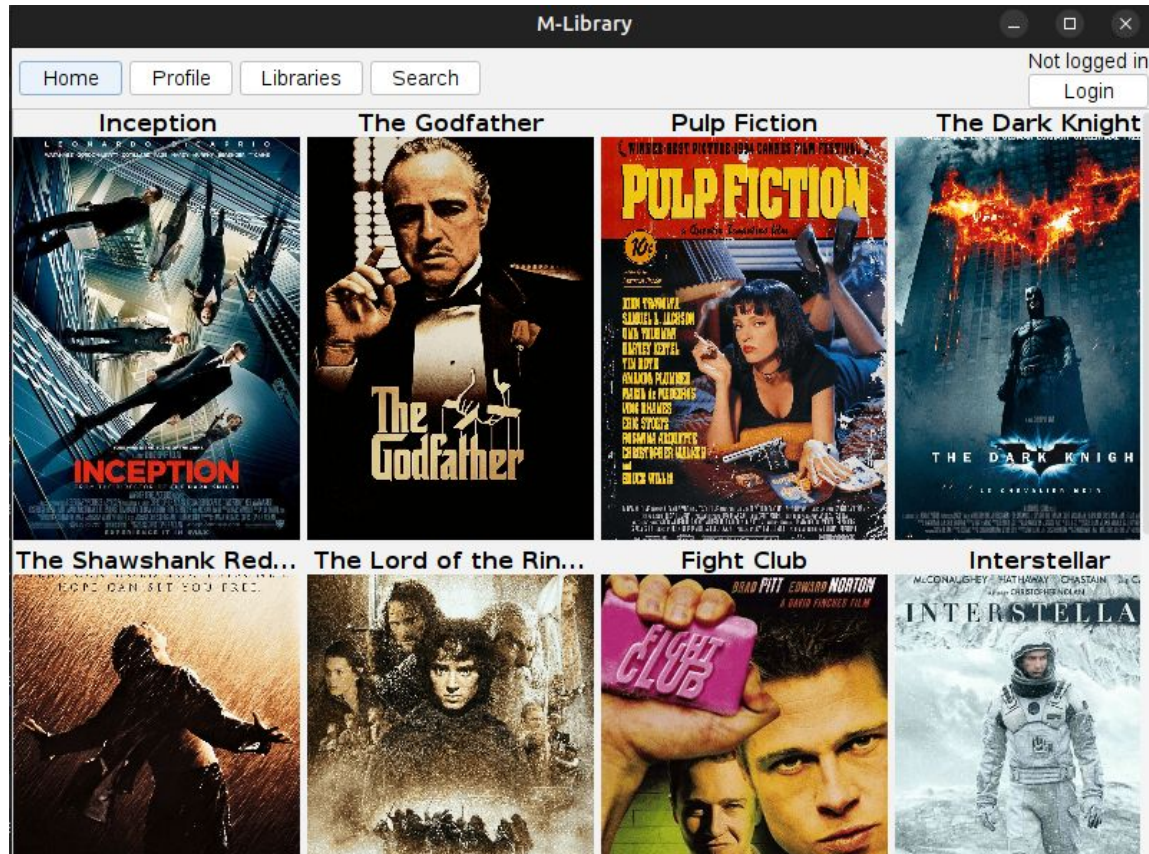


M-Library



M-Library

Application de gestion de bibliothèques de films :

- Accès à une base de données de films
- Création de librairies personnalisées utilisant les films de la base de données
- Accès au détail d'un film
- Accès au détail d'une librairie
- Visite du profil d'autres utilisateurs et de leurs librairies

Architecture Client-Serveur

Backend

Frontend

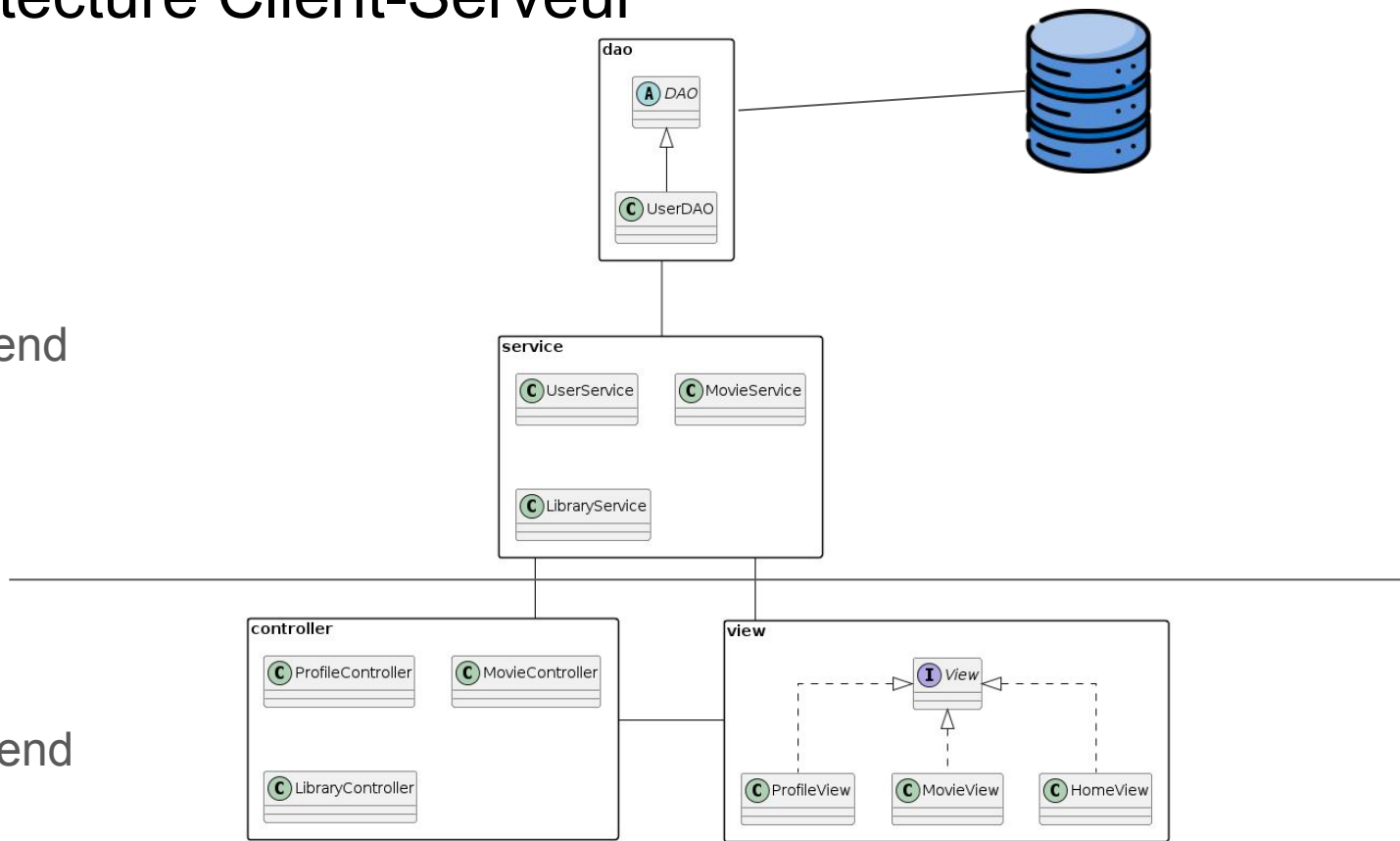


Figure 1 : Diagramme UML du projet

Client

Couche Vue

- La vue représente tous les éléments graphiques

Couche Contrôleur

- Les interactions avec l'utilisateur sont dans le controller

Couche Modèle

- Structures objet des données renvoyées du backend

Exemple Vue et Contrôleur

```
buttonDelete.addActionListener(e -> controller.deleteLibraryAction(modelId));
```

Figure 2 : controller.callback() dans la view

```
public void deleteLibraryAction(int libraryId) {  
    MainController.getInstance().show(ViewEnum.LIBRARIES, null);  
    libraryService.deleteLibrary(libraryId);  
}
```

Figure 3 : 1 callback = 1 interaction utilisateur

Serveur

Couche Service

- La couche Service fournit la logique métier
- Garant de l'état de l'application
- Renvoie des modèles au frontend

Couche DAO (accès BDD)

- La couche DAO interagit avec la base de données
- Pourrait être des fichiers .json

Exemple Service et DAO

```
public Library getById(int id) throws SQLException {  
    Library library = libraryDAO.getById(id);  
    if(library != null){  
        library.setOwner(this.userDAO.getById(library.getOwner().getId()));  
    }  
    return library;  
}
```

Figure 4 : Le service appelle le DAO

```
String query = "SELECT * FROM library WHERE library_id = ?";  
Library library = null;  
try (PreparedStatement stmt = databaseConnection.prepareStatement(query)) {  
    stmt.setInt(1, id);  
    ResultSet resultSet = stmt.executeQuery();  
    if (resultSet.next()) {  
        library = getLibraryFromResultSet(resultSet);  
    }  
}  
return library;
```

Figure 5 : Le DAO interroge la base de données

Base de données

BDD MySQL, hébergé sur un serveur Oracle

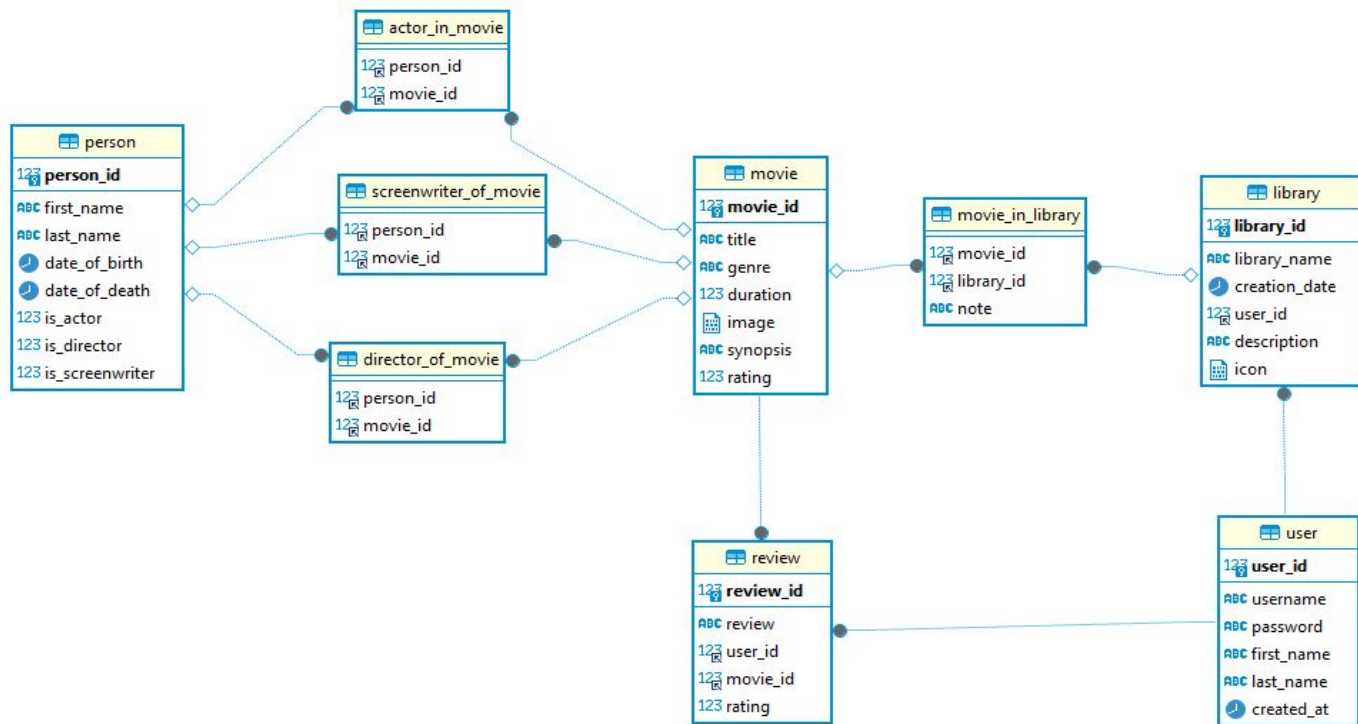


Figure 6 : Diagramme UML de la base de données

Gestion des pages

- Pages chargées au lancement
- Changement grâce au MainController

```
public void show(ViewEnum viewEnum, Integer modelId) {  
    if (SessionService.isSignedOut() && viewEnum.isAuthRequired()) {  
        this.mainView.setContent(ViewEnum.LOGIN);  
    } else {  
        this.views.get(viewEnum).refresh(modelId);  
        this.mainView.setContent(viewEnum);  
    }  
}
```

Figure 7 : Changement de vues

Problèmes rencontrés

- Changement de base de données
- Gestion des images
- GUI Designer très peu flexible

Pistes d'améliorations

- Séparer backend et frontend (Serveur web)
 - API REST
- Hachage et salage des mots de passe en base de données
- Design à améliorer