Spring Boot.

Spring Boot est un micro Framework dont le but est de faciliter la configuration d'un projet Spring et de réduire le temps de démarrage du projet.

Nous pouvons générer la structure d'un projet rapidement en y incluant toutes les dépendances Maven nécessaires à l'application à partir du site web (<https://start.spring.io/>).

2. Maven

Apache Maven, développé par l'organisation *Apache Software Foundation* est un moteur de production ou un gestionnaire de paquet ou en encore outils de construction de projet logiciel Java en général et Java EE en particulier.

Il utilise un paradigme connu sous le nom de **Project Object Model (POM)** afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production. [3]

3. Java Persistence API (JPA)

JPA est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java. Il repose essentiellement sur l'utilisation des annotations introduites dans java 5 comme le montre la figure suivante.

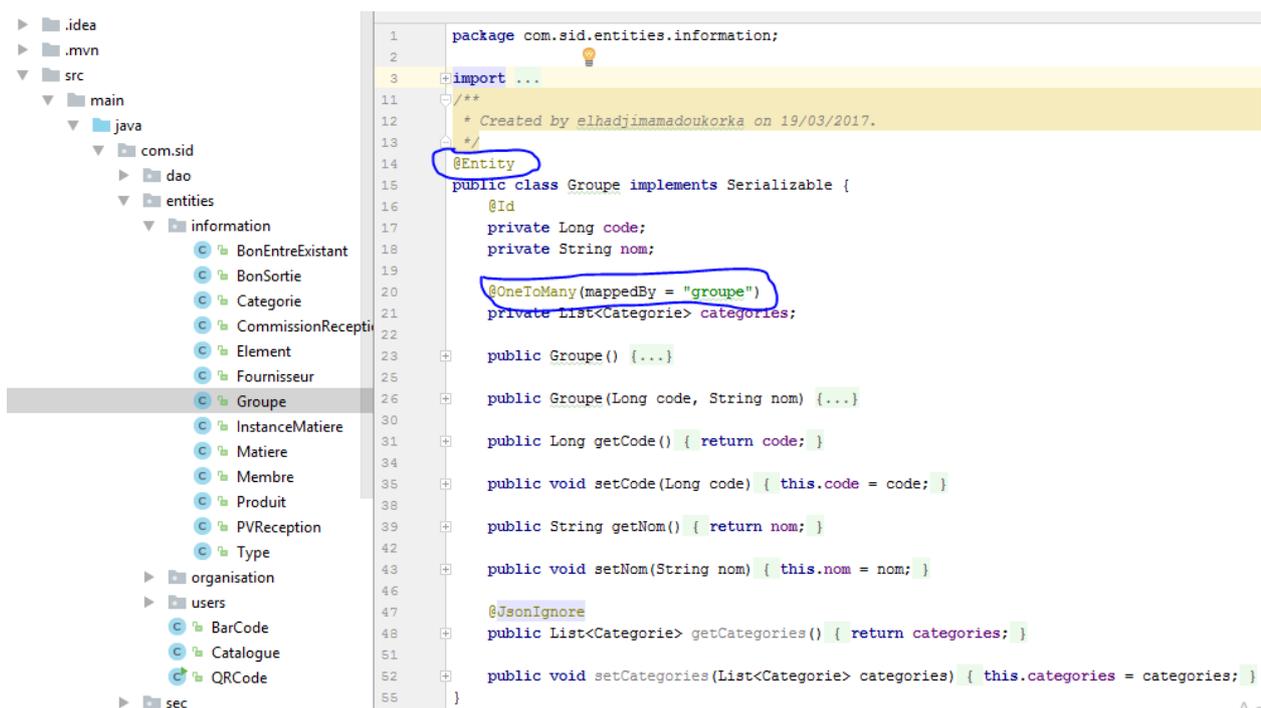


Figure 200 : Entité avec annotation JPA

4. Thymeleaf

Thymeleaf est un moteur de template, sous licence apache 2.0. Il est écrit en Java et génère du XML/XHTML/HTML5. Il a pour but d'être utilisé dans un environnement web pour la génération de vue pour les applications web basées sur le modèle MVC.

Il permet de traiter à la fois les fichiers appartenant à un site web ou non. Il n'y a pas de dépendance vis-à-vis de l'API Servlet.

Il est composé de plusieurs modules appelés dialecte :

- Les caractéristiques d'un dialecte (par exemple : évaluation, itération, etc.) s'appliquent à travers les balises et/ou les attributs des templates.
- Deux dialectes sont disponibles : Standard et le SpringStandard (pour les applications Spring MVC, mais en utilisant la même syntaxe que le dialecte standard).
- Les développeurs peuvent étendre les fonctionnalités des dialectes proposés ou bien créer leur propre dialecte. [4]

5. Bootstrap

C'est un Framework CSS qui embarque aussi des composants HTML et JavaScript. Il offre un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il propose un style pour les formulaires, les boutons, la navigation... il permet de gagner du temps en concevant un site web avec peu de ligne de code et sans connaissance du langage CSS. [5]

III. Outils de travail utilisé

Pour effectuer notre travail, nous avons fait appel aux outils suivants :

1. IntelliJ IDE

C'est un environnement de développement intégré (*IDE*) spécialisé dans le langage Java développé par *JetBrains*. Il fournit une pléthore d'outils nous permettant de développer nos applications java, et ce, sur plusieurs plateformes y compris mobiles. Il en existe deux versions disponibles en téléchargement, la *community Edition* libre et communautaire et l'*Ultimate Edition*.

2. StarUML

StarUML est un logiciel de modélisation UML cédé comme open source par son éditeur, à la fin de son exploitation commerciale sous une licence modifiée de GNU GPL. Il gère la plupart des diagrammes spécifiques dans la norme UML 2.0

3. Serveur Tomcat.

Apache Tomcat est un conteneur web libre de servlets et de JSP Java EE. C'est un des projets de l'*Apache Software Foundation*. Il est un serveur HTTP à part entière. Il gère les servlets et les JSP (par un compilateur Jasper compilant les pages JSP pour en faire des servlets). Tomcat a été écrit en langage Java, il peut donc s'exécuter à l'aide de la machine virtuelle Java sur n'importe quel système d'exploitation la supportant.

4. POS driver setup

C'est un outil qui permet d'installer les pilotes nécessaires pour que l'imprimante thermique puisse se connecter avec l'ordinateur et de configurer le système en créant des ports de communication.[6]

5. Zxing

ZXing prononcé *zebra crossing* est un outil open source de barre code. Il peut être utilisé pour créer et lire des codes Barre et des QR-code. Le projet Zxing est un générateur de code en ligne. Il permet de créer des QR codes qui peut contenir plusieurs sortes de données. Il a fourni aussi une application Android appelée *Barcode Scanner* qui utilise Zxing et utilise la camera du téléphone pour créer, partagé scanner et décoder des QR codes sans aucune communication avec le serveur. Dans notre cas nous allons utiliser sous forme de librairie Java (JAR) que nous allons ajouter à notre projet.

Il téléchargeable en suivant ce lien [7]

IV. Présentation de l'application

Après avoir défini tous les éléments nécessaires à notre projet nous nous sommes concentrés sur la phase d'implémentation de code. Ainsi différentes interfaces ont pu être élaboré dont nous allons présenter quelques une.

Avant tous, nous allons vous montrer une globale de notre application une fois que nous accédons à notre application. Quand le comptable des matières s'authentifie avec les bons paramètres, il La figure suivante en est une illustration.

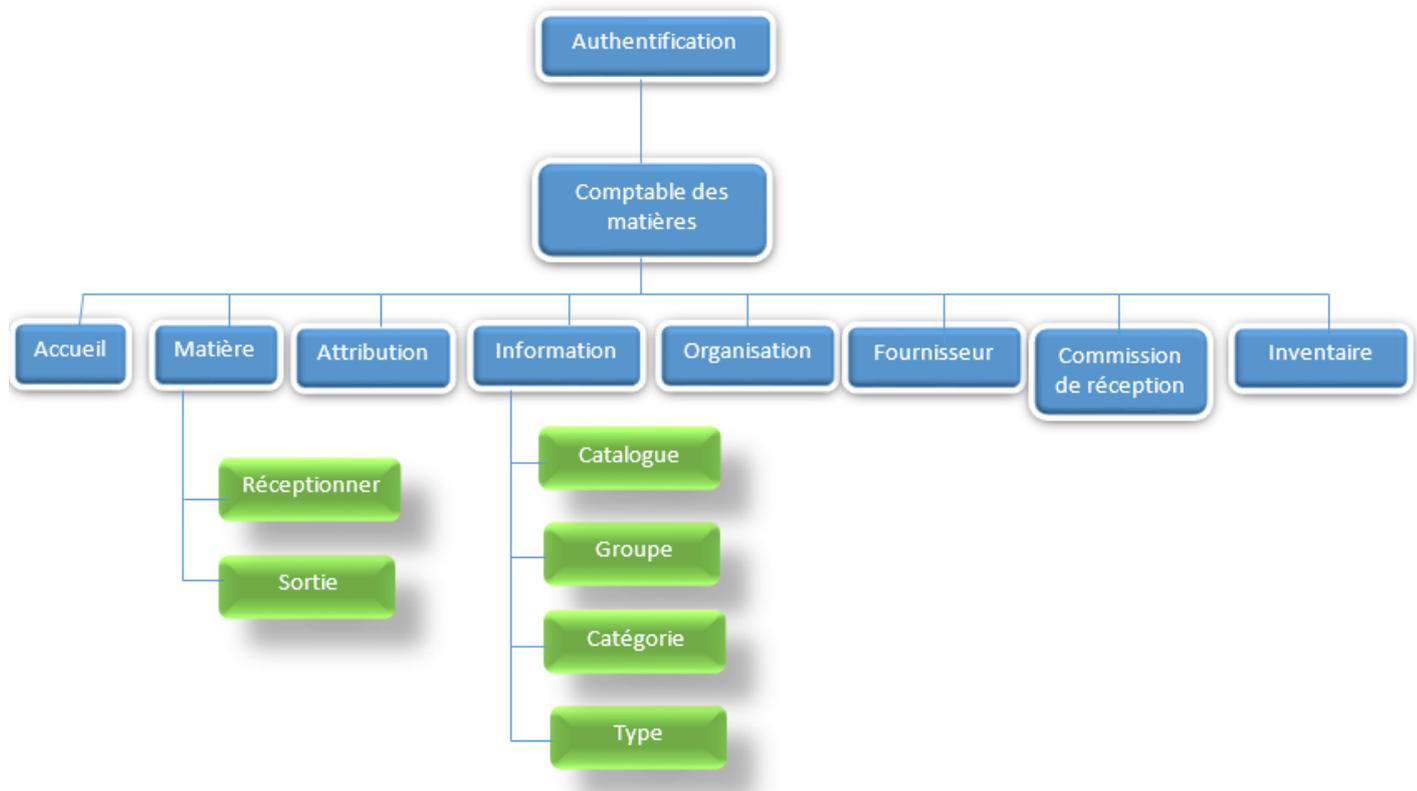


Figure 211 : Arborescence des menus de la page d'accueil

Pour explorer notre application, nous allons montrer le cas d'utilisation « gérer entrée ».

1. L'interface d'authentification

Pour accéder à notre application, le comptable des matières doit s'authentifier en saisissant deux paramètres qui sont le login et le mot de passe. Si ces paramètres sont corrects le comptable des matières à la page d'accueil avec les différents menus.

2. L'interface de la page d'accueil

Après avoir fourni les bons paramètres pendant l'authentification, le comptable des matières accède à la page d'accueil où il voit les différents menus et sous-menu.



Figure 222 : Page d'accueil

3. L'interface « d'ajout d'une réception »

Dans cette partie nous allons présenter l'interface de réception ou le comptable des matières va falloir choisir le type de bon d'entrée qui peut être : prêt, Commande ou Don. Dans un premier temps il va dans le menu *Matière* puis cliqué le sous-menu *Réception* (voir figure 39). Une fois là-bas cliqué sur le bouton *Ajouter*. Un pop-up s'affiche et demande de choisir le type de réception (figure 40).

Numero Bon	Type	Date	Produits		
7	Commande	2017-05-25	Details	Modifier	Supprimer
42	Don	2017-06-23	Details	Modifier	Supprimer
43	Commande	2017-06-24	Details	Modifier	Supprimer
44	pret	2017-07-20	Details	Modifier	Supprimer

Figure 233 : liste de toutes les réceptions

Sujet : Automatisation de la gestion des biens matériels de l'UFR des Sciences et Technologies

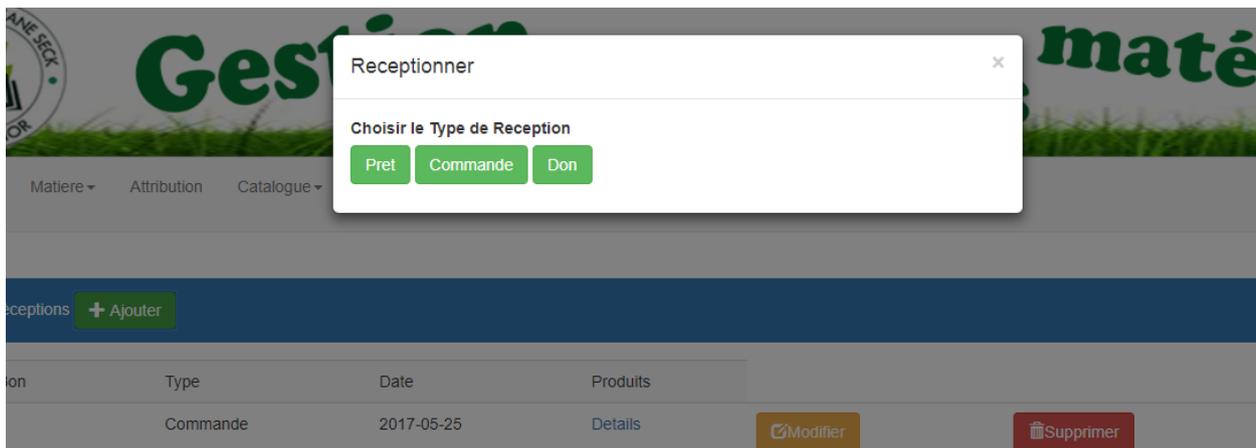


Figure 244 : Formulaire d'ajout d'une réception

Dans le cas où le choix du Type portera sur une commande, le comptable des matières aura ceci comme résultat :

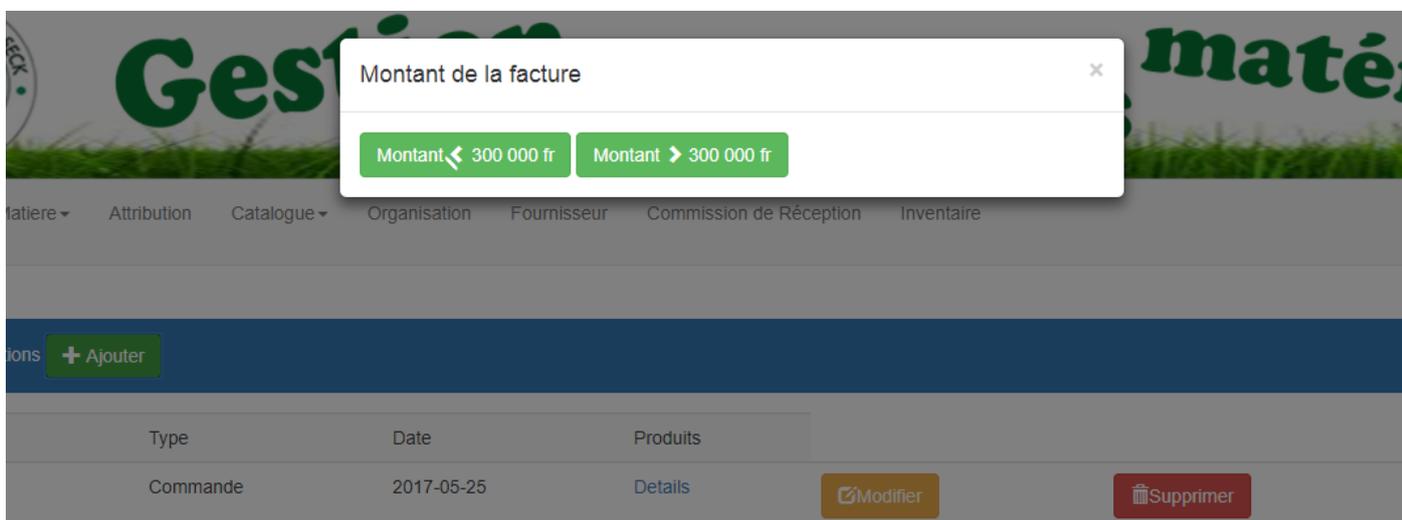


Figure 255 : Le choix du montant de la facture

Si le montant de la facture est supérieur à 300 000 F CFA alors le comptable des matières clique sur le deuxième bouton. Il accèdera enfin dans la figure suivante pour remplir les informations relatives à une réception de matières.

Ajout d'une réception

Type de réception
Commande

Montant > 300 000 Frs

Date 18 Mars 2018

Fournisseur : Sahra Materiaux

Close Save

Figure 266 : Formulaire d'ajout d'une réception de matière

Après avoir rempli les champs et validé le formulaire, le comptable des matières est redirigé vers la page correspondant à la figure (39). Pour éditer un bon d'entrée, il faut cliquer sur « *détail* » du type réception en question et une fois effectué il accède à l'interface suivante.

Bon numero : 43

Date : 2017-06-24

Liste des Produit + Ajouter

Visionner le Procé Verbal

Search Submit

Catalogue	quantite	Unite de Conditionnement	Prix Unitaire	Montant	Observation
-----------	----------	--------------------------	---------------	---------	-------------

Figure 27 : Interface d'édition de bon d'entrée

À partir de cette interface, le comptable des matières peut ajouter les différentes matières appartenant à ce bon d'entrée. Pour cela, il suffit de cliquer sur le bouton *Ajouter produit* comme le montre la figure suivante.

Ajout d'un Produit

Catalogue

Quantité

nature Unite (Unité de conditionnement)

Prix Unitaire

Observation

Figure 28 : Ajout du premier produit pour ce bon

Ajout d'un Produit

Catalogue

Quantité

nature Unite (Unité de conditionnement)

Prix Unitaire

Observation

Dashboard Background:
Accueil Matière Attribution Catalogue
Bon numero : 43
Date : 2017-06-24
Liste des Produit
Catalogue quantité Unite de Con
hp laserjet p2035 2 unité

Figure 29 : Ajout du deuxième produit pour ce bon

Une fois l'ajout effectué, le comptable des matières obtient le résultat suivant.

Sujet : Automatisation de la gestion des biens matériels de l'UFR des Sciences et Technologies

Bon numero :

Imprimer

43

Date :

2017-06-24

Liste des Produit [+ Ajouter](#) [Visionner le Procé Verbal](#)

Search Submit

Catalogue	quantite	Unite de Conditionnement	Prix Unitaire	Montant	Observation			
hp laserjet p2035	2	neant	150000.0	300000.0	ras	Code barre	Modifier	Supprimer
Ordinateur Dell	5	neant	300000.0	1500000.0	Bon état	Code barre	Modifier	Supprimer

Figure 270 : Bon d'entrée avec les différents produits

Pour chaque produit Ajouter, le comptable des matières peut visualiser ces codes-barres en cliquant sur le bouton Codes-barres. Maintenant à lui le choix de les imprimer ou pas.

Code barres du produit : imprimante

Imprimer



hp laserjet p2035
Numero 94

Imprimer



hp laserjet p2035
Numero 95

Imprimer

Figure 281 : Codes-barres du premier produit ajouté

Liste d'element

Imprimer



Ordinateur Dell
Numero 166

Imprimer



Ordinateur Dell
Numero 167

Imprimer



Ordinateur Dell
Numero 168

Imprimer



Ordinateur Dell
Numero 169

Imprimer



Ordinateur Dell
Numero 170

Imprimer

Figure 29 : Codes-barres du deuxième produit ajouté

Sujet : Automatisation de la gestion des biens matériels de l’UFR des Sciences et Technologies

Id	Année
1	2017

Nom	Prenom	Fonction	Commission
Camara	Fatou	Technicienne de laboratoire	2017
Diop	Ndickou	Secrétaire	2017
Tine	Fatou	Directeur des Finances	2017
Diallo	Tahirou	Directeur Marketing	2017
Badiane	Cheikh Souleymane	Comptable des matières	2017

Figure 302 : Commission de réception

Le comptable des matières peut maintenant créer le PV de réception en cliquant sur « visionner le PV de réception » (voir la figure 43). Ce qui va lui amener à la page de rédaction du PV de réception. La figure suivante en est une illustration.

UNIVERSITE ASSANE SECK
DE ZIGUINCHOR



UNITE DE FORMATION ET DE RECHERCHE
DES SCIENCES ET TECHNOLOGIES

PROCES - VERBAL DE RECEPTION

Budget : **Membre Commission:**
 Chapitre : fournisseur
 Type de Bon : **Commande**
 Arrete le présent Procès-Verbal à 7 Unités
 représentant une valeur de **1800000.0** francs
 Le : **2017-06-24**

Produits	quantite	Unite de Conditionnement	Prix Unitaire	Montant	Observation
hp laserjet p2035	2	unité	120000.0	300000.0	Bon état
Ordinateur Dell	5	neant	300000.0	1500000.0	Bon état

Figure 313 : Le PV de réception du nouveau Bon d’entrée

Le comptable des matières peut éditer le PV de réception en cliquant sur le bouton modifier. Ainsi il va renseigner les différents champs puis valider (voir figure 51) et le résultat peut être observé à la figure 52.

Figure 324 : Formulaire du Procès-Verbal

UNIVERSITE ASSANE SECK
DE ZIGUINCHOR



UNITE DE FORMATION ET DE RECHERCHE
DES SCIENCES ET TECHNOLOGIES

PROCES - VERBAL DE RECEPTION

Budget : **2016**

Chapitre : **1122**

Type de Bon : **Commande**

Arrete le présent Procès-Verbal à **7** Unités

représentant une valeur de **1800000.0** francs

Le : **2017-06-24**

Membre Commission:
Fatou Camara
Ndickou Diop
Fatou Tine
Tahirou Diallo
Cheikh Souleymane Badiane
fournisseur

+Modifier					
Produits	quantite	Unite de Conditionnement	Prix Unitaire	Montant	Observation
hp laserjet p2035	2	unité	120000.0	300000.0	Bon etat
Ordinateur Dell	5	neant	300000.0	1500000.0	Bon état

Figure 335 : Procès-Verbal complété

4. Attribution

Elle consiste à prendre des matières et a les affectés à un bénéficiaire. Cette fois-ci l'opération se fera avec l'aide de la douchette qui va lire les codes-barres collés sur les matières. Pour cela, le comptable des matières va se rendre sur le menu « Attribution », choisir le bénéficiaire puis de

Sujet : Automatisation de la gestion des biens matériels de l'UFR des Sciences et Technologies

scanner le QR-code et valide comme le montre la figure 53. Après l'avoir effectué, il observe le résultat sur la figure 54 qui montre les différentes attributions. Dans cette page il peut effectuer des recherches selon les critères suivants : département, bénéficiaire et produit.

The screenshot shows a web interface for assigning a material to a beneficiary. At the top, there is a dropdown menu for 'Beneficiaire' with 'Youssou' selected. Below it is a text input field for 'Matiere' containing '10.1.3_40'. At the bottom of the form are two buttons: 'Fermer' and 'Attribuer'. Below the form is a section titled 'Liste des Attribution' with a blue header. It contains three search filters: 'Matiere' with a 'Rechercher' button, 'Beneficiaire' with a 'Rechercher' button, and 'Departement' with a 'Rechercher' button. Below these filters is a table with three columns: 'Prenom', 'Nom', and 'Matériel'.

Figure 36 : Attribution d'une matière à un bénéficiaire

This screenshot shows the same web interface as Figure 36, but with the 'Attribuer' button highlighted in blue. The 'Liste des Attribution' section now displays a table with one row of data. The table has three columns: 'Prenom', 'Nom', and 'Matériel'. The row contains the values 'Youssou', 'DIENG', and 'hp pro 400 dn'. To the right of this row is a red button with a white 'x' icon and the text 'Supprimer'.

Prenom	Nom	Matériel
Youssou	DIENG	hp pro 400 dn

Figure 37 : Résultat de l'attribution

Conclusion générale

Le but que nous nous sommes fixé était de concevoir et d'implémenter une application optimale chargée d'automatiser la gestion des biens matériels de l'UFR ST. Pour atteindre ces objectifs, nous nous sommes appuyés sur une démarche bien structurée. Nous avons en premier lieu présenté notre institut qui est l'Université Assane Seck de Ziguinchor. Nous avons parlé de l'existant, des limites des solutions actuelles les problèmes rencontrés et proposé une solution pour pallier ces limites. Dans la deuxième partie, nous avons fait un exposé clair sur la méthodologie appliquée à notre projet.

Après, nous avons en troisième partie abordé la spécification des besoins ou nous avons listé toutes les fonctionnalités de notre solution informatique.

Ensuite nous avons entamé dans le troisième chapitre la conception de notre système après avoir donné son architecture pour finir par présenter le dictionnaire de donnée.

En fin, nous avons fait un tour sur les outils utilisés pour la réalisation de notre application pour par la fin faire un petit aperçu de notre solution informatique.

Cependant, certaines fonctionnalités n'ont pas pu être implémentées à savoir :

- localiser une matière au sein de l'UASZ ;
- déclarer un problème sur un bien matériel ;
- faire un suivi et une maintenance des biens matériels ;
- effectuer des études statiques sur les biens matériels ;
- étendre l'application pour toutes les UFR ;
- étendre pour les autres Universités.

Bibliographie et Webographie

- [1] crf.wallonie.be/compasinfo/images/site1/edit/Dossiers/SCRUM.pdf
- [2] « Créez votre application web avec Java EE - OpenClassrooms ». [En ligne]. Disponible sur: <https://openclassrooms.com/courses/creez-votre-application-web-avec-java-ee>. [Consulté le: 16-janvier-2018].
- [3] « Développons en Java - Maven ». [En ligne]. Disponible sur: <https://www.jmdoudoux.fr/java/dej/chap-maven.htm>. [Consulté le: 18-janvier-2018].
- [4] « Thymeleaf — Wikipédia ». [En ligne]. Disponible sur: <https://fr.wikipedia.org/wiki/Thymeleaf>. [Consulté le: 19-janvier-2018].
- [5] « Prenez en main Bootstrap - OpenClassrooms ». [En ligne]. Disponible sur: <https://openclassrooms.com/courses/prenez-en-main-bootstrap>. [Consulté le: 20-janvier-2018].
- [6] « Thermal Receipt Printer Driver - POS-X ». [En ligne]. Disponible sur: <https://www.pos-x.com/download/thermal-receipt-printer-driver-2/>. [Consulté le: 25-janvier-2018].
- [7] « GitHub - zxing/zxing: ZXing (“Zebra Crossing”) barcode scanning library for Java, Android ». [En ligne]. Disponible sur: <https://github.com/zxing/zxing>. [Consulté le: 23-janvier-2018].
- [8] « Guide de démarrage Scrum - L'Agiliste ». [En ligne]. Disponible sur: <https://www.agiliste.fr/guide-de-demarrage-scrum/>. [Consulté le: 13-janvier-2018].
- [9] « Gestion de projets: méthode Agile et Scrum pour les nuls ». [En ligne]. Disponible sur: <https://blog.trello.com/fr/methode-agile-scrum-gestion-projet>. [Consulté le: 13-janvier-2018].
- [10] « Introduction aux méthodes agiles et Scrum - L'Agiliste ». [En ligne]. Disponible sur: <https://www.agiliste.fr/introduction-methodes-agiles/>. [Consulté le: 15-janvier -2018].

Annexes

Annexe 1 : Création de la base de données

Pour la création de notre base de données, nous nous rendons sur le panneau d'administration graphique de WAMP Server qui est phpMyAdmin (*localhost/phpmyadmin*).

- Cliquer sur le menu Bases de données
- Puis écrire le nom de la base de données comme l'indique la figure ci-dessous.
- Enfin cliquer sur le bouton Créer.



Figure 38: Création de la base de données

Maintenant il reste à établir la liaison entre notre application et notre SGBDR. Pour cela, nous allons au niveau de notre IDE. Voici un aperçu sur l'arborescence de notre projet.

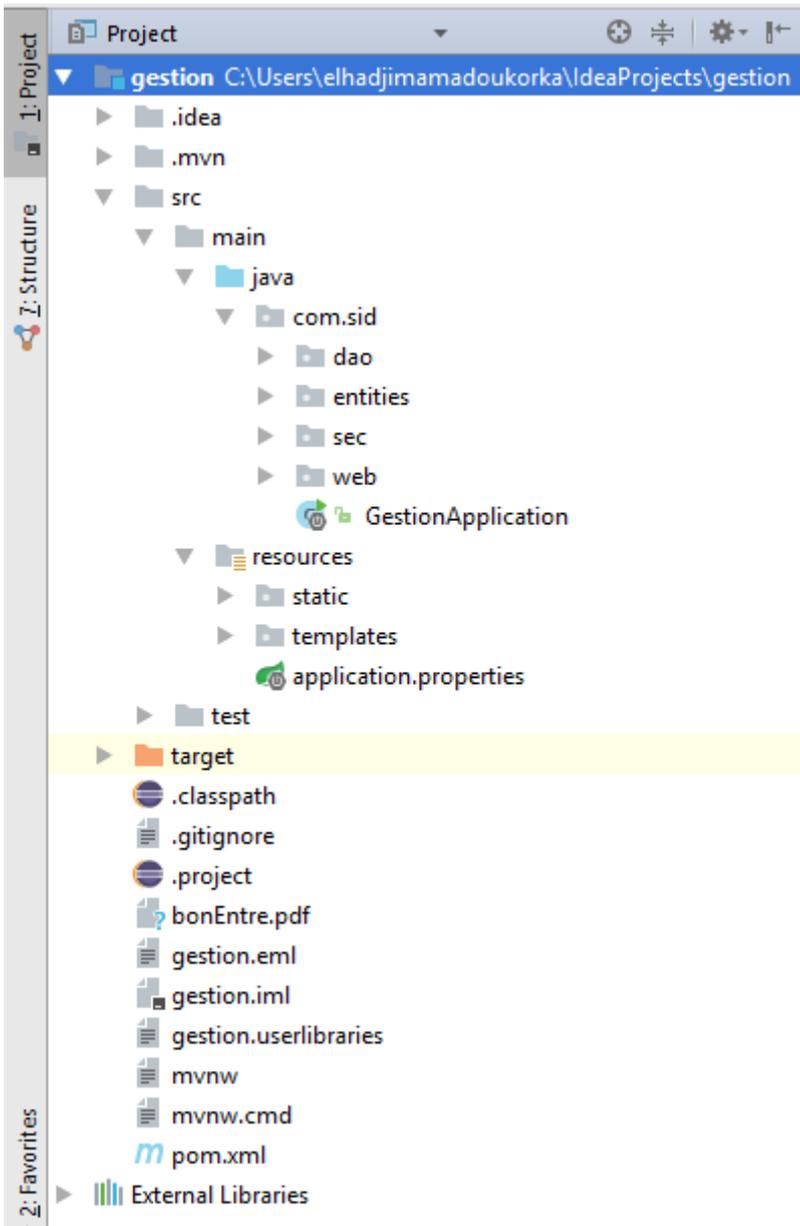


Figure 39 : Arborescence du projet

Une fois, là-bas éditez le fichier **application.properties** situé dans le répertoire **ressource**. En recopiant le code de la figure ci-dessous sans oublier de préciser à la fin de la première ligne le nom de la base de données.

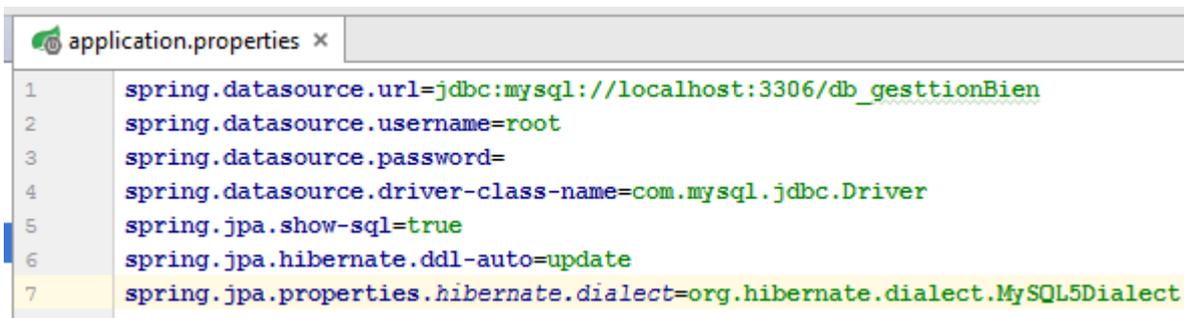


Figure 340 : Configuration de la base de donnée avec IntelliJ

Sujet : Automatisation de la gestion des biens matériels de l'UFR des Sciences et Technologies

Il faut également ajouter un connecteur MySQL à notre projet pour. Pour ce faire rendez-vous au menu **File** de notre IDE :

- Cliquer sur le sous-menu **Project Structure**
- Cliquer sur **Librairies** situées au le panel gauche de la fenêtre
- Cliquer sur le signe plus (+) pour aller sélectionner le fichier JAR de mysql_connector.jar.
- Enfin cliquer sur OK pour valider

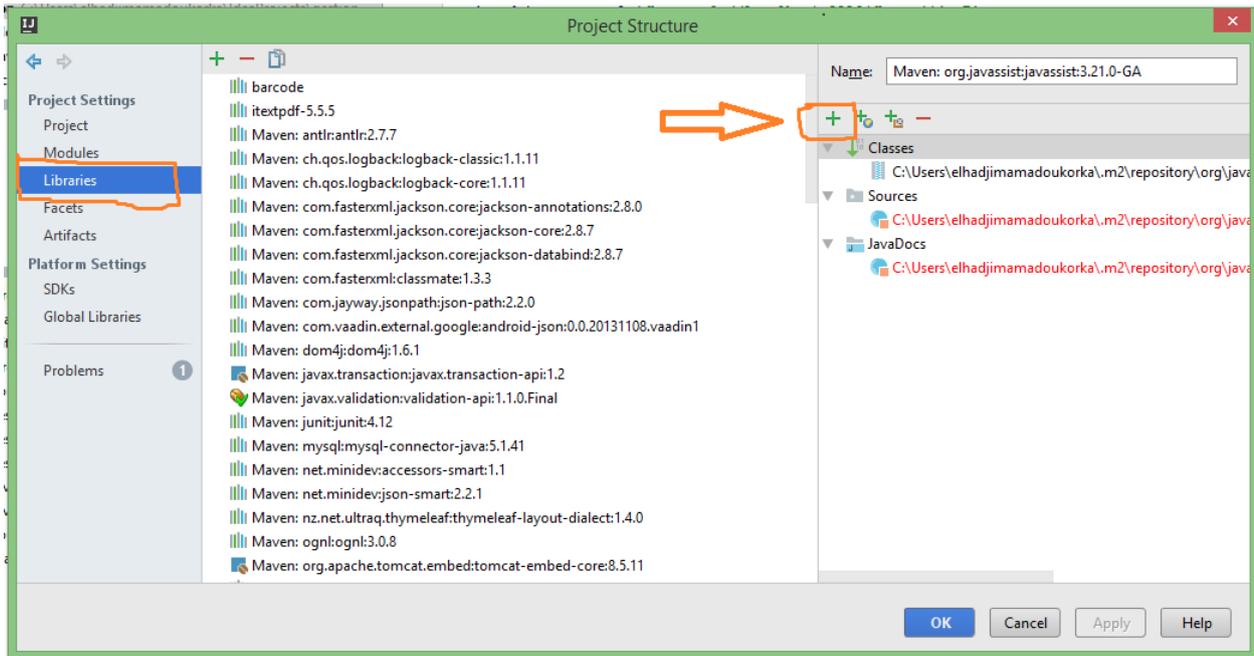


Figure 352: Configuration de MySQL connecteur

Une fois tout terminé, il suffit de créer les classes Java et exécuter l'application, les tables sont automatiquement générées comme la figure ci-dessous le montre.

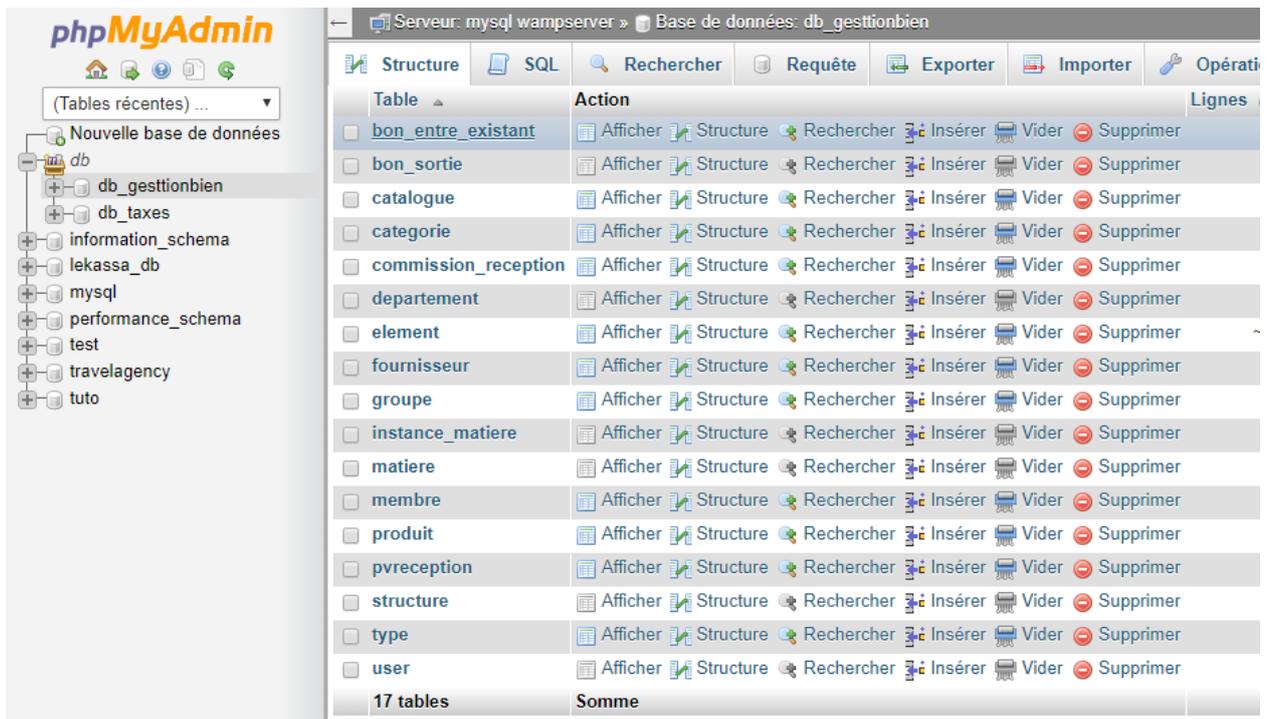


Figure 363 : Génération de la base de données

Annexe 2 : L'impression au niveau du contrôleur

Nous avons commencé d'abord par chercher le matériel qu'on veut imprimer son code-barres puis extraire le titre du produit qui va nous permettre de le retrouver parmi les images enregistrées. Ensuite nous avons instancié notre classe **BarCode** pour lui donner le chemin du fichier en question et après vient le code qui affiche l'interface d'impression de Java ou nous allons choisir notre imprimante thermique.

```
@RequestMapping(value = "/imprimerCodeBarre")
private String imprimerCodeBarre(@RequestParam(name = "f") Long id, Model model) {
    Element element=elementRepository.getOne(id);
    element.setId(id);
    Produit produit=element.getProduit();
    String titre=produit.getCatalogue().getDescription()+"-numero-"+element.getId();
    BarCode barCode=new BarCode();
    barCode.setPathname("C:\\Users\\elhadjimamadoukorka\\IdeaProjects\\gestion\\src" +
        "\\main\\resources\\static\\tmp\\" + titre + ".png");
    PrinterJob job=PrinterJob.getPrinterJob();
    job.setPrintable(barCode);
    boolean doPrint = job.printDialog();
    if (doPrint){
        try {
            // Effectue l'impression
            job.print();
        } catch (PrinterException ex) {
            ex.printStackTrace();
        }
    }
    return "/home";
}
```

Figure 374: Code Source de l'impression

- **L'algorithme d'impression de code barre**

Comme nous l'avons dit ci-dessus les codes-barres sont enregistrés sous forme d'image et à partir de ces fichiers que nous allons élaborer un algorithme qui va les imprimer à travers l'imprimante thermique. Pour l'algorithme d'impression nous allons d'abord montrer la class Java de personnalisation d'un code-barres puis le code du côté du contrôleur qui permet d'imprimer l'objet barre code.

- **La classe Java BarCode**

Elle implémente l'interface Printable et dans cette classe nous avons redimensionné la taille de l'image du code barre puis on va lire le fichier image du code-barre.

```
public class BarCode implements Printable {
    private String pathname;
    @Override
    public int print(Graphics graphics, PageFormat pageFormat, int pageIndex) throws PrinterException {
        int retValue = Printable.NO_SUCH_PAGE;
        switch(pageIndex){
            case 0 : {
                int x=50 ; //print start at 100 on x axes
                int y=80; //print start at 10 on y axes
                int imagewidth=50; //100
                int imageheight=50; //100
                BufferedImage read;
                try {
                    read = ImageIO.read(new File(pathname));
                    graphics.drawImage(read,x,y,imagewidth,imageheight, observer: null);
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                retValue = Printable.PAGE_EXISTS;
                break;
            }
            case 1 : {
                break;
            }
        }
        return retValue;
    }
    public String getPathname() { return pathname; }
    public void setPathname(String pathname) { this.pathname = pathname; }
}
```

Figure 385 : Code source de la classe BarCode

Annexe 3 : génération de code-barres

Pour la génération de code-barres, nous avons utilisé une librairie externe dénommée zxing qui permet de créer des codes-barres suivant différents formats (JPG PNG...). Voici l'algorithme qui nous a permis d'effectuer ce travail.

```
201 @RequestMapping(value = "/codeBarreProduit")
202 @ private String codeBarreProduit(Model model,@RequestParam(name = "f") Long id) throws Exception {
203     Produit produit=new Produit();
204     Produit pr=new Produit();
205     produit=produitRepository.findOne(id);
206     List<Element> listesElement=elementRepository.findByProduit(produit);
207     /*****/
208     for (int i = 0; i <listesElement.size() ; i++) {
209         Element im=listesElement.get(i);
210         pr=im.getProduit();
211         String titre=pr.getCatalogue().getDescription()+"-numero-"+im.getId();
212         ByteArrayOutputStream out= net.glxn.qrgen.QRCode.from
213             (pr.getCatalogue().getCode()+"_"+im.getId()).to(ImageType.PNG).stream();
214         File file=new File( pathname: "C:\\Users\\elhadjimamadoukorka\\IdeaProjects\\gestion" +
215             "\\src\\main\\resources\\static\\tmp\\"+titre+".png");
216         FileOutputStream fos=new FileOutputStream(file);
217         fos.write(out.toByteArray());
218         fos.flush();
219     }
```

Figure 396 : Code source de Génération de code-barre