



Cahier de Conception : Gestion d'une médiathèque

Groupe :
-GLA : M

Encadré par :
- Burkhardt Wolff

Réalise par :
-Echaimae Ouaddane
-Kebir Abdelhamid

Sommaire :

Introduction

I)Choix de la technologie

- 1)Choix du langage de programmation
- 2)Choix des bibliothèques PHP

II)Les Diagrammes de cas d'utilisation

- 1)Le Gestionnaire
 - a) Le Bibliothécaire
 - b) L'assistant du bibliothécaire
- 2)L'abonné et l'utilisateur
 - a) Le client : Abonné+Utilisateur
 - b) Diagramme de l'utilisateur + Diagramme de l'abonné
- 3) Diagramme de cas d'utilisation de la médiathèque complète

III)Le Diagramme de classe

- Explication de toutes les classes

IV)Les Diagrammes de séquence et d'état

- Énoncé de 7 scénarios exemples

V)Les Invariants

- Inv 1
- Inv 2
- Inv 3
- Inv 3
- Inv 4

VI)Modèle conceptuel de données

- 1) MCD
- 2) MLD

VII)Pseudo-code SQL

VIII) A quoi ressemblera notre site web ?

- 1)Diagramme d'enchaînement
- 2) Présentation en image de quelques fonctionnalités du site web

Introduction :

Objectifs de la conception :

Après avoir vu le cahier d'analyse de notre mediatheque , nous allons maintenant passer au cahier de conception de notre site web , avec le quel on aura une vision d'ensemble concrète et surtout technique du point de vue du code et des technologies a entreprendre , pour ainsi commencer l'implémentation de notre site web .

Par contre , nous devons prendre en compte les contraintes de conception , et tracer nos objectifs avec lesquelles , on pourra faire une bonne implémentation , notamment en prenant en compte les exigences de réalisation avec tout ce qui est performances , robustesse synchronisation , ainsi que le coté sécuritaire .

Nous devons aussi prévoir les méthodes et leur signatures , ainsi que des méthodes supplémentaire , en mettant en œuvre des shemas d'architecture logicielle .

Une fois notre diagramme de classe et de conception bien détaillé , nous choisirons donc un langage de programmation cible qui sera en adéquation avec nos methodes et nos classes , prenant en compte les concepts d'héritage et d'héritage multiple , tout en vérifiant la cohérence de nos choix , avec un debut d'implémentation et des squelettes de codes qui nous donnerons une vision globale sur l'implémentation.

On aura donc a choisir la représentation des associations entre les classes , et les tables de notre base de données , et cela en les illustrant a l'aide de methodes que nous définirons , nous verrons aussi les types et attributs que nous avons , avec sûrement des attributs complexes qui auront besoin d'etre décomposés ou optimisé , en précisant les types des classes , prédéfinis ou pas , enumeraions ou pas ainsi que la hiérarchie d'exceptions , et l'identification des methodes complexe avec la déduction des méthodes auxiliaires nécessaires . Nous feront aussi notre possible pour éviter a trop retoucher la signature des méthodes dans la partie implémentation , pour que note partie de conception soit la plus finale possible

Comment arriver a ces objectifs ?

Nous mettrons en t place quelques squelettes de code , ainsi que quelques screenshot de comment notre site web sera , du moins a quoi il ressemblera .

On précisera donc la technologie et les langages de programmation choisis pour la conception de notre site web , ainsi que les bibliothèques qui nous seront utiles pour coder , ainsi que les bibliothèques adéquates pour ce qu'on va entreprendre .

Nous allons donc avoir une vision concrète de notre mediatheque a l'aide des squelettes de codes , et les screenshot de notre site web , ainsi que les diagrammes de cas d'utilisations et les diagrammes de séquences qu'on a vu precedement dans notre cahier d'analyse , mais qui seront cette fois revisités , de manière a ce que nos methodes et nos classes soient en adequation avec nos diagrammes proposés precedement , meme chose pour les diagrammes de sequences qui seront cette fois illustrés plus ou moins avec des scénarios décrits par les méthodes et les classes que nous allons peut etre utiliser .

Nous allons donc avoir une vision plus technique grâce a ce cahier , qui nous permettra de commencer a l'implémentation de notre site web .

Nous allons aussi utiliser un Modèle conceptuel de données pour illustrer notre base de données qui nous servira s'appuie pour notre code et pour notre site web .

Les points importants coté analyse :

Nous continuerons a s'intéresser et a détailler a l'analyse vue precedement dans le cahier d'analyse en précisant les points suivants :

- les fonctionnalités et les services que la mediatheque offre aux internautes , c'est a dire , tout ce qui concerne les types de documents et de media a emprunter , les loisirs , ainsi que les projections de films et vidéos et tout ce qui va avec
- Les différents acteurs qui peuvent accéder a notre site web en précisant plus concrètement les acteurs interne de la mediatheque qui sont bien entendu les employés , et les acteurs externe de la mediatheque qui sont personne d'autre que les abonnés .
- Le processus d'emprunt de media et de document , en détaillant cela avec quelques scénarios qui donnerons des exemple sur comment se passe l'emprunt de la part d'un client , et autre .
- Le processus d'inscription , qui nous donnera une idée sur l'inscription de abonné a la mediatheque , la création de son compte , en utilisant un scenario et des diagrammes pour illustrer cela
- La gestion de stock , qui est très important dans la mediatheque , qui nous permettra de savoir comment les pertes seront gérés a la fin du mois , et comment calculet cela .
- La gestion d'authentification , qui illustrera la méthode d'authentification des abonnés et des employés de la mediatheque , avec leur compte utilisateur
- Et pour finir , la gestion des comptes utilisateurs pour notamment savoir comment ils seront créer , par qui , et quand .

Les points importants cotés conception : (taches TODO)

- Définir la technologie et les langages de programmation
- Définir les bilbliothèques adéquates a notre langage de programmation choisis
- Revisiter les diagrammes vues dans le cahier d'analyse afin d'être en adéquation avec notre technologie choisi
- Définir un modèle conceptuel de données pour illustrer nos tables dans la base de données
- Illustrer la structure de notre site web avec des screenshot de notre mediatheque , en mettant en evidance les barres de recherche , l'accueil du site , et ainsi de suite

Donc après avoir déterminé les principales tâches à détailler ci-dessous dans notre projet, on va utiliser quelques outils qui nous accompagneront tout au long de cette conception et analyse, et cela en utilisant bien-sûr notre cahier des charges et d'analyse qui ont été conçus précédemment, mais principalement en utilisant la méthode d'analyse unifiée UML qui nous permettra d'avoir une bonne conception d'objet, et surtout une conception concrète, à l'aide de différents diagrammes et scénarios qui illustreront plusieurs exemples, qui seront cette fois-ci détaillés à l'aide des méthodes et leur signatures qui précisent les associations et les définiront comme il le faut.

Les diagrammes qui seront utilisés seront les suivants :

- Diagramme de classe, qui nous donnera une idée sur notre base de données, et sur comment on structurera notre code
- diagramme de cas d'utilisation, on en fera un pour chaque acteur de la médiathèque, puis un autre plus globale qui regroupera tous les acteurs de la médiathèque, et cela pour illustrer les différents rôles de chaque acteur
- Les diagrammes de séquences, qui nous donneront plusieurs exemples et scénarios d'utilisation de la médiathèque, et cela concernant l'authentification, les emprunts etc
...

Exigences de realisation :

Sécurisation :

Du point de vue sécuritaire , notre site web ne demande pas de grandes exigences , vu qu'il ne contiendra pas des données de cartes bancaires de clients , ou autres , tout ce qu'il y a a éviter , sont les spams , et les injections SQL, et la préservation de l'anonymat des utilisateurs , et tout cela pourrait etre évité grace a une bilbliothèque ou librairie dédiée a ça

Performance :

Les reponses aux recherches des utilisateurs doivent etre rapides , precises , se rapprochant le plus possible de ce qu'ils recherche , ou dans le meilleure cas , leur trouvant exactement la ressource qu'ils recherchent si toutefois , elle est existante dans la database de la mediatheque

Robustesse :

Le site web doit pouvoir tenir avec les recherches simultanées des utilisateurs ,

Synchronisation :

Le site web doit pouvoir communiquer avec les serveurs web tel que mozilla , google chrome , ou autres , ainsi que d'avoir une bonne complémentarité avec la data base , ce qui se fera correctement vu notre choix de technologie qu'on va enoncer dans les pages a venir

I- Technologie a adopter pour concevoir notre projet ?

1) Choix du langage de programmation :

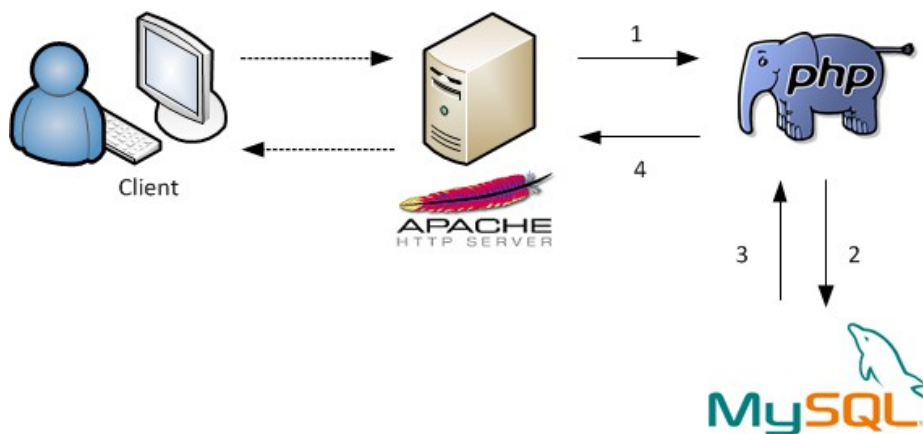
Dans ce projet , nous cherchons a concevoir un site web pour une mediatheque , nous nous tournons naturellement vers le php , qui est a notre avis , le langage de programmation le plus adapté pour concevoir un site web dynamique , que nous pourrons toute fois marier a quelques bouts de code de java script , si besoin , notamment pour implémenter des boutons .

Tout d'abord , nous allons s'appuyer sur une base de données qu'on va bien évidemment administrer en utilisant des requetes MySql (qui sera aussi un langage de programmation qu'on va aussi utiliser)

Et comme le php gère tres bien les requetes MySql contrairement aux autres langages , nous allons l'utiliser principalement pour ça , nous pourrons donc facilement écrire des programmes qui affichent des données extraites de bases SQL , ou qui stockent des données postées par un formulaire dans une table SQL , en ajoutant le fait que le langage PHP sait communiquer avec presque tous les SGBD contrairement aux autres langages qui sont plus ou moins limités dans ce point la . Nous pourrons donc utiliser des SGBD tel qu'Oracle , MySql , Db2 , Informix , Postgresql , SQL Server , Acces etc .. , on voit donc que le choix s'ouvre a nous avec de multiples options dans un niveau d'abstraction concernant les bases de données .

Il y'a aussi le fait que les plateformes Linux/Apache/PHP , sont d'une très grande stabilité , notre site web pourra donc tourner des mois et des mois sans pour autant avoir besoin d'un reboot .

Autre critère qu'on a pris en considération pour choisir la technologie PHP est le critère de maintenance , en effet , après une longue hésitation entre JAVA et PHP , et en effectuant de petits tests , nous avons conclu que la maintenance en PHP se fait plus ou moins plus simplement qu'en JAVA , en effet un bug dans PHP est moins bloquant et handicapant qu'un bug en JAVA , vu qu'essentiellement , un bug en JAVA arrête carrément le traitement , la ou un bug en PHP , peut juste bloquer une partie de l'affichage , ce qui n'est pas vraiment handicapant , le seul handicap concernant la maintenance et les bugs auxquelles on fera face , c'est par rapport a la detection , en effet , le PHP etant un langage qui n'est pas compilé , un certain nombre de bug se presenteront seulement lors de l'execution , mais on essaiera de faire avec ce probleme en optimisant la qualité de notre developpement .



2) Choix des bibliothèques PHP :

Dans cette partie , nous allons vous donner une idée des librairies que nous allons utiliser , on distinguera les bibliothèques par catégories , qui sont notamment ceux consacrés aux bases de données , développement , documents , mails , HTML/CSS , formulaires , Images/Media , JAVA SCRIPT , Microformats , Flux , Sécurité et enfin des bibliothèques consacrés aux tests .

Dans chaque catégorie , nous allons choisir plusieurs bibliothèques , en utiliser une seule généralement , ou d'autres si besoin durant l'implémentation du code , ce qui est important , c'est d'envisager plusieurs options de librairies en cas de besoin ou de défaillance ou de non concordance , donc nous n'avons pas pris le risque d'en choisir seulement une seule pour chaque catégorie .

a) Librairies de base de données :

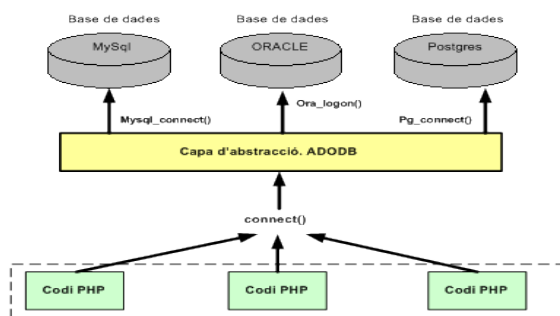
Dans cette catégorie , nous allons proposer des librairies relatives à la gestion de notre base de données , dans notre cas de la médiathèque , notre base de données sera plutôt simpliste , du coup nous privilégions l'utilisation de Mimesis , mais toute fois , nous gardons en secours , ADOdb qui est aussi très bien et plus développée

ADODB

- Une librairie orientée objet écrite en PHP qui abstrait les relations à la base de données pour la portabilité.
- Elle est modelée sur la librairie ADO de Microsoft, mais a beaucoup d'améliorations qui la rendent unique comme les table pivots, le support d'active records, la génération de code HTML pour la pagination de jeux de données avec les liens suivants/précédents, jeux de données en cache, la génération de menus HTML, etc.).
- Supporte un large nombre de base de données incluant : MySQL, PostgreSQL, Interbase, Firebird, Informix, Oracle, MS SQL, Foxpro, Access, ADO, Sybase, FrontBase, DB2, SAP DB, SQLite, Netezza, LDAP, and generic ODBC, ODBTP.

Mimesis

- Mimesis est une API bas niveau de base données par fichiers plat écrite en PHP et Open Source créée pour agir comme backend pour les scripts côté serveur qui demandent les fonctionnalités d'une base de données.
- Au lieu de parser des états SQL, Mimesis utilise des constructions orientées objet de PHP pour fournir une classe distincte avec différentes méthodes de manipulation de base de données.



b) Librairie de developpement :

Nous avons bien évidemment besoin d'un outil de développement tel que la JAVADOC par exemple , nous proposons donc ces deux librairies :

PhpCodeSniffer

- PHP CodeSniffer est un script PHP5 qui segmente et «sniff» le code PHP pour détecter les violations d'un standard de code défini.
 - C'est un outil de développement essentiel qui assure la qualité du code.
- Il peut aussi aider à prévenir des erreurs sémantiques de code des développeurs.

PhpDocumentor

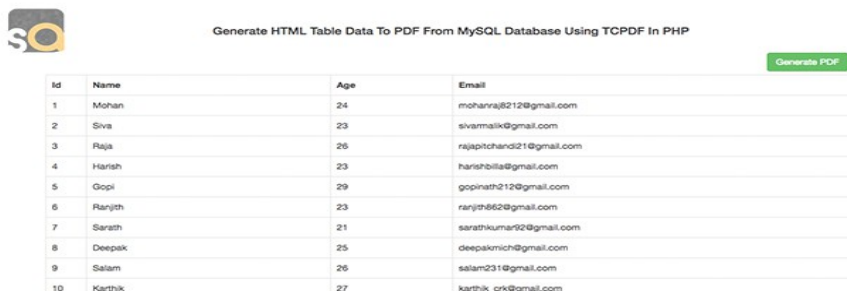
- Similaire à Javadoc, écrit en PHP, phpDocumentor peut être utilisé depuis la ligne de commande ou via une interface web pour créer des documentations professionnelles à partir du code PHP.
- phpDocumentor contient un support pour la liaison à la documentation, incorporant des documents de niveau utilisateur comme les tutoriaux la création de code source surligné avec des références croisées avec la documentation PHP.

c) Librairie de document :

Ces librairies concernent notamment l'encodage des caractères et des documents qui pourrait être défectueux , ou qui ne s'encode pas correctement , nous proposons donc ces deux librairies.

TCPDF

- Une classe PHP Open Source pour générer des documents PDF.
- Ne requiert pas de librairie externe pour les fonctions basiques; supporte tous les formats de page ISO y compris UTF-8, Unicode, les langages RTL et HTML.



The screenshot shows a web application titled "Generate HTML Table Data To PDF From MySQL Database Using TCPDF in PHP". It features a table with 10 rows of data. A green button labeled "Generate PDF" is located in the top right corner of the table area.

Id	Name	Age	Email
1	Mohan	24	mohanna1212@gmail.com
2	Siva	23	sivamalik@gmail.com
3	Raja	26	rajapitchand21@gmail.com
4	Harish	23	harishbilla@gmail.com
5	Gopi	29	gopinath212@gmail.com
6	Ranjith	23	ranjith662@gmail.com
7	Sarath	21	sarathkumar92@gmail.com
8	Deepak	25	deepakmich@gmail.com
9	Salam	26	salam231@gmail.com
10	Karthik	27	karthik_crk@gmail.com



d) Librairie de mail :

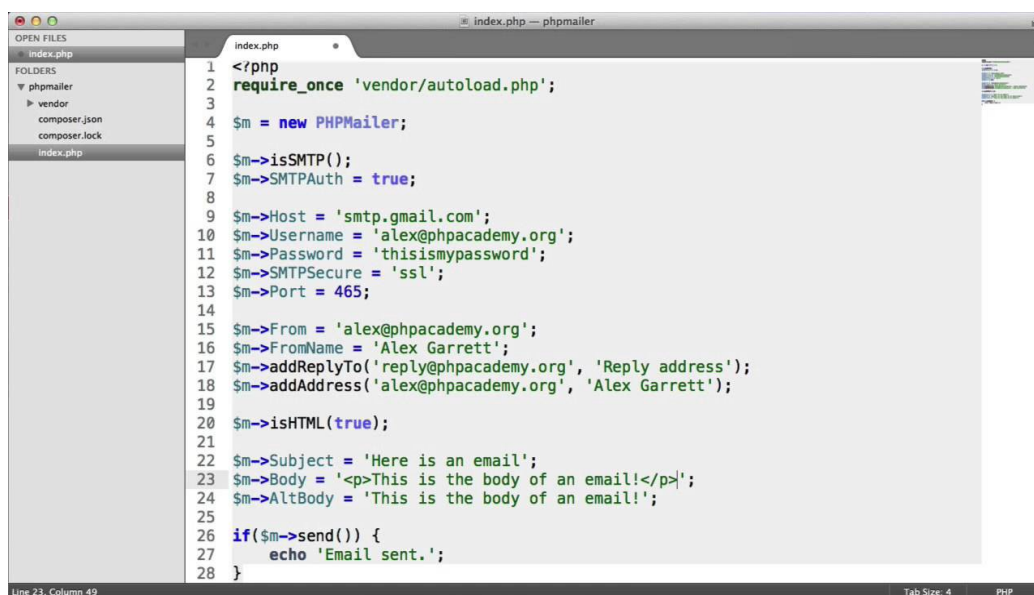
Notre Cahier de charge mentionne le fait qu'en cas de retard de retour de documents , ou en cas de vol , des mails automatiques s'enverront a l'abonné pour le prévenir , et le rappeler , nous proposons ces librairies pouvant effectuer ce genre d'actions , et offrant ainsi la sécurité , et l'anonymat

Swift mailer

- Swift Mailer s'intègre dans n'importe quelle application web écrite en PHP5, offrant une approche objet élégante et flexible pour envoyer des mails avec un multitude de fonctions : envoyer des emails en utilisant un serveur SMTP, SendMail , Postfix, ou n'importe quelle implémentation de transport de votre choix.
 - Supporte les serveurs qui requierent un user/mot de passe et/ou l'encryptage.
 - Protège contre les injections de headers sans détériorer le contenu des headers d'origine.
 - Envoi des mails multipartie/HTML compatible MIME, utilise des plugins orientés événements pour customiser la librairie.
- Maîtrise les grosses pièces jointes et les images en ligne/attachées avec une utilisation mémoire peu gourmande.

PHPMailer

- Classe de transfert d'email pour PHP pleine de fonctions.
- Les fonctions de PHPMailer : Supporte les emails signés digitalement avec le cryptage S/MIME,
 - supporte les emails avec de multiple destinataires, CC, BCC et REPLY-TO,
 - supporte les emails en HTML et en texte brut, les images embarquées, les emails multipartie/alternatif pour les clients qui ne lisent pas les mails HTML, les serveurs SMTP redondants, l'authentification SMTP.
- Elle a été testée sur de multiples serveurs SMTP : SendMail, qmail, Gmail, Imail, Exchange, etc...

A screenshot of a code editor window titled 'index.php - phpmailer'. The editor shows the code for index.php, which uses PHPMailer to send an email. The code is as follows:

```
1 <?php
2 require_once 'vendor/autoload.php';
3
4 $m = new PHPMailer;
5
6 $m->isSMTP();
7 $m->SMTPAuth = true;
8
9 $m->Host = 'smtp.gmail.com';
10 $m->Username = 'alex@phpacademy.org';
11 $m->Password = 'thisismypassword';
12 $m->SMTPSecure = 'ssl';
13 $m->Port = 465;
14
15 $m->From = 'alex@phpacademy.org';
16 $m->FromName = 'Alex Garrett';
17 $m->addReplyTo('reply@phpacademy.org', 'Reply address');
18 $m->addAddress('alex@phpacademy.org', 'Alex Garrett');
19
20 $m->isHTML(true);
21
22 $m->Subject = 'Here is an email';
23 $m->Body = '<p>This is the body of an email!</p>';
24 $m->AltBody = 'This is the body of an email!';
25
26 if($m->send()) {
27     echo 'Email sent.';
28 }
```

The left sidebar shows the file explorer with 'index.php' selected. The status bar at the bottom indicates 'Line 23, Column 49', 'Tab Size: 4', and 'PHP'.

e) Librairie pour les formulaires :

Dans notre projet , nous serons amenés à manipuler les formulaires de façon très fréquente , nous avons donc pensé à deux bibliothèques pouvant faciliter la manipulation des formulaires , d'une manière plus automatisée , pour éviter la perte de temps , et les spams

- **Securimage**

Un CAPTCHA Open Source et gratuit pour générer des images complexes et des codes CAPTCHA pour protéger les formulaires contre le spam et les abus.

- Il peut être facilement ajouté à vos formulaires sur votre site pour vous protéger des robots de spam.

- **PhpObjectForms**

phpObjectForms est une bibliothèque entièrement orientée objet, hautement personnalisable, facile à étendre, pour créer et manipuler des formulaires HTML sans écrire de code HTML.

- Les fonctions principales :
 - le support de tous les éléments standards des formulaires, l
 - la validation des saisies **côté serveur** basée sur les expressions régulières,
- la pré-validation côté client en JavaScript optionnelle, le support pour les formulaires basés sur des templates.
- La présentation du formulaire de sortie est entièrement gérée par CSS, vous pouvez donc facilement l'améliorer sans écrire de code mais uniquement en changeant la feuille de style.

f) Html CSS :

Vu notre forte utilisation de l'html/css , nous aurons encore une fois besoin d'un outil permettant d'automatiser ça , notamment au niveau du CSS , vu qu'on sera amené à raffiner notre site web , pour qu'il est au moins une bonne présentation , Ainsi qu'au niveau du HTML , qui aura sûrement besoin d'une purification , et de filtrage

HTML Purifier

HTML Purifier est une bibliothèque de filtres conforme aux standards HTML écrite en PHP.

- HTML purifier ne se contentera pas de supprimer le code malicieux (mieux connu comme XSS) avec une liste blanche permissive et hautement audité, il vérifiera aussi que vos documents soient compatibles avec les standards, quelque chose de seulement réalisable avec une connaissance complète des spécifications du W3C.

GeSHi - Generic Syntax Highlighter

GeSHi est un colorateur syntaxique pour le HTML, écrit en PHP.

- Typiquement, vous entrez la source et le nom du langage que vous voulez traiter, GeSHi retourne le résultat syntaxiquement coloré.
 - Mais il ne s'arrête pas là, GeSHi possède plusieurs fonctions puissantes et uniques incluant :
 - la possibilité de changer le style de n'importe quel élément coloré à la volée,
 - l'utilisation de classes CSS pour réduire la charge de sortie produite (GeSHi peut aussi produire une feuille de style pour être utilisée sur un langage à la volée),

g) Librairie pour image media fichier :

Nous aurons besoin de manipuler les images , les redimensionner , notamment pour les couvertures de livres , ou encore les icones d'authentification , ou encore d'emprunt ou l'abonné pourra cliquer pour y accéder .

- **PHP Thumb**

Une librairie de manipulation d'images légère qui vise la génération de miniatures.

- Elle donne la possibilité de redimensionner par la hauteur, la largeur, un pourcentage, de créer des recadrages personnalisés, ou de calculer les recadrages depuis le centre, de tourner les images.
- vous pouvez aussi ajouter facilement des fonctionnalités à la librairie par le biais de plugins.
- La librairie donne aussi la possibilité de créer de multiples manipulations par instances (aussi connu sous le nom de «chaining »), sans avoir besoin de sauvegarder et réinitialiser la classe à chaque manipulation.

- **class.upload.php**

Ce script PHP upload les fichiers et manipule les images facilement. Le script parfait pour générer des miniatures ou créer une galerie photo. Il peut convertir, redimensionner et travailler sur les images uploadées de plusieurs manières, ajouter des libellés, ajouter un filigrane et de la réflexion et d'autres fonctions d'édition. Vous pouvez les utiliser pour uploader des fichiers à travers un formulaire HTML, un uploader Flash, ou un fichier local.

- **getID3()**

Un script PHP qui extrait les informations utiles, (comme les tags ID3, le bit-rate, la durée, etc...) de fichiers mp3 et d'autres formats multimédia (Ogg, WMA, WMV, ASF, WAV, AVI, AAC, VQF, FLAC, MusePack, Real, QuickTime, Monkey's Audio, MIDI et plus).



h) Java Script :

Cette librairie nous permettra d'avoir plus ou moins quelques facilités à injecter du code java script dans du php

PHPLiveX

- PHPLiveX est une librairie PHP-JavaScript légère qui vous aide à intégrer de l'AJAX dans vos scripts PHP. Sans recharger votre page, vous pouvez appeler vos fonctions dans la même page.
- Soumettre vos formulaires à une URL et envoyer une requête à une autre page avec n'importe quel type de donnée.

I) Microformat :

Une librairie qui pourra nous être utile dans le cas d'extraction de microformat de notre page web, toutefois, on n'est pas sûr de l'utiliser, mais nous l'avons quand même proposée.

iCalcreator

iCalcreator est une classe de gestion PHP pour formater des fichiers iCal pour les systèmes qui ne sont pas des calendriers comme les CMS, les projets de système de gestion, et les autres applications capables de créer des informations calendaires.

- iCalcreator crée, parse, édite et sélectionne les calendriers et les composants de calendrier.

Hkit

- Un outil simple en PHP5 pour extraire les microformats communs d'une page.
- La page peut appeler une chaîne de caractères ou une URL, et le résultat est retourné en tant que tableau PHP.
 - hKit utilise SimpleXML pour parser et requiert donc PHP5.



j) Flux RSS

Une librairie pour gérer les flux RSS , ainsi que la traduction d'encodage des caractères

SimplePie

Une classe PHP Open Source qui fournit une API facile à utiliser pour gérer tout le travail rébarbatif lors de la récupération, la mise en cache, le parsing, la normalisation des données entre les flux RSS et Atom,

- gérant aussi la traduction d'encodage des caractères, et en nettoyant les données qui en résultent.

K) Sécurité

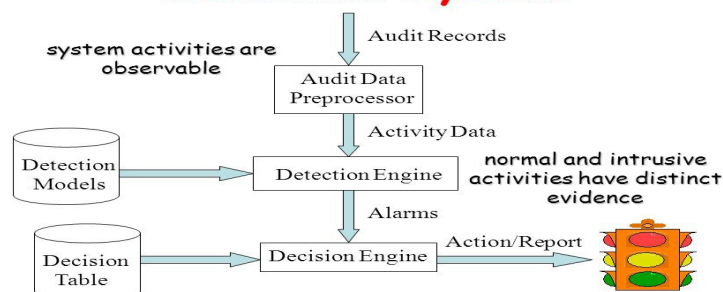
Notre site web n'aura pas vraiment besoin d'une sécurité de niveau élevé dans le sens où il ne contiendra pas des informations sensibles comme des numéros de carte bancaire , ou autres , mais nous avons quand même besoin d'une librairie pouvant nous garantir une bonne sécurité pour éviter les injections SQL et tout ce qui va avec , en les détectant .

PHP Intrusion Detection System (PHPIDS)

Une couche de sécurité, simple à utiliser, bien structurée rapide et à l'état de l'art pour les applications basées sur PHP.

- L'IDS ne déshabille, ni n'assainit, ni ne filtre une saisie malveillante, il reconnaît simplement quand un attaquant essaye de casser votre site et réagit exactement comme vous le souhaitez.
- Actuellement PHPIDS détecte toutes les sortes de XSS, injections SQL, injections des headers, traversée de répertoire, RFE/LFI, attaques DOS et LDAP.
- Basé sur un jeu de filtres approuvé et lourdement évalué il juge et donne à n'importe quelle attaque une évaluation d'impact numérique qui facilite le choix de l'action à suivre après la tentative de piratage.
- Cela pourrait s'étendre de l'enregistrement simple à l'émission d'un mail de secours à l'équipe de développement, l'affichage d'un message d'alerte pour l'attaquant ou même la fin de la session de l'utilisateur.

Components of Intrusion Detection System

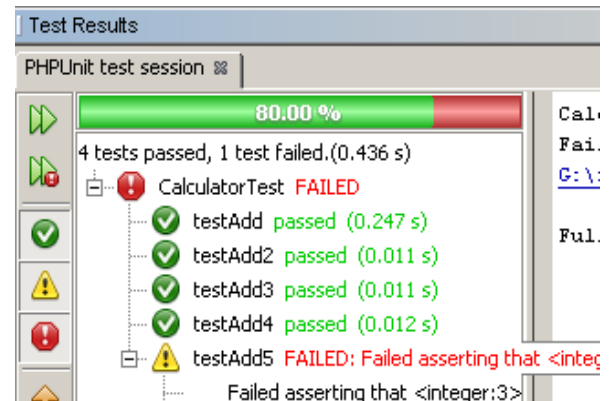
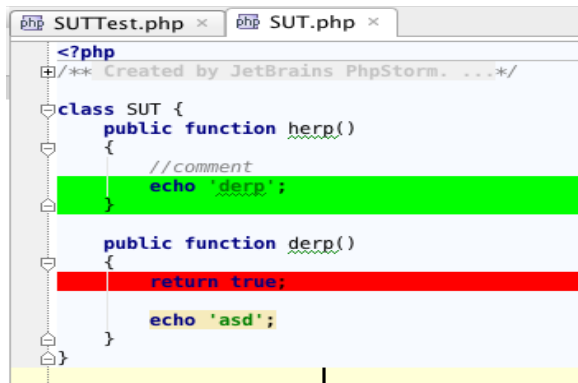


L) Tests

Bien évidemment , a la fin de notre implémentation nous devrons tester notre code , pour voir et couvrir tout les cas possibles , nous avons donc penser a PHPUnit , qui fonctionne pratiquement de la meme maniere que Junit , pour tester notre code

PHPUnit

PHPUnit est un membre de la famille xUnit des Frameworks de test et fournit aussi bien un Framework qui permet l'écriture des tests facilement et l'exécution des fonctions de test que l'analyse de leurs résultats.



M) Debugage :

Allant dans la continuité de PHPUnit , nous aurons aussi besoin de Debugger en cas de problemes , et donc une librairie nous permettant cela comme c'est le cas pour PHP Debug

PHP Debug

Fournit de l'assistance dans le débuge de code PHP, par les traces, l'affichage des variables, le chronométrage des process, les fichiers inclus, les requêtes exécutées.

- Ces informations sont rassemblées pendant l'exécution du script et sont affichées à la fin du script (dans une belle div flottante ou une table HTML) pour qu'elles soient lues et utilisées à nimporte quel moment

II) Les diagrammes de cas d'utilisation :

Grâce a ce diagramme , nous allons détailler les principaux acteurs de la mediatheque et leurs rôles , dans un premier temps , on va associer un seul diagramme de cas d'utilisation a chaque acteur afin de bien visualiser son rôle , puis ensuite , nous allons faire un diagramme de cas d'utilisation global pour visualiser la complémentarité des rôles de chaque acteur dans un seul diagramme.

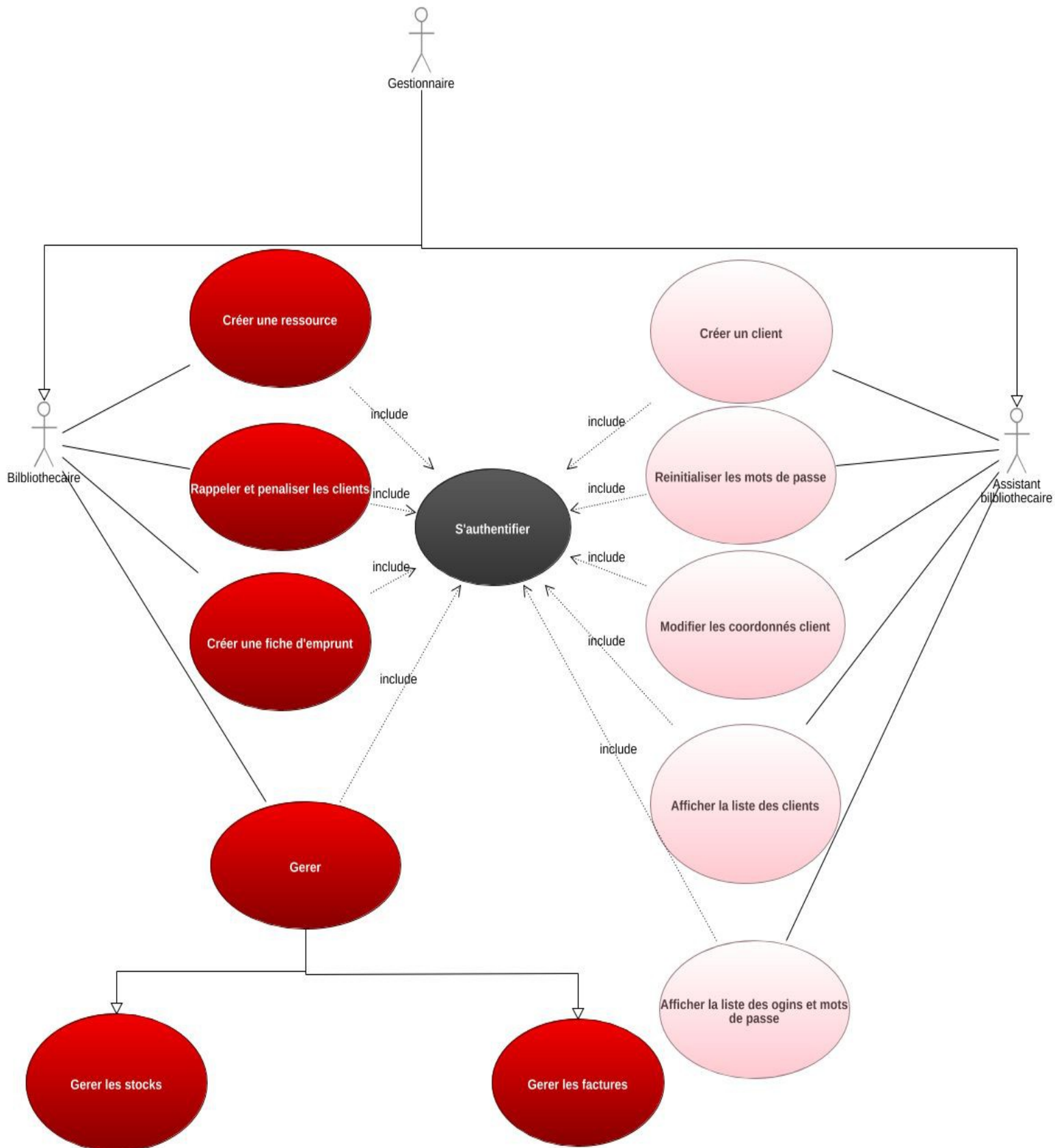
Le système de gestion de la médiathèque est accessible par 4 acteurs :

- Le gestionnaire
- l'administrateur
- l'abonné
- l'usager

ou :

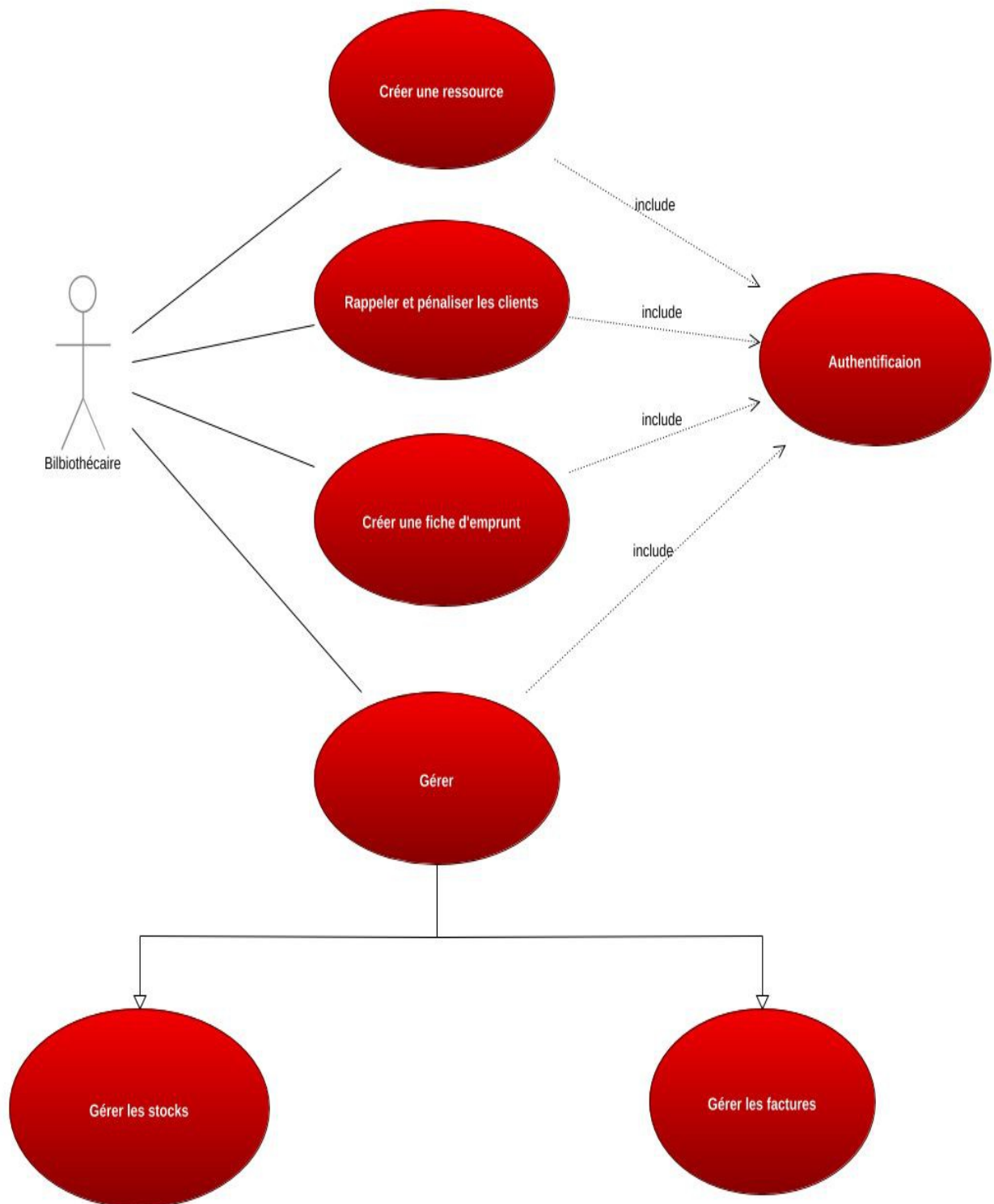
- le gestionnaire est l'employé de la mediatheque se divisant en bilbliothecaire , et son assistant
- L'administrateur jouera le rôle du serveur
- L'abonné et l'usager sont tout deux des clients de la mediatheque

1) Le Gestionnaire :



Comme on peut le voir , les rôles du gestionnaire se divisent entre le bibliothécaire , et l'assistant , commençons par le bibliothécaire :

a) le bibliothécaire :



Commentaires sur le diagramme :

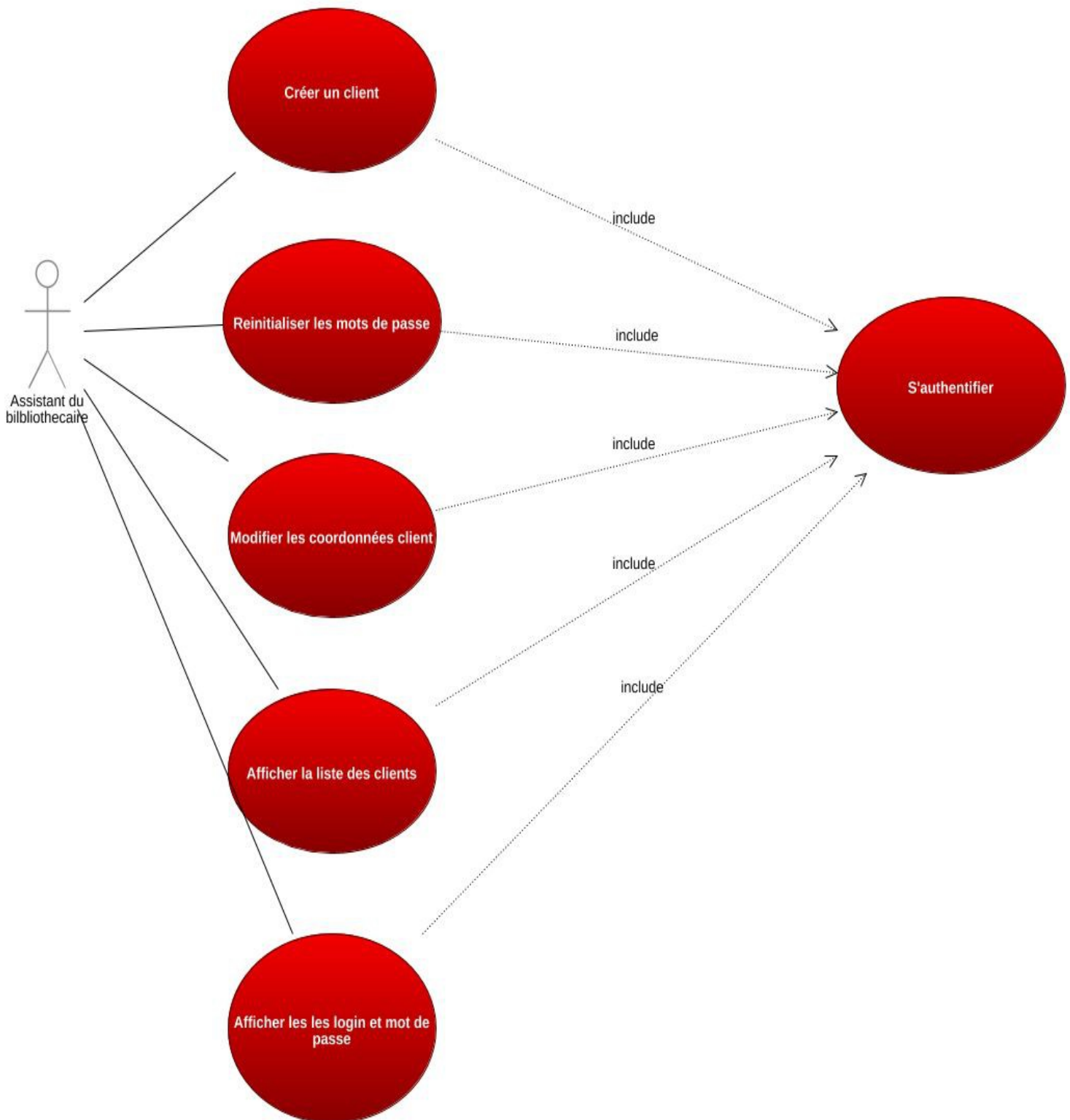
Dans ce diagramme , on illustre les différents rôles du bibliothécaire , qui est lui une variante du gestionnaire , en effet , le bibliothécaire aura un rôle principalement fixé sur la gestion des ressources de la mediatheque il pourra donc :

- Créer ou ajouter une nouvelle ressource a la bibliothèque , que ça soit un document , media , livre , vidéo ou autre .
- Rappeler les clients les plus en retard en leur envoyant des lettres de pénalités , et en leur donnant des délai pour le rendu des documents , dans le cas échéant , il pourra donc les pénaliser
- Gérer les stocks , voir les documents restant pour ainsi gerer les pertes générés par la mediatheque .
- Gérer les factures des clients , ce qui inclus le fait de gerer les clients en retard , et les pénaliser .

Plus globalement :

Un bibliothécaire peut créer des ressources et des fiches d'emprunt et les ajoutées dans le site de la médiathèque, il gère aussi les stocks et les factures ainsi qu'il peut pénaliser les clients en cas d'emprunt retardé ou non rendu. L'accès à ces données ne peut se faire qu'après une authentification de la part du bibliothécaire.

b) l'assistant du bilbiothecaire :



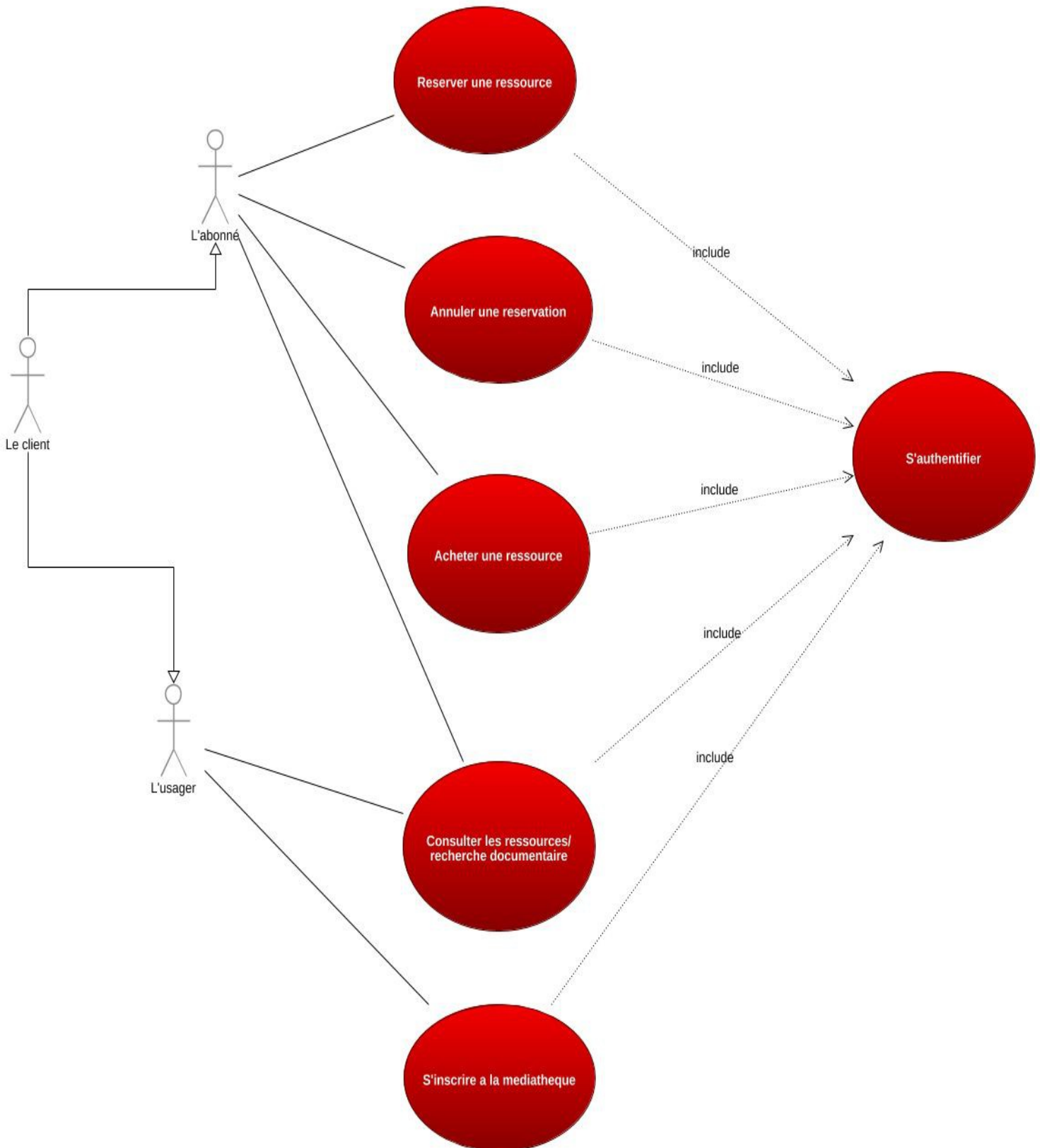
Commentaires sur le diagramme :

Precedement , nous avons vu que le bibliothécaire avait pour rôle , la gestion des ressources , maintenant on passe a son assistant qui lui a pour rôle la gestion des clients , il peut donc :

- Créer de nouveaux clients qui lui donne leurs informations pour la creation de comptes utilisateur de la mediatheque
- Il peut aussi réinitialiser les mots de passes des comptes des abonnés en cas de perte de mot de passe , ou en cas de changement de mot de passe .
- Modifier les coordonnées des abonnés en cas de changement de numero de telephone d'adresse mail ou autre
- Le droit a la consultation de la liste des clients et toutes les relations relatives à leurs comptes (Login, Mot de passe)

2) L'abonné et l'utilisateur :

a) Le Client : Abonné + Usager :



Commentaires sur le diagramme :

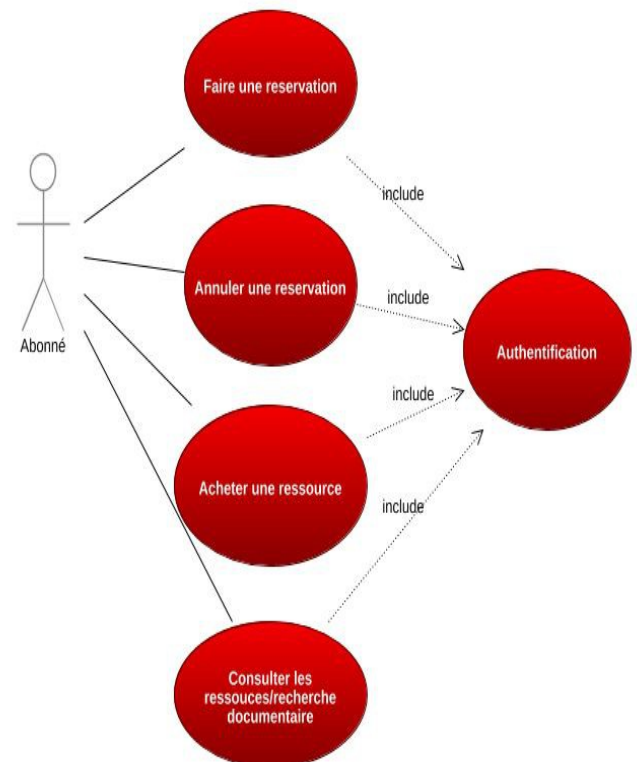
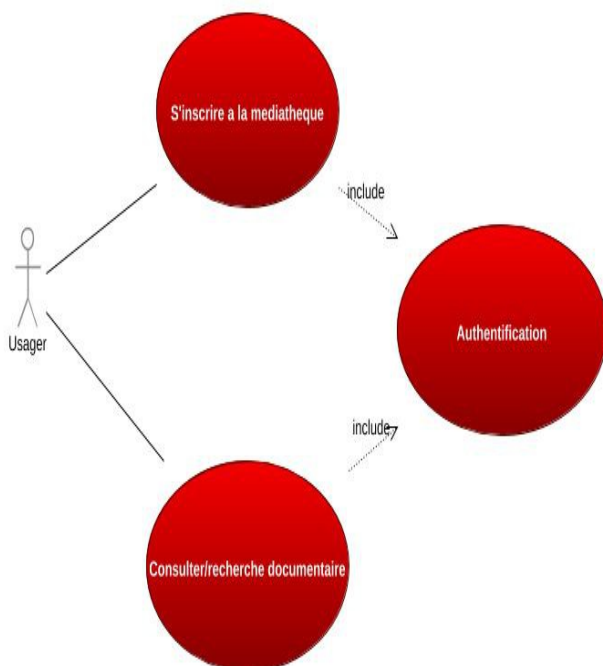
Un client de la mediatheque peut être soit un usager , soit un abonné , ce dernier a le droit a l'utilisation de toutes les fonctionnalités de la mediathèque , qui sont notamment illustrés dans le diagramme de cas d'utilisation :

- Il peut réserver une ressource si disponible
- Annuler la réservation de cette dernière si il veut
- Acheter une ressource si elle figure dans les ressources a vendre
- Utiliser les outils de consultations de ressources , ainsi que l'outil de recherche documentaire se trouvant a l'acueil du site de la mediatheque

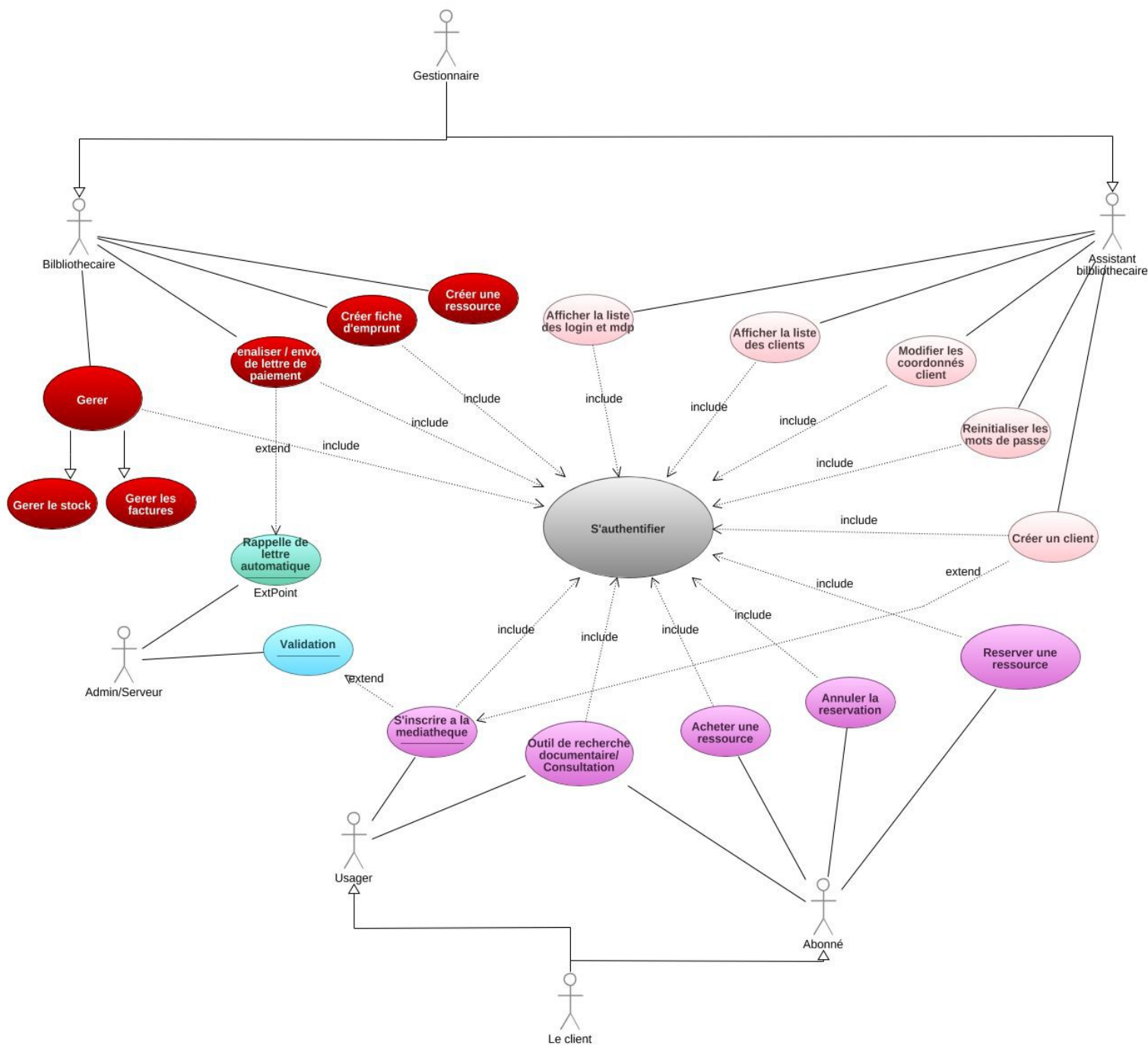
Concernant l'usager , ses fonctionnalités sont restreintes , vu qu'il doit s'inscrire pour beneficier des fonctionnalités de l'abonn é , il a donc seulement les deux droits suivants :

- -S'inscrire a la mediatheque , et devenir donc un abonné de la mediatheque et ainsi beneficier de toutes les fonctionnalités
- -Avoir accès a l'outil de recherche documentaire se trouvant a l'accueil du site de la mediatheque , et consulter les catalogues , sans pour autant pouvoir reserver , ou encore acheter .

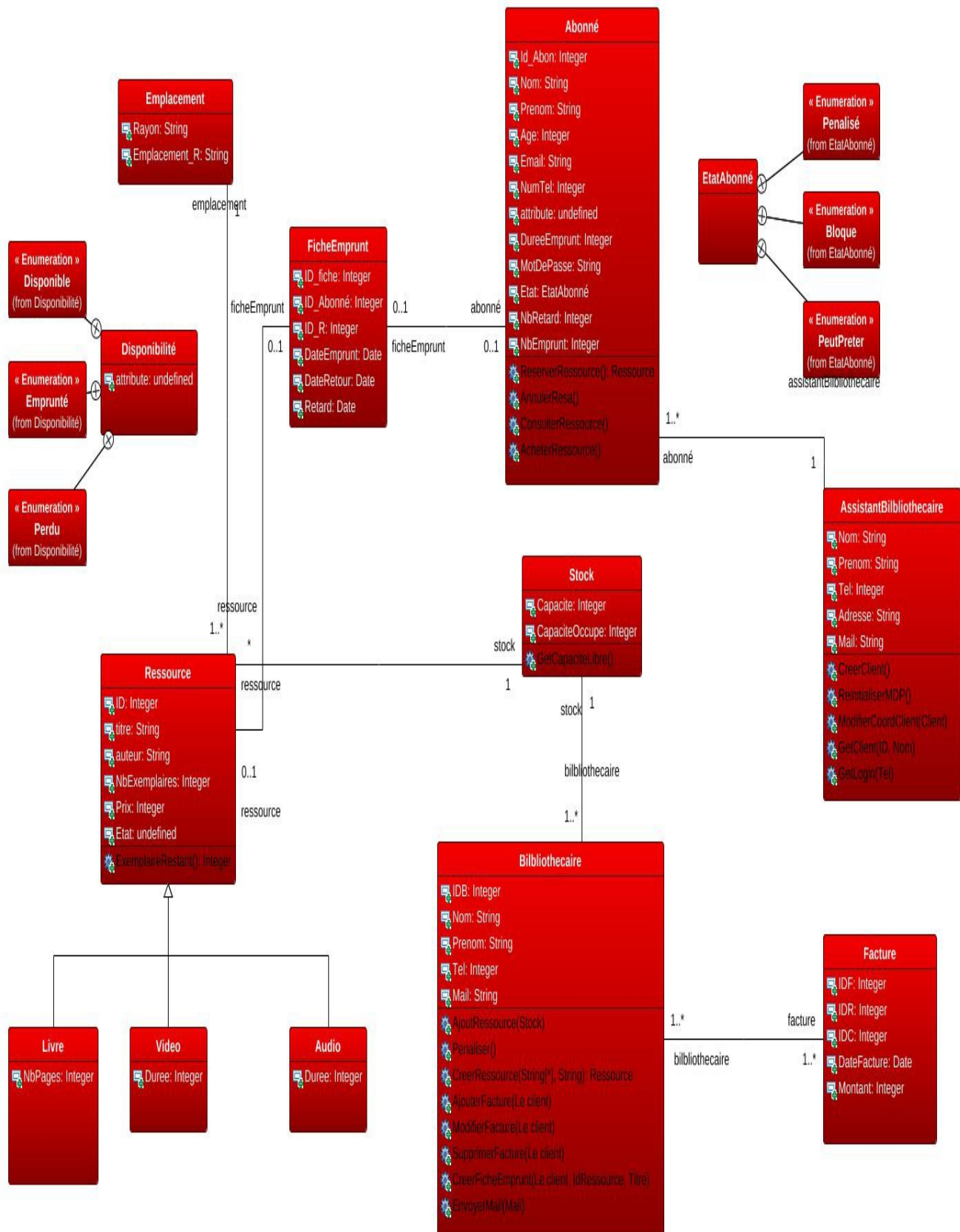
b) Diagramme de l'usager + diagramme de l'abonné :



3) Diagramme de cas d'utilisation de la mediatheque complet :



III) Diagramme de classe :



Comme le diagramme le montre ci-dessus on a constaté 11 classes :

- Abonné
- AssistantBilbliothecaire
- Bibliothécaire
- Ressources
- Audio
- Vidéo
- Livre
- Stock
- FicheEmprunt
- Emplacement
- Facture

Chaque table est caractérisée par des attributs et des opérations :

1) Classe Abonné :

a) Attributs :

IdAbonné, Nom, Prénom, Age, Email, Adresse, NumTel, MotDePasse, Etat, NbrRetards, DuréeEmprunt, NombreEmprunt.

b) Opérations :

- Réserver une ressource : Pour réserver une ressource déjà existante dans la mediatheque
- Acheter une ressource : Pour acheter une ressource , que ça soit une video , un livre ou un audio , si et seulement si elle est a vendre
- Consulter une ressource : Pour consulter les ressources et le catalogue de la bilbliothèque , et pour accéder a la recherche documentaire
- Annuler une réservation : Pour annuler une reservation déjà effectuée

2) Classe AssistantBilbliothecaire :

a) Attributs :

IdSec, Nom, Prénom, Tel, Adresse, Email.

b) Opérations :

- CreerClient : Pour créer un nouveau client , qui passera de « Usager » a « abonné »
- ,RéinitialiserMotDePasse : Pour changer ou reinitialiser le mot de passe d'un abonné
- ModifierCoordonnées ; Pour modifier les coordonnées d'un client donné , (Numero de telephone , adresse , mail , ou autre)

- GetClient : Pour consulter les informations sur un client
- GetLoginMDP. : Pour consulter les mots de passe et les login d'un client donné

3) Classe Bibliothécaire :

a) Attributs :

IdB, Nom, Prénom, Tel, Email, Adresse.

b) Opérations :

- Ajouter une ressource au stock : Pour ajouter une nouvelle ressource au stock de la mediatheque
- Pénaliser un client ; Pour pénaliser un client en cas de retard ou autre
- Créer une ressource : Pour créer une nouvelle ressource non existante dans la mediatheque
- (Ajouter, Modifier, Supprimer) une facture ; Pour Editer la facture d'un client
- Créer une fiche d'emprunt : Pour créer une fiche d'emprunt avec toutes les informations du client
- Envoyer un mail à un client : Pour envoyer un mail de rappel au client qui se fera via le serveur/admin

4) Classe Ressource :

a) Attributs :

Id, Titre, Auteur, NbrExemplaire, NbrEmprunts, NbrExemplaire, Prix, Etat.

b) Opérations :

- Afficher les exemplaires restants : Pour voir le nombre d'exemplaire restant et consulter les exemplaires

5) Classe Livre :

a) Attributs :

NbrDePages.

6) Classe Audio :

a) Attributs :

Durée.

7) Classe Vidéo :

a) Attributs :

Durée.

8) Classe Stock :

a) Attributs :

Capacité, CapacitéOccupée.

b) Opérations :

- getCapacitéLibre : Qui nous renvoi la capacité restante dans le stock.

9) Classe FicheEmprunt :

a) Attributs :

Id_fiche, Id_Abonne, Id_ressource, Date d'emprunt, Date de retour, Retard.

10) Classe Emplacement :

a) Attributs :

Rayon, Emplacement de la ressource.

11) Classe Facture :

a) Attributs :

Id_Facture, Id_Client, Id_Ressource, Date_Facture, Montant, Règlement.

IV) Diagrammes de sequences et diagrammes d'etat :

Afin de décrire comment se déroulent les actions entre les acteurs et le système, on a établi des diagrammes de séquences qui nous permettront de visualiser l'enchaînement des actions dans le temps et pour montrer les interactions d'objets.

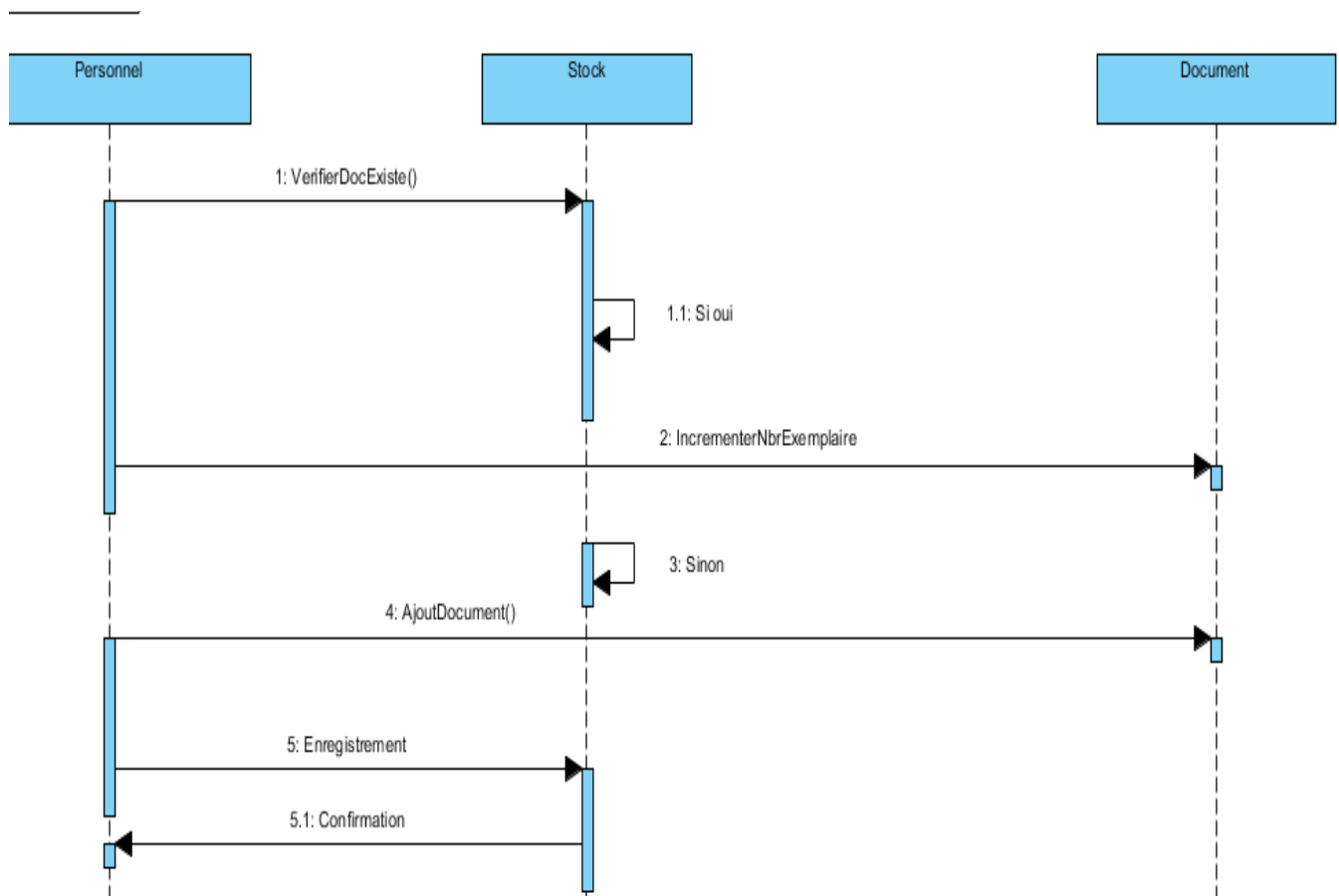
1) Ajouter une ressource au stock :

1^{ère} étape :

- Le personnel vérifie si la nouvelle ressource existe déjà dans le stock.
- Si la ressource existe alors le personnel l'ajoute et incrémente le nombre d'exemplaire de cette ressource.
- Si la ressource n'existe pas, le personnel crée une nouvelle ressource avec ses informations (Id,titre,auteur,NbrExemplaire,Prix), et l'ajoute à la liste des ressources.

2^{ème} étape :

Le personnel enregistre l'ajout de la ressource dans le système de la médiathèque.



2) Authentication :

1^{ère} étape :

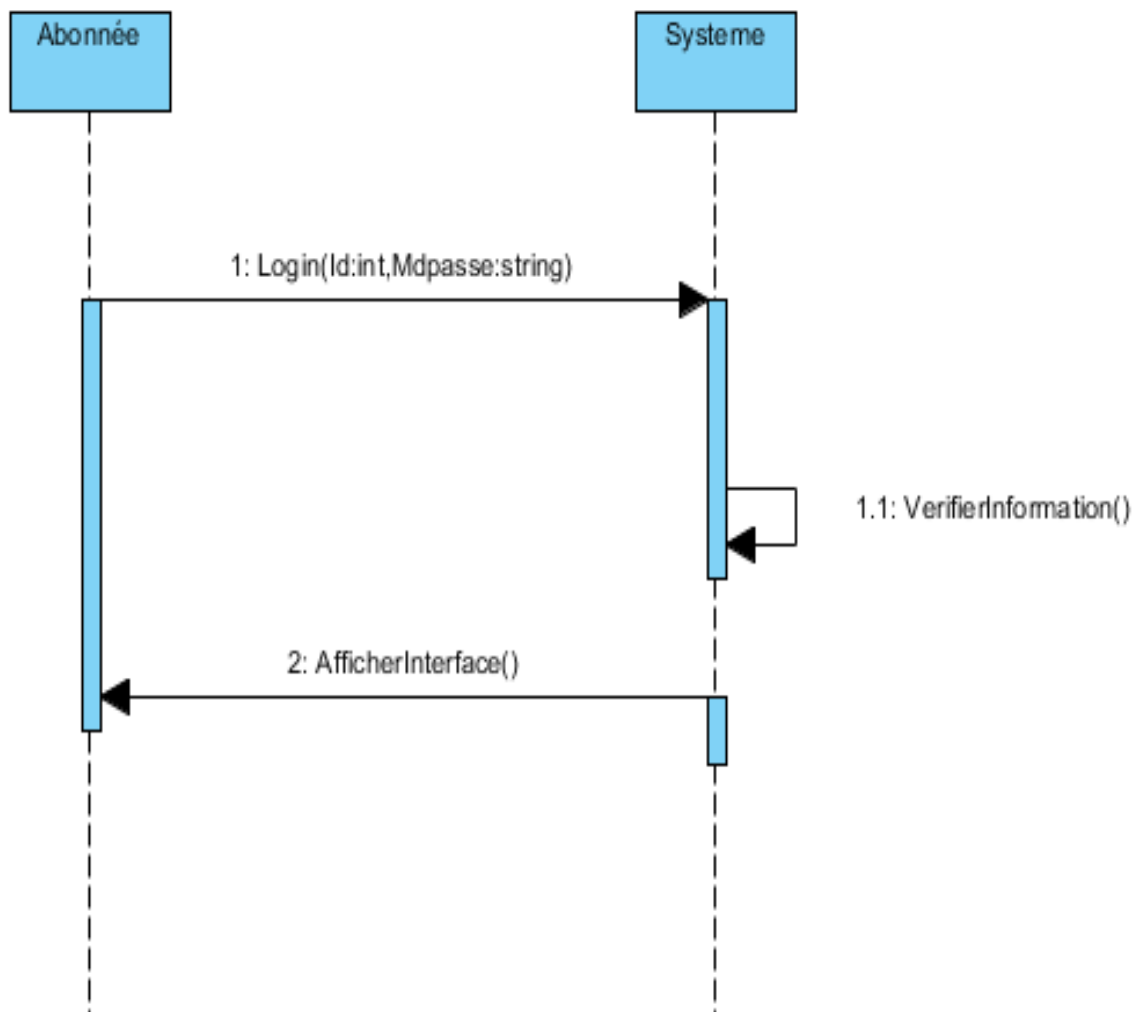
- L'abonnée saisi ses coordonnées (Identifiant, mot de passe) pour s'authentifier.

2^{ème} étape :

- Le système vérifie les informations saisies par l'utilisateur si elles sont correctes, si c'est le cas, le système valide l'authentification de l'abonné(e).

3^{ème} étape :

- Le système transfère l'abonné(e) à la page d'accueil du site web de la médiathèque.



3)Emprunter :

1^{ère} étape :

- L'abonné demande l'emprunt d'une ressource.

2^{ème} étape :

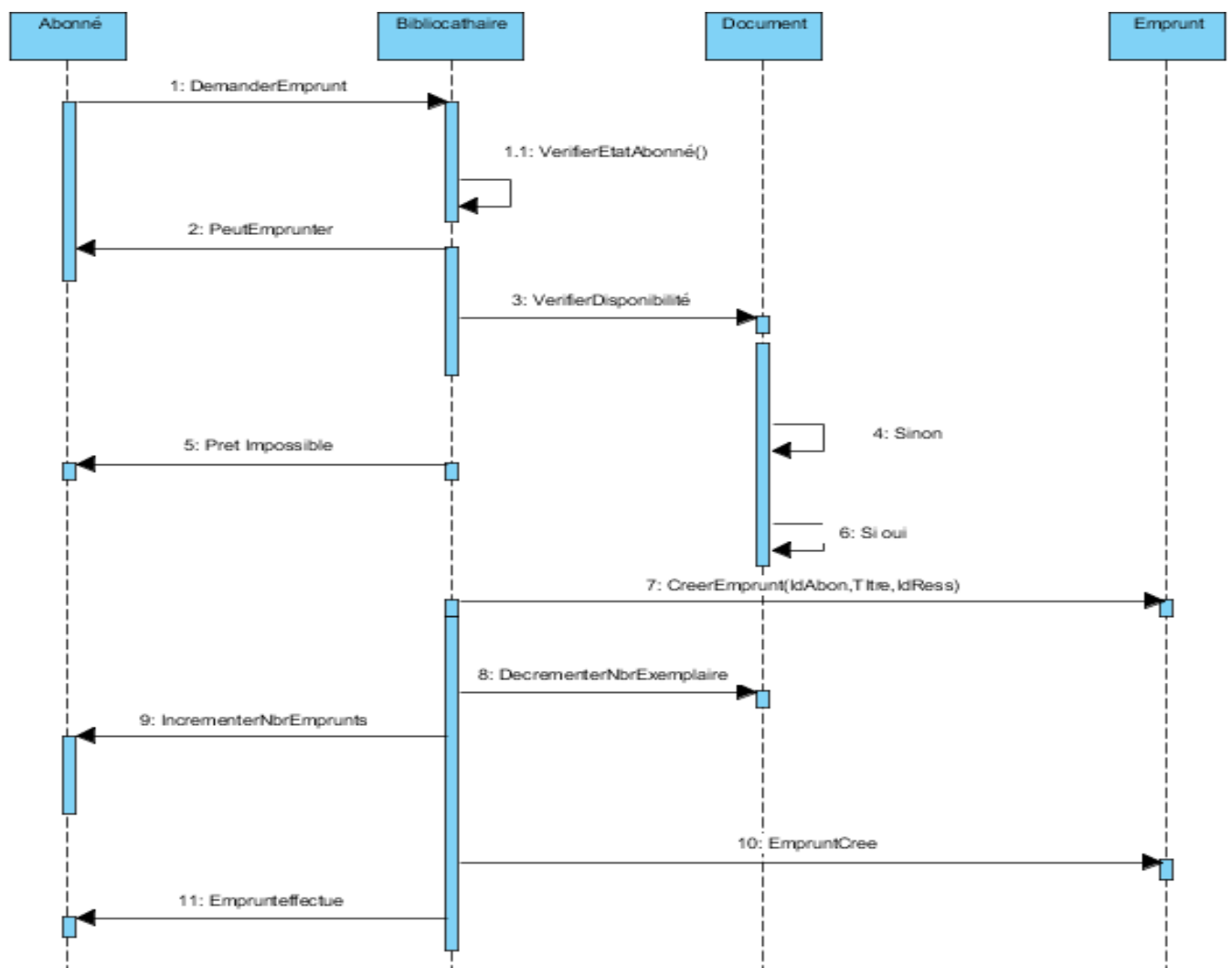
- Le système/Bibliothécaire vérifie si l'abonné(e) a le droit d'emprunter ou non, ensuite le système vérifie la disponibilité de la ressource. Si elle n'est pas disponible l'emprunt ne pourra pas aboutir sinon le bibliothécaire va créer une nouvelle fiche d'emprunt attribuée à l'abonné(e) et à la ressource choisie.

3^{ème} étape :

- Le Bibliothécaire va décrémenter le nombre d'exemplaire de cette ressource, et incrémentera le nombre d'emprunts de l'abonné.

4^{ème} étape :

- L'emprunt sera créé et validé.



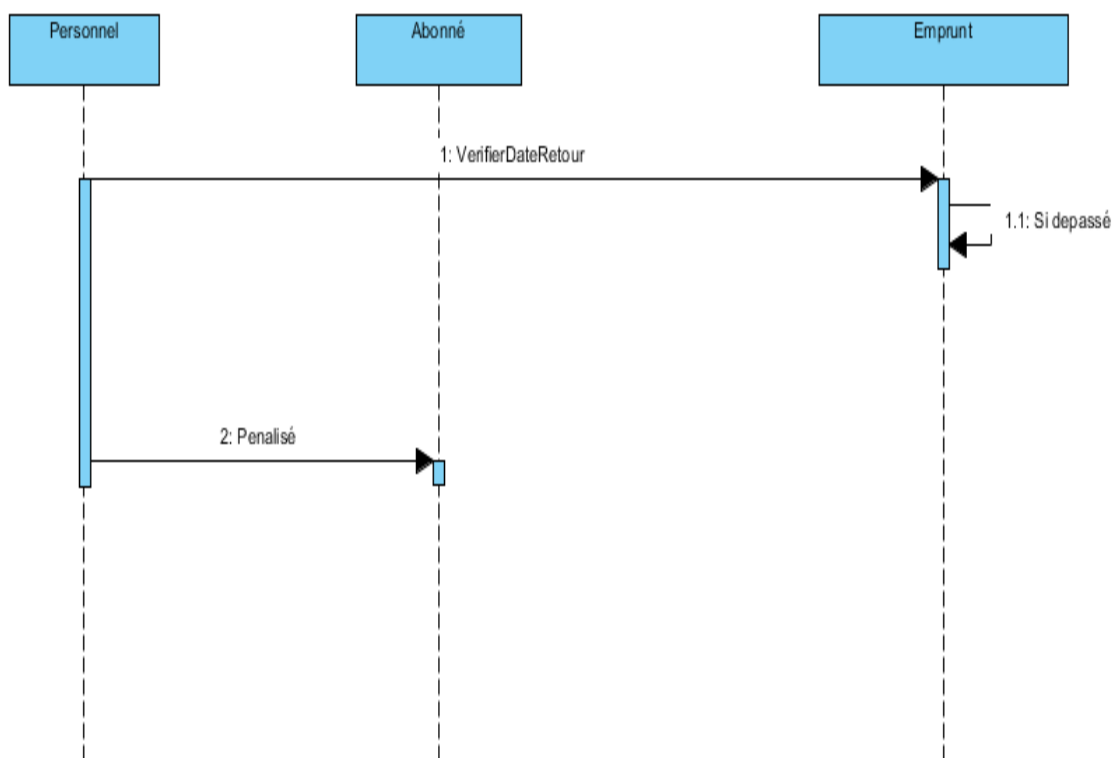
4)Retard d'emprunt :

1^{ère} étape :

- Le personnel vérifie si il y'a des documents qui devaient être rendu ce jour là mais qui ne le sont pas.

2^{ème} étape :

- Si la date de retour est dépassée, l'emprunteur sera pénalisé.



5) Retour d'emprunt :

1^{ère} étape :

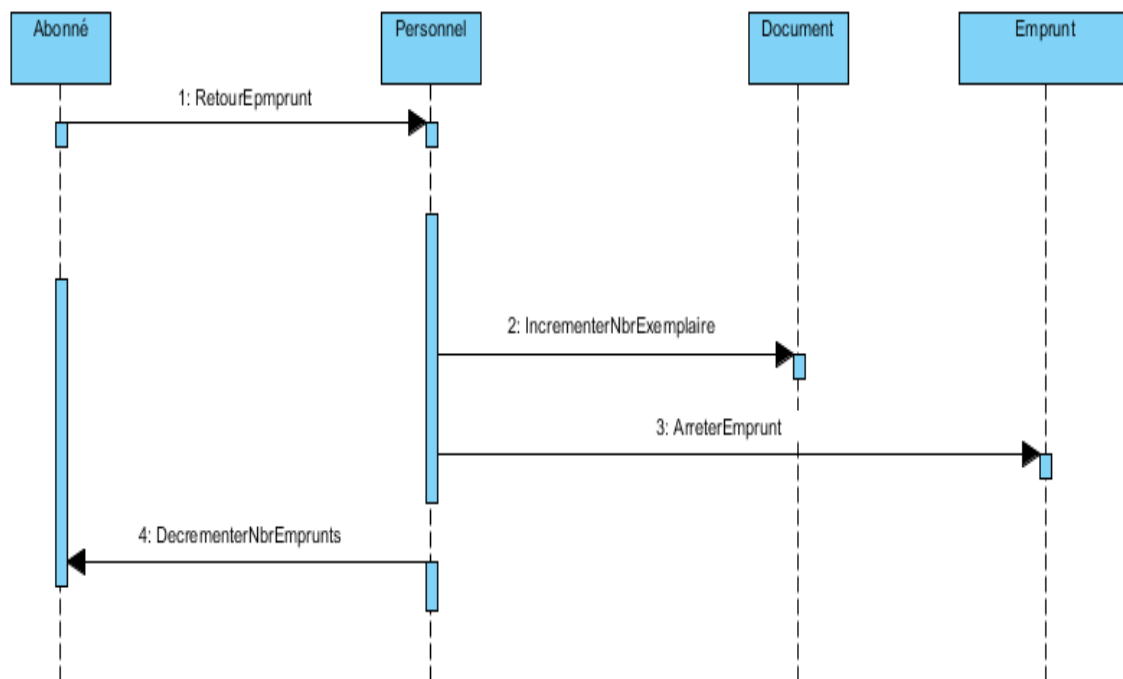
- L'abonné va rendre le document emprunté.

2^{ème} étape :

- Le personnel va enregistrer la ressource dans le stock, et incrémentera le nombre d'exemplaires de cette ressource.

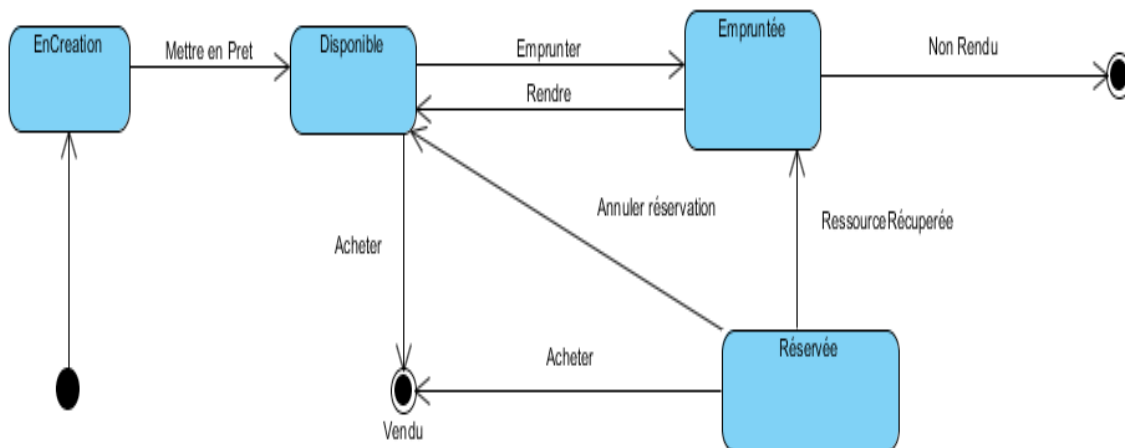
3^{ème} étape :

- Le personnel retire l'emprunt de l'abonné et décrémente son nombre d'emprunts.



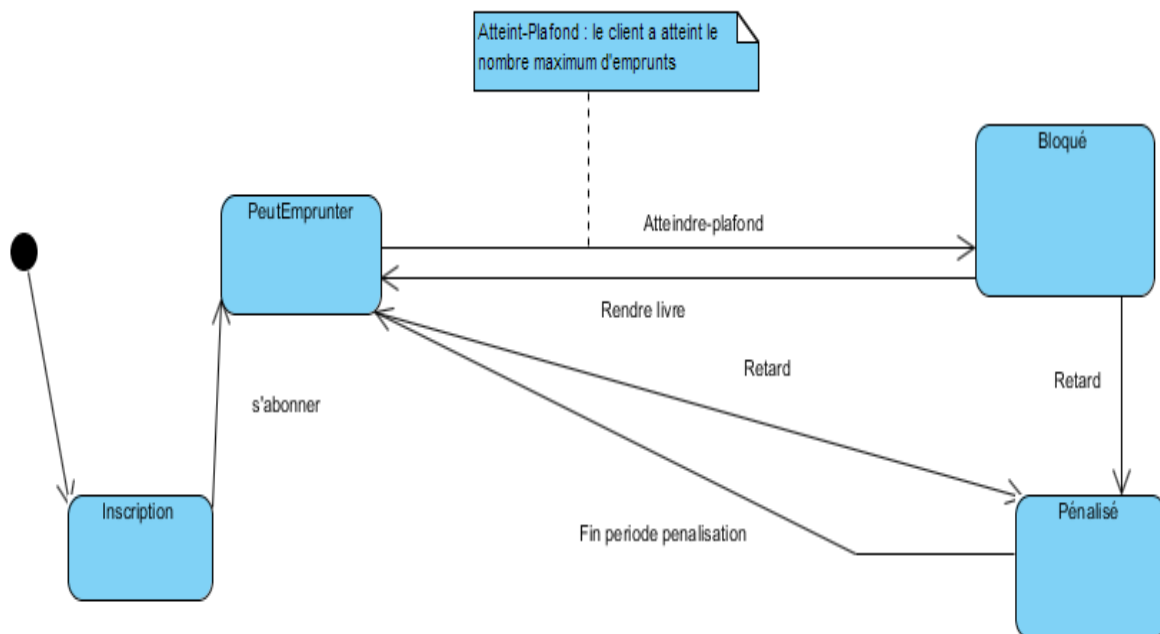
6)Etat de la ressource :

- La première étape consiste à créer une ressource et l'enregistrer dans la médiathèque.
- Ensuite on met cette ressource en disponibilité pour que les abonnés puissent l'emprunter.
- Si une personne emprunte ou réserve une ressource, elle passe à l'état emprunté ou réservé.
- Si une personne rend la ressource ou annule la réservation, elle revient à son état de base "Disponible".
- L'état final d'un emprunt est soit :
 - Vendu : si un abonné achète la ressource.
 - Perdu : si un abonné ne rend pas une ressource



7) Etat d'un client :

- Au début on crée un client après son inscription auprès du secrétariat de la médiathèque.
- Suite à son inscription, l'abonné aura le droit d'emprunter des ressources.
- Si le client atteint le nombre maximum des emprunts il passera à l'état de : "bloqué", et dès qu'il rend une ressource il pourra emprunter à nouveau.
- Si le client ne rend pas une ressource à sa date de retour, il aura une pénalisation de ne pas emprunter pour une période égale au nombre de jours de retard, et l'emprunt ne sera possible que quand cette période sera terminée.



V) Invariants :

Inv 1 :

- ***Le nombre d'emprunts maximum est 5 emprunt.***
 - en OCL : Context Abonnée
 - inv : EmpruntMaximum
 - *this.NombreEmprunts <= 5*

Inv 2 :

- ***La durée Maximum d'emprunt est de 1 mois:***
 - en OCL : Context Abonée
 - inv : Durée emprunt
 - *this.duréeEmprunt = 30;*

Inv 3 :

- ***Le prix d'une Ressource est positif***
 - en OCL : Context Ressource
 - inv : PrixRessource
 - *This.Prix >0;*

Inv 4 :

- ***Le nombre d'exemplaire d'une Ressource est positif***
 - en OCL : Context Ressource
 - inv : NombreExemplairePositif
 - *This.NombreExemplaire > 0;*

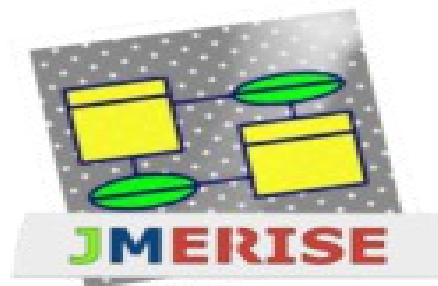
Inv 5 :

- ***Une Ressource a trois état***
 - en OCL : Context Ressource
 - inv : EtatRessource
 - *This.Etat = "Disponible" or his.Etat = "emprunté" or This.Etat="Perdu"*

VI) MCD & MLD:

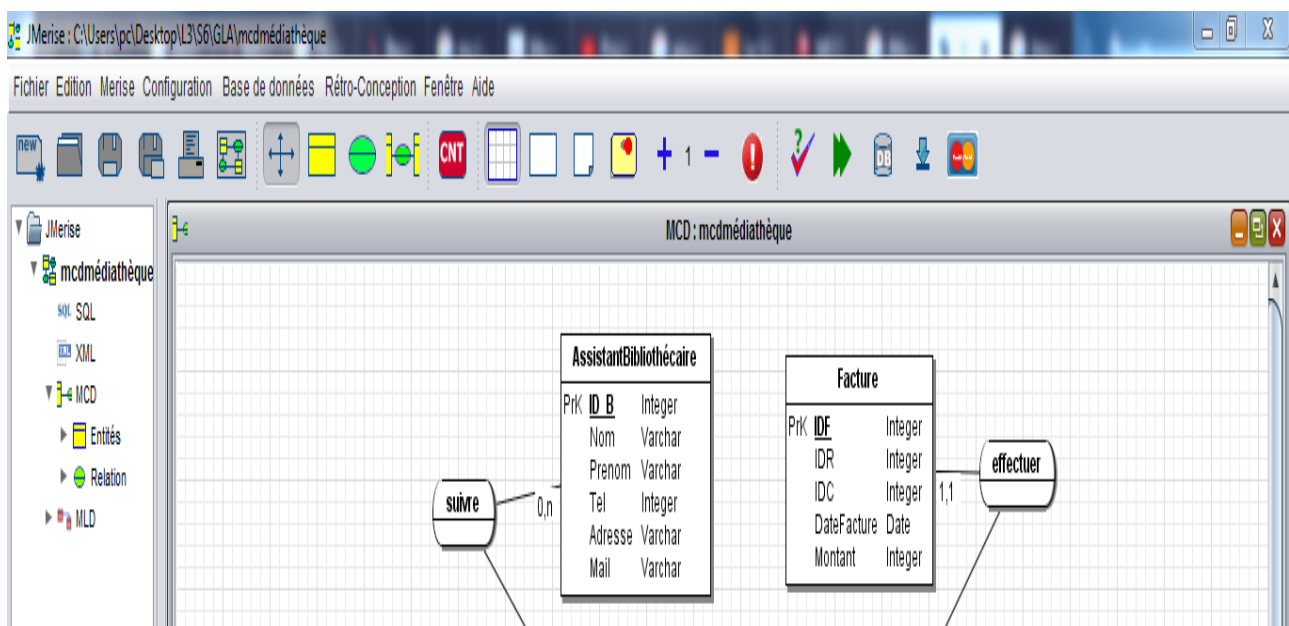
1) MCD :

Le Modèle Conceptuel des Données a pour but de décrire de façon formelle les données qui seront utilisées par le système d'information. Il décrit la sémantique c'est-à-dire le sens attaché à ces données et à leurs rapports et non à l'utilisation qui peut en être faite. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités et des relations entre elles.



Pour réaliser ce MCD, on a utilisé l'outil JMerise qui est un logiciel dédié à la modélisation des modèles conceptuels de données (MCD) pour Merise il permet la généralisation et la spécialisation des entités, la création des relations et des cardinalités ainsi que la généralisation des modèles logiques de données (MLD) et des scripts SQL...

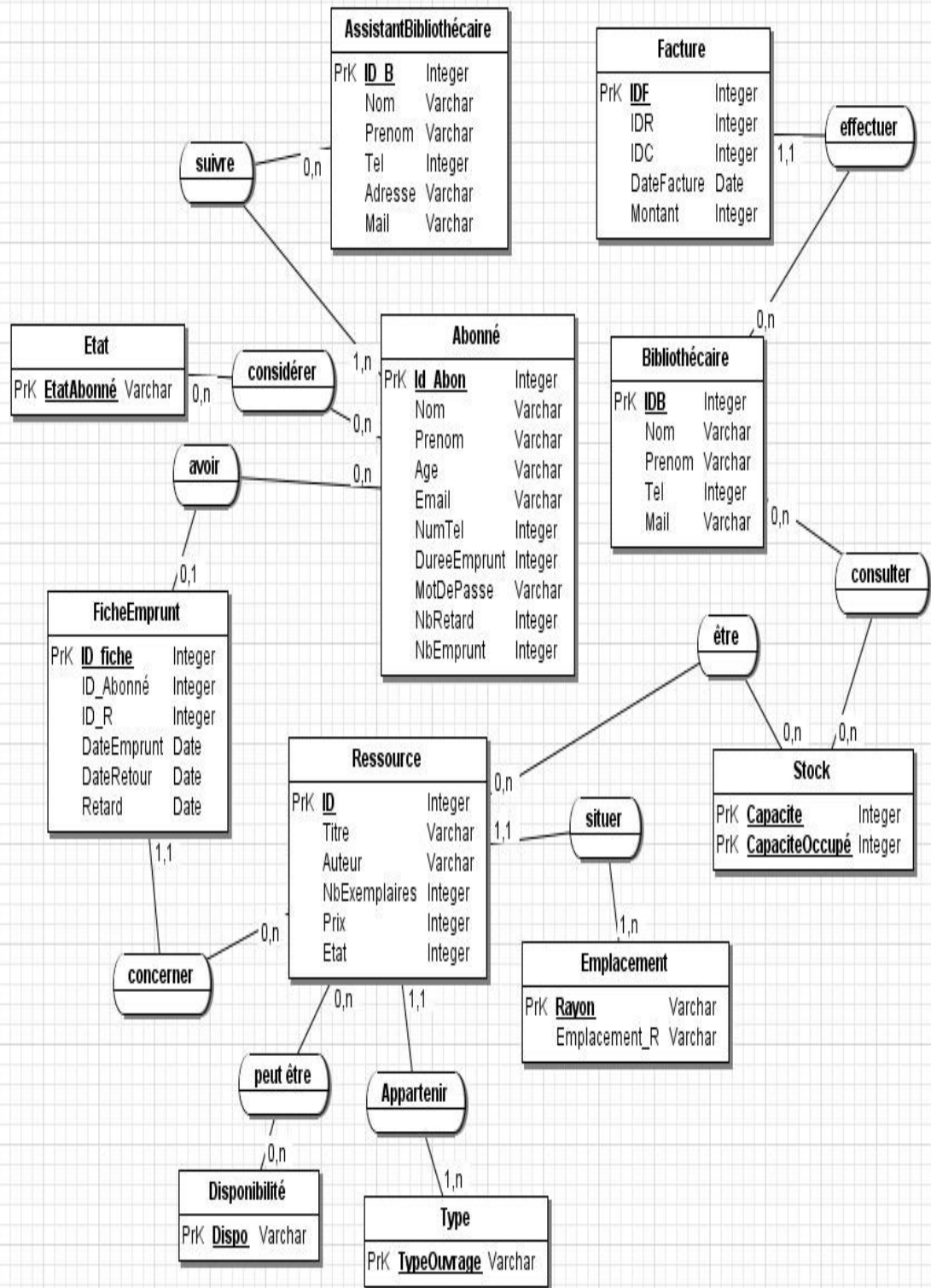
En modélisant les différentes entités entre elles, en veillant à créer les relations et les cardinalités nécessaires pour que le schéma puisse bien englober toutes les notions. Voici, un aperçu de l'outil JMerise utilisé :



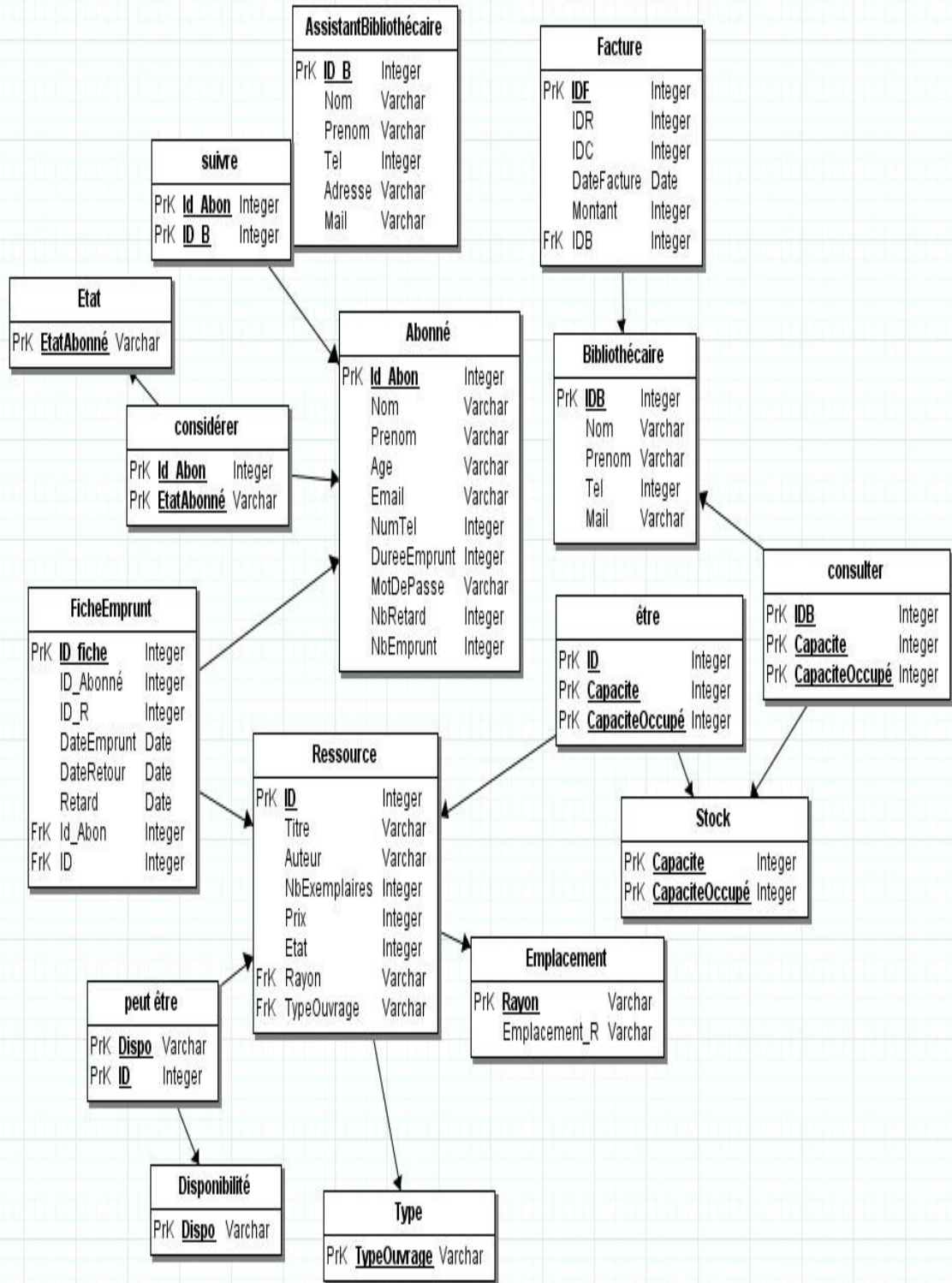
Ici, on retrouve le MCD créé et à travers lequel on peut générer par la suite le MLD et le script SQL qu'on utilisera dans la suite dans l'implémentation.

Désormais, MERISE permet aussi de modéliser l'héritage entre les entités. L'héritage a du sens lorsque plusieurs entités possèdent des propriétés similaires. Mais ici, on regroupe les différents types de ressources en une nouvelle entité qui s'appellera TypeOuvrage pour éviter l'héritage.

L'état d'un abonné on le modélise en une nouvelle entité nommée Etat et ayant comme attribut EtatAbonné, tout comme la disponibilité d'une ressource on le regroupe dans une nouvelle entité nommée Disponibilité et ayant comme attribut Dispo.



2) MLD :



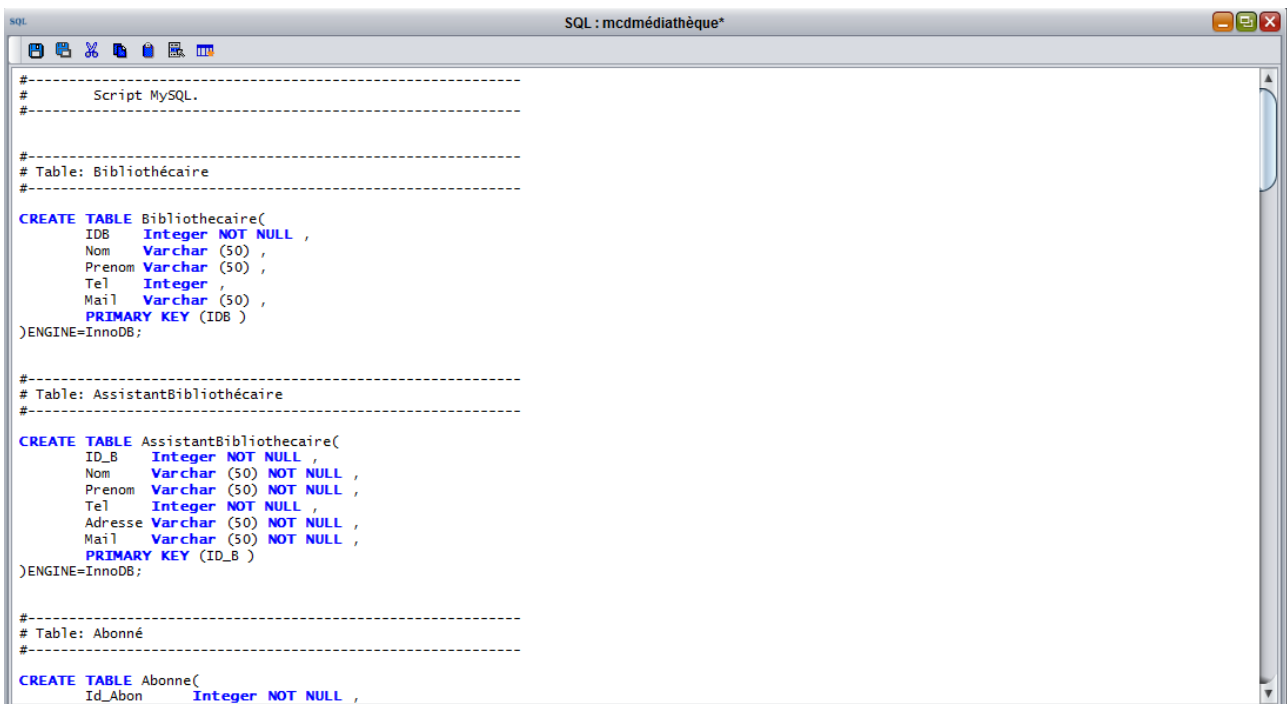
VII) Pseudo-code SQL :

Les règles de passage au SQL sont assez simples :

- chaque relation devient une table
- chaque attribut de la relation devient une colonne de la table correspondante
- chaque clef primaire devient une **PRIMARY KEY**
- chaque clef étrangère devient une **FOREIGN KEY**

En règle générale, toute entité du MCD devient une relation dont la clé est l'identifiant de cette entité. Chaque propriété de l'entité devient un attribut de la relation correspondante.

Voici ce que cela donnerait :



```
SQL : mcdmédiathèque*

#-----
#      Script MySQL.
#-----

# Table: Bibliothécaire
#-----

CREATE TABLE Bibliothécaire(
  IDB      Integer NOT NULL ,
  Nom      Varchar (50) ,
  Prenom   Varchar (50) ,
  Tel      Integer ,
  Mail     Varchar (50) ,
  PRIMARY KEY (IDB )
)ENGINE=InnoDB;

#-----
# Table: AssistantBibliothécaire
#-----

CREATE TABLE AssistantBibliothécaire(
  ID_B      Integer NOT NULL ,
  Nom      Varchar (50) NOT NULL ,
  Prenom   Varchar (50) NOT NULL ,
  Tel      Integer NOT NULL ,
  Adresse  Varchar (50) NOT NULL ,
  Mail     Varchar (50) NOT NULL ,
  PRIMARY KEY (ID_B )
)ENGINE=InnoDB;

#-----
# Table: Abonné
#-----

CREATE TABLE Abonné(
  Id_Abon   Integer NOT NULL ,
```

```
SQL : mcdmédiathèque*

Nom          Varchar (50) NOT NULL ,
Prenom       Varchar (50) NOT NULL ,
Age          Varchar (10) ,
Email        Varchar (50) ,
NumTel       Integer ,
DureeEmprunt Integer ,
MotDePasse   Varchar (25) ,
NbRetard     Integer ,
NbEmprunt    Integer ,
PRIMARY KEY (Id_Abon )
)ENGINE=InnoDB;

#-----
# Table: FicheEmprunt
#-----

CREATE TABLE FicheEmprunt(
  ID_fiche    Integer NOT NULL ,
  ID_Abonne   Integer ,
  ID_R        Integer ,
  DateEmprunt Date ,
  DateRetour  Date ,
  Retard      Date ,
  Id_Abon     Integer ,
  ID          Integer ,
  PRIMARY KEY (ID_fiche )
)ENGINE=InnoDB;

#-----
# Table: Emplacement
#-----

CREATE TABLE Emplacement(
  Rayon       Varchar (25) NOT NULL ,
  Emplacement_R Varchar (25) ,
  PRIMARY KEY (Rayon )
)ENGINE=InnoDB;
```

```
SQL : mcdmédiathèque*

#-----
# Table: Ressource
#-----

CREATE TABLE Ressource(
  ID          Integer NOT NULL ,
  Titre       Varchar (25) ,
  Auteur      Varchar (25) ,
  NbExemplaires Integer ,
  Prix        Integer ,
  Etat        Integer ,
  Rayon       Varchar (25) ,
  TypeOuvrage Varchar (25) NOT NULL ,
  PRIMARY KEY (ID )
)ENGINE=InnoDB;

#-----
# Table: Stock
#-----

CREATE TABLE Stock(
  Capacite      Integer NOT NULL ,
  CapaciteOccupe Integer NOT NULL ,
  PRIMARY KEY (Capacite ,CapaciteOccupe )
)ENGINE=InnoDB;

#-----
# Table: Facture
#-----

CREATE TABLE Facture(
  IDF         Integer NOT NULL ,
  IDR         Integer ,
  IDC         Integer ,
  DateFacture Date ,
  Montant     Integer ,
  IDB         Integer ,
  PRIMARY KEY (IDF )
)ENGINE=InnoDB;
```

```
SQL : mcdmédiathèque*

CapaciteOccupe Integer NOT NULL ,
PRIMARY KEY (IDB ,Capacite ,CapaciteOccupe )
)ENGINE=InnoDB;

#-----
# Table: être
#-----

CREATE TABLE etre(
ID Integer NOT NULL ,
Capacite Integer NOT NULL ,
CapaciteOccupe Integer NOT NULL ,
PRIMARY KEY (ID ,Capacite ,CapaciteOccupe )
)ENGINE=InnoDB;

#-----
# Table: suivre
#-----

CREATE TABLE suivre(
Id_Abon Integer NOT NULL ,
ID_B Integer NOT NULL ,
PRIMARY KEY (Id_Abon ,ID_B )
)ENGINE=InnoDB;

#-----
# Table: considérer
#-----

CREATE TABLE considerer(
Id_Abon Integer NOT NULL ,
EtatAbonne Varchar (25) NOT NULL ,
PRIMARY KEY (Id_Abon ,EtatAbonne )
)ENGINE=InnoDB;

#-----
```

```
SQL : mcdmédiathèque*

PRIMARY KEY (IDF )
)ENGINE=InnoDB;

#-----
# Table: Type
#-----

CREATE TABLE Type(
TypeOuvrage Varchar (25) NOT NULL ,
PRIMARY KEY (TypeOuvrage )
)ENGINE=InnoDB;

#-----
# Table: Etat
#-----

CREATE TABLE Etat(
EtatAbonne Varchar (25) NOT NULL ,
PRIMARY KEY (EtatAbonne )
)ENGINE=InnoDB;

#-----
# Table: Disponibilité
#-----

CREATE TABLE Disponibilite(
Dispo Varchar (25) NOT NULL ,
PRIMARY KEY (Dispo )
)ENGINE=InnoDB;

#-----
# Table: consulter
#-----

CREATE TABLE consulter(
IDB Integer NOT NULL ,
Capacite Integer NOT NULL ,
```

```
SQL: mcdmediathèque*

# Table: considérer
#-----
CREATE TABLE considerer(
    Id_Abon Integer NOT NULL ,
    EtatAbonne Varchar (25) NOT NULL ,
    PRIMARY KEY (Id_Abon ,EtatAbonne )
)ENGINE=InnoDB;

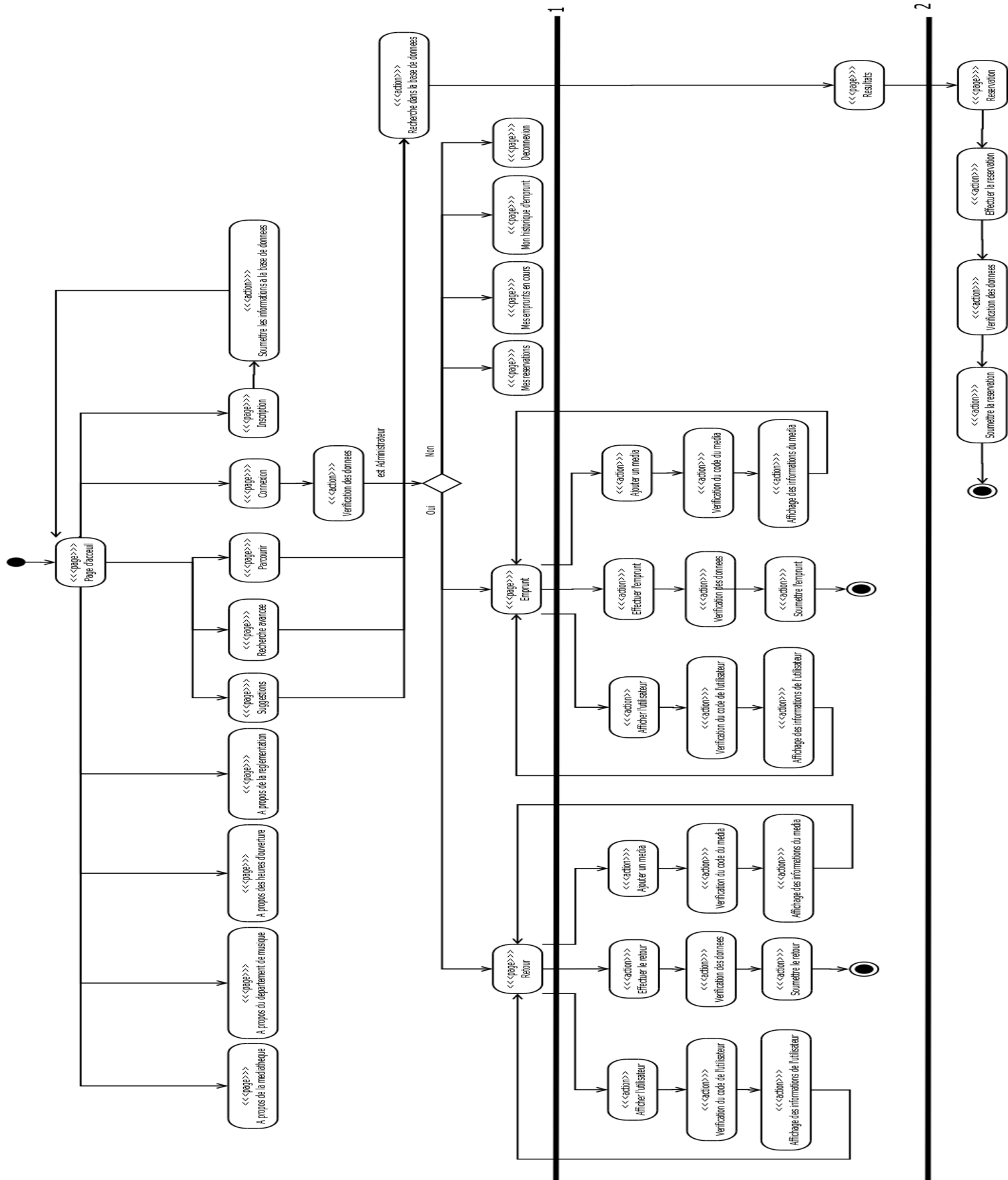
#-----
# Table: peut être
#-----
CREATE TABLE peut_etre(
    Dispo Varchar (25) NOT NULL ,
    ID Integer NOT NULL ,
    PRIMARY KEY (Dispo ,ID )
)ENGINE=InnoDB;

ALTER TABLE FicheEmprunt ADD CONSTRAINT FK_FicheEmprunt_Id_Abon FOREIGN KEY (Id_Abon) REFERENCES Abonne(Id_Abon);
ALTER TABLE FicheEmprunt ADD CONSTRAINT FK_FicheEmprunt_ID FOREIGN KEY (ID) REFERENCES Ressource(ID);
ALTER TABLE Ressource ADD CONSTRAINT FK_Ressource_Rayon FOREIGN KEY (Rayon) REFERENCES Emplacement(Rayon);
ALTER TABLE Ressource ADD CONSTRAINT FK_Ressource_TypeOuvrage FOREIGN KEY (TypeOuvrage) REFERENCES Type(TypeOuvrage);
ALTER TABLE Facture ADD CONSTRAINT FK_Facture_IDB FOREIGN KEY (IDB) REFERENCES Bibliothecaire(IDB);
ALTER TABLE consulter ADD CONSTRAINT FK_consulter_IDB FOREIGN KEY (IDB) REFERENCES Bibliothecaire(IDB);
ALTER TABLE consulter ADD CONSTRAINT FK_consulter_Capacite FOREIGN KEY (Capacite) REFERENCES Stock(Capacite);
ALTER TABLE consulter ADD CONSTRAINT FK_consulter_CapaciteOccupe FOREIGN KEY (CapaciteOccupe) REFERENCES Stock(CapaciteOccupe);
ALTER TABLE etre ADD CONSTRAINT FK_etre_ID FOREIGN KEY (ID) REFERENCES Ressource(ID);
ALTER TABLE etre ADD CONSTRAINT FK_etre_Capacite FOREIGN KEY (Capacite) REFERENCES Stock(Capacite);
ALTER TABLE etre ADD CONSTRAINT FK_etre_CapaciteOccupe FOREIGN KEY (CapaciteOccupe) REFERENCES Stock(CapaciteOccupe);
ALTER TABLE suivre ADD CONSTRAINT FK_suivre_Id_Abon FOREIGN KEY (Id_Abon) REFERENCES Abonne(Id_Abon);
ALTER TABLE suivre ADD CONSTRAINT FK_suivre_ID_B FOREIGN KEY (ID_B) REFERENCES AssistantBibliothecaire(ID_B);
ALTER TABLE considerer ADD CONSTRAINT FK_considerer_Id_Abon FOREIGN KEY (Id_Abon) REFERENCES Abonne(Id_Abon);
ALTER TABLE considerer ADD CONSTRAINT FK_considerer_EtatAbonne FOREIGN KEY (EtatAbonne) REFERENCES Etat(EtatAbonne);
ALTER TABLE peut_etre ADD CONSTRAINT FK_peut_etre_Dispo FOREIGN KEY (Dispo) REFERENCES Disponibilite(Dispo);
ALTER TABLE peut_etre ADD CONSTRAINT FK_peut_etre_ID FOREIGN KEY (ID) REFERENCES Ressource(ID);
```

VIII) A quoi ressemblera notre site web ? (screens)

Dans cette partie du cahier de conception , nous allons vous illustrer comment le site web se présentera au préalable , en commençant par un diagramme d'enchaînement , qui nous illustrera les composantes du site , ainsi que les sous-composantes de chaque composant , et comment y accéder , pour finir enfin avec quelques screens qui illustrerons quelques fonctionnalités du site web , afin que vous ayez une idée de ce qui vous attend , mais tout d'abord commençant avec la partie diagramme d'enchaînement , que vous trouverez dans la page suivante , vu sa dimension :

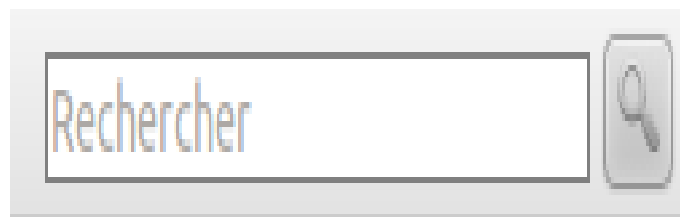
1) Diagramme d'enchainement :



2) *Presentation en image de quelques fontionalités du site web :*

-Barre de recherche :

sur ce screen nous illustrons une barre de recherche toute simple que l'abonné trouvera a l'accueil du site web :



-Barre de recherche avancé :

Dans la continuité de la barre de recherche simple , nous avons illustrer cette fois une barre de recherche avancé qui permettra a l'abonné de faire des recherches plus détaillées :

Rechercher :

Titre du média*

Contient

Indiana Jones

et

Année de publication

(>) Est plus grand que

1988

et

Mots-clés

Contient

[+]

Rechercher

Effacer

-Barre de suggestions :

Qui sera utilisé par l'abonné dans le cas où il voudra que la médiathèque mette en ligne des ressources inexistantes

Suggestions

Titre :

Artiste :

Type de support :

Commentaire :

-Resultats de recherche :

Qui apparaîtra après la recherche effectuée par l'abonné dans le cas où les ressources qu'il recherche sont bien existantes dans la médiathèque :

Affinez votre recherche

Année de publication

1987
1990

Recherche demandée: Les médias ayant le mot clé: «Indiana Jones», 2 média(s) trouvé(s)

Précédent 1 Suivant



Indiana Jones et la Dernière Croisade

Éditeur: Lucasfilm
Année de publication: 1987
Catégorie: Vidéo - VHS
Genre: Aventure
Action:



Indiana Jones et le Temple maudit

Éditeur: Lucasfilm
Année de publication: 1990
Catégorie: Vidéo - VHS
Genre: Aventure
Notes: Film extraordinaire
Action:

Précédent 1 Suivant

-Champs cliquable de suggestion :

Ici , on illustrera un champ cliquable de suggestions ou l'abonné pourra consulter les ressources présentes dans la mediatheque et cliquer dessus directement

Affinez votre recherche

Artiste
Jacques Brel
Les cowboys fringants*
Polémil Bazar
Afficher plus


Genre
Country
Folk
Musique classique
Afficher plus

Année de publication
1988
1997
1998
Afficher plus


Éditeur
Barclay
Deutsche Grammophon
Éditeur indépendant
Afficher plus

-Reservation :

Après que l'abonné aies choisi sa ressource , il pourra donc la réserver pour que le gestionnaire valide l'emprunt qu'on illustrera dans le screen d'après , la réservation se présentera donc de cette façon :



Réservation



Break syndical
Éditeur: La tribune
Année de publication: 2002
Catégorie: Audio - CD
Genre: Country
Artiste: Les cowboys fringants

Réserver
☐ Dès que possible
☒ Période déterminée
Du 2011-06-08 ▼ au 2011-06-08 ▼

-Emprunt :

Qui se présentera de cette façon au gestionnaire apres qu'un abonné aies réserver une ressource , le gestionnaire devra donc valider l'emprunt qui se presentera de cette façon la :



Emprunt

Matricule de
l'utilisateur :
123456789
Bob Gratton
514-777-8888
bob@gmail.com

Du au

No. Référence : - +

CD-C00005 | Motel capri

-Retour :

De meme pour le retour , le gestionnaire devra le valider , et cela se presentera de cette façon la :



Retour

Matricule de
l'utilisateur :
123456789
Bob Gratton
514-777-8888
bob@gmail.com

No. Référence : - +

CD-C00005 | Motel capri

