



**Lucas BENITO**  
**Tanguy Prudat**  
**Pierre-Antoine MONTEZ**  
**Alexis Urvoy**

**GI01-02**

**Rapport de LO43 - P2011**

**UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD**

# **Logiciel de gestion d'une bibliothèque**

<b>I. SUJET DU PROJET DE LO43 .....</b>	<b>3</b>
1. OBJECTIFS .....	3
2. LE SUJET .....	3
3. PREMIERE APPROCHE DU SUJET .....	4
4. PREMIERE APPROCHE « TECHNIQUE DU SUJET » .....	4
<b>II. SOLUTIONS TECHNIQUES DU LOGICIEL .....</b>	<b>5</b>
1. L'IMPLEMENTATION LOGICIELLE .....	5
2. UTILISATION DU XML .....	6
<b>III. MODELISATION UML .....</b>	<b>7</b>
1. DIAGRAMME DE CAS D'UTILISATION .....	7
2. DIAGRAMME DE SEQUENCE .....	8
3. DIAGRAMME DES FONCTIONS.....	9
4. DIAGRAMME DES CLASSES DU LOGICIEL.....	10
5. DIAGRAMME DES CLASSES DE L'INTERFACE GRAPHIQUE .....	12
<b>IV. PRESENTATION DU LOGICIEL .....</b>	<b>13</b>
<b>IV. PROBLEMES RENCONTRES.....</b>	<b>20</b>
<b>V. AMELIORATIONS POSSIBLES.....</b>	<b>20</b>
<b>VI. CONCLUSION .....</b>	<b>20</b>

# I. Sujet du projet de LO43

## 1. Objectifs

- Faire l'analyse UML du projet en utilisant les diagrammes nécessaires
- Proposer une implémentation logicielle de la gestion d'une bibliothèque
- Rédiger un rapport final
- Présentation et démonstration de notre logiciel

## 2. Le sujet

L'objectif de ce projet est de développer une application en Java permettant la gestion des prêts des documents. Plus précisément, il s'agit de développer les fonctionnalités suivantes :

### • L'ajout et les mises à jour de documents

Il existe en fait plusieurs types de documents: journaux, livres et magazines, CDs, etc. Une catégorisation des documents est nécessaire. Chaque document est caractérisé par son identifiant, catégorie, titre, auteurs, éditeur, l'année, etc.

### • La recherche documentaire

Cette fonction permet aux emprunteurs de chercher un document en utilisant, l'identifiant, les noms des auteurs, le titre, etc. Si le document n'est pas disponible pour le moment, une date de disponibilité et de réservation sera fournie à l'emprunteur. L'emprunteur n'a le droit qu'à un nombre fixe d'emprunts, Il ne pourra effectuer un emprunt que s'il a rendu les documents déjà empruntés.

### • La gestion des comptes des emprunteurs

Par exemple, les étudiants/enseignants empruntent des livres, des rapports, ou des articles de la bibliothèque. L'emprunt ne doit pas dépasser deux mois. Pour chaque emprunt, le nom de l'emprunteur, le document emprunté, la date d'emprunt, etc., doivent être enregistrés. Des interfaces seront aussi développées pour consulter l'état des prêts et/ou les réservations en cours. Par exemple, les prêts doivent être classés par nombre de jours de retard afin de prévenir les emprunteurs concernés.

• Une interface graphique permettant la gestion des flux d'E/S sera développée. Les données sont stockées dans des fichiers. Au démarrage de l'application, ces données doivent être enregistrées dans une structure de données pour faciliter la recherche et l'ajout. Les structures de données permettant de faire la recherche et l'ajout avec un coût minimal sont des tables de hachage. A la fermeture de l'application, toutes les données stockées dans la structure de données seront à nouveau stockées dans les fichiers correspondants.

### 3. Première approche du sujet

L'objectif de ce projet a été de créer une unique application offrant toutes les fonctionnalités nécessaires pour la gestion d'une bibliothèque. Son rôle principal est de tenir un état de lieux permanent de la bibliothèque, afin de surveiller les entrées et sorties de documents. Pour se faire, toutes les informations concernant les documents et leurs emprunts sont stockées dans une base de données.

Cette application doit fournir à la fois les services destinés aux emprunteurs ainsi que ceux permettant la bonne direction de la bibliothèque par leurs responsables. Ainsi, à l'ouverture du programme, l'interface et les fonctionnalités diffèrent selon si l'utilisateur est un simple emprunteur ou un bibliothécaire.

Les emprunteurs ont à leur disposition une interface leur permettant uniquement d'avoir accès à la liste des documents. Ils peuvent y effectuer une recherche et choisir le ou les documents qu'ils veulent emprunter. Un récapitulatif de leurs emprunts les informe sur leur situation.

Les bibliothécaires ont aussi accès à la liste des emprunts et des emprunteurs. Ils peuvent décider de valider ou non un emprunt. Ils peuvent aussi ajouter un document dans la bibliothèque.

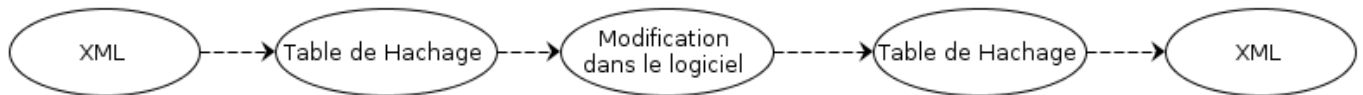
Les règles de la bibliothèque sont implicitement contenus dans le programme, de telle manière qu'un emprunteur ne pourra pas dépasser plus de cinq emprunts simultanés. La date limite de retour d'un document est fixé à deux mois.

Le langage utilisé est le langage Java et les données sont stockées dans des fichiers XML. Chaque modification faite par un utilisateur entraîne à la fermeture de l'application une modification de la base de données. Le stockage intermédiaire des données se fait grâce à des tables de hachages. L'interface graphique utilise les bibliothèques Swing et AWT.

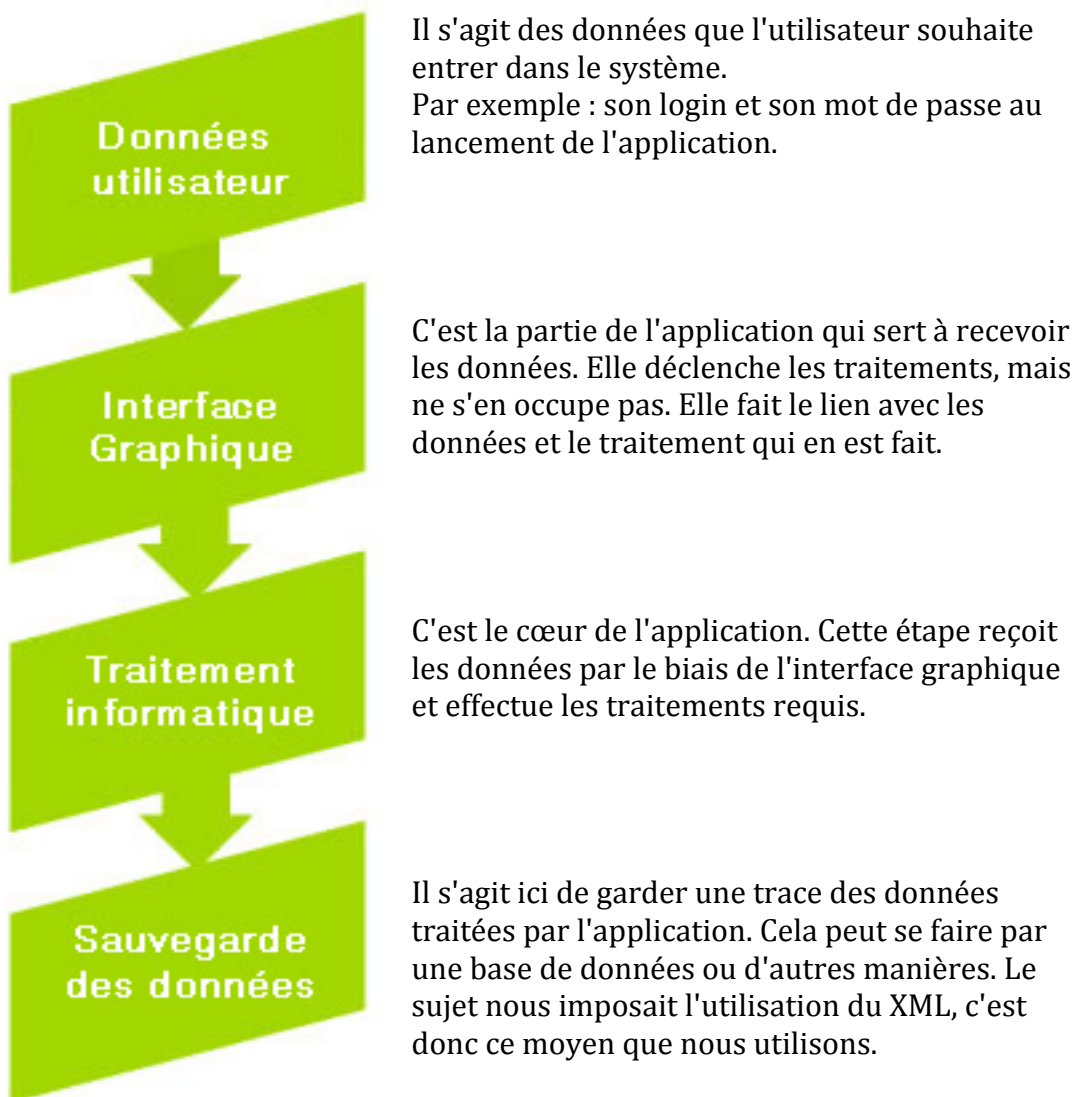
### 4. Première approche « technique du sujet »

Nous avons décidé pour plus de clarté et une meilleure utilisation de logiciel d'utiliser des fichiers XML. C'est à dire que toutes les données (Documents, Utilisateurs et emprunts effectués) seront stockées dans des fichiers XML. Lors du lancement du logiciel, toutes ces données seront exportées dans des tables de hachage et les modifications se feront directement au niveau de ces tables. Lors de la fermeture, on stockera les tables de hachage (modifiées ou non) dans ces mêmes fichiers XML.

## II. Solutions techniques du logiciel



### 1. L'implémentation logicielle



## 2. Utilisation du XML

Tout au long de l'utilisation de l'application, nous disposons de 3 fichiers XML :

utilisateurs.xml  
documents.xml  
emprunts.xml

Le fichier utilisateurs.xml est celui qui contient toutes les informations concernant les utilisateurs du système comme le login, le mot de passe, le nom, etc. Ces informations sont essentielles car elles permettent de connecter un utilisateur au démarrage de l'application (comparaison entre les login et mot de passe entrés dans l'interface graphique et ceux du fichier XML). Une fois connecté, ces informations sont encore utiles pour un bibliothécaire qui voudrait modifier les informations d'un utilisateur par exemple. La mise à jour du fichier XML sera traitée plus loin.

Le fichier documents.xml quant à lui contient les informations relatives à tous les documents de la bibliothèque, sur le même principe que le fichier utilisateurs.xml.

Enfin, le fichier emprunts.xml recense tous les emprunts effectués à la bibliothèque, selon l'id de l'utilisateur qui a effectué cet emprunt, la date d'emprunt et la date de retour.

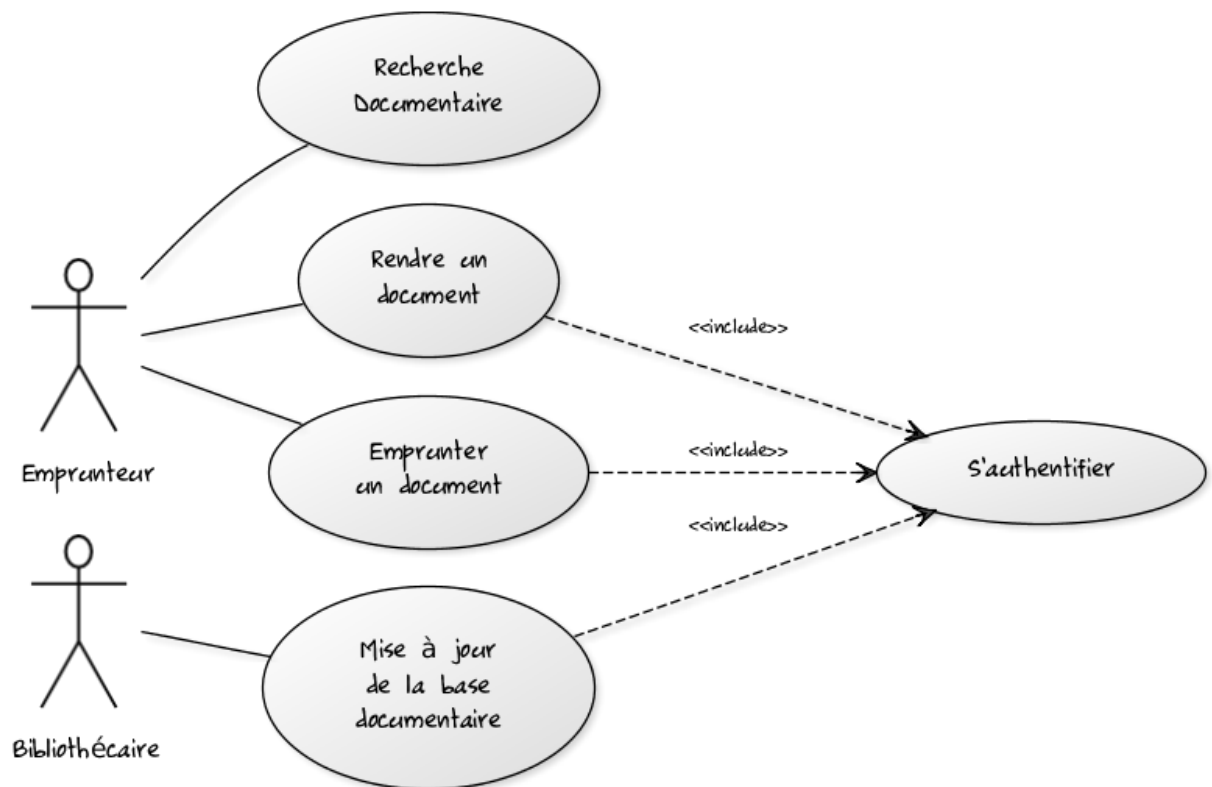
Maintenant que nous savons à quoi correspond chaque fichier, voyons comment ils sont utilisés dans l'application.

Au lancement de l'application, avant la moindre connexion de la part d'un utilisateur, chaque fichier XML est lu et les données qu'il contient sont stockées dans une HashMap. Ceci pour des raisons de rapidité d'exécution et pour une facilité d'accès aux données depuis l'application. Lorsqu'un changement survient, les HashMaps sont directement modifiées, mais pas les fichiers XML, toujours dans un souci de rapidité d'exécution. Ce n'est que lors de la fermeture de l'application (lors d'un appui sur le bouton « Quitter ») que les fichiers XML sont mis à jour. En pratique, un nouveau fichier XML est généré contenant toutes les données de chaque HashMap qui vient remplacer les anciens fichiers.

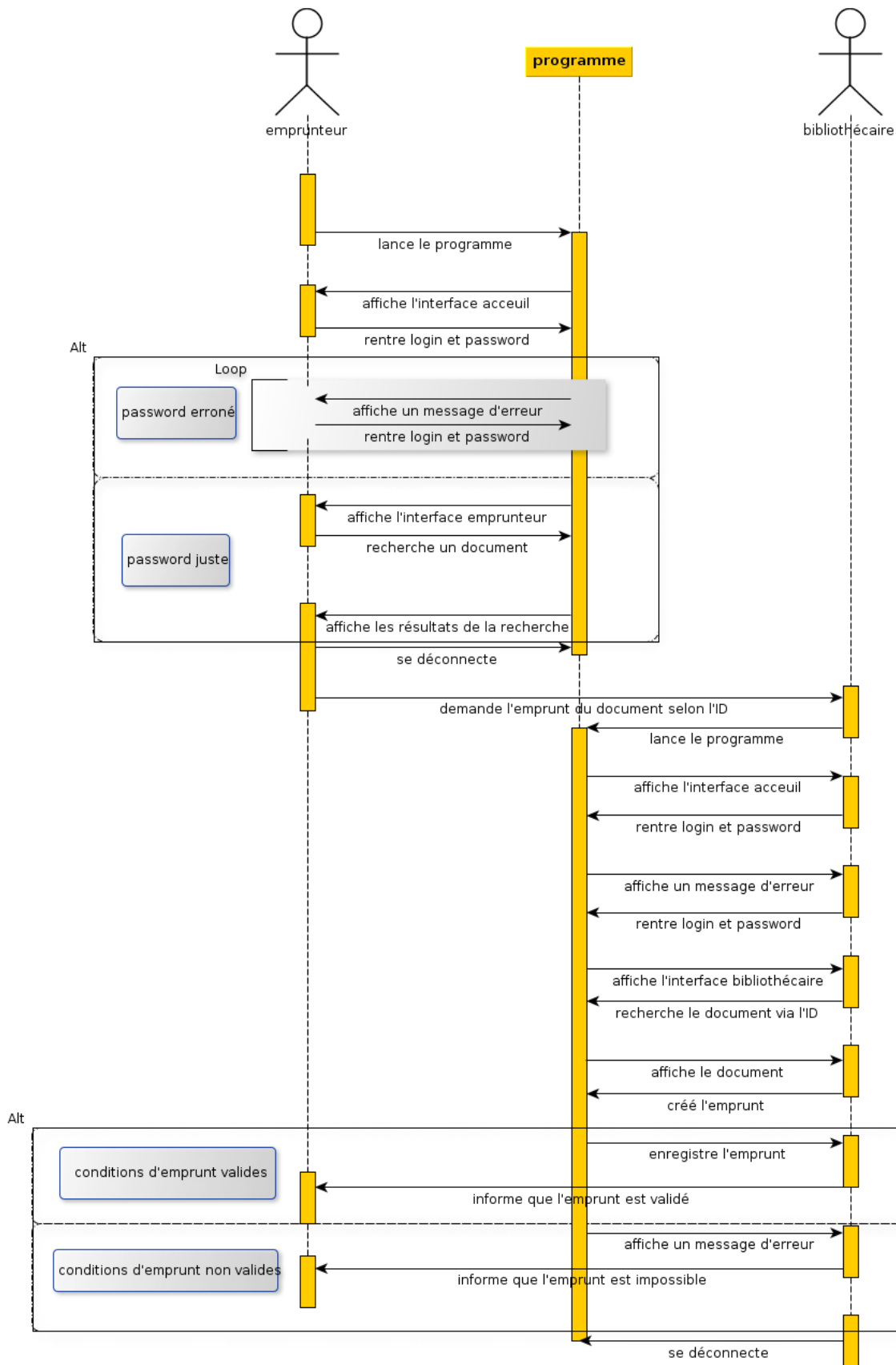


### III. Modélisation UML

#### 1. Diagramme de cas d'utilisation

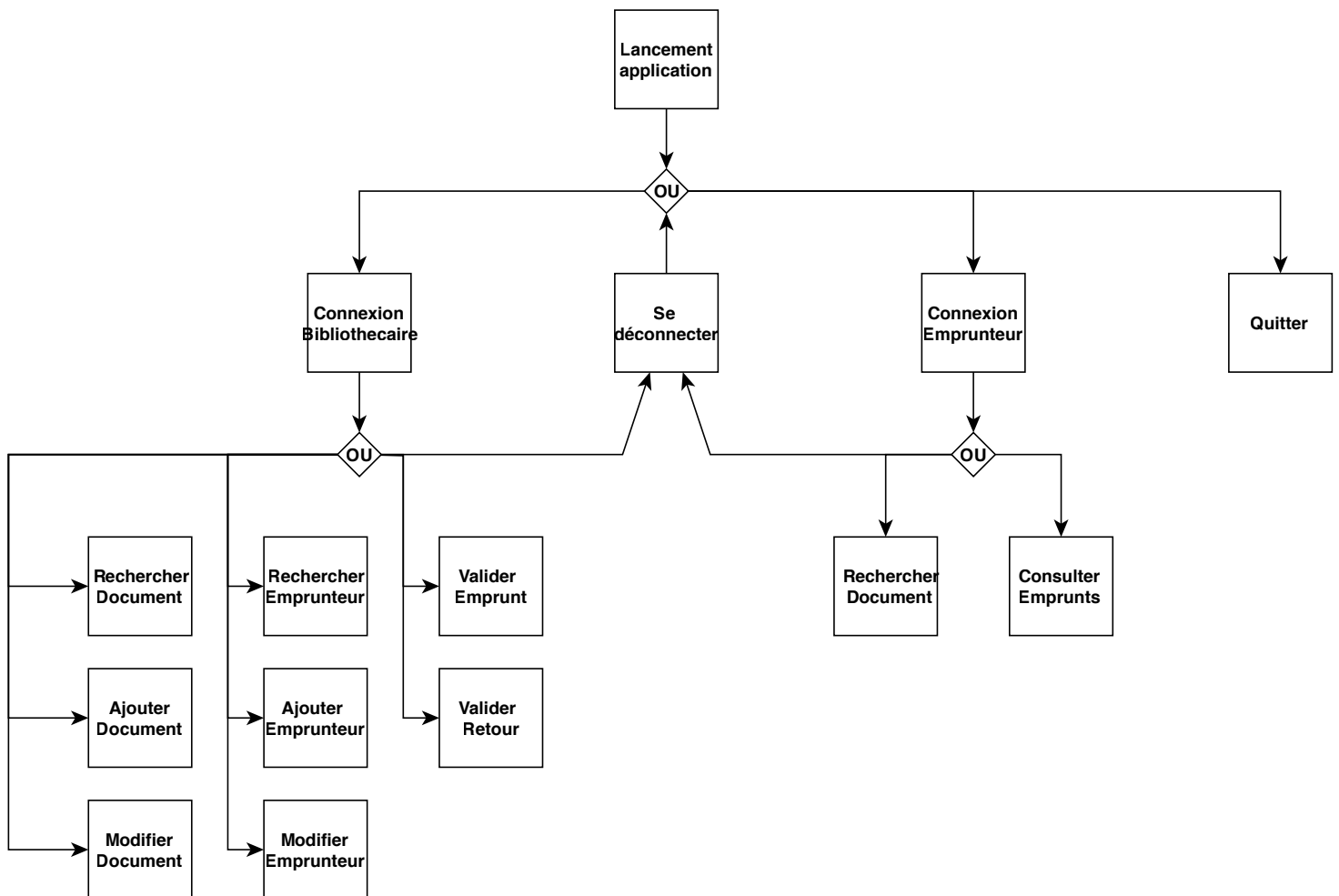


## 2. Diagramme de séquence

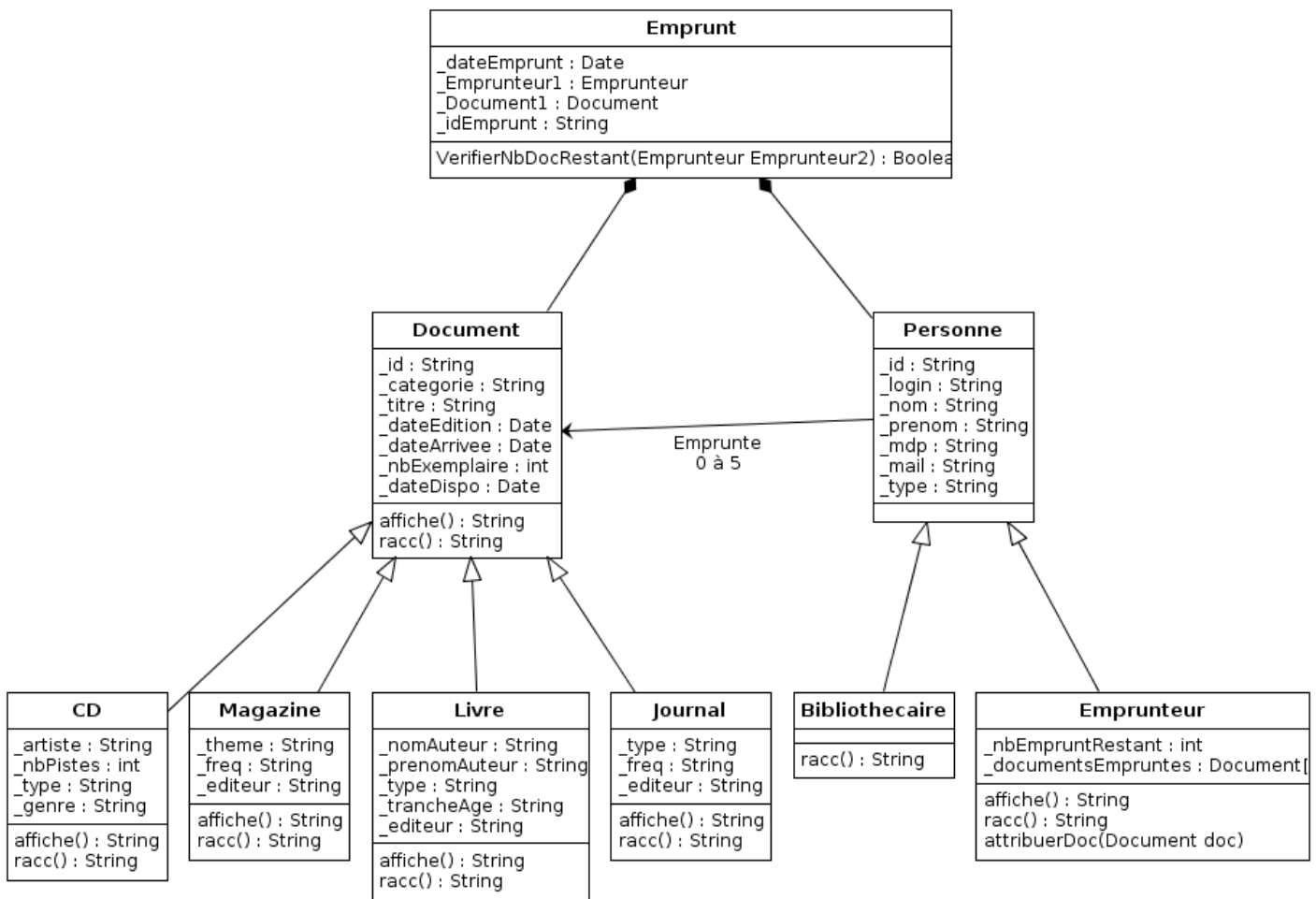




### 3. Diagramme des fonctions



#### 4. Diagramme des classes du logiciel



La classe Document est une classe abstraite qui contient sept attributs : l'identifiant (id) du document, son titre, sa date d'édition, sa catégorie (livre, magazine, journal ou CD), le nombre d'exemplaires disponibles en bibliothèque, sa date d'arrivée en bibliothèque (date du premier exemplaire) et sa date de disponibilité (date à laquelle le document sera disponible s'il est en cours d'emprunt). Les classes Livre, Magazine, Journal et CD héritent toutes de la classe Document.

La classe Livre contient cinq attributs : le nom et le prénom de l'auteur, le type du livre (policier, fantastique...), la tranche d'âge conseillée (par exemple 12-16 ans) et l'éditeur (par exemple Robert Laffont)

La classe Magazine contient trois attributs : le thème du magazine (vulgarisation scientifique, mode...), sa fréquence (quotidien, hebdomadaire...) et l'éditeur.

La classe Journal contient trois attributs : le type de journal (politique, économique...), sa fréquence et l'éditeur.

La classe CD contient quatre attributs : l'artiste, le type (compilation, single...), le genre (classique, rock...) et le nombre de pistes.

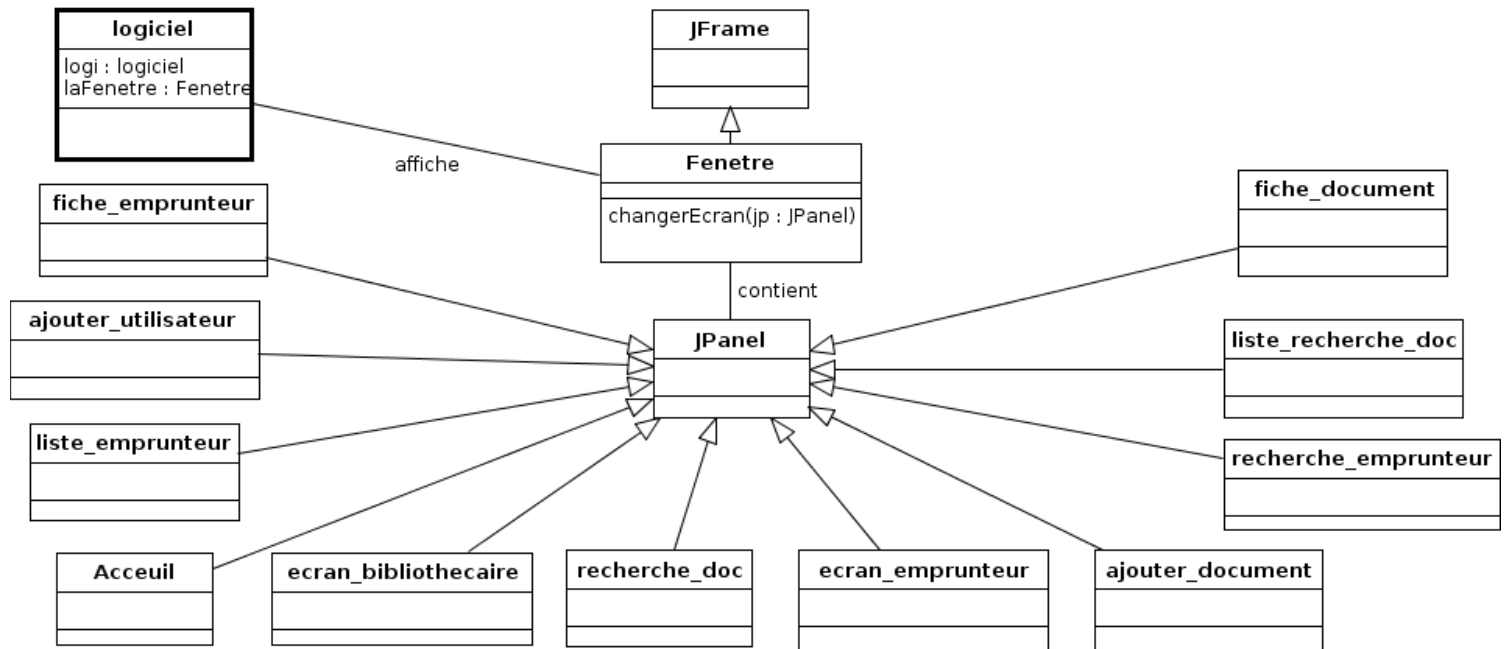
La classe Personne est une classe abstraite qui contient sept attributs : l'identifiant (id) de la personne, son login (par exemple jean.dupont), son nom, son prénom, son mot de passe, son mail et son type (Emprunteur ou Bibliothécaire). Les classes Emprunteur et Bibliothécaire héritent de la classe Personne.

La classe Emprunteur contient deux attributs : le nombre d'emprunts restants (égal au nombre d'emprunts maximum fixé à 5 moins le nombre d'emprunts en cours) et un tableau contenant les documents empruntés. La méthode `accessDocument(i:Integer)` renvoie le document placé en i-ème position dans le tableau contenant les documents empruntés. La méthode `attribuerDoc(doc:Document)` valide l'emprunt d'un document et modifie le nombre d'emprunts et le tableau d'emprunts seulement si l'emprunteur n'a pas dépassé le nombre maximal d'emprunts, auquel cas elle affiche un message d'erreur.

Les méthodes `affiche()` et `racc()` sont codées pour chacune de ces classes : la première retourne une description détaillée du document (ou de la personne) et la seconde une description plus concise.

La classe Emprunt contient les attributs utiles à la mémorisation d'un emprunt : la date de celui-ci, le document emprunté, son emprunteur ainsi que l'identifiant de l'emprunt. La méthode `VerifierNbDocRestant(emprunteur:Emprunteur)` renvoie VRAI s'il l'emprunteur peut encore réserver des documents.

## 5. Diagramme des classes de l'interface graphique



Lors de la construction d'un objet logiciel, on instancie un objet « Fenetre » qui va contenir tout ce que l'on pourra voir à l'écran. « Fenetre » hérite de la classe « JFrame » et doit contenir un objet de type « JPanel ». Cet objet est un conteneur qui place tout les éléments sur la fenêtre (texte, zone de saisie, bouton...).

Nous avons donc créé pleins de classes héritant de « JPanel » contenant les différents éléments nécessaires pour les différents écrans de notre logiciel. Plutôt que de devoir changer de contenu dans une seule fenêtre nous avons ainsi décidé de préparer plusieurs conteneurs différents et de changer le conteneur d'une seule fenêtre. Nous créons un nouveau conteneur et il vient remplacer l'ancien.

Certains conteneurs sont utilisés à la fois pour un bibliothécaire et pour un emprunteur (comme le conteneur « fiche\_emprunteur »). Ceci dit, les fonctionnalités ne doivent pas être les mêmes en fonctions de quel type d'utilisateur est en train de manipuler le logiciel. Pour faire cette différence, un paramètre est passé au constructeur pour distinguer le type de personne qui a appelé ce conteneur. En fonction de ce paramètre, on affichera ou non certains éléments.

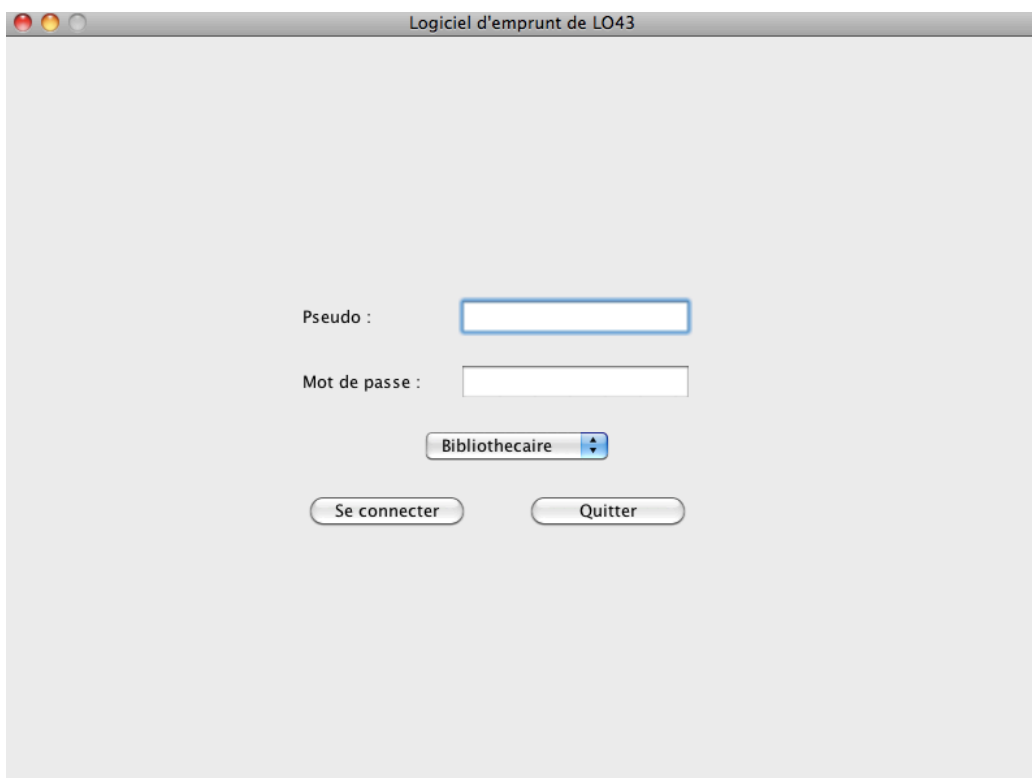
## IV. Présentation du logiciel

### Se connecter

Cette fonction est la première disponible pour un utilisateur. Elle lui permet d'accéder au contenu de l'application. L'utilisateur doit fournir son login et son mot de passe et cliquer sur le bouton « se connecter » ou sur la touche « Entrée ». Pour obtenir ce login et ce mot de passe, l'utilisateur doit d'abord être inscrit auprès de la bibliothèque. Cette étape ne sera pas détaillée ici.

Si le login ou le mot de passe est incorrect, un message avertit l'utilisateur et celui doit alors recommencer jusqu'à ce que le couple login/mot de passe soit correct.


Un fichier XML (Extensible Markup Language) contenant tous les utilisateurs du système est stocké à la racine de l'application. Pour des raisons de sécurité, les mots de passe ne sont pas stockés en clair dans le fichier. Seul un « hash » SHA1 (Secure Hash Algorithm) du mot de passe de chaque utilisateur est stocké.



Logiciel d'emprunt de LO43

Pseudo :

Mot de passe :

Bibliothécaire 

Se connecter    Quitter

## Rechercher un document

Il s'agit de la fonction principale de notre application. C'est sur cet écran qu'arrivent les utilisateurs après s'être authentifiés. Plusieurs choix s'offrent alors à l'utilisateur. D'abord il doit choisir le type de document qu'il souhaite rechercher parmi lesquels : cd, livre, journal, et magazine. Il peut aussi choisir « Tout » s'il souhaite faire une recherche sur tout les types de documents.

Pour affiner sa recherche, il peut alors choisir de renseigner un champ particulier d'un document (son id, son titre, etc). Notez que cette étape n'est pas obligatoire, et qu'une recherche sans préciser de mot clé retournera tous les documents pour le type choisi.

Pour valider la recherche, un simple clique sur le bouton « Rechercher » suffit. Cette action affiche la liste des documents trouvés pour la recherche effectuée, sous forme de tableau. A partir de là, l'utilisateur peut choisir de faire une autre recherche s'il n'est pas satisfait du résultat, ou bien double-cliquer sur la ligne du document qui l'intéresse.

Cette dernière action a pour effet d'afficher la fiche du document en question. L'utilisateur y retrouve alors toutes les informations concernant ce document. A partir d'ici, deux cas se distinguent : l'emprunteur, qui peut décider d'emprunter ce document, ou le bibliothécaire qui peut modifier la fiche du document. Ces deux actions seront détaillées plus loin.

The screenshot displays the 'Logiciel d'emprunt de LO43' application. The top navigation bar includes 'Recherche documentaire' (selected), 'Fiche d'emprunteur', 'Ajouter un document', and 'Ajouter un utilisateur'. The search interface on the left allows filtering by document type (set to 'Tout') and search criteria (radio buttons for 'Titre', 'ID', and 'Catégorie', with 'Titre' selected). A search button labeled 'Rechercher' is at the bottom left. The main area shows a table of search results with columns 'id', 'document', and 'Catégorie'. The table lists 20 items, including CDs, journals, magazines, and books. At the bottom, there are buttons for 'Effectuer une autre recherche' and 'Se déconnecter'.

id	document	Catégorie
08	Alive	cd
09	Le monde n°1524	journal
17	Science et vie n°00001	magazine
04	Le seigneur des anneaux : Les deux t...	livre
18	Science et vie n°00004	magazine
05	Hit2010	cd
15	Micro Hebdo du 12/11/2009	magazine
06	The Fame	cd
16	Micro Hebdo du 13/11/2009	magazine
07	Still Dre	cd
13	La depeche du 21/12/2008	journal
14	Micro Hebdo du 11/11/2009	magazine
01	20000 Lieues sous les mers	livre
11	La croix du 15/04/2011	journal
02	Meutre à MontMartre	livre
12	Le petit quotidien du 12/12/2008	journal
03	Harry Potter à l'école des sorciers	livre
10	Le canard enchainé du 12/02/2010	journal

## Ajouter un document

Cette action n'est disponible que pour les bibliothécaires. Lors d'un ajout de document, tous les champs affichés sur l'écran doivent être remplis. Une fois toutes les informations rentrées correctement, un clique sur le bouton « Enregistrer » ajoute le document à la liste des documents disponibles. Il peut alors être emprunté directement par un emprunteur.

## Modifier un document

Un bibliothécaire peut avoir besoin de modifier un document (modifier le nombre d'exemplaire par exemple, ou une faute d'orthographe dans le titre). Pour cela, après avoir trouvé le document désiré (via l'interface de recherche de document), il a la possibilité de modifier chaque champ du document. Une fois les modifications terminées, il suffit de cliquer sur le bouton « Valider » pour que les modifications soient prises en compte. C'est aussi sur cet écran qu'on pourra supprimer le document, par un clique sur le bouton « Supprimer ».

The screenshot displays the 'Logiciel d'emprunt de LO43' interface. The top navigation bar includes 'Recherche documentaire', 'Fiche d'emprunteur', 'Ajouter un document' (highlighted), and 'Ajouter un utilisateur'. The 'Ajouter un document' form contains the following fields:

- ID :
- Catégorie :
- Titre :
- Date d'édition :
- Date d'arrivée :
- Nombre d'exemplaire disponible :
- Date de disponibilité :
- Artiste :
- Type :

A second window, also titled 'Logiciel d'emprunt de LO43', shows a document with the following details:

- ID : 11
- Catégorie : journal
- Titre : La croix du 15/04/2011
- Date d'édition : 15/04/2011
- Date d'arrivée :
- Nombre d'exemplaire disponible : 2
- Type : Religieux
- Editeur : Bayard presse

At the bottom of this window are buttons for 'Effectuer une autre recherche', 'Valider', and 'Se déconnecter'. A modal dialog box titled 'Enregistrer ?' is overlaid on the second window, asking 'Voulez-vous enregistrer les modifications ?' with 'Non' and 'Oui' buttons.



## Rechercher un utilisateur

De la même manière qu'une recherche documentaire, une recherche d'emprunteur est disponible. Cette fonction ne peut être utilisée que par les bibliothécaires. Deux types de recherches sont disponibles : soit la recherche d'un utilisateur « normal », basique. Soit la recherche d'un utilisateur n'ayant pas respecté les délais de retour pour un emprunt (dans la version actuelle de l'application, cette fonction n'est pas disponible).

Pour préciser sa recherche, le bibliothécaire pourra renseigner l'id de l'utilisateur qu'il recherche ou son nom, afin de réduire le nombre d'utilisateurs correspondants à la recherche.

Logiciel d'emprunt de LO43

Recherche documentaire **Fiche d'emprunteur** Ajouter un document Ajouter un utilisateur

Type de recherche :

☒ Emprunteur normal

☐ Emprunteur n'ayant pas respecté les délais d'emprunts

Veuillez saisir au moins un des deux champs suivants :

ID :

Nom :

Rechercher

Logiciel d'emprunt de LO43

id	nom	prenom	mail	Type
04	montez	pierre	pam@lo43.utbm.fr	Emprunteur
01	Nom	Lucas	lucas@lo43.utbm.fr	Emprunteur
02	Nom	Lucas	lucas@lo43.utbm.fr	Bibliothécaire
03	Urvoy	Alexis	lucas@lo43.utbm.fr	Bibliothécaire

Effectuer une autre recherche

Se déconnecter

## Ajouter un utilisateur

Cette fonction n'est pas disponible dans la version actuelle de l'application. De la même manière que l'ajout d'un document, le bibliothécaire doit remplir tous les champs de l'interface et cliquer sur le bouton « Ajouter ». Ainsi, un nouvel utilisateur est ajouté au système et il pourra se connecter avec son login et son mot de passe fournis par le bibliothécaire.

## Modifier un utilisateur

Encore une fois, seul un bibliothécaire peut effectuer cette action. Elle lui permet de modifier les informations d'un utilisateur comme son nom, son prénom ou son adresse e-mail.

The screenshot displays the 'Logiciel d'emprunt de LO43' application interface. At the top, there are four tabs: 'Recherche documentaire', 'Fiche d'emprunteur', 'Ajouter un document', and 'Ajouter un utilisateur' (which is highlighted). The 'Ajouter un utilisateur' form contains the following fields:

- ID : [Empty text box]
- Type d'utilisateur : [Dropdown menu showing 'Emprunteur']
- Pseudo : [Empty text box]
- Mot de passe : [Empty text box]
- Nom : [Empty text box]
- Prénom : [Empty text box]
- Mail : [Empty text box]

Below the form, the application shows the details of the user being added:

- ID : 03
- Nom : Urvoy
- Prénom : Alexis
- Mail : lucas@lo43.utbm.fr

A confirmation dialog box titled 'Enregistrer ?' is displayed in the foreground, asking 'Voulez-vous enregistrer les modifications ?' with 'Non' and 'Oui' buttons. The 'Oui' button is highlighted. In the background, the text 'Aucun emprunt à votre a' is visible. At the bottom of the application window, there are four buttons: 'Valider la modification', 'Modifier la fiche', 'Se déconnecter', and 'Effectuer une autre recherche'.

### Valider un emprunt

Lorsqu'un emprunteur désire emprunter un document, il va voir le bibliothécaire, qui utilise l'id du document (généralement marqué sur le livre) pour le rechercher, et l'ajouter à la liste des emprunts de la personne (via l'id sur sa carte de bibliothèque par exemple). Les informations comme la date d'emprunt et la date de retour sont enregistrés pour permettre la gestion des prêts.

The screenshot shows a software window titled "Logiciel d'emprunt de LO43". It contains a form with the following fields and values:

- ID : 07
- Catégorie : cd
- Titre : Still Dre
- Date d'édition : 2008
- Date d'arrivée : (empty)
- Nombre d'exemplaire di : 1
- Artiste : Dr Dre
- Type : Single
- Genre : Rap

At the bottom of the form are five buttons: "Effectuer une autre recherche", "Emprunter", "Modifier", "Se déconnecter", and "Supprimer".

A modal dialog box titled "Saisie" is overlaid on the form. It contains the text "Veuillez rentrer le pseudo de l'utilisateur :", an input field, and two buttons: "Annuler" and "OK". There is also a small icon of a coffee cup and a pen in the dialog box.

### Valider un retour de document

Lors d'un retour de document, le bibliothécaire doit le signaler afin de mettre à jour la liste des emprunts d'un utilisateur. La liste des emprunts pour un utilisateur est donc affichée, et une fois l'emprunt sélectionné, il suffit de cliquer sur le bouton « Retour » pour indiquer au système que le document à été rendu par l'utilisateur.

### Consulter liste des emprunts

Cette action n'est possible que pour un emprunteur. Sur cet écran, l'utilisateur à accès a tous ces emprunts en cours avec leur date de rendu. Il peut ainsi savoir ou il en est, et combien de documents il peut encore emprunter (maximum 5, pour une durée de 30 jours par document). Cette fonctionnalité n'est pas encore disponible dans la version actuelle de l'application.

Cette fonctionnalité devra aussi être disponible pour les bibliothécaires, qui pourront ainsi visualiser les emprunts en retard, et prévenir les emprunteurs en leur envoyant un mail par exemple.

The screenshot shows a web application window titled "Logiciel d'emprunt de LO43". The interface includes a form for user identification and a table of active loans.

**User Information Form:**

- ID : 04
- Nom : montez
- Prenom : pierre
- Mail : pam@lo43.utbm.fr

**Loan Table:**

id	document	Catégorie	Date de retour
02	Meutre à MontMartre	livre	11/11/2011

**Navigation Buttons:**

- Supprimer la fiche
- Modifier la fiche
- Se déconnecter
- Effectuer une autre recherche

## IV. Problèmes rencontrés

Tout d'abord, il a fallu apprendre à créer une interface graphique. N'ayant jamais utilisé les bibliothèques AWT et Swing, il a fallu un certains temps avant de réussir à maîtriser et faire des interfaces pratiques pour la programmation et pour l'utilisation du logiciel.

De plus, la gestion de la base de données ne fut pas simple. Nous sommes tout de suite partis sur le principe de gérer toutes les informations dans des fichiers XML car ce langage nous était familier. Mais l'exploitation de ce type de fichier à partir du langage Java est délicate. D'autant plus que nous avions comme contrainte d'exploiter des tables de hachages, objet que nous ne savions pas maîtriser au début du projet.

Enfin, la répartition du travail dans le groupe et l'organisation de séances pour travailler ensemble furent également une difficulté.

## V. Améliorations possibles

Actuellement, lors du chargement de l'écran d'accueil, nous chargeons les fichiers XML dans des tables de hachages. L'étape inverse s'effectue uniquement lorsque nous quittons l'application à l'aide du bouton « quitter » de l'écran d'accueil. Pour éviter toute perte de données non voulue, nous avons empêché la fermeture de la fenêtre grâce à la croix rouge en haut à droite de toutes applications. Fermer le logiciel uniquement lors d'un retour à l'écran d'accueil est un peu long. Nous pourrions ainsi essayer de gérer le stockage des tables de hachages dans le XML lors d'une fermeture intempestive de l'application.

## VI. Conclusion

Ce projet a été bénéfique du fait de toutes les connaissances que nous avons dû apprendre sur le langage Java et la modélisation UML.

Il fallait bien réfléchir à la solution que nous voulions mettre en place pour ensuite commencer à modéliser une solution envisageable.

Nous avons ainsi découvert beaucoup de choses sur le langage Java comme la lecture de fichiers XML à l'aide d'une API JDOM, la gestion de tables de hachage et les outils graphiques Swing et AWT.

Enfin, l'organisation et la séparation du travail nous a permis de mieux appréhender un travail sur projet que l'on pourrait rencontrer dans nos futurs stages ou emplois. Gérer les aptitudes et points forts de chacun pour réussir à tirer le meilleur de nous et créer une application simple mais efficace fut une tâche enrichissante.