

Mini projet java: Gestion d'une bibliothèque

Travail réalisé par Yosr BARGHOUTH et Maissa DRIDI (L2CS02)
16 Décembre 2021

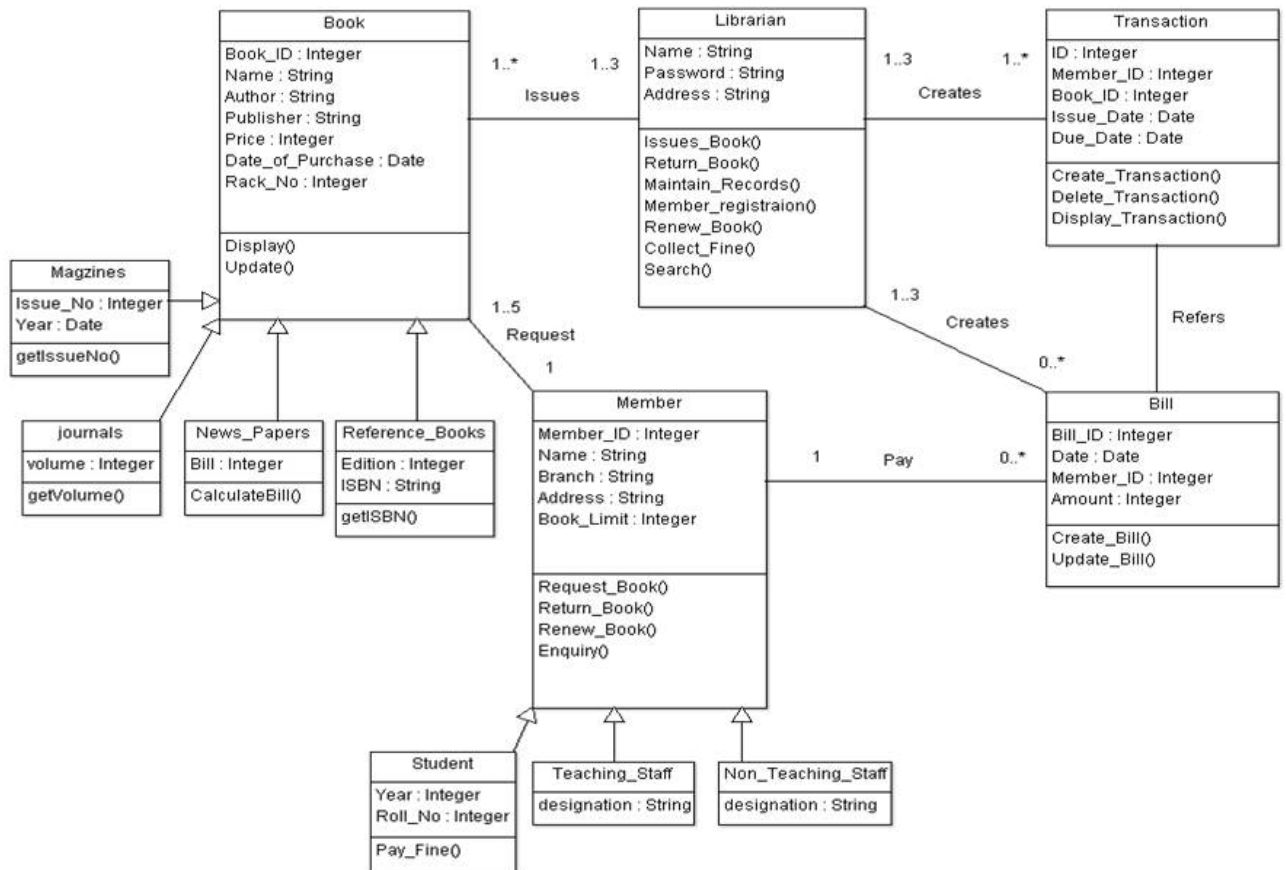
I. Introduction

Notre projet vise une gestion **facile** et **efficace** d'une bibliothèque. Cela assurera le bon fonctionnement des événements divers (consulter les livres disponibles, emprunter un livre, rendre un livre...) et d'autres événements peuvent être réalisés selon les privilèges attribués.

II. Diagramme de cas d'utilisation (cas général)



III. Diagramme de classes (traité dans le projet)



dig. Class Diagram for Library Management System

IV. Fonctionnement:

A. Configuration des détails de la connexion:

- Pour se connecter à MySQL, l'utilisateur doit fournir les paramètres suivants: username, password, numéro de port et nom de la base de données.
- Ces paramètres permettent de créer un objet **java.sql.Connection** qui assure la communication entre la base de données sur MySQL et l'application Java.
- Si l'utilisateur n'a pas une base de données sur MySQL, il peut en créer une. En cliquant sur "initialiser la base de données", les tables **users**, **livres** et **emprunts** sont créées et des lignes sont ajoutées à users et livres.

- Si l'utilisateur dispose de la base de données avec les tables et les lignes déjà prêtes, il peut passer à l'interface du Login.

B. Login:

- Un utilisateur s'identifie avec son "username" qui est le numéro de la CIN et le mot de passé associé à son compte.
- Lors de l'initialisation de la base de données, deux utilisateurs sont créés:
 - Un admin avec cin=12345678 et mdp=admin
 - Un étudiant (utilisateur régulier) avec cin=87654321 et mdp=etudiant
- L'interface de login dispose d'un processus de vérification faisant appel à la base de données des utilisateurs inscrits dans la bibliothèque.
 - Si le mot de passé entré est erroné ou bien le n° de la CIN ne se trouve pas dans la base de données, l'application signale à l'utilisateur le problème.
- Suivant le statut de l'utilisateur, ce dernier est soit dirigé vers l'interface d'accueil de l'admin (authentication.HomepageAdmin) soit l'interface d'accueil d'un utilisateur régulier (authentication.HomepageRegular).

C. Interface Admin:

1. Gestion utilisateurs:

- a) L'admin de la bibliothèque peut voir les utilisateurs inscrits (cin, nom, prenom, email, statut, mot de passe).
- b) L'admin peut chercher un utilisateur par le numéro de la cin
- c) L'admin peut ajouter un utilisateur en fournissant les informations nécessaires à condition que le n° de la cin ne figure pas dans la table users sinon il s'agit d'une violation de la contrainte "clé primaire".
- d) L'admin peut supprimer un utilisateur s'il existe en fournissant son n° de CIN
- e) L'admin peut mettre à jour les données d'un utilisateur. En fournissant son n° de CIN, les champs sont remplis avec les données préexistantes. L'admin apporte les modifications et valide en cliquant sur **modifier**

2. Gestion Emprunts:

- a) L'admin peut voir l'ensemble des emprunts effectués au sein de la bibliothèque.
- b) L'admin peut chercher l'ensemble des emprunts effectués par un utilisateur y compris l'emprunt courant.
- c) L'admin peut chercher l'ensemble des emprunts dont fait l'objet un livre particulier identifié par un isbn.

3. Gestion Livres

- a) L'admin peut voir tous les livres de la bibliothèque identifiés par leur isbn, leurs titres, leurs auteurs et leurs disponibilités
- b) L'admin peut chercher un livre soit par l'isbn, le titre ou l'auteur. Les résultats sont renvoyés grâce à des requêtes à la base de données.
- c) L'admin peut modifier, ajouter ou supprimer un livre.

D. Interface utilisateur régulier

1. Consulter livres + recherche:

- a) Un utilisateur régulier peut voir tous les livres de la bibliothèque identifiés par leur isbn, leurs titres, leurs auteurs et leurs disponibilités.
- b) Un utilisateur régulier peut chercher un livre soit par l'isbn, le titre ou l'auteur.

2. Emprunt d'un livre:

- Un utilisateur régulier ne peut emprunter qu'un seul livre à la fois.
- Pour effectuer un emprunt, l'utilisateur fournit l'ISBN du livre qu'il veut emprunter et peut vérifier s'il s'agit bien du livre qu'il veut en obtenant le titre, l'auteur et la disponibilité.
- L'application vérifie si l'utilisateur a un emprunt courant.
(cin, isbn, dateemprunt, date retour, retour)
- Si l'utilisateur a déjà un emprunt courant ou si le livre n'est pas disponible, l'application annule l'emprunt.
- En effectuant l'emprunt d'un livre disponible dans la librairie, la date de retour est calculée par défaut en fonction de la date du jour. La durée de l'emprunt est de 15 jours.

3. Rendre un livre emprunté:

- Un utilisateur régulier ne peut rendre un livre qu'après l'avoir emprunté.
- L'interface "Rendre un livre" permet à l'utilisateur de vérifier les détails de l'emprunt courant.
- En rendant le livre, la date de retour est actualisée et devient la date du jour, le livre redevient disponible pour l'emprunt et l'emprunt est supposé finalisé (la valeur de retour est 1).

4. Historique des emprunts:

- Un utilisateur régulier peut voir l'ensemble des emprunts qu'il a effectués depuis son inscription y compris l'emprunt courant. La table comprend les détails de l'emprunt tels l'isbn du livre, la date de l'emprunt, la date de retour ou bien le deadline et le statut de l'emprunt.

V- Structure du projet:

- Package authentication

- Classe **User.java** : attributs de l'utilisateur, constructeurs, getter et setter.

- **UsersDB.java** : classe établissant la connexion avec mysql. Ses méthodes permettent le retour des résultat des requêtes SQL (LID) ou bien la mise à jour de la base de données (LMD)
ex: GetAllUsers retourne une liste d'objets de classe User, AddUser ajoute un User dans la table users..
- Interface graphique **init.java**: permet à l'utilisateur de paramétrer la connexion avec MySql, de tester la connexion et d'initialiser la base de données.
- Classe **initdb.java** qui crée l'objet de Connection (de la librairie prédéfinie java.sql) utilisé au cours du programme pour assurer la connexion avec MySQL.
- Interface graphique **Loggingin.java** qui assure l'accès au compte de l'utilisateur.
- Interface graphique **HomepageRegular.java** : interface d'accueil pour l'utilisateur régulier.
- Interface graphique **HomepageAdmin.java** : interface d'accueil pour l'utilisateur Admin.
- Interface graphique **UsersAdmin.java** : interface de gestion des utilisateurs pour l'Admin.
- Interface graphique **UserFileUI.java** qui affiche les détails d'un utilisateur.
- **Package emprunts**
 - **Classe Emprunt.java**: attributs de l'emprunt, constructeurs, getter et setter.
 - **Classe EmpruntsGestion.java** : classe établissant la connexion avec mysql. Ses méthodes permettent le retour des résultat des requêtes SQL (LID) ou bien la mise à jour de la base de données (LMD)
ex: GetAllEmprunts retourne une liste d'objets de classe Emprunt, AddEmprunt ajoute un Emprunt dans la table emprunts..
 - Interface graphique **EmpruntsAdmin.java**: interface de gestion des emprunts par l'Admin
 - Interface graphique **EmpruntRetourUI.java**: interface de retour du livre emprunté par un utilisateur régulier
 - Interfaces graphiques renvoyant les tables comportant des résultats de recherche **EmpruntFileCinUI.java** et **EmpruntFilelsbnUI.java**
- **Package livres**
 - **Classe Livre.java**: attributs du livre constructeurs, getter et setter.
 - **Classe LivresGestion.java** : classe établissant la connexion avec mysql. Ses méthodes permettent le retour des résultat des requêtes SQL (LID) ou bien la mise à jour de la base de données (LMD)

ex: GetAllLivres retourne une liste d'objets de classe Emprunt, AddLivre ajoute un Livre dans la table livres..

- Interface graphique **LivresRegular.java** interface permettant à un utilisateur régulier de consulter les livres et faire la recherche.
- Interface graphique **LivresAdmin.java** interface permettant à un admin de gérer les livres
- Interfaces graphiques **LivreFileISbnUI.java**, **LivreFileAuteurUI.java**, **LivreFileTitreUI.java** renvoyant les résultats de recherche des livres.