2016-2017

ENCEV SIMON, ID BOURHIM FARID

Système de gestion d’une bibliothèque

**PROJET SGBD**

Table des matières

[Cas d’étude 3](#_Toc472089120)

[Schéma Entité-association 4](#_Toc472089121)

[Analyse du schéma 5](#_Toc472089122)

[Contraintes d’intégrité 5](#_Toc472089123)

[Schéma relationnel 6](#_Toc472089124)

[Schéma applicatif 7](#_Toc472089125)

[Sécurité 7](#_Toc472089126)

[Présentation graphique 10](#_Toc472089127)

[Winform : Choix du mode de connexion 10](#_Toc472089128)

[Winform : GUI Administrateur 10](#_Toc472089129)

[Gestion des livres 10](#_Toc472089130)

[Gestion des exemplaires 11](#_Toc472089131)

[Gestion des emprunts et des retards 12](#_Toc472089132)

[Volet Lecteurs 12](#_Toc472089133)

[Winform : Gui Lecteur 13](#_Toc472089134)

[Section d’emprunts 13](#_Toc472089135)

[Section des réservations 13](#_Toc472089136)

[Section d’historique 14](#_Toc472089137)

[Section de gestion de profil 15](#_Toc472089138)

[Conclusion 15](#_Toc472089139)

# Cas d’étude

Dans le cadre du projet de développement SGBD de cette année scolaire, nous avons été mené à réaliser un projet autour d’une application de gestion de bibliothèques.

L’application qui fait l’objet de ce rapport a pour but de permettre une bonne gestion fonctionnelle au sein d’une bibliothèque en matière de gestion de livres, des membres, des emprunts.

On gardera les informations des livres empruntés, par qui, quand et rentré quand, ainsi que le prix payé pour chaque emprunt afin de sortir les statistiques par livre emprunté.

L’application comprend 3 parties :

* Une application serveur
* Une base de données
* Une interface client pour les lecteurs ou bibliothécaires

L’application serveur gèrera toutes transactions avec la base de données et les clients.

Elle sera composée de :

* 2 DAL. Une en ADO et l’autre utilisant Entity Framework.
* Une Librairie Business Layer
* Une Librairie Entities
* Une Librairie Error
* Une Librairie service WCF

L’application Client sera quant à elle composée d’une application WinForms liée au service WCF de l’application serveur.

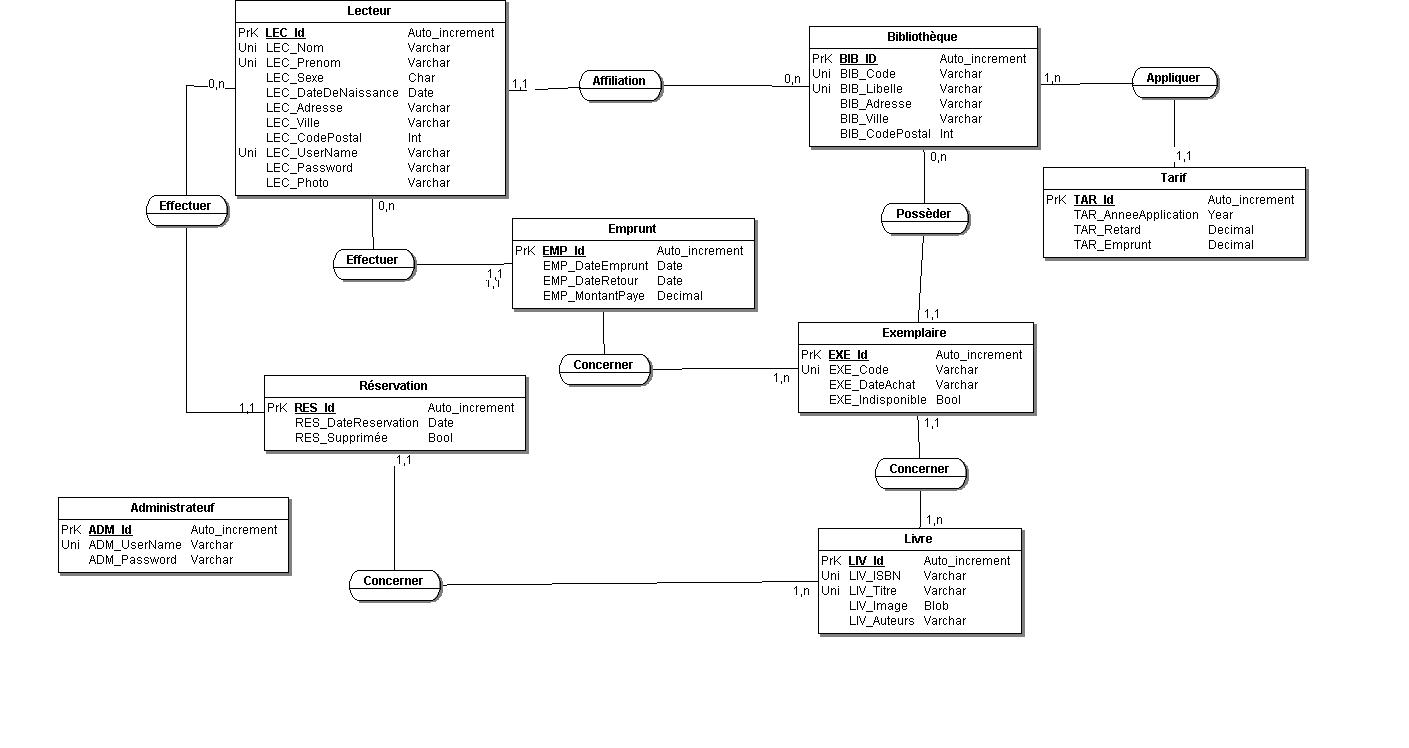
L’application cotée client pour les bibliothécaires doit permettre de :

* Rechercher et créer des exemplaires de livres
* Ajouter un livre par son ISBN via l’API GoogleBooks
* Retrouver la liste des retardataires et le prix qu’ils devraient payer au moment de la consultation de cette liste et cela pour l’ensemble des bibliothèques.
* Pouvoir gérer le retour d’un exemplaire d’un emprunteur et indiquer le montant à payer.

L’application cotée client pour les lecteurs doit permettre de :

* S’identifier
* Emprunter ou réserver un livre depuis la bibliothèque à laquelle il s’est connecté.
* Pouvoir rechercher un livre dans le but de le réserver, l’emprunter ou avoir des informations sur celui-ci.
* Avoir la liste des disponibilités des réservations en fonction de la bibliothèque ou on est connectée et emprunter en fonction de cette bibliothèque
* Avoir la liste des réservations du lecteur pour toutes les bibliothèques.
* Lors d'un emprunt, si le l'exemplaire est indisponible, proposer de faire une réservation
* Pouvoir consulter la liste des retards du lecteur et combien il doit payer au total et par retard
* Avoir la liste de réservation du lecteur valable dans toute les bibliothèques et lorsqu’un exemplaire est disponible, celui-ci lui est proposé lorsqu’il se connecte à l’application.

# Schéma Entité-association



## Analyse du schéma

## Contraintes d’intégrité

1. Entité : Lecteur

* Le champ « BIB\_Id » est obligatoire puisqu’il réfère la bibliothèque principale du lecteur.
* Le champ «Sexe » ne peut prendre que la valeur ‘M’ ou ‘F’.
* Le champ « usename » est unique pour assurer l’unicité de la donnée contenue dans ce dernier.

1. Entité : Bibliothèque

* Le champ « BIB\_Code » obligatoire.
* Le libellé « BIB\_libelle » et unique pour assurer la contrainte d’unicité de champ.

1. Entité : Administrateur

* Le champ « BIB\_Id » ne permet pas nul puisqu’il réfère la bibliothèque principale que l’administrateur gère.

1. Entité : Tarif

* Un tarif à une validité d’un an avec date de début et date de fin.
* Le champ « BIB\_Id » ne peut pas être nul puisqu’il réfère la bibliothèque qui applique ce tarif.

1. Entité : Livre

* Le numéro ISBN à une longueur de 10 entiers.

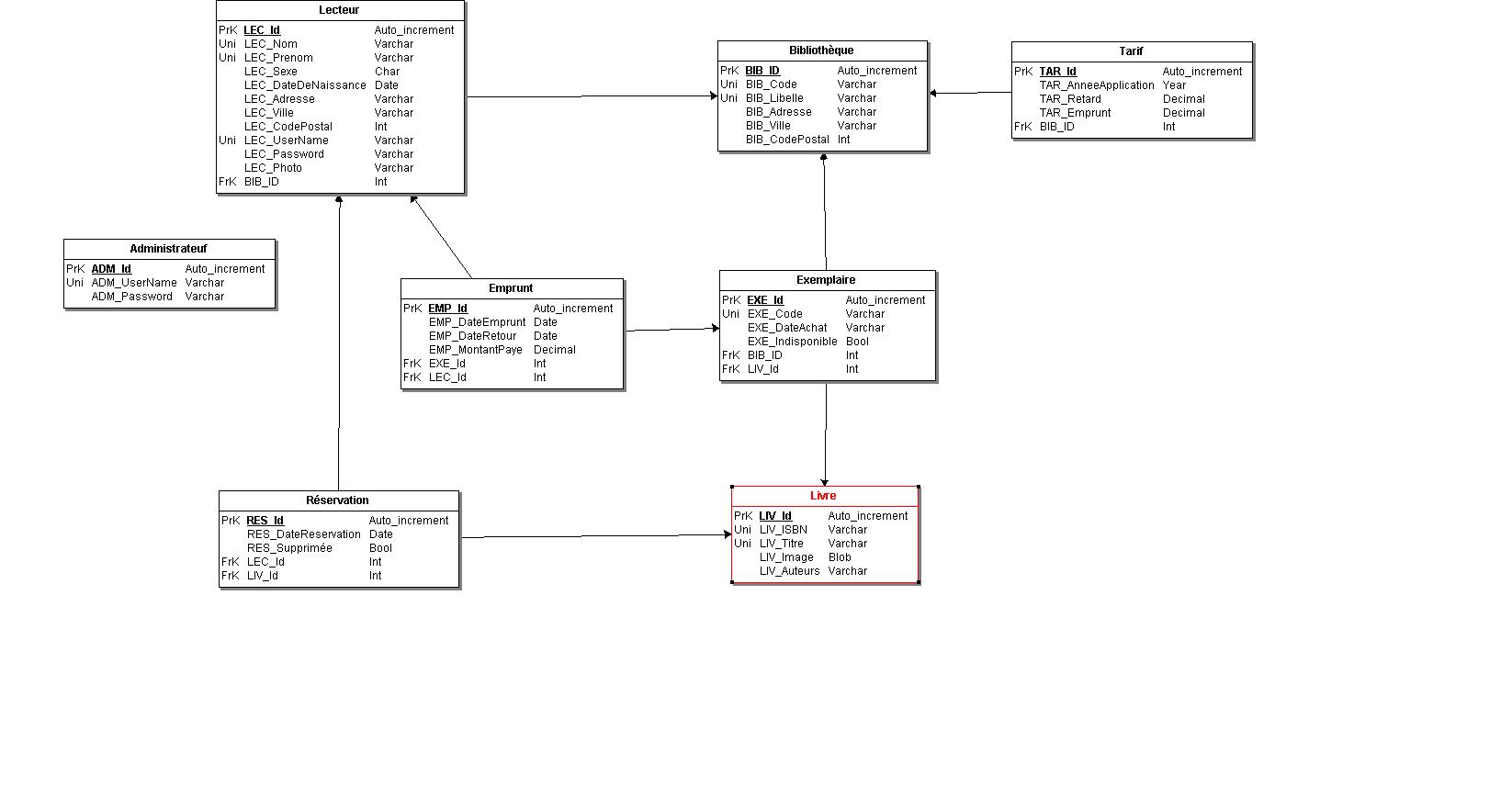
1. Entité : Exemplaire

* Le champ « Date d’achat » ne permet pas la valeur nulle et prend le format : jj-mm-aa.
* Le champ « LIV\_Id » ne permet pas la valeur nulle et doit référer un livre existant.
* Le champ  « BIB\_Id » ne permet pas la valeur nulle et doit référer la bibliothèque qui le possède.

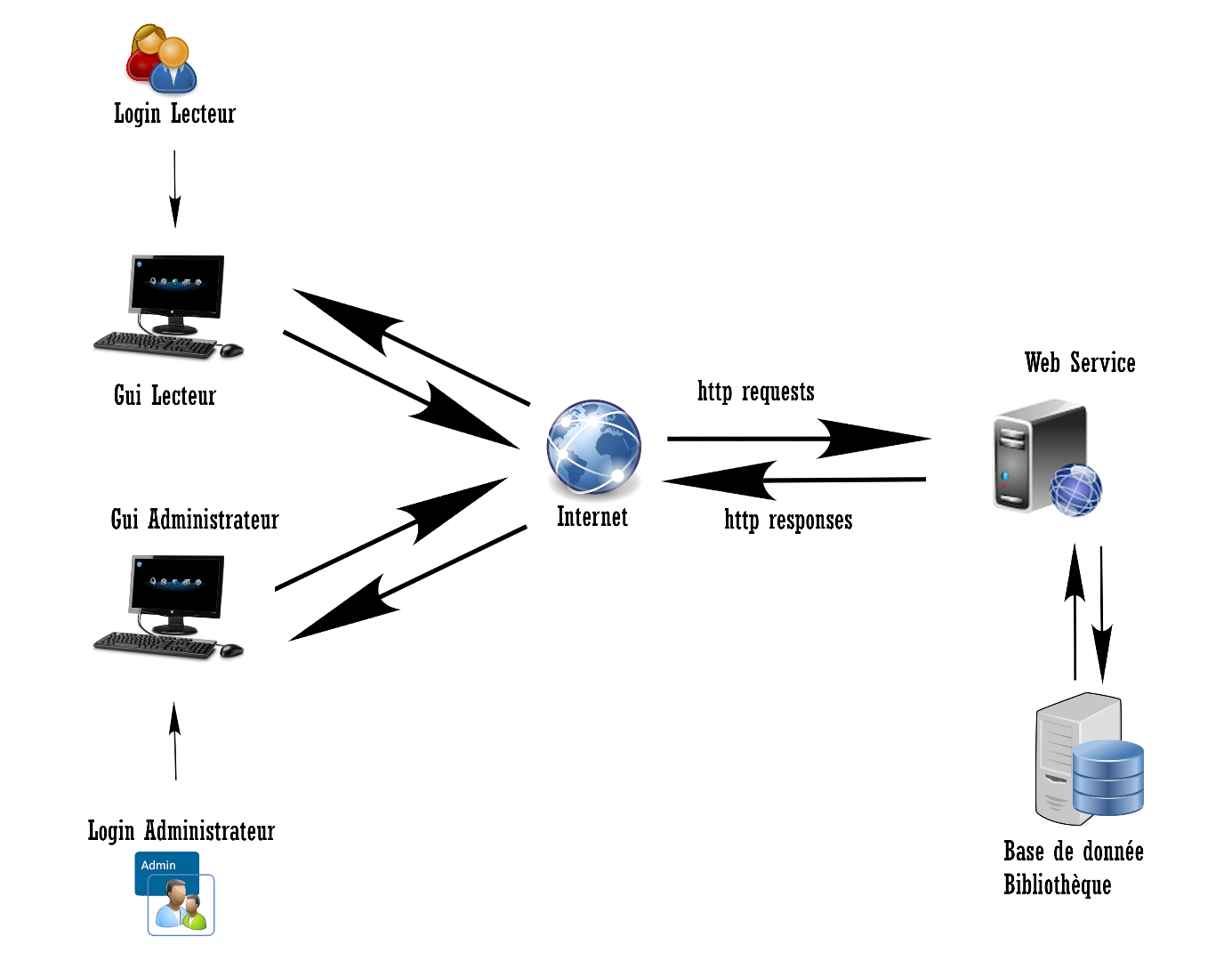
1. Entité : Emprunt

* LEC\_Id ne permet pas la valeur nulle car il réfère le lecteur qui a effectué l’emprunt.
* EXE\_Id ne permet pas la valeur nulle puisqu’il refère l’Id de l’exemplaire emprunté.
* RES\_Id ne permet pas la valeur nulle car il refère la reservation qui a généré l’emprunt.

# Schéma relationnel



# Schéma applicatif



# Sécurité

L’application de gestion a été conçue sur deux axes majeurs « administrateur » et  « lecteur ». pour assurer une bonne sécurité et bonne protection de données, nous avons opté a n’autorisez l’accès a notre base de donnée qu’à travers des procédures stockées, et ce sous le schéma respectif de chaque type d’utilisateur qui a été aussi conçu pour cette fin.

Au niveau de la gestion des connexions à la base des données, tous les utilisateurs de type « lecteur » utilisent une ‘’connectionstring’’ préalablement établie pour permettre la connexion en mode lecteur. Ainsi que les utilisateurs de type « administrateur » utilisent celle qui permet la connexion en mode administrateur.

Dans le tableau suivants on trouve la liste des utilisateurs qui peuvent accéder a la base de données et leurs type d’utilisateur ;

|  |  |  |  |
| --- | --- | --- | --- |
| Utilisateur | Type | Nom d’utilisateur | Mot de passe |
| Toto | Administrateur | toto | toto |
| Admin | Administrateur | admin | admin |
| Farid | Lecteur | farid | farid |
| Simon | Lecteur | simon | simon |
| Gerard | Lecteur | gerard | gerard |
| Etienne | Lecteur | etienne | etienne |

# Gestion des erreurs

On trouve dans le tableau suivant la liste des erreurs importantes qui sont soulevée par l’application.

|  |
| --- |
| Erreur |
| RAISERROR (20000,-1,-1, 'L"exemplaire que vous essayez d"emprunter est actuellement indisponible') |
| RAISERROR (21000,-1,-1, 'vous avez emprunté trop de livres') |
| RAISERROR (22000,-1,-1, 'Vous ne pouvez pas emprunter de livre tant que vous n''avez pas reglé vos retards') |
| RAISERROR (24000,-1,-1, 'Action impossible, vous avez déjà reservé ce livre') |
| RAISERROR (25000,-1,-1, 'Login ou password incorrect') (LEC\_Login) |
| RAISERROR (26000,-1,-1, 'Login ou password incorrect') (ADM\_Login) |

# Procédures stockées

On trouve dans les deux sections suivantes, la liste des procédures stockées importantes qui sont utilisées et leur résultat.

## Côté administrateur

|  |  |
| --- | --- |
| Procédure stockée | Résultat attendu |
| ADM\_Login | Vérifie si l’utilisateur existe en tant qu’administrateur. |
| EMP\_ALLEmpruntsEnCours | Affiche tous les emprunts en cours dans l’ensemble des bibliothèques. |
| EMP\_ListRetardatairesAllBib | Affiche tous les retardataires au niveau de toutes les bibliothèques. |
| EMP\_NbEmpruntsByLec\_Id | Affiche le résultat du calcul des emprunts d’un lecteur passé en paramètre. |
| EMP\_NbRetardByLec\_Id | Affiche le résultat du clacul des retards d’un lecteur passé en paramètre. |
| EXE\_AllExemplaireBybBib\_Id | Affiche les exemplaires disponibles dans une bibliothèque passée en paramètre. |
| EXE\_AllExemplaireByLibelle | Affiche le résultat de la recherche des exemplaires par libellé d’exemplaire passé en paramètre. |
| EXE\_AllExemplaireByExe\_Code | Affiche le résultat de la recherche des exemplaires par code d’exemplaire passé en paramètre. |
| EXE\_AllExemplaireByTitre | Affiche le résultat de la recherche des exemplaire par titre (passé en paramètre) du livre auquel il fait référence. |
| EXE\_CreerExemplaire | Procédure qui crée un exemplaire via éléments qui constituent ce dernier, ceux-ci sont passé en paramètre |
| EXE\_RetournerExemplaire | Procédure qui permet de rendre disponible un exemplaire en clôturant l’emprunt de ce dernier. |
| Liv\_AjoutLivre | Permet d’ajouter un livre grâce aux éléments qui le constituent et passés en paramètre. |
| LIV\_UpdateLivre | Met à jour un livre grâce aux éléments passés en paramètre qui le constituent. |
| RES\_ListeReservatioEnCours | Affiche la liste des réservations des lecteurs pour l’ensemble des bibliothèques. |

## Côté lecteur

|  |  |
| --- | --- |
| Procédures | Résultat attendu |
| EMP\_AllEmpruntsByLec\_Id | Affiche tous les emprunts d’un lecteur connecté à sa session. |
| EMP\_AllEmpruntsEnCoursByLec\_Id | Affiche tous les emprunts en cours d’un lecteur connecté. |
| EMP\_EllEmpruntsRenduByLec\_Id | Afficher tous les emptunts clôturés d’un lecteur connecté. |
| EMP\_ALlRetardsAllBibByLec | Afficher les emprunts en retard d’un lecteur connecté. |
| EXE\_ALlExemplairesByBib | Afficher les exemplaires détenus par la bibliothèque à laquelle le lecteur est connecté. |
| EMP\_RetardsAllBibByLec | Afficher les retards d’un lecteur connecté par rapport à la bibliothèque à laquelle il est connecté. |
| EXE\_EmprunterExememplaire | Permet d’emprunter un exemplaire. |
| LEC\_Login | Vérifie le login d’un lecteur. |
| RES\_ReserverLivre | Procédure qui permet de réserver un livre. |
| RES\_ListeReservationByLec\_Id | Afficher la liste des réservations d’un lecteur connecté. |

# Présentation graphique

## Winform : Choix du mode de connexion

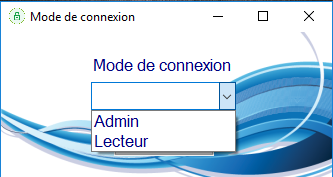


Figure - Fenêtre de choix du mode de connexion

Au lancement de notre application, cette fenêtre est le premier à s’afficher pour permettre à utilisateur de choisir son mode de connexion en fonction de son rôle applicatif. Une fois le mode sélectionné, l’utilisateur est dirigé soit vers l’application correspondante a son mode de connexion et ce en passant bien évidement par une phase d’authentification.

## Winform : GUI Administrateur

### Gestion des livres

La fenêtre suivante représente le volet de premier contact avec le bibliothécaire. Il y trouve la liste des livres que l’ensemble des bibliothèques gèrent. Il peut aussi effectuer des recherches de livre sur bas de son titre ou de son ISBN ainsi qu’ effectuer des ajouts de livre que ce soit manuellement, en introduisant les informations du livre , ou en passant par le Web Api de Google books qui permet l’ajout automatique de livres depuis un ISBN.

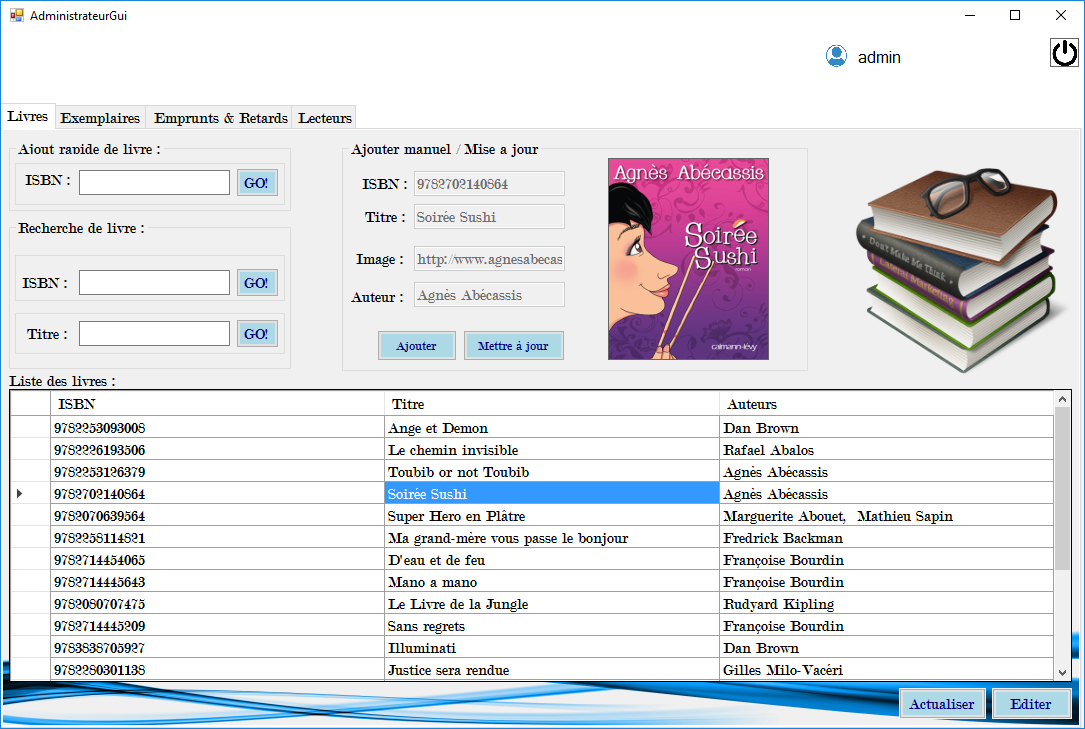


Figure - Fenêtre : Gestion des livres (côté admin)

### Gestion des exemplaires

Dans cette section, le bibliothécaire peut ajouter des exemplaires de livres, il peut aussi parcourir la liste des exemplaires et effectuer une recherche parmi ceux-ci que ce soit par code d’exemplaire ou son titre.

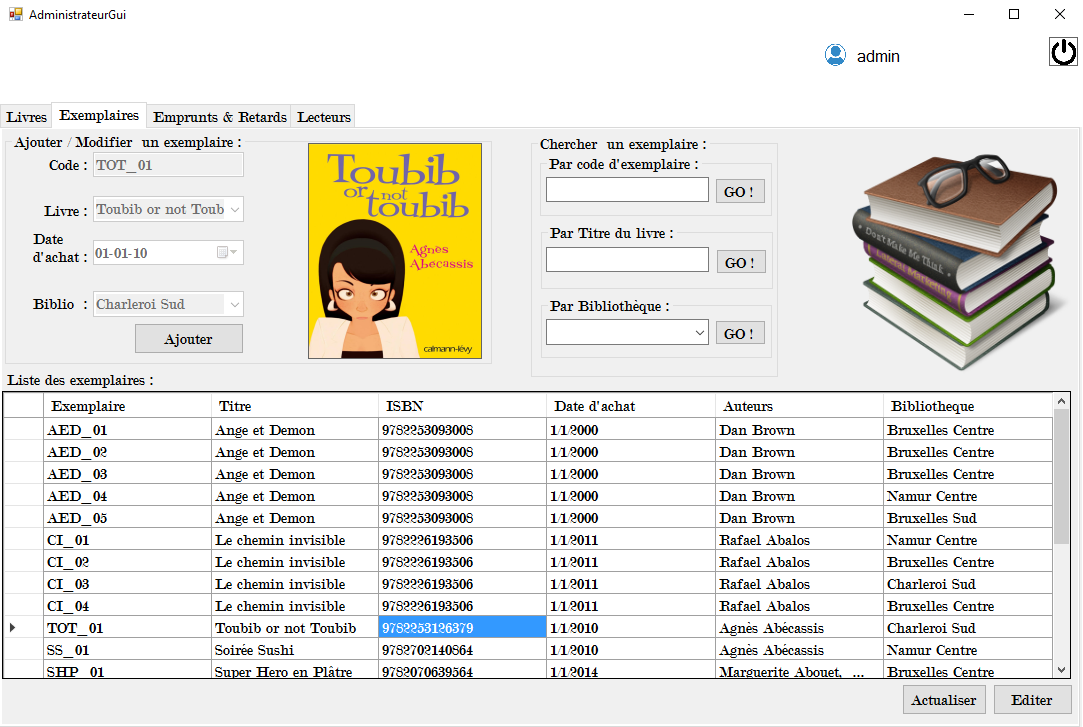


Figure – Fenêtre : Gestion des exemplaires (côté admin)

### Gestion des emprunts et des retards

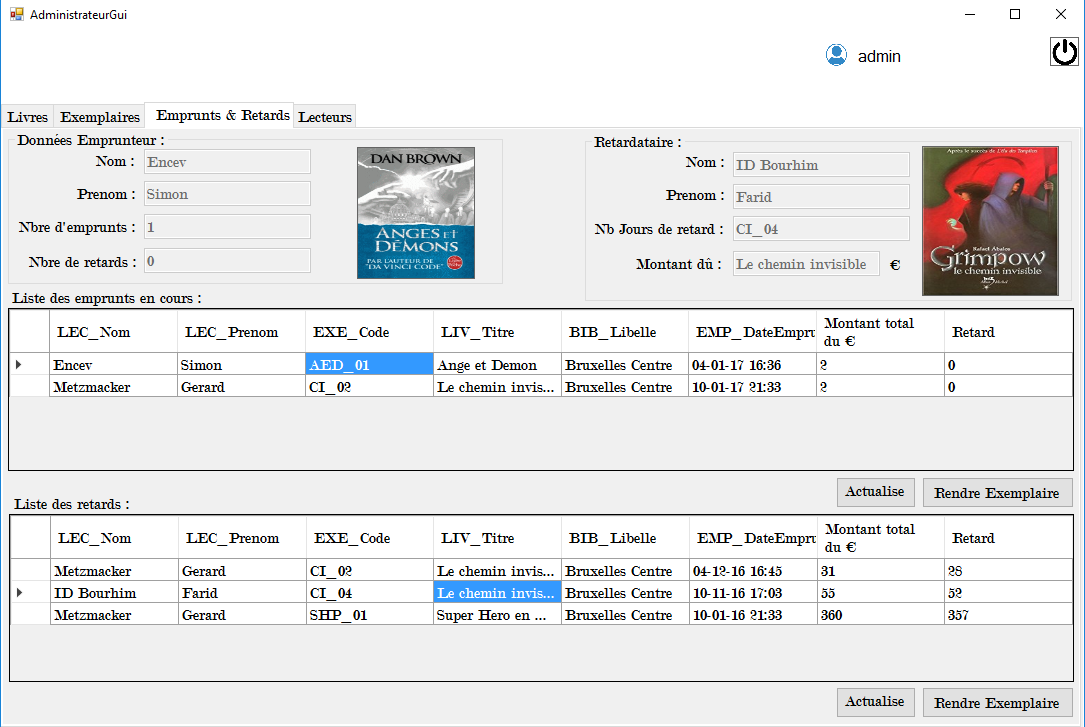
Cette section permet au bibliothécaire d’avoir la liste des emprunts en cours et les retards ainsi qu’enregistrer les exemplaires retournés.

Figure – Fenêtre : Gestion des emprunts (côté admin)

### Volet Lecteurs

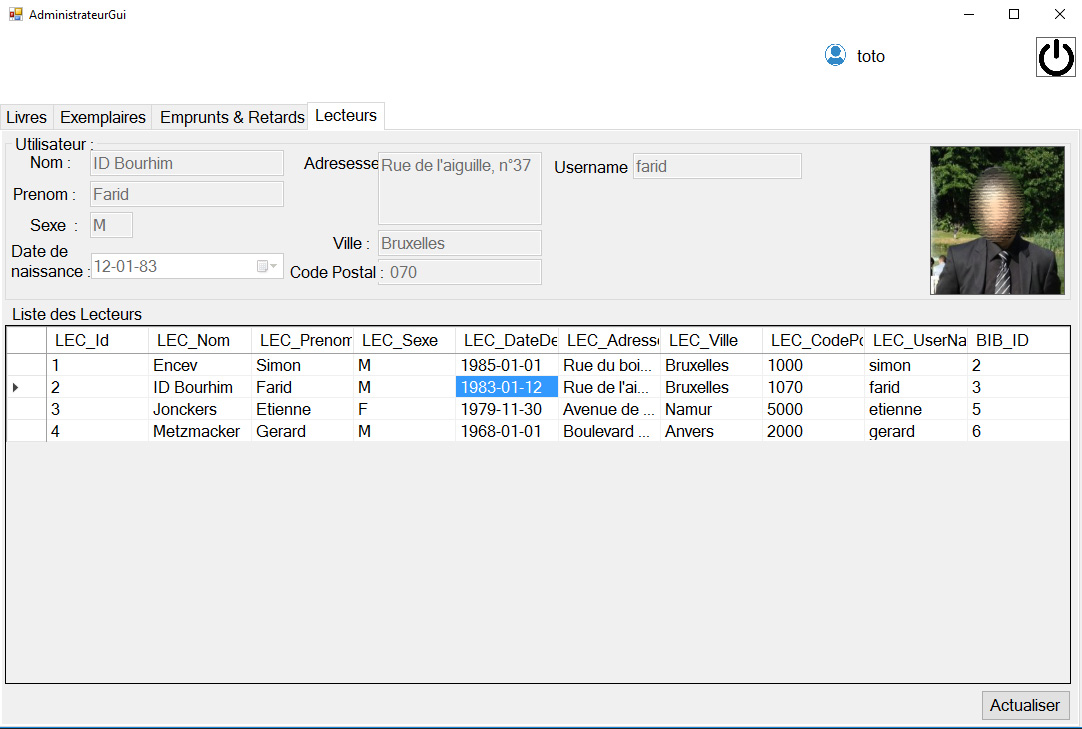
A travers ce volet, le bibliothécaire peut retrouver la liste des lecteurs pour toutes les bibliothèques ainsi que leurs informations.

Figure Fenêtre :Gestion des lecteurs (côté admin)

## Winform : Gui Lecteur

### Section d’emprunts

A travers cette section, le lecteur peut effectuer des recherches de livres par ISBN ou par titre pour la bibliothèque à laquelle il est connecté et les emprunter. Si l’exemplaire emprunté fait l’objet d’une réservation, celle-ci est automatiquement supprimée. Si l’exemplaire en question n’est pas disponible, l’application lui propose de réserver le livre correspondant. Le lecteur à une vue sur ses emprunts en cours et leurs montants ainsi que les retards.

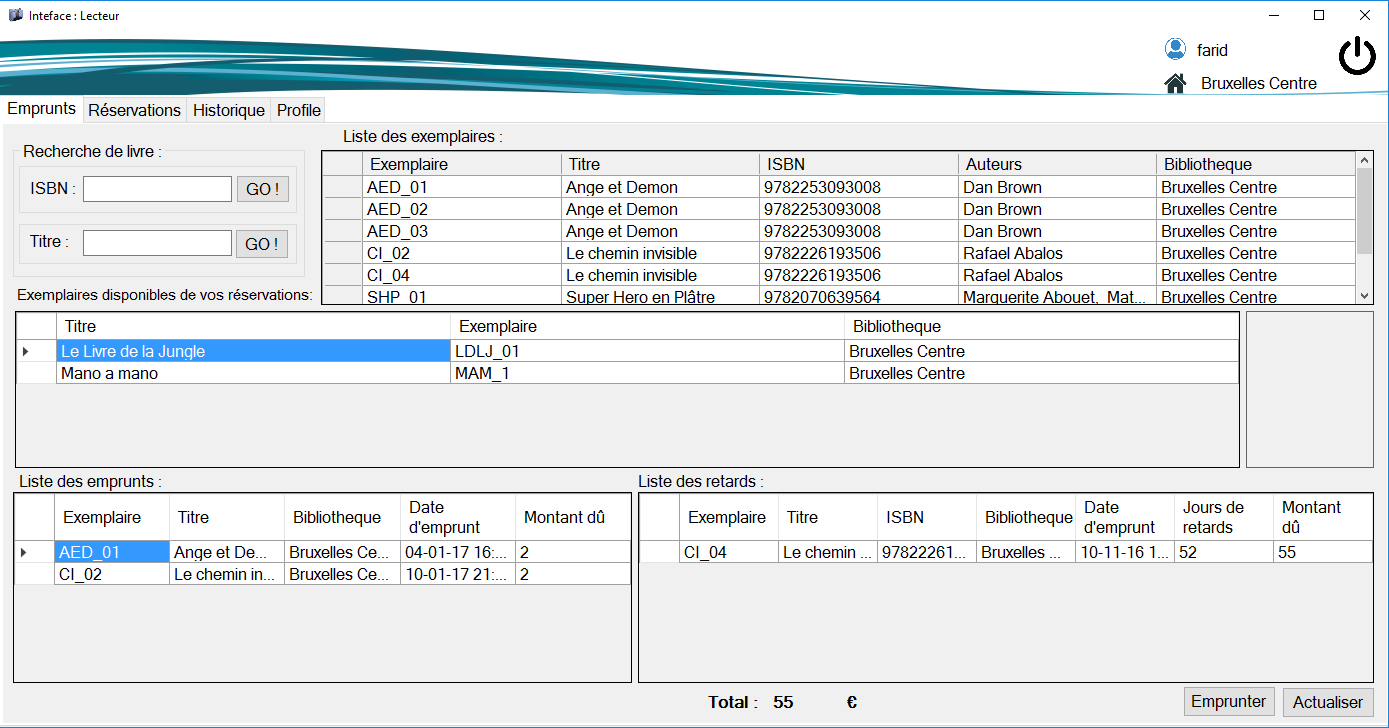


Figure - Fenêtre :Emprunts (côté lecteur)

### Section des réservations

A travers cette section, le lecteur peut effectuer des recherches sur les livres via leur ISBN ou leur titre afin de les réserver. Il y voit aussi la liste de réservations en cours.

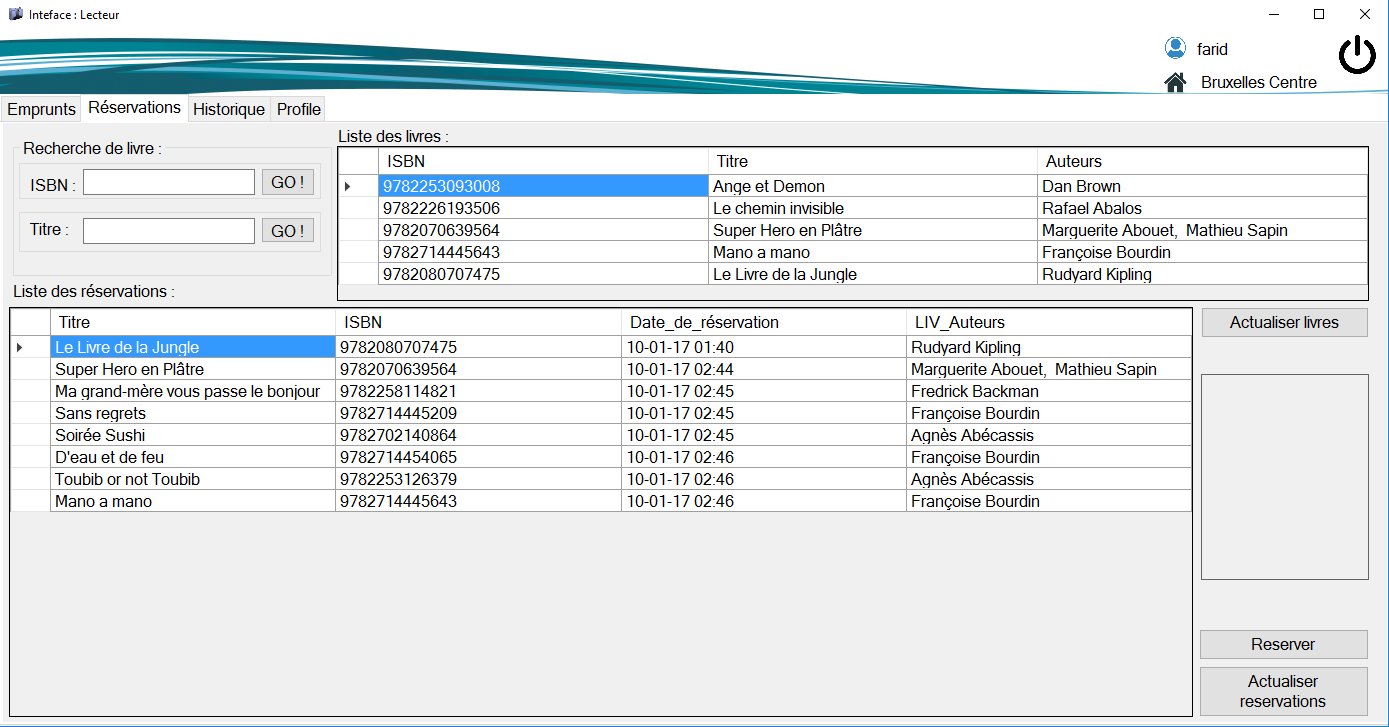


Figure - Fenêtre : Réservations (côté lecteur)

### Section d’historique

A travers cette section, le lecteur peut voir l’historique de tous ses emprunts clôturés ainsi que leurs montants.

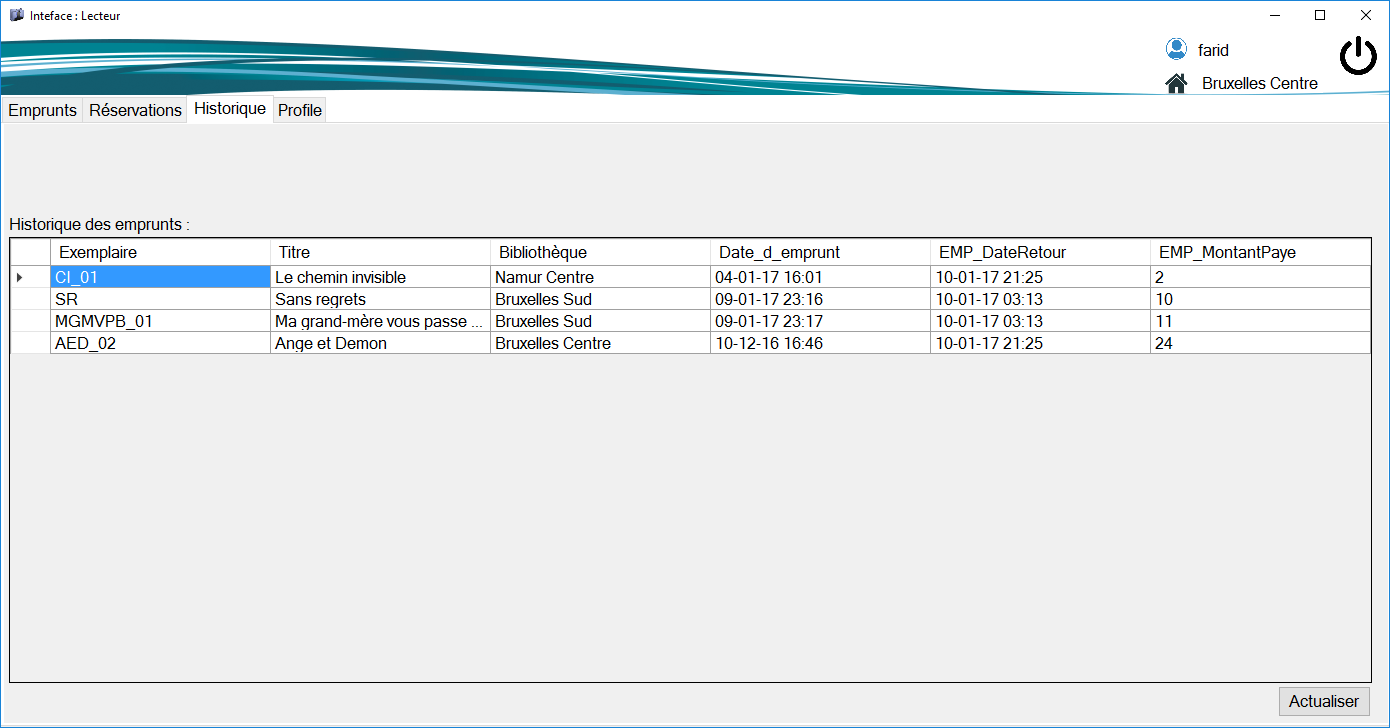


Figure Fenêtre : Historique (côté lecteur)

### Section de gestion de profil

Le lecteur peut, à travers cette section, voire ses informations personnelles et les modifier en cas de besoin.



Figure - Fenêtre : Profil (côté lecteur)

# Conclusion

Au cours de ce projet, nous avons pu mettre en application les connaissances acquises pendant les séances de module « Projet SGBD » en vue de la création d’une base de données sécurisée et une interface utilisateur pouvant interagir avec cette dernière.

Ce projet qu’on a considéré au début d’être simple a réalisé s’est avéré plus difficile qu’il en avait l’air. La grande partie du temps que le projet pris était consacré à l’établissement d’une base données complète et efficace en matière de gestion des couts d’accès au donné ainsi que les redondances.

Ce projet nous permet de mieux s’organiser pour aboutir la réalisation d’un projet en travaillant sur un projet partagé sur ‘’Github’’ qui permet vraiment de bien et voir le projet se compléter avec l’apport de chaque membre participant à ce projet.