

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 2
Nom, prénom : Epagneaud Julien		N° candidat : 02149044890
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : ..10. / ..05. /..2026..
Organisation support de la réalisation professionnelle Médiathèque municipale fictive		
Intitulé de la réalisation professionnelle Conception et développement d'une application Java Swing de gestion de médiathèque — MediaGest BTS		
Période de réalisation : **/01/2026 - 10/05/2026 Lieu : Domicile et Lycée		
Modalité : <input type="checkbox"/> Seul(e) <input checked="" type="checkbox"/> En équipe		
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données		
Conditions de réalisation⁵ (ressources fournies, résultats attendus) Projet réalisé dans le cadre du BTS SIO option SLAM. La réalisation consiste à concevoir et développer une application client lourd de gestion de médiathèque. L'application permet à un agent de médiathèque de gérer les adhérents, les documents, les emprunts et les demandes d'emprunt. Elle propose aussi un accès adhérent pour déposer une demande et un accès visiteur pour consulter le catalogue. Le projet attendu est une application Java Swing fonctionnelle, reliée à une base de données MySQL, avec authentification, jeu d'essai, interfaces graphiques, diagrammes UML, MCD.		
Description des ressources documentaires, matérielles et logicielles utilisées⁶ Les ressources documentaires utilisées sont la documentation du projet MediaGest BTS, l'expression du besoin, les user stories, le backlog produit, les diagrammes UML, le modèle conceptuel de données, le cahier de tests fonctionnels, le manuel utilisateur et le manuel administrateur. Les ressources logicielles utilisées sont Java 8, Java Swing, JDBC, MySQL/MariaDB, MySQL Connector/J 8.4.0, Eclipse et Laragon.		
Modalités d'accès aux productions⁷ et à leur documentation⁸ Les productions sont accessibles dans le dossier du projet "Projet_PHP"		

⁵ En référence aux conditions de réalisation et ressources nécessaires du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

⁶ Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

⁷ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁸ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

**ANNEXE VII-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**

Épreuve E6 - Conception et développement d'applications (option SLAM)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Documentation complète du projet MediaGest

BTS

Projet de fin d'études - BTS SIO SLAM Application Java Swing de gestion de médiathèque

Sommaire

1. Analyse

- 1.1 Expression de besoin contextualisée
- 1.2 User stories et backlog produit
- 1.3 Calendrier de réalisation selon la méthode Agile
- 1.4 Diagramme UML des cas d'utilisation

1. Conception

- 2.1 Schéma d'architecture logicielle et technique
- 2.2 Diagramme UML de classes
- 2.3 Modèle conceptuel de données
- 2.4 Description des IHM, écrans et navigation

1. Réalisation

- 3.1 Présentation du code source
- 3.2 Tests unitaires et vérifications techniques
- 3.3 Fichier d'exportation de la base de données

1. Documentation logicielle

- 4.1 Cahier de tests fonctionnels
- 4.2 Manuel administrateur
- 4.3 Manuel utilisateur

1. Éléments à compléter avant le rendu

Introduction générale

Le projet **MediaGest BTS** est une application Java fenêtrée destinée à la gestion d'une médiathèque municipale. Elle permet à un agent de médiathèque de se connecter à une interface d'administration, de consulter un tableau de bord, de gérer les adhérents, de gérer le catalogue de documents, de suivre les emprunts et de traiter les demandes d'emprunt. Elle propose aussi un accès adhérent et un accès visiteur au catalogue.

L'application répond au besoin d'une petite structure souhaitant centraliser la gestion de ses documents et de ses adhérents dans une base de données relationnelle. Elle remplace une gestion manuelle ou dispersée par un outil simple, accessible depuis un poste informatique standard. L'authentification utilise un hachage PBKDF2 afin de ne pas stocker les mots de passe en clair.

Le projet a été réalisé avec **Java Swing** pour l'interface graphique, **JDBC** pour l'accès aux données et **MySQL/MariaDB** pour la persistance. Il constitue la situation professionnelle client lourd de l'option **BTS SIO SLAM**, en complément d'une autre situation professionnelle réalisée avec une technologie web.

1. Analyse

1.1 Expression de besoin contextualisée

Contexte du projet

La médiathèque municipale souhaite informatiser la gestion de ses adhérents, de son catalogue de documents et de ses emprunts. Jusqu'à présent, ces informations peuvent être suivies de manière manuelle, dans des tableaux ou dans des documents séparés. Cette organisation peut entraîner des erreurs de suivi, des oublis de retour, des difficultés à savoir si un document est disponible et un manque de visibilité sur l'activité globale.

L'application **MediaGest BTS** propose une interface de gestion simple à destination du personnel de la médiathèque. Elle permet de centraliser les informations dans une base de données unique et d'effectuer les opérations courantes depuis une application de bureau.

Organisation cliente

L'organisation cliente est une médiathèque municipale fictive. Elle accueille des usagers inscrits, appelés adhérents, et met à leur disposition différents types de documents :

- livres
- DVD
- magazines
- mangas
- CD
- documents liés à l'informatique, à la jeunesse, au cinéma ou à la bande dessinée

Le personnel de la médiathèque utilise l'application pour administrer le catalogue, suivre les emprunts et traiter les demandes. Les adhérents peuvent se connecter à leur espace personnel pour demander l'emprunt d'un document disponible. Les visiteurs peuvent consulter le catalogue disponible sans créer de demande.

Problèmes rencontrés

La médiathèque rencontre plusieurs difficultés :

- le suivi des adhérents peut être fastidieux lorsqu'il est réalisé manuellement
- la disponibilité des documents n'est pas toujours visible immédiatement
- les emprunts et les retours doivent être suivis avec précision
- les documents peuvent être classés dans différentes catégories
- les agents ont besoin d'une vue rapide sur l'activité de la médiathèque
- les adhérents doivent pouvoir déposer une demande sans que le personnel saisisse leur nom à leur place
- les visiteurs doivent pouvoir consulter les documents disponibles sans compte
- les erreurs de saisie ou les oublis peuvent rendre le suivi moins fiable

Besoin exprimé

Le besoin principal est de disposer d'une application de bureau permettant au personnel de :

- se connecter avec un compte administrateur
- consulter un tableau de bord
- visualiser les statistiques principales
- gérer les adhérents
- gérer les documents
- classer les documents par catégorie
- rechercher rapidement un document
- enregistrer un emprunt
- marquer un document comme retourné
- suivre les emprunts en cours, terminés ou en retard
- valider ou refuser les demandes d'emprunt déposées par les adhérents

Utilisateurs concernés

Administrateur / agent de médiathèque

L'utilisateur principal est un agent de médiathèque. Il dispose d'un compte de connexion et peut accéder aux fonctionnalités de gestion.

Il peut :

- se connecter à l'application
- consulter le tableau de bord
- ajouter, modifier ou supprimer un adhérent
- ajouter, modifier ou supprimer un document
- rechercher des documents
- enregistrer un emprunt
- enregistrer le retour d'un document
- consulter les emprunts existants

Adhérent

L'adhérent représente une personne inscrite à la médiathèque et autorisée à emprunter des documents. Il peut disposer d'un compte de connexion pour accéder à son espace personnel.

Un adhérent possède :

- un nom
- un prénom
- une adresse e-mail
- un numéro de téléphone
- une date d'inscription
- un compte utilisateur associé lorsque l'adhérent s'inscrit depuis l'application

Il peut :

- se connecter à son espace personnel
- rechercher un document disponible

- déposer une demande d'emprunt sans sélectionner son nom dans une liste.

Visiteur

Le visiteur est une personne qui consulte le catalogue sans être connectée.

Il peut :

- accéder au catalogue disponible depuis l'écran de connexion
- rechercher un document par titre, auteur ou catégorie
- voir les documents disponibles
- consulter sans pouvoir déposer de demande d'emprunt.

Objectifs attendus

Les objectifs du projet sont :

- centraliser les données de la médiathèque
- simplifier les opérations courantes de gestion
- éviter les documents empruntés deux fois en même temps
- permettre une recherche rapide dans le catalogue
- fournir une base de données relationnelle structurée
- proposer une inscription adhérent et un accès visiteur
- sécuriser les comptes avec un mot de passe haché
- proposer une application client lourd opérationnelle pour l'épreuve BTS
- séparer le code en couches lisibles : modèle, DAO, service, interface

Contraintes du projet

Contraintes techniques

- application développée en Java
- interface graphique réalisée avec Swing
- accès aux données réalisé avec JDBC
- base de données MySQL ou MariaDB
- connecteur utilisé : `mysql-connector-j-8.4.0.jar`
- projet importable et exécutable depuis Eclipse
- base locale nommée `mediagest_bts`

Contraintes fonctionnelles

- l'application doit proposer une authentification
 - les mots de passe ne doivent pas être stockés en clair
 - l'application doit permettre l'inscription d'un adhérent
 - l'application doit proposer un accès visiteur limité à la consultation
 - l'application doit proposer au moins trois fenêtres
 - les données doivent être stockées dans une base relationnelle
 - la base doit contenir au moins quatre tables reliées par clés étrangères
 - l'application doit fournir un jeu de données réaliste
 - les opérations principales doivent être accessibles depuis l'interface
-

1.2 User stories et backlog produit

Acteurs du projet

Acteur	Description
Agent de médiathèque	Utilisateur connecté qui gère les adhérents, les documents, les emprunts et les demandes.
Adhérent	Personne inscrite à la médiathèque, pouvant se connecter et déposer une demande d'emprunt.
Visiteur	Personne non connectée pouvant consulter les documents disponibles.
Administrateur	Rôle applicatif utilisé pour accéder à toutes les fonctionnalités de gestion.

User stories

ID	User story	Priorité	Statut
US01	En tant qu'agent, je veux me connecter afin de sécuriser l'accès à l'application.	Haute	Réalisé
US02	En tant qu'agent, je veux consulter un tableau de bord afin d'avoir une vue globale de l'activité.	Haute	Réalisé
US03	En tant qu'agent, je veux ajouter un adhérent afin d'inscrire une nouvelle personne.	Haute	Réalisé
US04	En tant qu'agent, je veux modifier les informations d'un adhérent afin de garder ses données à jour.	Haute	Réalisé
US05	En tant qu'agent, je veux supprimer un adhérent afin de nettoyer la base lorsque c'est possible.	Moyenne	Réalisé
US06	En tant qu'agent, je veux ajouter un document afin d'enrichir le catalogue.	Haute	Réalisé
US07	En tant qu'agent, je veux modifier un document afin de corriger ses informations.	Haute	Réalisé
US08	En tant qu'agent, je veux rechercher un document afin de le retrouver rapidement.	Haute	Réalisé
US09	En tant qu'agent, je veux enregistrer un emprunt afin de suivre les documents sortis.	Haute	Réalisé
US10	En tant qu'agent, je veux empêcher l'emprunt d'un document déjà indisponible afin d'éviter les conflits.	Haute	Réalisé
US11	En tant qu'agent, je veux marquer un emprunt comme retourné afin de rendre le document disponible.	Haute	Réalisé
US12	En tant qu'agent, je veux rechercher un emprunt afin de retrouver rapidement une opération.	Moyenne	Réalisé
US13	En tant qu'agent, je veux me déconnecter afin de quitter ma session.	Moyenne	Réalisé
US14	En tant qu'adhérent, je veux créer un compte afin d'accéder à mon espace personnel.	Haute	Réalisé
US15	En tant qu'adhérent, je veux me connecter afin de déposer une demande d'emprunt.	Haute	Réalisé
US16	En tant qu'adhérent, je veux rechercher un document disponible afin de le demander plus rapidement.	Haute	Réalisé
US17	En tant que visiteur, je veux consulter le catalogue afin de voir les documents disponibles.	Moyenne	Réalisé
US18	En tant que visiteur, je veux rechercher un document afin de trouver rapidement un ouvrage.	Moyenne	Réalisé

ID	User story	Priorité	Statut
US19	En tant qu'agent, je veux valider ou refuser les demandes afin de contrôler les emprunts.	Haute	Réalisé
US20	En tant qu'utilisateur, je veux que mon mot de passe soit protégé afin d'éviter son stockage en clair.	Haute	Réalisé

Backlog produit

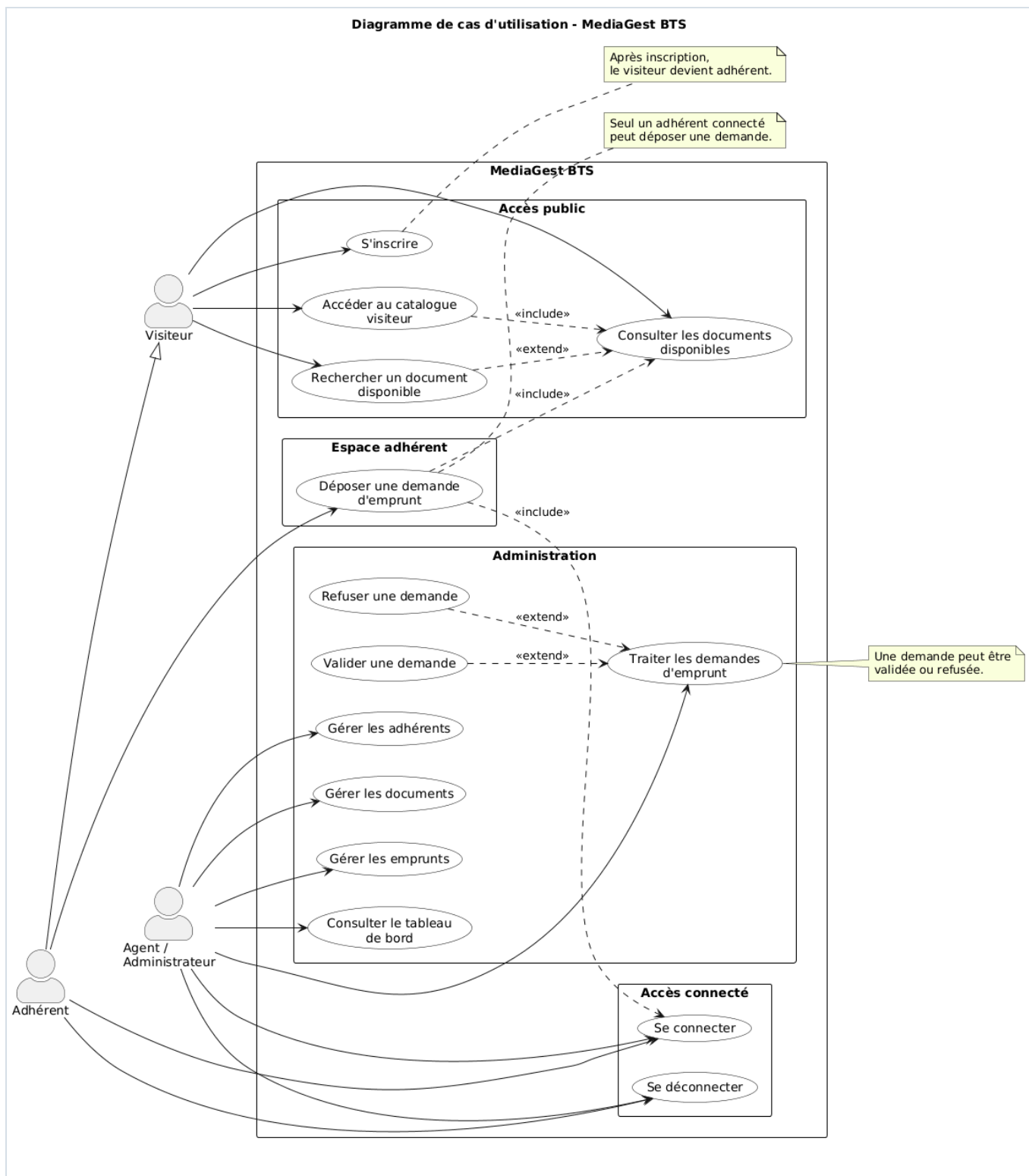
Fonctionnalité	Description	Version
Authentification	Connexion avec identifiant et mot de passe.	V1
Tableau de bord	Statistiques et derniers emprunts.	V1
Gestion des adhérents	CRUD complet des adhérents.	V1
Gestion des documents	CRUD complet des documents et recherche.	V1
Gestion des emprunts	Création, recherche, retour et suppression d'emprunts.	V1
Demandes d'emprunt	Dépôt par l'adhérent, validation ou refus par le personnel.	V1
Espace adhérent	Connexion adhérent, recherche de document disponible et dépôt de demande.	V1
Accès visiteur	Consultation et recherche dans le catalogue des documents disponibles.	V1
Inscription adhérent	Création d'une fiche adhérent et d'un compte utilisateur associé.	V1
Gestion des catégories	Catégories utilisées dans les listes et dans la base de données.	V1
Gestion avancée des droits	Ajouter des permissions plus fines que les rôles actuels <code>administrateur</code> et <code>adherent</code> .	Évolution possible
Hachage des mots de passe	Stocker les mots de passe sous forme de hash PBKDF2 avec sel.	V1
Gestion des pénalités	Calculer des pénalités en cas de retard.	Évolution possible

1.3 Calendrier de réalisation selon la méthode Agile

Le projet peut être présenté comme ayant été organisé en itérations courtes. Chaque itération produit une partie testable de l'application.

Période	Objectif	Tâches réalisées	Résultat
Sprint 1	Analyse et cadrage	Identification du besoin, choix du contexte, définition des entités principales.	Contexte et périmètre fonctionnel.
Sprint 2	Base de données	Création des tables, clés primaires, clés étrangères et jeu d'essai.	Scripts SQL <code>init_database.sql</code> et <code>demo_data.sql</code> .
Sprint 3	Architecture Java	Création des packages <code>modele</code> , <code>dao</code> , <code>service</code> , <code>ihm</code> , <code>configuration</code> , <code>utilitaire</code> .	Structure applicative claire.
Sprint 4	Authentification et tableau de bord	Création de l'écran de connexion, hachage des mots de passe et écran principal.	Accès sécurisé et statistiques.
Sprint 5	Gestion des adhérents et documents	Développement des écrans CRUD et des DAO associés.	Gestion opérationnelle du catalogue et des adhérents.
Sprint 6	Gestion des emprunts	Création, recherche, retour de document et mise à jour de disponibilité.	Cycle d'emprunt fonctionnel.
Sprint 7	Espace adhérent et visiteur	Inscription, accès visiteur, connexion adhérent, recherche et dépôt de demande.	Fonctionnalités adhérent et visiteur opérationnelles.
Sprint 8	Tests et documentation	Vérifications fonctionnelles, compilation et rédaction de la documentation.	Projet présentable à l'épreuve.

1.4 Diagramme UML des cas d'utilisation



Acteurs représentés

- **Agent de médiathèque / administrateur** : utilisateur interne de l'application. Il se connecte avec un compte personnel et accède aux fonctionnalités de gestion : adhérents, documents, emprunts et demandes d'emprunt.
- **Adhérent** : utilisateur inscrit à la médiathèque. Il peut créer un compte, se connecter à son espace personnel, consulter les documents disponibles, rechercher un document et déposer une demande d'emprunt.

- **Visiteur** : utilisateur non connecté. Il peut uniquement consulter le catalogue des documents disponibles et effectuer une recherche. Il ne peut pas déposer de demande d'emprunt.
- **Base de données** : système externe utilisé par l'application pour enregistrer les comptes, les adhérents, les documents, les emprunts et les demandes.

Cas d'utilisation principaux

Le diagramme de cas d'utilisation doit montrer les droits de chaque type d'utilisateur. Les fonctionnalités ne sont pas les mêmes selon le profil connecté.

Cas de l'agent / administrateur

Cas d'utilisation	Description
Se connecter	L'agent saisit son identifiant et son mot de passe. Si le rôle est administrateur, le tableau de bord s'ouvre.
Consulter le tableau de bord	L'agent visualise les statistiques principales : adhérents, documents, emprunts en cours, emprunts terminés et derniers emprunts.
Gérer les adhérents	L'agent peut afficher, ajouter, modifier et supprimer des fiches adhérents.
Gérer les documents	L'agent peut afficher, rechercher, ajouter, modifier et supprimer des documents du catalogue.
Gérer les emprunts	L'agent peut créer un emprunt, rechercher un emprunt et enregistrer un retour.
Traiter les demandes d'emprunt	L'agent consulte les demandes déposées par les adhérents.
Valider une demande	L'agent accepte une demande. L'application crée alors un emprunt et rend le document indisponible.
Refuser une demande	L'agent refuse une demande. Aucun emprunt n'est créé.
Se déconnecter	L'agent revient à l'écran de connexion.

Cas de l'adhérent

Cas d'utilisation	Description
S'inscrire	L'adhérent remplit un formulaire avec son nom, prénom, email, téléphone, identifiant et mot de passe. L'application crée une fiche adhérent et un compte utilisateur.
Se connecter	L'adhérent saisit son identifiant et son mot de passe. Si le rôle est <code>adherent</code> , son espace personnel s'ouvre.
Consulter les documents disponibles	L'adhérent visualise uniquement les documents disponibles.
Rechercher un document disponible	L'adhérent peut rechercher par titre, auteur ou catégorie.
Déposer une demande d'emprunt	L'adhérent sélectionne un document disponible et dépose une demande. Il n'a pas besoin de choisir son nom dans une liste, car il est déjà identifié par sa connexion.

Cas du visiteur

Cas d'utilisation	Description
Accéder au catalogue visiteur	Le visiteur ouvre le catalogue depuis l'écran de connexion sans compte utilisateur.
Consulter les documents disponibles	Le visiteur voit les documents actuellement disponibles.
Rechercher un document disponible	Le visiteur peut rechercher un document par titre, auteur ou catégorie.

Remarques pour le diagramme :

- le cas **Se connecter** peut être relié à l'agent et à l'adhérent
 - le cas **S'inscrire** concerne uniquement l'adhérent
 - le cas **Accéder au catalogue visiteur** concerne uniquement le visiteur
 - le cas **Rechercher un document disponible** peut être partagé par l'adhérent et le visiteur
 - le cas **Déposer une demande d'emprunt** doit être relié uniquement à l'adhérent connecté
 - le cas **Traiter les demandes d'emprunt** peut être spécialisé en **Valider une demande** et **Refuser une demande**
 - le visiteur ne doit pas être relié au cas **Déposer une demande d'emprunt**
 - la base de données peut être représentée comme système externe, mais elle n'est pas obligatoire dans un diagramme de cas d'utilisation simplifié
-

2. Conception

2.1 Schéma d'architecture logicielle et technique

Architecture générale

L'application est organisée selon une architecture en couches :

```
Utilisateur
|
v
Interfaces Swing
|
v
Services métier
|
v
DAO JDBC
|
v
Base de données MySQL / MariaDB
```

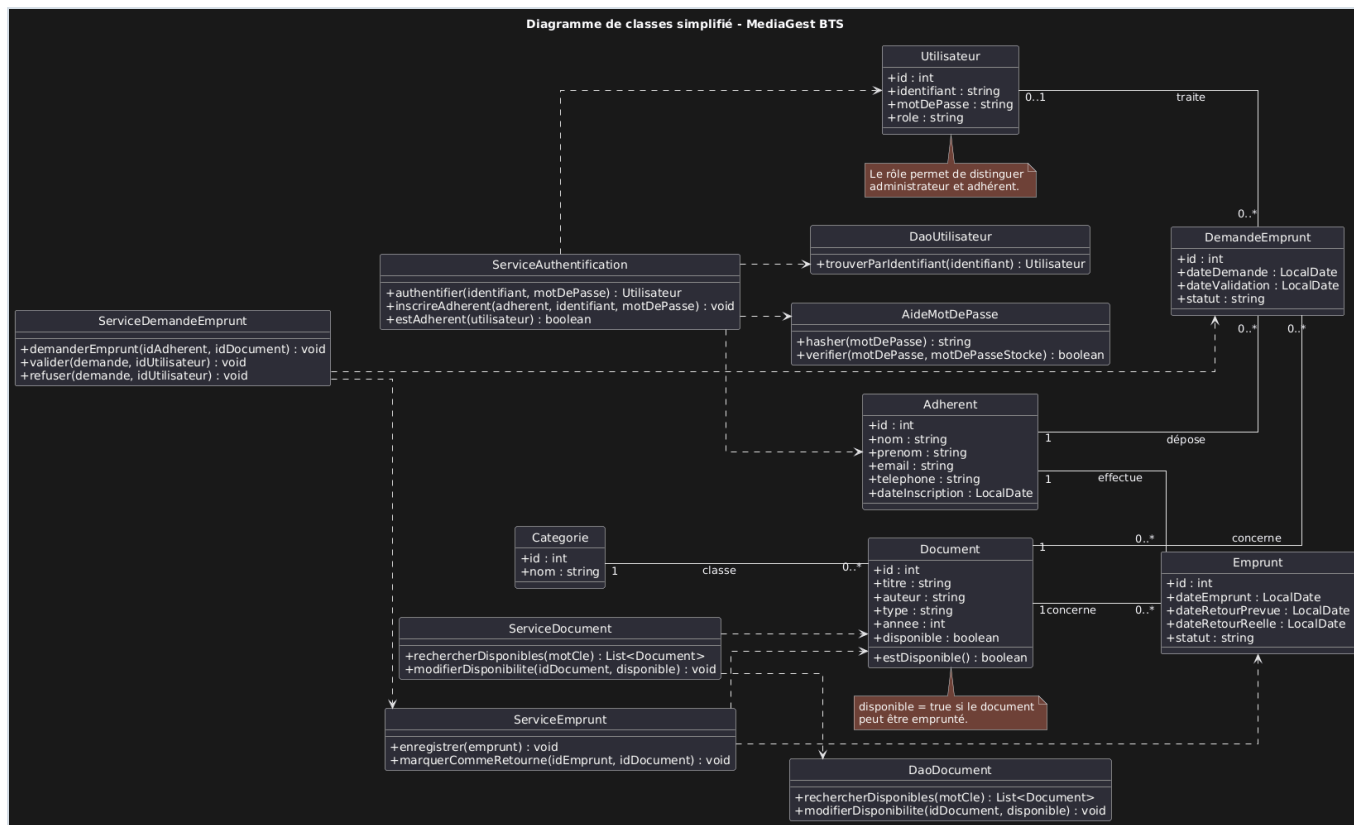
Rôle des couches

Couche	Package	Rôle
Application	<code>com.mediagest.application</code>	Point d'entrée de l'application.
Interface graphique	<code>com.mediagest.ihm</code>	Fenêtres Swing et interaction utilisateur.
Services	<code>com.mediagest.service</code>	Logique applicative entre l'IHM et les DAO.
DAO	<code>com.mediagest.dao</code>	Requêtes SQL et accès aux données.
Modèles	<code>com.mediagest.modele</code>	Classes métiers représentant les entités.
Configuration	<code>com.mediagest.configuration</code>	Paramètres de connexion à la base de données.
Utilitaires	<code>com.mediagest.utilitaire</code>	Aide à l'affichage, aux messages et au hachage des mots de passe.

Architecture technique d'exécution

Élément	Technologie utilisée
Langage	Java 8
Interface	Java Swing
Accès base de données	JDBC
SGBD	MySQL ou MariaDB
Connecteur	MySQL Connector/J 8.4.0
Sécurité mot de passe	PBKDF2WithHmacSHA256 avec sel aléatoire
IDE	Eclipse
Type d'application	Application fenêtrée client lourd
Base de données	mediagest_bts
Hôte base de données	localhost
Port	3306
Utilisateur MySQL	root
Mot de passe MySQL	vide dans la configuration actuelle

2.2 Diagramme UML de classes



Le diagramme de classes ci-dessus représente les principales classes de l'application Java. Il met en évidence les classes métiers, les fenêtres importantes, les services applicatifs et l'utilitaire de hachage des mots de passe.

- FenetreConnexion, FenetreInscriptionAdherent, FenetreEspaceAdherent et FenetreGestionDemandesEmprunt
- ServiceAuthentification avec l'authentification, l'inscription adhérent et la vérification du rôle
- AideMotDePasse avec les méthodes hasher et verifier
- DaoUtilisateur.trouverParIdentifiant
- ServiceDocument.rechercherDisponibles et DaoDocument.rechercherDisponibles
- les classes métiers Utilisateur, Adherent, Catégorie, Document, Emprunt et DemandeEmprunt.

2.3 Modèle conceptuel de données

Le MCD retenu est celui fourni dans l'image MCD.png. La description ci-dessous se base uniquement sur ce schéma : les emprunts et les demandes y sont modélisés comme des associations porteuses d'attributs, et non comme des entités séparées.

Les ajouts récents de l'application, comme l'inscription adhérent, l'accès visiteur, la recherche de documents disponibles et le hachage des mots de passe, ne nécessitent pas de nouvelle entité dans le MCD. Ils s'appuient sur les entités déjà présentes : utilisateur, adherent, document, categorie et les associations Demander et Emprunter.

Entités du MCD

UTILISATEUR

Représente un compte pouvant se connecter à l'application. Le rôle permet de distinguer un compte du personnel et un compte adhérent.

Attribut	Description
id_utilisateur	Identifiant unique de l'utilisateur.
identifiant	Nom de connexion.
mot_de_passe	Hash du mot de passe de connexion.
role	Rôle de l'utilisateur.

ADHERENT

Représente une personne inscrite à la médiathèque.

Attribut	Description
id_adherent	Identifiant unique de l'adhérent.
nom	Nom de famille.
prenom	Prénom.
email	Adresse e-mail.
telephone	Numéro de téléphone.
date_inscription	Date d'inscription à la médiathèque.

CATEGORIE

Représente une catégorie de classement des documents.

Attribut	Description
id_categorie	Identifiant unique de la catégorie.
nom	Nom de la catégorie.

DOCUMENT

Représente un document disponible dans la médiathèque.

Attribut	Description
id_document	Identifiant unique du document.
titre	Titre du document.
auteur	Auteur ou créateur.
type	Type de document : livre, DVD, magazine, manga, CD.
annee	Année de publication ou de sortie.
disponible	Indique si le document peut être emprunté.

Associations du MCD

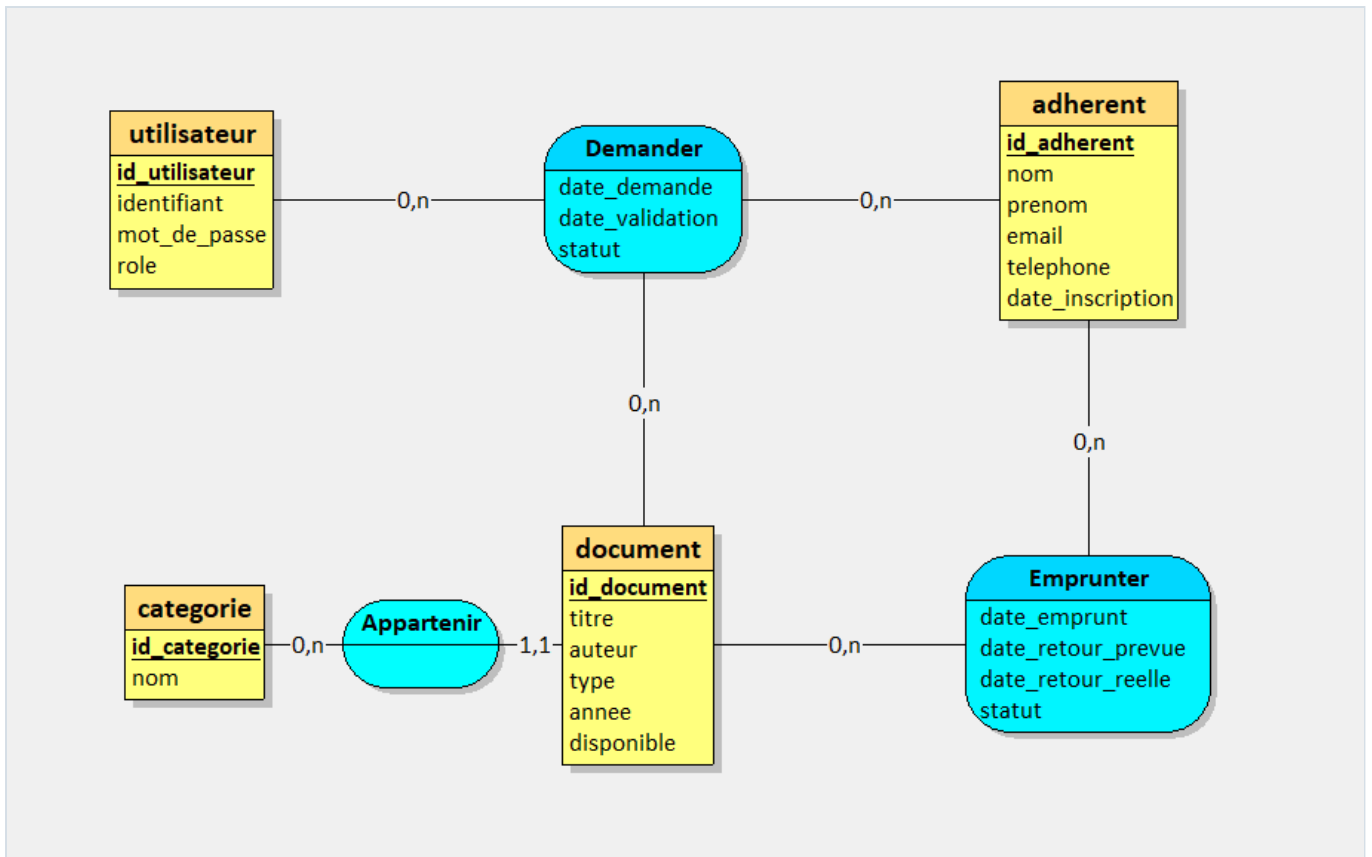
Association	Entités reliées	Cardinalités	Explication
Appartenir	CATEGORIE - DOCUMENT	CATEGORIE 0,n, DOCUMENT 1,1	Une catégorie peut contenir plusieurs documents. Un document appartient à une seule catégorie.
Emprunter	ADHERENT - DOCUMENT	ADHERENT 0,n, DOCUMENT 0,n	Un adhérent peut emprunter plusieurs documents. Un document peut être emprunté plusieurs fois dans le temps.
Demander	UTILISATEUR - ADHERENT - DOCUMENT	UTILISATEUR 0,n, ADHERENT 0,n, DOCUMENT 0,n	L'association relie le personnel, l'adhérent et le document concerné par une demande.

Attributs des associations

Association	Attributs	Rôle
Emprunter	date_emprunt, date_retour_prevue, date_retour_reelle, statut	Décrit le cycle de vie d'un emprunt.
Demander	date_demande, date_validation, statut	Décrit le dépôt et le traitement d'une demande d'emprunt.

Schéma MCD

L'image de référence du MCD est le fichier `MCD.png`. Si d'anciens exports existent encore dans le dossier, ils ne sont pas utilisés pour cette documentation.



Choix de modélisation retenu

Dans le MCD, `Emprunter` et `Demander` ne sont pas représentées comme des entités autonomes. Ce sont des associations avec attributs, car elles décrivent des actions réalisées entre plusieurs entités.

L'association `Emprunter` porte les informations liées au suivi d'un emprunt : dates et statut. L'association `Demander` porte les informations liées au suivi d'une demande : date de demande, date de validation et statut.

MLD proposé

```

UTILISATEUR(
  id_utilisateur,
  identifiant,
  mot_de_passe,
  role
)

ADHERENT(
  id_adherent,
  nom,
  prenom,
  email,
  telephone,
  date_inscription
)

CATEGORIE(
  id_categorie,
  nom
  )
  
```

```

)

DOCUMENT(
  id_document,
  titre,
  auteur,
  type,
  annee,
  disponible,
  id_categorie
)

EMPRUNTER(
  id_adherent,
  id_document,
  date_emprunt,
  date_retour_prevue,
  date_retour_reelle,
  statut
)

DEMANDER(
  id_utilisateur,
  id_adherent,
  id_document,
  date_demande,
  date_validation,
  statut
)

```

MRD correspondant

```

document.id_categorie -> categorie.id_categorie
emprunter.id_adherent -> adherent.id_adherent
emprunter.id_document -> document.id_document
demander.id_utilisateur -> utilisateur.id_utilisateur
demander.id_adherent -> adherent.id_adherent
demander.id_document -> document.id_document

```

2.4 Description des IHM, écrans et navigation

Écran de connexion

Classe concernée : `src/com/mediagest/ihm/FenetreConnexion.java`

L'écran de connexion permet de choisir entre une connexion du personnel, une connexion adhérent, un accès visiteur au catalogue et une inscription adhérent. Après validation, l'application vérifie les informations en base de données. Si le rôle est `administrateur`, le tableau de bord du personnel s'ouvre. Si le rôle est `adherent`, l'espace personnel de l'adhérent s'ouvre directement.

La vérification du mot de passe se fait avec un hash PBKDF2 stocké en base de données. Le mot de passe saisi n'est pas comparé directement en clair.

Fonctionnalités :

- saisie de l'identifiant
- saisie du mot de passe
- contrôle des champs vides
- vérification auprès du service d'authentification
- redirection selon le rôle : personnel ou adhérent
- accès visiteur au catalogue sans authentification
- ouverture du formulaire d'inscription adhérent
- affichage d'un message en cas d'erreur.

Capture d'écran :



The screenshot displays the MediaGest login interface. At the top, a dark teal header contains the logo 'MediaGest' and the subtitle 'Gestion de médiathèque municipale'. Below the header, a white form area is framed by a thin orange border. It features two input fields: 'Identifiant :' and 'Mot de passe :'. At the bottom of the form, there are three dark teal buttons with white text: 'Connexion', 'Acces visiteur', and 'Inscription adherent'.

Inscription adhérent

Classe concernée : `src/com/mediagest/ihm/FenetreInscriptionAdherent.java`

Cette fenêtre permet à un nouveau lecteur de créer son compte. L'inscription crée à la fois une fiche `adherent` et un compte `utilisateur` avec le rôle `adherent`. Pour rester conforme au MCD existant sans ajouter de nouvelle association, l'identifiant d'un compte adhérent correspond à son email.

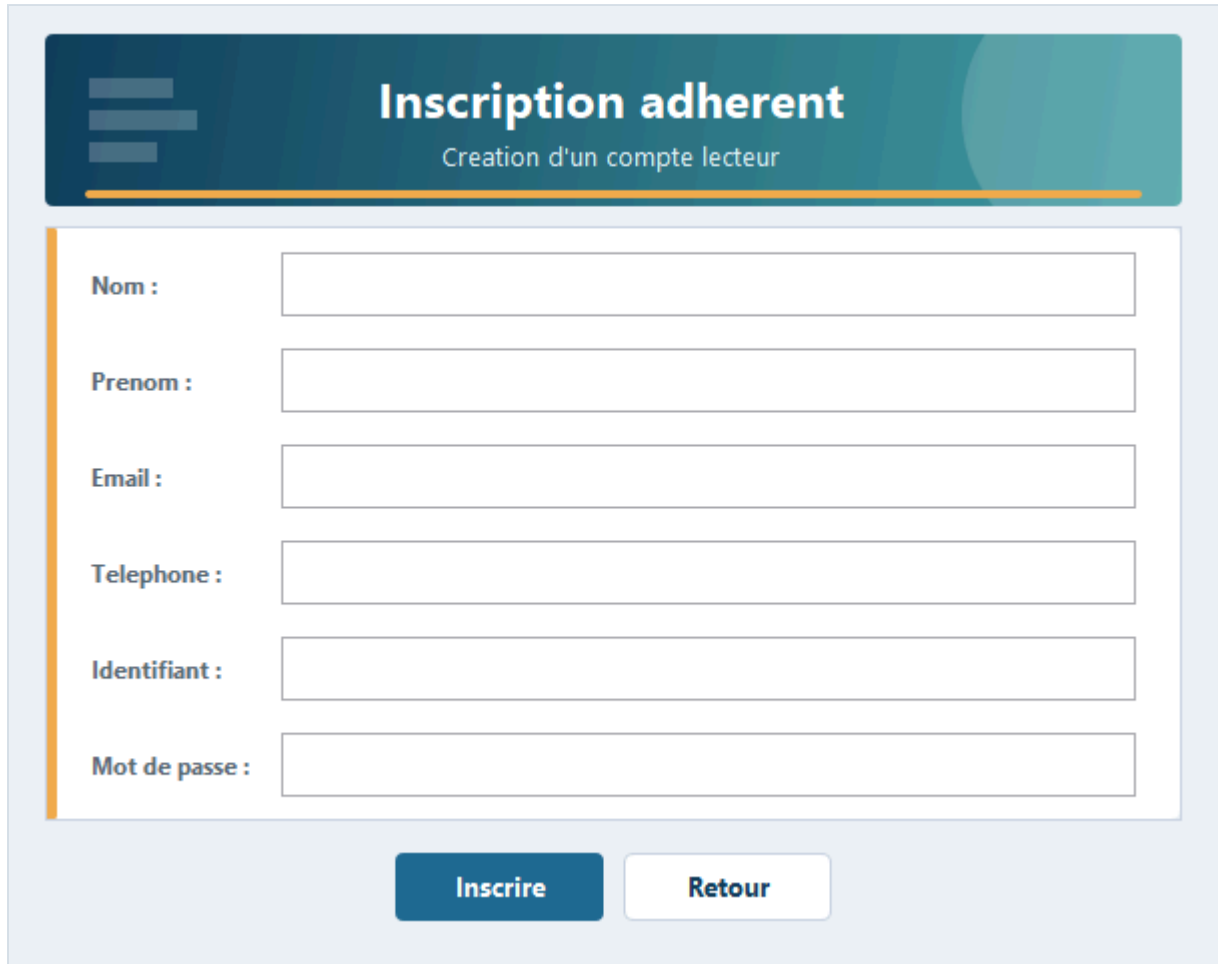
Le mot de passe choisi pendant l'inscription est haché avant l'insertion en base de données.

Champs disponibles :

- nom
- prénom
- email
- téléphone
- identifiant

- mot de passe.

Capture d'écran :



Inscription adherent
Creation d'un compte lecteur

Nom :

Prenom :

Email :

Telephone :

Identifiant :

Mot de passe :

Inscrire **Retour**

Tableau de bord

Classe concernée : `src/com/mediagest/ihm/FenetreTableauBord.java`

Le tableau de bord est l'écran principal de l'application. Il affiche des indicateurs et permet d'accéder aux différentes fenêtres de gestion.

Informations affichées :

- nombre d'adhérents
- nombre de documents
- nombre d'emprunts en cours
- nombre d'emprunts terminés
- derniers emprunts.

Actions possibles :

- ouvrir la gestion des adhérents
- ouvrir la gestion des documents
- ouvrir la gestion des emprunts
- ouvrir la validation des demandes d'emprunt
- se déconnecter.

Capture d'écran :



Gestion des adhérents

Classe concernée : `src/com/mediagest/ihm/FenetreGestionAdherents.java`

Cette fenêtre permet de gérer les personnes inscrites à la médiathèque.

Fonctionnalités :

- afficher la liste des adhérents
- sélectionner un adhérent dans un tableau
- ajouter un adhérent
- modifier un adhérent
- supprimer un adhérent
- vider le formulaire
- revenir au tableau de bord.

Champs disponibles :

- nom
- prénom
- email
- téléphone
- date d'inscription.

Capture d'écran :

Gestion des adhérents

Ajout, modification et suivi des membres

ID	Nom	Prénom	Email	Téléphone	Date inscription
2	Bernard	Emma	emma.bernard@exam...	0600000002	2025-10-02
5	Durand	Hugo	hugo.durand@exampl...	0600000005	2026-02-20
1	Martin	Lucas	lucas.martin@example....	0600000001	2025-09-10
6	Morel	Camille	camille.morel@exampl...	0600000006	2026-03-12
3	Petit	Nathan	nathan.petit@example....	0600000003	2025-11-15
4	Robert	Léa	lea.robert@example.co...	0600000004	2026-01-08

ID	Nom *	Prénom *	Email *	Téléphone	Date inscription
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="2026-05-18"/>

Ajouter

Modifier

Supprimer

Vider

Retour

Gestion des documents

Classe concernée : `src/com/mediagest/ihm/FenetreGestionDocuments.java`

Cette fenêtre permet de gérer le catalogue de la médiathèque.

Fonctionnalités :

- afficher les documents
- rechercher un document par titre, auteur ou catégorie
- ajouter un document
- modifier un document
- supprimer un document
- associer un document à une catégorie
- indiquer si le document est disponible ou emprunté.

Champs disponibles :

- titre
- auteur
- type
- catégorie
- année
- disponibilité.

Capture d'écran :

Gestion des documents

Catalogue, catégories et disponibilité

Titre, auteur ou catégorie :

Rechercher

Réinitialiser

ID	Titre	Auteur	Type	Catégorie	Année	Disponibilité
7	Astérix chez les Bretons	Goscinnny et Uderzo	Livre	Bande dessinée	1966	Emprunté
4	Aventure au collège	Sophie Martin	Manga	Jeunesse	2020	Disponible
5	Cinéma français volum...	Collectif	DVD	Cinéma	2019	Emprunté
8	Initiation aux réseaux	Nadia Petit	Livre	Informatique	2023	Disponible
2	Java pour débutants	Claire Dubois	Livre	Informatique	2022	Disponible
10	Le mystère de la ville	Julien Moreau	Livre	Roman	2024	Emprunté
1	Le Petit Prince	Antoine de Saint-Exup...	Livre	Jeunesse	1943	Disponible

ID	Titre *	Auteur	Type	Catégorie	Année	Disponibilité
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Livre"/>	<input type="text" value="Bande dessinée"/>	<input type="text"/>	<input type="text" value="Disponible"/>

Ajouter

Modifier

Supprimer

Vider

Retour

Gestion des emprunts

Classe concernée : `src/com/mediagest/ihm/FenetreGestionEmprunts.java`

Cette fenêtre permet de suivre les emprunts des adhérents.

Fonctionnalités :

- afficher les emprunts
- rechercher un emprunt par nom d'adhérent ou titre de document
- enregistrer un nouvel emprunt
- empêcher l'emprunt d'un document déjà indisponible
- marquer un emprunt comme retourné
- supprimer un emprunt
- mettre à jour la disponibilité du document.

Champs disponibles :

- adhérent
- document
- date d'emprunt
- date de retour prévue
- date de retour réelle
- statut.

Capture d'écran :

Gestion des emprunts

Emprunts en cours, retours et recherches

Nom d'adhérent ou titre de document :

Rechercher

Réinitialiser

ID	Adhérent	Document	Date emprunt	Retour prévu	Retour réel	Statut
6	Camille Morel	Le mystère de la ville	2026-04-28	2026-05-19		En cours
3	Nathan Petit	Astérix chez les Bretons	2026-04-25	2026-05-16		En cours
1	Lucas Martin	Java pour débutants	2026-04-20	2026-05-11	2026-05-10	Retourné
2	Emma Bernard	Cinéma français volum...	2026-04-12	2026-05-03		En retard
4	Léa Robert	Le Petit Prince	2026-03-02	2026-03-23	2026-03-20	Retourné

ID	Adhérent	Document	Statut
<input type="text"/>	Emma Bernard	Astérix chez les Bretons (Emprunté)	En cours
Date emprunt	Retour prévu	Retour réel	
2026-05-18	2026-06-08	<input type="text"/>	

Ajouter emprunt

Marquer comme retourné

Supprimer

Vider

Retour

Espace adhérent - demande d'emprunt

Classe concernée : `src/com/mediagest/ihm/FenetreEspaceAdherent.java`

Cette fenêtre possède deux modes. En mode visiteur, elle affiche uniquement le catalogue des documents disponibles. En mode adhérent connecté, elle affiche l'espace personnel de l'adhérent et permet de déposer une demande d'emprunt sans choisir son nom dans une liste. La demande ne crée pas immédiatement un emprunt : elle doit ensuite être validée ou refusée par le personnel de la médiathèque.

Fonctionnalités :

- afficher uniquement les documents disponibles
- rechercher un document disponible par titre, auteur ou catégorie
- réinitialiser la recherche pour revenir au catalogue complet des documents disponibles
- identifier automatiquement l'adhérent connecté
- sélectionner un ouvrage
- déposer une demande d'emprunt
- bloquer la demande d'emprunt en mode visiteur
- éviter les demandes en double pour le même adhérent et le même document.

Capture d'écran :

Espace adhérent

Sélection d'un ouvrage à demander

Adhérent connecté : Lucas Martin

Document : **Rechercher** Reinitialiser

ID	Titre	Auteur	Type	Categorie	Annee
4	Aventure au collège	Sophie Martin	Manga	Jeunesse	2020
8	Initiation aux réseaux	Nadia Petit	Livre	Informatique	2023
2	Java pour débutants	Claire Dubois	Livre	Informatique	2022
1	Le Petit Prince	Antoine de Saint-Exup...	Livre	Jeunesse	1943
3	Les bases de MySQL	Marc Leroy	Livre	Informatique	2021

Demander l'emprunt **Actualiser** Retour

Capture du catalogue visiteur :

Catalogue visiteur

Consultation des documents disponibles

Mode visiteur : consultation du catalogue disponible uniquement.

Document : **Rechercher** Reinitialiser

ID	Titre	Auteur	Type	Categorie	Annee
4	Aventure au collège	Sophie Martin	Manga	Jeunesse	2020
8	Initiation aux réseaux	Nadia Petit	Livre	Informatique	2023
2	Java pour débutants	Claire Dubois	Livre	Informatique	2022
1	Le Petit Prince	Antoine de Saint-Exup...	Livre	Jeunesse	1943
3	Les bases de MySQL	Marc Leroy	Livre	Informatique	2021

Actualiser Retour

Validation des demandes d'emprunt

Classe concernée : `src/com/mediagest/ihm/FenetreGestionDemandesEmprunt.java`

Cette fenêtre est accessible depuis le tableau de bord du personnel. Elle permet de consulter les demandes déposées par les adhérents, puis de les valider ou de les refuser.

Fonctionnalités :

- afficher les demandes d'emprunt

- consulter l'adhérent, le document, la date de demande et le statut
- valider une demande
- créer automatiquement un emprunt après validation
- refuser une demande
- enregistrer l'utilisateur du personnel ayant traité la demande
- mettre à jour la disponibilité du document après validation.

Capture d'écran :

Demandes d'emprunt						
Validation par le personnel						
ID	Adherent	Document	Date demande	Statut	Traitee par	Date validation
4	Emma Bernard	Aventure au collèg	2026-05-18	En attente		
3	Hugo Durand	Musique classique d...	2026-05-06	En attente		
2	Emma Bernard	Initiation aux réseaux	2026-05-04	En attente		
1	Lucas Martin	Les bases de MySQL	2026-05-02	En attente		

Valider l'emprunt
Refuser
Actualiser
Retour

Navigation

flowchart TD

```

Login[Connexion] --> Dashboard[Tableau de bord]
Login --> Inscription[Inscription adhérent]
Login --> Visiteur[Catalogue visiteur]
Login --> EspaceAdherent[Espace adhérent connecté]
Dashboard --> Adherents[Gestion des adhérents]
Dashboard --> Documents[Gestion des documents]
Dashboard --> Emprunts[Gestion des emprunts]
Dashboard --> Demandes[Demandes d'emprunt]
Inscription --> Login
Visiteur --> Login
EspaceAdherent --> Login
Adherents --> Dashboard
Documents --> Dashboard
Emprunts --> Dashboard
Demandes --> Dashboard
Dashboard --> Login

```

3. Réalisation

3.1 Présentation du code source

Organisation du projet

```
projet2.0/  
├── lib/  
│   └── mysql-connector-j-8.4.0.jar  
├── sql/  
│   ├── init_database.sql  
│   ├── demo_data.sql  
│   ├── reset.sql  
│   └── looping_retroconception_mcd.sql  
├── src/  
│   └── com/  
│       └── mediagest/  
│           ├── application/  
│           ├── configuration/  
│           ├── dao/  
│           ├── modele/  
│           ├── service/  
│           ├── ihm/  
│           └── utilitaire/  
├── MCD.png  
├── MCD-Java.loo  
├── diagramme_classes.puml  
└── MediaGestBTS.launch
```

Point d'entrée de l'application

Fichier : `src/com/mediagest/application/ApplicationPrincipale.java`

La classe `ApplicationPrincipale` lance l'application avec `SwingUtilities.invokeLater`. Elle configure l'apparence graphique via `AideInterface` puis ouvre la fenêtre de connexion.

Configuration de la base de données

Fichier : `src/com/mediagest/configuration/ConfigurationBaseDonnees.java`

La classe `ConfigurationBaseDonnees` centralise les informations de connexion :

```
URL : jdbc:mysql://localhost:3306/mediagest_bts  
Utilisateur : root  
Mot de passe : vide
```

Elle charge le pilote MySQL `com.mysql.cj.jdbc.Driver` et retourne une connexion JDBC.

Classes modèles

Les classes du package `modele` représentent les principales entités du domaine :

Classe	Rôle
Utilisateur	Représente un compte de connexion.
Adherent	Représente une personne inscrite à la médiathèque.
Categorie	Représente une catégorie de document.
Document	Représente un document du catalogue.
Emprunt	Représente un emprunt ou un retour.
DemandeEmprunt	Représente une demande déposée par un adhérent et traitée par le personnel.

DAO

Les DAO du package `dao` contiennent les requêtes SQL et utilisent JDBC.

Classe	Rôle
DaoBase	Fournit la connexion à la base de données.
DaoUtilisateur	Création et recherche d'un utilisateur par identifiant pour l'authentification.
DaoAdherent	CRUD des adhérents et recherche par email.
DaoCategorie	Lecture des catégories.
DaoDocument	CRUD, recherche globale et recherche des documents disponibles.
DaoEmprunt	CRUD, recherche et statistiques des emprunts.
DaoDemandeEmprunt	Lecture, création et traitement des demandes d'emprunt.

Les requêtes sont préparées avec `PreparedStatement`, ce qui limite les risques d'injection SQL.

Services

Les services assurent le lien entre les fenêtres Swing et les DAO.

Classe	Rôle
ServiceAuthentification	Authentification avec vérification du hash, inscription adhérent et association compte/fiche adhérent.
ServiceAdherent	Logique liée aux adhérents.
ServiceDocument	Logique liée aux documents, catégories et recherches dans le catalogue.
ServiceEmprunt	Logique liée aux emprunts et à la disponibilité des documents.
ServiceDemandeEmprunt	Logique liée aux demandes, validations et refus d'emprunt.

Exemple de règle métier :

- lors de la création d'un emprunt, l'application vérifie que le document est disponible

- après création de l'emprunt, le document devient indisponible
- lorsqu'un emprunt est marqué comme retourné, le document redevient disponible.

Sécurité de l'authentification

Les mots de passe ne sont pas stockés en clair. Lors de l'inscription, l'application génère un hash PBKDF2 avec un sel aléatoire. La base de données stocke une valeur au format :

```
PBKDF2$iterations$sel$hash
```

Lors de la connexion, l'application récupère l'utilisateur par son identifiant, recalcule le hash à partir du mot de passe saisi et compare le résultat avec la valeur stockée.

Interfaces graphiques

Fenêtre	Rôle
FenetreConnexion	Connexion à l'application.
FenetreInscriptionAdherent	Création d'un compte adhérent.
FenetreTableauBord	Tableau de bord et accès aux modules.
FenetreGestionAdherents	Gestion des adhérents.
FenetreGestionDocuments	Gestion des documents.
FenetreGestionEmprunts	Gestion des emprunts.
FenetreEspaceAdherent	Catalogue visiteur avec recherche, ou dépôt d'une demande d'emprunt par un adhérent connecté.
FenetreGestionDemandesEmprunt	Validation ou refus des demandes par le personnel.

Utilitaires

Classe	Rôle
AideInterface	Centralise la création et le style des composants Swing.
AideMessage	Centralise l'affichage des messages utilisateur.
AideMotDePasse	Centralise le hachage PBKDF2 et la vérification des mots de passe.

3.2 Tests unitaires et vérifications techniques

Tests unitaires

Le projet ne contient pas actuellement de framework de tests unitaires comme JUnit. Les vérifications sont donc principalement réalisées par tests fonctionnels depuis l'interface graphique.

Évolution possible :

- ajouter JUnit au projet
- tester les services
- tester les DAO avec une base de test
- tester les règles métier liées aux emprunts.

Vérifications techniques réalisées

Élément vérifié	Résultat attendu
Présence du connecteur MySQL	Le fichier <code>lib/mysql-connector-j-8.4.0.jar</code> existe.
Présence des scripts SQL	Les fichiers <code>init_database.sql</code> , <code>demo_data.sql</code> et <code>reset.sql</code> existent.
Présence des classes métiers	Les classes <code>Utilisateur</code> , <code>Adherent</code> , <code>Categorie</code> , <code>Document</code> , <code>Emprunt</code> et <code>DemandeEmprunt</code> existent.
Présence des fenêtres	Les fenêtres de connexion, inscription, tableau de bord, adhérents, documents, emprunts, espace adhérent et demandes existent.
Présence des DAO	Les classes DAO existent pour les entités principales.
Hachage des mots de passe	Les mots de passe de test sont stockés sous forme de hash PBKDF2 dans les scripts SQL.
Recherche catalogue visiteur/adhérent	La recherche ne retourne que les documents disponibles.

Vérification de compilation

La compilation a été vérifiée avec le JDK 1.8 installé sur le poste et le connecteur MySQL `mysql-connector-j-8.4.0.jar`.

3.3 Fichier d'exportation de la base de données

Scripts disponibles

Fichier	Description
<code>sql/init_database.sql</code>	Crée la base de données et les tables.
<code>sql/demo_data.sql</code>	Insère les données de démonstration.
<code>sql/reset.sql</code>	Recrée la base et réinsère les données.
<code>sql/looping_retroconception_mcd.sql</code>	Script de rétroconception ; l'image <code>MCD.png</code> reste la référence du MCD retenu.

Tables principales

Table	Description
utilisateur	Comptes de connexion à l'application, avec mot de passe stocké sous forme de hash PBKDF2.
adherent	Personnes inscrites à la médiathèque.
categorie	Catégories des documents.
document	Catalogue de documents.
emprunt	Table issue de l'association Emprunter du MCD : emprunts et retours des documents.
demande_emprunt	Table issue de l'association Demander du MCD : demandes d'emprunt et traitement par le personnel.

Données de test

Le jeu d'essai contient :

- un compte administrateur
- des comptes adhérents rattachés à des fiches adhérents par leur email
- plusieurs adhérents réalistes
- plusieurs catégories
- plusieurs documents
- plusieurs emprunts avec des statuts différents : en cours, en retard, retourné.

Identifiants de test :

Identifiant : admin

Mot de passe : admin123

Identifiant adhérent : lucas.martin@example.com

Mot de passe : adherent123

Ces mots de passe sont ceux à saisir dans l'application. Dans les scripts SQL, ils sont stockés sous forme de hash PBKDF2 et non en clair.

4. Documentation logicielle

4.1 Cahier de tests fonctionnels

ID	Fonction testée	Préconditions	Étapes	Résultat attendu
TF01	Connexion réussie	Base initialisée, compte admin présent	Saisir admin et admin123, cliquer sur Connexion	Le tableau de bord s'ouvre.
TF02	Connexion refusée	Base initialisée	Saisir un mauvais identifiant ou mot de passe	Un message d'erreur apparaît.
TF03	Ajout d'un adhérent	Être connecté	Ouvrir la gestion des adhérents, remplir les champs obligatoires, cliquer sur Ajouter	L'adhérent apparaît dans le tableau.
TF04	Modification d'un adhérent	Être connecté, adhérent existant	Sélectionner un adhérent, modifier une valeur, cliquer sur Modifier	Les informations sont mises à jour.
TF05	Suppression d'un adhérent sans emprunt	Être connecté, adhérent sans emprunt	Sélectionner l'adhérent, cliquer sur Supprimer	L'adhérent est supprimé.
TF06	Suppression d'un adhérent avec emprunt	Être connecté, adhérent lié à un emprunt	Sélectionner l'adhérent, cliquer sur Supprimer	Un message indique que la suppression est impossible.
TF07	Ajout d'un document	Être connecté	Ouvrir la gestion des documents, remplir le formulaire, cliquer sur Ajouter	Le document apparaît dans le catalogue.
TF08	Recherche d'un document	Documents existants	Saisir un mot-clé puis cliquer sur Rechercher	Les documents correspondants sont affichés.
TF09	Modification d'un document	Document existant	Sélectionner un document, modifier une valeur, cliquer sur Modifier	Le document est mis à jour.
TF10	Création d'un emprunt	Adhérent et document disponible existants	Sélectionner un adhérent, un document disponible, cliquer sur Ajouter emprunt	L'emprunt est créé et le document devient indisponible.
TF11	Emprunt d'un document indisponible	Document déjà emprunté	Sélectionner le document indisponible et tenter l'emprunt	L'application refuse l'opération.
TF12	Retour d'un document	Emprunt en cours existant	Sélectionner l'emprunt puis cliquer sur Marquer comme retourné	Le statut passe à retourné et le

ID	Fonction testée	Préconditions	Étapes	Résultat attendu
				document redevient disponible.
TF13	Recherche d'un emprunt	Emprunts existants	Saisir un nom ou un titre puis cliquer sur Rechercher	Les emprunts correspondants sont affichés.
TF14	Déconnexion	Être sur le tableau de bord	Cliquer sur Déconnexion	L'écran de connexion s'affiche.
TF15	Accès visiteur	Base initialisée	Cliquer sur Accès visiteur	Le catalogue des documents disponibles s'ouvre sans bouton de demande.
TF16	Connexion adhérent	Compte adhérent existant	Saisir <code>lucas.martin@example.com</code> et <code>adherent123</code>	L'espace adhérent s'ouvre avec Lucas Martin connecté.
TF17	Demande adhérent connecté	Être connecté comme adhérent	Sélectionner un document disponible puis cliquer sur Demander l'emprunt	La demande est créée sans choisir l'adhérent dans une liste.
TF18	Inscription adhérent	Email non utilisé	Cliquer sur Inscription adhérent, remplir le formulaire, valider	Un compte utilisateur et une fiche adhérent sont créés.
TF19	Recherche catalogue visiteur/adhérent	Être sur le catalogue visiteur ou l'espace adhérent	Saisir un titre, auteur ou catégorie puis cliquer sur Rechercher	Seuls les documents disponibles correspondant au mot-clé sont affichés.
TF20	Vérification du hachage des mots de passe	Base initialisée	Consulter la table <code>utilisateur</code> puis tester une connexion	La base contient des hashes PBKDF2 et la connexion fonctionne avec le mot de passe original.

4.2 Manuel administrateur

Installation de la base de données

Le projet est prévu pour fonctionner avec la base MySQL locale fournie par Laragon. La configuration utilisée par l'application est la suivante :

```
Hôte : localhost
Port : 3306
Base de données : mediagest_bts
Utilisateur : root
Mot de passe : vide
```

1. Ouvrir Laragon.
2. Cliquer sur `Démarrer` ou `Tout démarrer` pour lancer le service MySQL.
3. Ouvrir le client SQL de Laragon (`Database/HeidiSQL`) ou phpMyAdmin si celui-ci est installé.
4. Exécuter le script `sql/init_database.sql` afin de créer la base `mediagest_bts` et ses tables.
5. Exécuter ensuite le script `sql/demo_data.sql` afin d'ajouter les comptes et données de démonstration.

Pour supprimer les données existantes et repartir d'une base propre, exécuter uniquement :

```
sql/reset.sql
```

Importation du projet dans Eclipse

1. Ouvrir Eclipse.
2. Aller dans `File > Import`.
3. Sélectionner `Existing Projects into Workspace`.
4. Choisir le dossier du projet.
5. Valider l'importation.

Configuration du connecteur MySQL

Le connecteur est fourni dans :

```
lib/mysql-connector-j-8.4.0.jar
```

Si Eclipse ne le charge pas automatiquement :

1. Clic droit sur le projet.
2. `Build Path > Add External Archives`.
3. Sélectionner le fichier JAR.
4. Valider.

Lancement de l'application

Classe principale :

```
src/com/mediagest/application/ApplicationPrincipale.java
```

Une configuration Eclipse est également disponible :

```
MediaGestBTS.launch
```

Configuration actuelle de la base

```
Hôte : localhost
Port : 3306
Base : mediagest_bts
Utilisateur : root
Mot de passe : vide
```

Comptes de test

```
Identifiant : admin
Mot de passe : admin123
Rôle : administrateur

Identifiant : lucas.martin@example.com
Mot de passe : adherent123
Rôle : adherent
```

Dans la base de données, ces mots de passe ne sont pas visibles en clair. Ils sont stockés dans la table `utilisateur` sous forme de hash PBKDF2.

Maintenance

L'administrateur technique peut :

- modifier les paramètres de connexion dans `ConfigurationBaseDonnees.java`
- réinitialiser la base avec `reset.sql`
- enrichir le jeu d'essai dans `demo_data.sql`
- ajouter de nouvelles catégories dans la table `categorie`
- modifier le compte administrateur dans la table `utilisateur`, en générant un nouveau hash si le mot de passe change.

4.3 Manuel utilisateur

Connexion

1. Lancer l'application.
2. Saisir l'identifiant.
3. Saisir le mot de passe.
4. Cliquer sur **Connexion**.

En cas d'erreur, vérifier les identifiants ou l'état de la base de données.

Consulter le tableau de bord

Après connexion, le tableau de bord affiche :

- le nombre d'adhérents
- le nombre de documents

- le nombre d'emprunts en cours
- le nombre d'emprunts terminés
- les derniers emprunts.

Gérer les adhérents

Pour ajouter un adhérent :

1. Cliquer sur **Gérer les adhérents**.
2. Remplir le nom, le prénom et l'e-mail.
3. Renseigner éventuellement le téléphone et la date d'inscription.
4. Cliquer sur **Ajouter**.

Pour modifier un adhérent :

1. Sélectionner une ligne dans le tableau.
2. Modifier les champs souhaités.
3. Cliquer sur **Modifier**.

Pour supprimer un adhérent :

1. Sélectionner une ligne.
2. Cliquer sur **Supprimer**.
3. Confirmer la suppression.

La suppression peut être refusée si l'adhérent possède des emprunts liés.

Gérer les documents

Pour ajouter un document :

1. Cliquer sur **Gérer les documents**.
2. Renseigner le titre.
3. Compléter l'auteur, le type, la catégorie, l'année et la disponibilité.
4. Cliquer sur **Ajouter**.

Pour rechercher un document :

1. Saisir un mot-clé dans le champ de recherche.
2. Cliquer sur **Rechercher**.
3. Cliquer sur **Réinitialiser** pour afficher tous les documents.

Gérer les emprunts

Pour déposer une demande d'emprunt comme adhérent :

1. Depuis l'écran de connexion, saisir l'identifiant et le mot de passe d'un compte adhérent.
2. Cliquer sur **Connexion**.
3. Rechercher éventuellement un ouvrage par titre, auteur ou catégorie.
4. Sélectionner un ouvrage disponible.
5. Cliquer sur **Demander l'emprunt**.

Pour consulter le catalogue comme visiteur :

1. Depuis l'écran de connexion, cliquer sur **Accès visiteur**.

2. Consulter la liste des documents disponibles.
3. Saisir un mot-clé puis cliquer sur **Rechercher** pour filtrer le catalogue.
4. Cliquer sur **Reinitialiser** pour revenir à la liste complète des documents disponibles.

Pour créer un compte adhérent :

1. Depuis l'écran de connexion, cliquer sur **Inscription adhérent**.
2. Renseigner le nom, le prénom, l'email, le téléphone, l'identifiant et le mot de passe.
3. Utiliser l'email comme identifiant.
4. Valider l'inscription.

Pour valider une demande d'emprunt comme personnel :

1. Se connecter avec le compte administrateur.
2. Cliquer sur **Demandes d'emprunt** depuis le tableau de bord.
3. Sélectionner une demande en attente.
4. Cliquer sur **Valider l'emprunt**.

Après validation, l'emprunt est créé et le document devient indisponible.

Pour refuser une demande :

1. Ouvrir **Demandes d'emprunt**.
2. Sélectionner la demande.
3. Cliquer sur **Refuser**.

Pour créer un emprunt :

1. Cliquer sur **Gérer les emprunts**.
2. Sélectionner un adhérent.
3. Sélectionner un document disponible.
4. Vérifier la date d'emprunt et la date de retour prévue.
5. Cliquer sur **Ajouter emprunt**.

Pour enregistrer un retour :

1. Sélectionner un emprunt en cours.
2. Cliquer sur **Marquer comme retourné**.
3. Le statut est mis à jour et le document redevient disponible.

Pour rechercher un emprunt :

1. Saisir le nom d'un adhérent ou le titre d'un document.
2. Cliquer sur **Rechercher**.

Déconnexion

Depuis le tableau de bord, cliquer sur **Déconnexion** pour revenir à l'écran de connexion.

5. Éléments à compléter avant le rendu

Les éléments suivants doivent être ajoutés ou vérifiés avant de fournir la documentation à la commission :

- vérifier que les captures d'écran intégrées correspondent bien à la version finale de l'application

- vérifier depuis Eclipse que le projet compile et se lance correctement
- vérifier que MySQL ou MariaDB est démarré avant la démonstration
- vérifier que l'image `MCD.png` est bien présente dans le dossier du projet avant l'export de la documentation
- exporter une version PDF finale de cette documentation après correction
- ajouter éventuellement le diagramme de classes au format image si demandé par le lycée
- corriger les éventuels problèmes d'encodage visibles dans certains anciens fichiers texte.