

GLO : Logiciel de gestion d’une médiathèque

MIAGE nantes | Master 1 alternance

Rapport intermediaire 1

Sebastien heuzé, Cyril Le Driant

2013

Table des matières

[1. Introduction 3](#_Toc377585977)

[2. Analyse des besoins 4](#_Toc377585978)

[2.1. Sous système gestion des médias 4](#_Toc377585979)

[2.1.1. Diagramme de cas d’utilisation 4](#_Toc377585980)

[2.1.2. Cartouches 5](#_Toc377585981)

[2.1.3. Diagrammes d’activité 6](#_Toc377585982)

[2.1.4. Maquettes d’écran 8](#_Toc377585983)

[2.2. Sous-système utilisateurs 9](#_Toc377585984)

[2.2.1. Diagramme de cas d’utilisation 9](#_Toc377585985)

[2.2.2. Cartouches 10](#_Toc377585986)

[2.2.3. Maquettes d’écran 11](#_Toc377585987)

[3. Analyse 13](#_Toc377585988)

[3.1. Diagrammes de collaboration 13](#_Toc377585989)

[3.2. Diagramme de classes. 14](#_Toc377585990)

[3.3. Diagrammes de séquence 15](#_Toc377585991)

[3.3.1. Ajout d’un média 15](#_Toc377585992)

[3.3.2. Emprunter un média 15](#_Toc377585993)

[3.3.3. Suppression d’un média 17](#_Toc377585994)

[3.4. Diagrammes d’états 18](#_Toc377585995)

[3.4.1. Diagramme d’états pour le média 18](#_Toc377585996)

[4. Conception 19](#_Toc377585997)

[4.1. Architecture générale 19](#_Toc377585998)

[4.2. Diagramme de composants 20](#_Toc377585999)

[4.3. Diagramme de déploiement 21](#_Toc377586000)

[5. Implémentation 22](#_Toc377586001)

[5.1. Un diagramme de classes affiné 22](#_Toc377586002)

[6. Conclusion générale 23](#_Toc377586003)

[6.1. Etat des lieux des développements 23](#_Toc377586004)

[6.2. Bilan 23](#_Toc377586005)

[Figure 1 - Diagramme de CU, SS médias 4](file:///C:\Users\Cyril\Documents\GitHub\GLOEiffelProject\Diagrammes\partie1_rapport%20(1).docx#_Toc377586006)

[Figure 2 - Diagramme d'activité : emprunt d'un média 7](file:///C:\Users\Cyril\Documents\GitHub\GLOEiffelProject\Diagrammes\partie1_rapport%20(1).docx#_Toc377586007)

[Figure 3 - Menu gestion des médias 8](#_Toc377586008)

[Figure 4 - Menu gestion des livres 8](#_Toc377586009)

[Figure 5 – Menu de gestion des DVD 8](#_Toc377586010)

[Figure 6 - Emprunt d'un DVD 8](#_Toc377586011)

[Figure 7 - Diagramme de CU, SS utilisateurs 9](file:///C:\Users\Cyril\Documents\GitHub\GLOEiffelProject\Diagrammes\partie1_rapport%20(1).docx#_Toc377586012)

[Figure 8 - Menu gestion utilisateurs 11](#_Toc377586013)

[Figure 9 - Ajout d'un utilisateur 11](#_Toc377586014)

[Figure 10 - Modification d'un utilisateur 11](#_Toc377586015)

[Figure 11 - Suppression d'un utilisateur 12](#_Toc377586016)

[Figure 12 - Recherche d'un utilisateur 12](#_Toc377586017)

[Figure 13 - Diagramme de collaboration 13](#_Toc377586018)

[Figure 14 - Diagramme de classe 14](#_Toc377586019)

[Figure 15 - Diagramme de séquence - Ajout d'un média 15](file:///C:\Users\Cyril\Documents\GitHub\GLOEiffelProject\Diagrammes\partie1_rapport%20(1).docx#_Toc377586020)

[Figure 16 - Diagramme de séquence - Emprunter média 16](#_Toc377586021)

[Figure 17 - Diagramme de séquence, suppression d'un média 17](file:///C:\Users\Cyril\Documents\GitHub\GLOEiffelProject\Diagrammes\partie1_rapport%20(1).docx#_Toc377586022)

[Figure 18 - Diagramme d'états - média 18](#_Toc377586023)

[Figure 19 - Diagramme de composants 20](file:///C:\Users\Cyril\Documents\GitHub\GLOEiffelProject\Diagrammes\partie1_rapport%20(1).docx#_Toc377586024)

[Figure 20 - Diagramme de déploiement 21](#_Toc377586025)

[Figure 21 - Diagramme de classes affiné 22](#_Toc377586026)

# Introduction

Actuellement en master MIAGE première année, nous suivons un module de formation au génie logiciel à objets dont l’objectif est de nous apprendre à construire des logiciels fiables et de qualité.

Pour atteindre cet objectif, le module se donne pour moyens :

* L’introduction aux problèmes du génie logiciel et aux méthodes de développement
* Les différentes méthodes de spécification et de conception
* L’utilisation du langage UML comme structurant des idées

Afin de mettre en pratique les connaissances acquises durant le module, un projet d’application clôture le module.

Dans ce projet, les étudiants vont devoir mettre en application l’ensemble des connaissances acquises durant le module. Aussi bien les connaissances des méthodes de gestion de cycles de développement, que d’architecture applicative via UML et les modèles de conception.

Le projet se propose de construire une application de gestion d’une médiathèque en utilisant dans un premier temps les outils de conception proposés lors des cours (diagrammes de CU, d’activité, de composants et de déploiement, …) afin de spécifier et concevoir le logiciel. Dans un second temps, nous allons devoir implémenter le logiciel conçu dans le langage Eiffel, conçu selon les paradigmes de la programmation orientée objet.

L’équipe de projet est donc composée par Sébastien Heuzé et Cyril Le Driant, tous deux alternants à Cap Gemini.

Le présent rapport a pour but d’exposer les choix retenus durant les phases d’analyse des besoins, de spécifications et de conception.

Bonne lecture.

# Analyse des besoins

L’analyse des besoins a pour but de mettre en avant les fonctionnalités que doit proposer le logiciel final. Cette partie va contenir l’ensemble des diagrammes compréhensibles par des personnes sans connaissances particulières dans le domaine de l’informatique.

Le logiciel à concevoir est donc un système de gestion d’une médiathèque. Celui-ci couvre donc au moins deux domaines fonctionnels : la gestion des médias proposés et la gestion des adhérents pouvant emprunter les médias.

## Sous système gestion des médias

Le sous-système de gestion des médias propose toutes les fonctionnalités qui sont propres au domaine fonctionnel des médias.

### Diagramme de cas d’utilisation

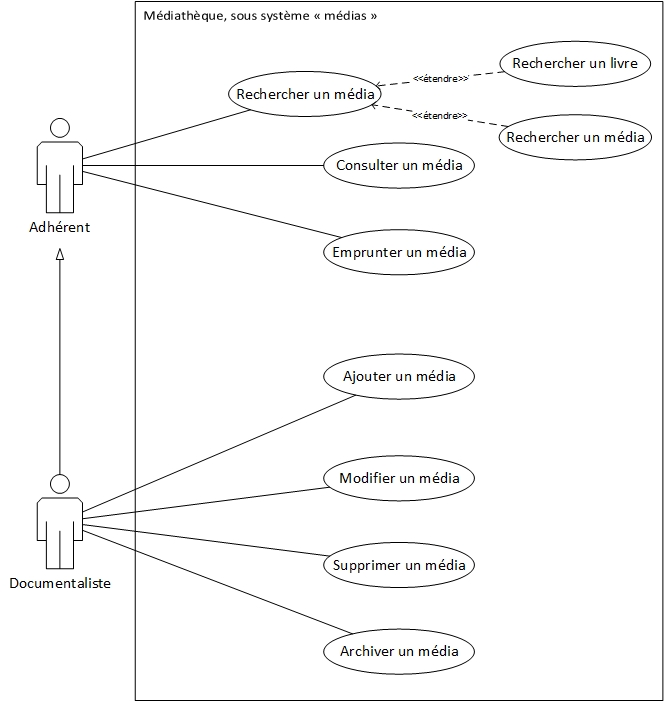
Dans ce diagramme, tous les cas d’utilisation incluant « média » sont étendus par l’application aux DVD et aux livres. Pour une question de lisibilité ils ne sont affichés qu’une seule fois. De plus tous les cas d’utilisation incluent la recherche de média

Figure 1 - Diagramme de CU, SS médias

Le diagramme de cas d’utilisation ci-dessus présente les différentes fonctionnalités que doit permettre le sous-système de gestion des médias.

De manière classique on retrouve l’ensemble des opérations dites CRUD : création de média, lecture des informations portées par un média, modification de média et suppression de média.

Le système permet de faire la différence entre un profil « adhérent » et un profil « documentaliste ». Le documentaliste possède des droits de gestion de la collection de médias en plus des droits classiques de l’adhérent.

### Cartouches

Les cartouches de cas d’utilisation permettent de donner du sens aux cas d’utilisation exprimés dans le diagramme de CU. Pour cela, une description textuelle du cas d’utilisation est faite, en précisant les principales interactions. Les cartouches proposés dans ce rapport prendront la forme suivante :

|  |  |
| --- | --- |
| **Nom du cas d’utilisation** | |
| Acteur principal | L’acteur initialisant l’action |
| Acteurs secondaires | Le ou les acteurs participant à la réalisation de l’action (Facultatif) |
| Description générale | 2 ou 3 phrases décrivant le cas d’utilisation de manière générale |
| Pré condition | Etat du système avant l’action |
| Post condition | Etat du système après l’action |

|  |  |
| --- | --- |
| **Rechercher un média** | |
| Acteur principal | Utilisateur (Adhérent ou documentaliste) |
| Acteurs secondaires | Aucun |
| Description générale | La recherche d’un média permet de retrouver un média existant dans la collection. |
| Pré condition | Aucune |
| Post condition | La recherche retourne bien un résultat, pouvant être vide si la collection de médias ne contient pas le média recherché |

|  |  |
| --- | --- |
| **Consulter un média** | |
| Acteur principal | Utilisateur (adhérent ou documentaliste) |
| Acteurs secondaires | Aucun |
| Description générale | Après la recherche d’un média, il est possible d’en consulter les principales informations. |
| Pré condition | Le média est existant dans la base de données |
| Post condition | Aucune |

|  |  |
| --- | --- |
| **Emprunter un média** | |
| Acteur principal | Utilisateur (adhérent ou documentaliste) |
| Acteurs secondaires | Aucun |
| Description générale | Emprunt d’un média disponible dans la collection de la médiathèque pour un adhérent |
| Pré condition | Le média demandé existe et est disponible à l’emprunt, l’adhérent faisant la demande existe |
| Post condition | Le nombre d’exemplaires disponibles a été diminué du nombre d’exemplaires empruntés. L’emprunt a été ajouté à l’utilisateur pour le tracer. |

|  |  |
| --- | --- |
| **Ajouter un média** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | Afin d’étoffer la collection de la médiathèque, le documentaliste peut ajouter une nouvelle référence dans la collection |
| Pré condition | Aucune |
| Post condition | Le média a bien été ajouté dans la collection |

|  |  |
| --- | --- |
| **Modifier un média** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | Lorsqu’un média contient des informations erronées, le documentaliste doit être en mesure de les corriger. |
| Pré condition | Le média existe dans la collection |
| Post condition | Les modifications que le documentaliste a apportées ont bien été prises en compte. |

|  |  |
| --- | --- |
| **Supprimer un média** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | La suppression d’un média permet de retirer une référence de la collection de médias disponibles de la médiathèque. |
| Pré condition | Le média existe dans la collection |
| Post condition | Le média n’existe plus dans la collection |

|  |  |
| --- | --- |
| **Archiver un média** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | L’archivage d’un média permet de rendre inactif un média. Lors de l’affichage de la liste des médias disponibles, les médias archivés ne sont pas affichés. |
| Pré condition | Le média existe dans la collection et n’est pas archivé |
| Post condition | Le média existe dans la collection et est archivé |

### Diagrammes d’activité

Diagramme d’activité pour l’emprunt d’un média

Ce diagramme présente la succession d’activités permettant de mener à l’emprunt d’un média. Pour que le scénario s’exécute, le média demandé doit être disponible, sinon un message informe le demandeur que le média n’est pas disponible à l’emprunt.

Dans un premier temps, le système affiche les informations sur le média demandé. Si le média affiché ne correspond pas au média recherché, alors on sort du processus.

Dans un second temps, le processus d’emprunt requière les identifiants du demandeur d’emprunt afin de lui affecter cet emprunt de média. On ne gère pas de limite d’emprunts pour une personne.

En fin de processus, le système demande une confirmation à l’utilisateur que le couple « média / emprunteur » est bien correct. Si c’est le cas, le système va enregistrer l’emprunt. Pour l’adhérent, l’emprunt va être ajouté dans la collection de ses emprunts.

Si l’utilisateur choisi d’annuler la transaction, une demande de confirmation apparaît afin de lui faire confirmer son choix.

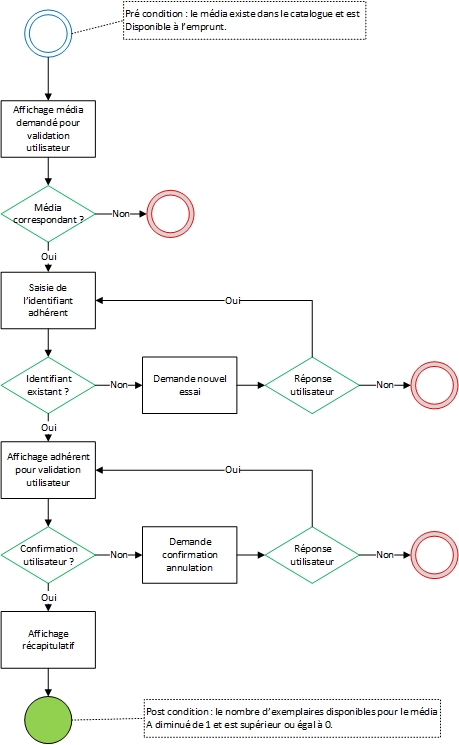


Figure 2 - Diagramme d'activité : emprunt d'un média

### Maquettes d’écran

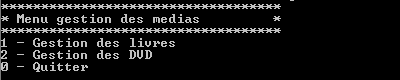


Figure 3 - Menu gestion des médias



Figure 4 - Menu gestion des livres

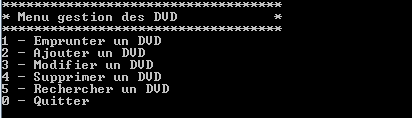


Figure 5 – Menu de gestion des DVD

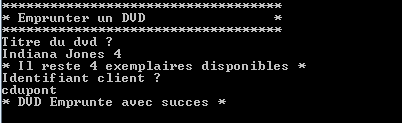


Figure 6 - Emprunt d'un DVD

## Sous-système utilisateurs

### Diagramme de cas d’utilisation

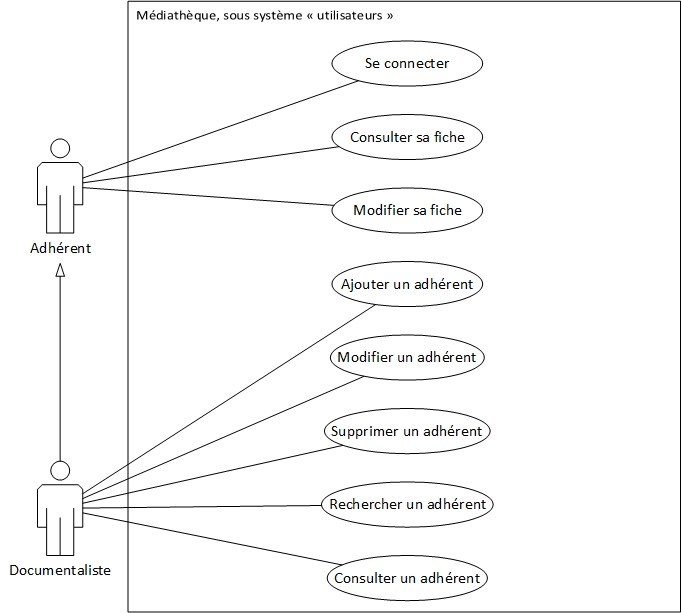
*Dans ce diagramme, tous les cas d’utilisation incluent le fait de se connecter. Pour des raisons de lisibilité on ne les affiche donc pas*.

Figure 7 - Diagramme de CU, SS utilisateurs

Le sous-système des utilisateurs propose également des cas d’utilisation permettant de manière traditionnelle de se connecter, de consulter et modifier sa fiche. Le profil documentaliste vient étendre les fonctionnalités proposées à l’adhérent par un ensemble de possibilités constituant la gestion du répertoire des adhérents. Ainsi le documentaliste va pouvoir créer de nouveaux adhérents, rechercher, modifier, supprimer, consulter les adhérents.

Précision importante : l’ensemble des cas d’utilisation donnés ne peuvent être accomplis qu’après la connexion de l’utilisateur. Pour ne pas surcharger le diagramme, nous avons choisi de ne pas faire figurer toutes les flèches d’inclusion depuis tous les CU vers le CU « se connecter ».

### Cartouches

|  |  |
| --- | --- |
| **Se connecter** | |
| Acteur principal | Utilisateur (adhérent ou documentaliste) |
| Acteurs secondaires | Aucun |
| Description générale | La connexion permet d’accorder des droits d’exécution de fonctionnalités dans le système. |
| Pré condition | Aucun utilisateur n’est connecté |
| Post condition | Le système a enregistré l’utilisateur connecté dans une variable globale |

|  |  |
| --- | --- |
| **Consulter sa fiche** | |
| Acteur principal | Utilisateur |
| Acteurs secondaires | Aucun |
| Description générale | Permet à un utilisateur de consulter les informations qui lui sont propres (nom, prénom, emprunts, etc…) |
| Pré condition | L’utilisateur est connecté |
| Post condition | L’utilisateur n’a pas été modifié |

|  |  |
| --- | --- |
| **Modifier sa fiche** | |
| Acteur principal | Utilisateur |
| Acteurs secondaires | Aucun |
| Description générale | Lorsque des informations d’un utilisateur ne sont pas correctes, l’utilisateur peut modifier ces informations. |
| Pré condition | L’utilisateur est connecté |
| Post condition | Les modifications ont été sauvegardées |

|  |  |
| --- | --- |
| **Ajouter un adhérent** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | Permet au documentaliste d’ajouter un nouvel adhérent à la médiathèque. |
| Pré condition | L’adhérent n’existe pas (pas de doublon dans la base) |
| Post condition | Le nouvel adhérent existe dans la base de données |

|  |  |
| --- | --- |
| **Modifier une fiche d’adhérent** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | Permet au documentaliste de modifier la fiche d’un adhérent |
| Pré condition | Le documentaliste est connecté, la fiche adhérent existe |
| Post condition | Les modifications ont bien été prises en compte |

|  |  |
| --- | --- |
| **Supprimer un adhérent** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | Permet au documentaliste de supprimer un adhérent |
| Pré condition | Le documentaliste est connecté et la fiche adhérent existe |
| Post condition | La fiche adhérent a été supprimée |

|  |  |
| --- | --- |
| **Rechercher un adhérent** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | Permet au documentaliste de rechercher un adhérent |
| Pré condition | Le documentaliste est connecté et la fiche adhérent existe |
| Post condition | Aucune |

|  |  |
| --- | --- |
| **Consulter un adhérent** | |
| Acteur principal | Documentaliste |
| Acteurs secondaires | Aucun |
| Description générale | Permet d’afficher la fiche de l’adhérent |
| Pré condition | Le documentaliste est connecté et l’adhérent existe |
| Post condition | Aucune |

### Maquettes d’écran



Figure 8 - Menu gestion utilisateurs

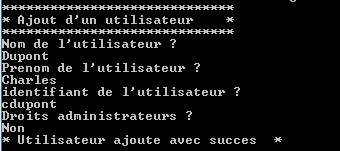


Figure 9 - Ajout d'un utilisateur

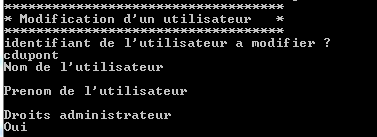


Figure 10 - Modification d'un utilisateur

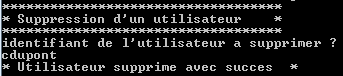


Figure 11 - Suppression d'un utilisateur

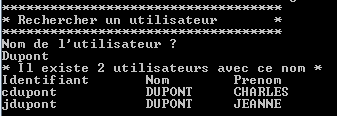


Figure 12 - Recherche d'un utilisateur

Le reste des écrans CRUD n’étant qu’une suite de questions suivant le même principe que l’ajout, modification, suppression de l’utilisateur, ils ne seront pas représentés afin de ne pas encombrer ce rapport.

# Analyse

Principes et objectifs

L’analyse du système a pour but de modéliser le domaine fonctionnel couvert par le futur logiciel. Le besoin est exprimé sous forme de diagrammes respectant des normes de modélisation afin d’être à la fois rigoureux et compréhensibles par des personnes n’ayant pas de compétences particulières dans les techniques de modélisation ni dans le domaine fonctionnel.

Dans cette partie vont être présentés différents diagrammes UML permettant de décrire certaines fonctionnalités du logiciel ainsi que son organisation logique.

## Diagrammes de collaboration

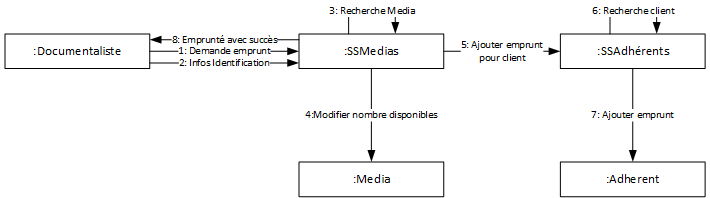


Figure 13 - Diagramme de collaboration

Ce diagramme de collaboration reprend le cas « emprunt d’un média », cela permet de voir comment les différentes classes de l’application communiquent entre elles.

Pour plus d’informations, se référer aux [diagrammes de séquence](#_Diagrammes_de_séquence) de la partie analyse.

## Diagramme de classes.

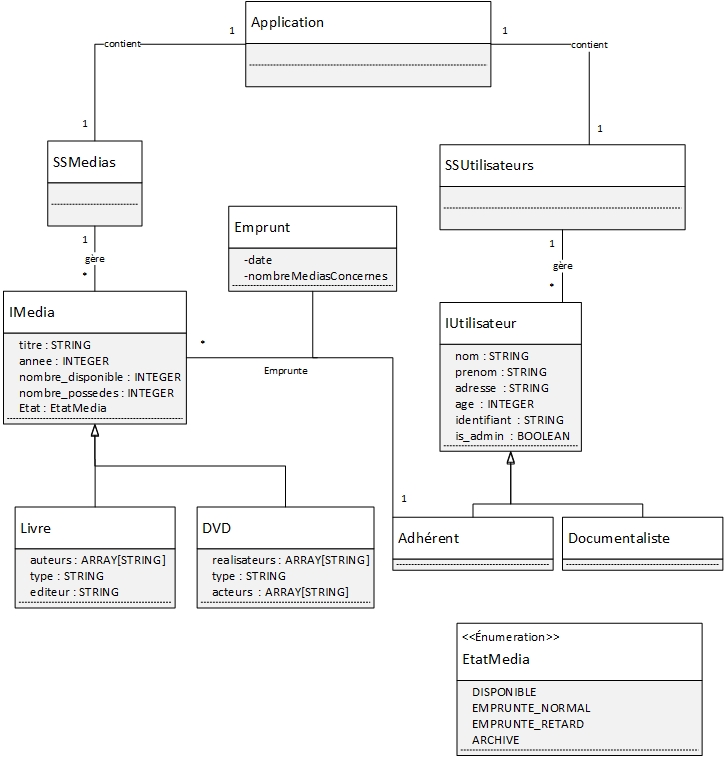


Figure 14 - Diagramme de classe

Le programme est structuré comme suit :

* Une classe principale : Application.
* Deux classes intermédiaires : SSMedias et SSUtilisateurs, ces classes vont permettre à la classe principale d’interagir avec les différentes classes de type IMedia et IPersonne (Gestion, ajout, suppression)
* Deux interfaces : IMedia et IUtilisateur, ces deux classes sont les deux interfaces qui représentent les deux principaux éléments de l’application :
  + Medias (Livres, DVD)
  + Utilisateurs (Adhérents, Documentaliste)
* La classe permettant de tracer les emprunts des utilisateurs.

## Diagrammes de séquence

### Ajout d’un média

Dans ce cas d’utilisation, la documentaliste interagit avec le système afin d’ajouter un média (existant ou non) dans la collection de la médiathèque.

Lorsque le média existe déjà dans la collection, alors on incrémente le nombre de médias disponibles, sinon on crée un nouveau média et on l’ajoute dans la collection des médias proposés.

### Emprunter un média

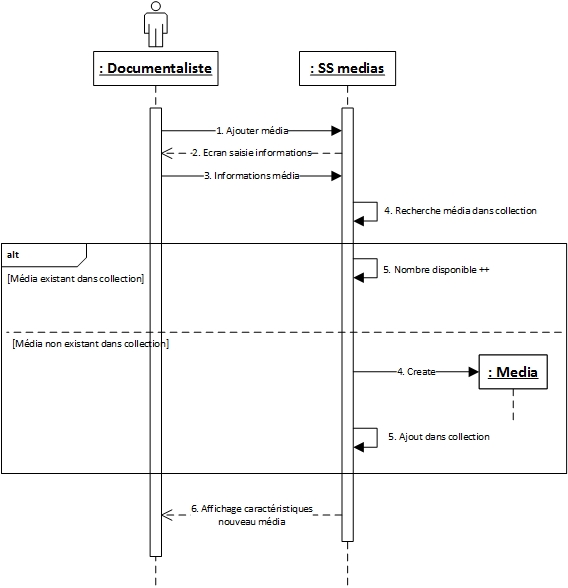


Figure 15 - Diagramme de séquence - Ajout d'un média

Lors de l’emprunt d’un média, le système demande deux informations à l’utilisateur : l’identifiant de l’utilisateur et l’identifiant du média. Lors de l’emprunt, le système recherche le média demandé, et diminue le nombre d’exemplaires disponibles. On stocke également dans l’adhérent la liste des médias empruntés, on va donc lui ajouter ce média. Pour finir, l’application affiche une validation que l’emprunt a bien été affecté à l’adhérent. Un affichage pourra être réalisé afin d’informer l’utilisateur du nouveau nombre d’exemplaires disponibles ainsi que de la liste des emprunts réalisés par l’adhérent.

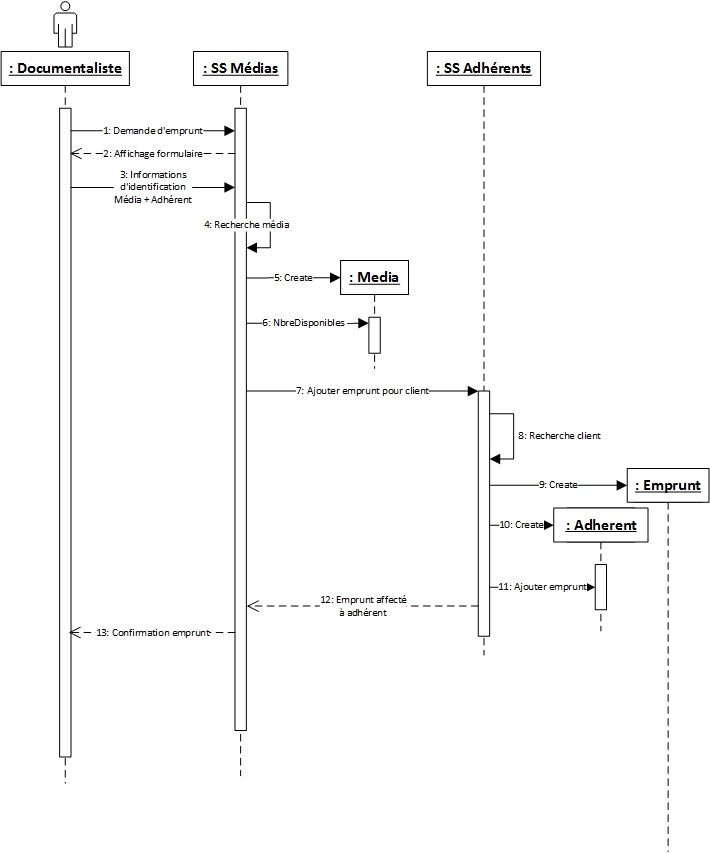


Figure 16 - Diagramme de séquence - Emprunter média

### Suppression d’un média

La suppression d’un média correspond à la suppression définitive d’un média de la collection, qu’importe le nombre d’exemplaires disponibles pour ce média. Voici le diagramme de séquence correspondant :

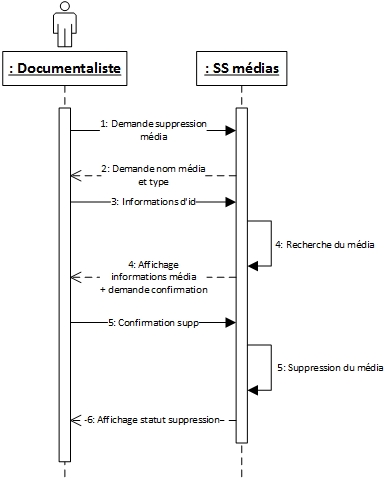


Figure 17 - Diagramme de séquence, suppression d'un média

C’est la documentaliste qui lance le processus de suppression du média. Le système répond par une demande d’informations permettant d’identifier le média dans la collection. Le média est recherché selon ces informations puis affiché à la documentaliste pour demande de confirmation. Si la confirmation est faite, le système de gestion des médias supprime le média et affiche le statut de la suppression à la documentaliste.

## Diagrammes d’états

Dans le système, seul le média est soumis à un état. Il peut passer de l’état disponible à l’état emprunté et inversion.

### Diagramme d’états pour le média

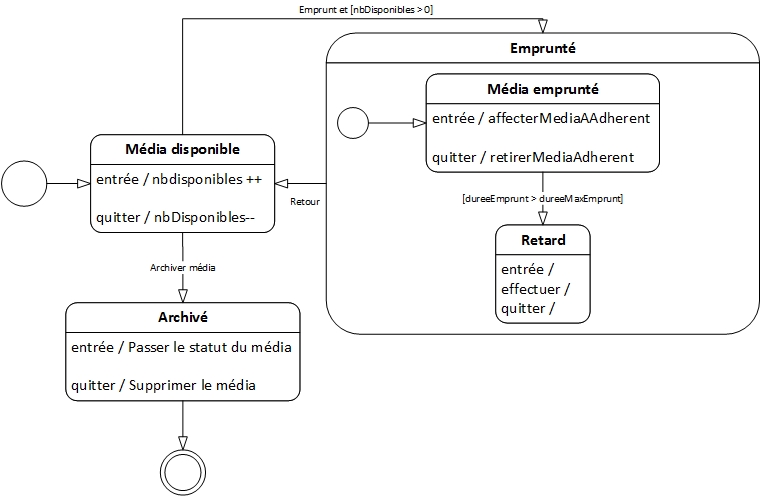


Figure 18 - Diagramme d'états - média

Lorsqu’un média est ajouté dans la collection, il devient disponible à l’emprunt. Si le nombre disponible de média est supérieur à zéro, alors on peut l’emprunter. Lors de l’emprunt, le média passe dans le super état « emprunté ». Si l’emprunt vient d’être fait, le média est dans l’état « emprunté ». Lorsque la durée d’emprunt maximale est dépassée, le média passe dans l’état « retard ». Lors du retour du média, l’état repasse en « disponible ». Le média peut être archivé lorsqu’il n’est plus utilisé afin de ne plus pouvoir l’emprunter. Seul un média archivé peut être supprimé définitivement de la collection.

# Conception

La conception a pour but de répondre de manière technique aux besoins posés dans la phase d’analyse. Pour cela, des diagrammes d’architecture (logicielle et technique) seront utilisés. Une décomposition modulaire de l’application va également être proposée.

La conception peut être générale (abstraite) ou détaillée, avec une décomposition qui rentre dans les détails. Cette partie du document propose une décomposition générale permettant une vue d’ensemble sur le système.

## Architecture générale

L’application se décompose en trois grands modules principaux : le module médias, le module utilisateurs puis le module services.

Le module utilisateur et médias sont deux modules dits « métiers », permettant de stocker des informations contenues dans le système de fichiers, qui nous sert de base de données. Le module utilisateur permet de créer et utiliser les différents types d’utilisateur, et le module médias permet de manipuler les différents médias utilisés par la médiathèque.

Chacun de ces modules expose une interface définissant des méthodes généralistes et communes aux différents types d’utilisateurs et médias permettant de manipuler les objets.

Le module « services » propose un ensemble de fonctions permettant de manipuler les objets métiers. Les fonctions sont par exemple : l’ajout d’un média dans la médiathèque, l’emprunt ou à la suppression du média.

## Diagramme de composants

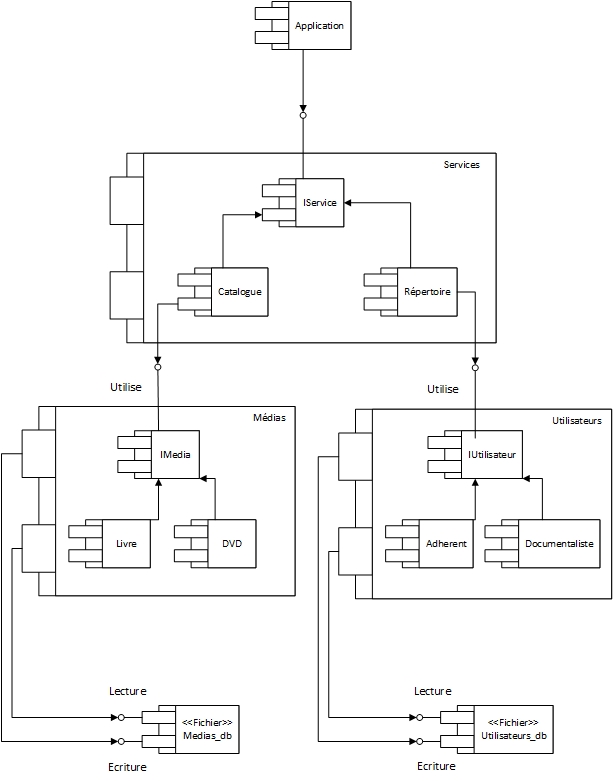


Figure 19 - Diagramme de composants

Le système se décompose donc en trois grands modules : les services, les médias et les utilisateurs. Le composant médias propose des objets représentant le concept de média : livre, film, etc.

L’interface IMedia propose des méthodes et attributs communs à tous les médias : titre, auteur, année, …

Pour les utilisateurs, c’est l’interface IUtilisateur qui propose des propriétés propres à tous les utilisateurs : nom, prénom, âge, type, …

Ces deux composants sont gérés par le troisième, qui se compose d’objets exposant des services permettant de gérer les deux types de composants manipulés : ajout / suppression / emprunt / restitution de média.

La séparation en couches permet de cloisonner les responsabilités des différents composants. L’application est le composant qui va dialoguer avec l’utilisateur, la couche services est celle qui propose la manipulation des objets métiers. Cette décomposition est une architecture en couches, chaque couche utilisant seulement la couche qui lui est immédiatement inférieure.

## Diagramme de déploiement

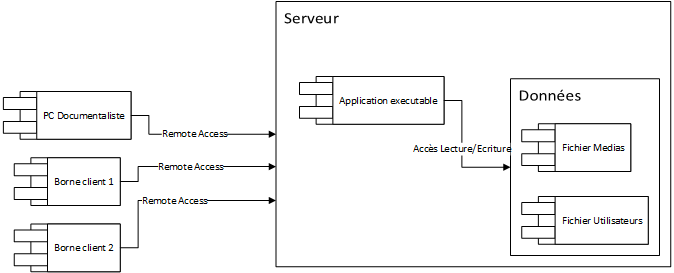


Figure 20 - Diagramme de déploiement

L’application sera exécutée sur un serveur, celle-ci pourra être exécutée à distance par différents PC.

* PC documentaliste : le poste de la documentaliste qui permettra de gérer les médias
* Bornes clients : les bornes clients permettront aux clients d’effectuer une recherche de médias, voir les détails de ce média, le nombre de médias disponibles.

Différentes méthodes d’accès à distance existent, pour une application en mode console, psexec sera suffisante.

L’application étant ouverte plusieurs fois, il faudra s’assurer de mettre à jour continuellement le fichier des médias et utilisateurs qui servira de base de données.

# Implémentation

L’implémentation consiste à traduire dans un langage de programmation les différents concepts modélisés lors de la phase de conception. Dans notre cas le langage de programmation choisi est Eiffel, langage orienté objet.

Cette phase se mêle avec la phase de testing durant laquelle chaque nouvelle fonctionnalité doit être testée pour le cas nominal, mais également pour les cas aux limites afin de s’assurer du bon comportement du logiciel dans des cas pour lesquels il n’est pas censé fonctionner.

## Un diagramme de classes affiné

Lors de l’implémentation, nous nous sommes aperçu que la modélisation était incomplète, car ne permettant pas de gérer correctement les emprunts. Nous avons donc revu le diagramme de classes afin de le rendre plus facile à intégrer et plus maintenable (une personne doit pouvoir travailler sur du code qui n’est pas le sien).

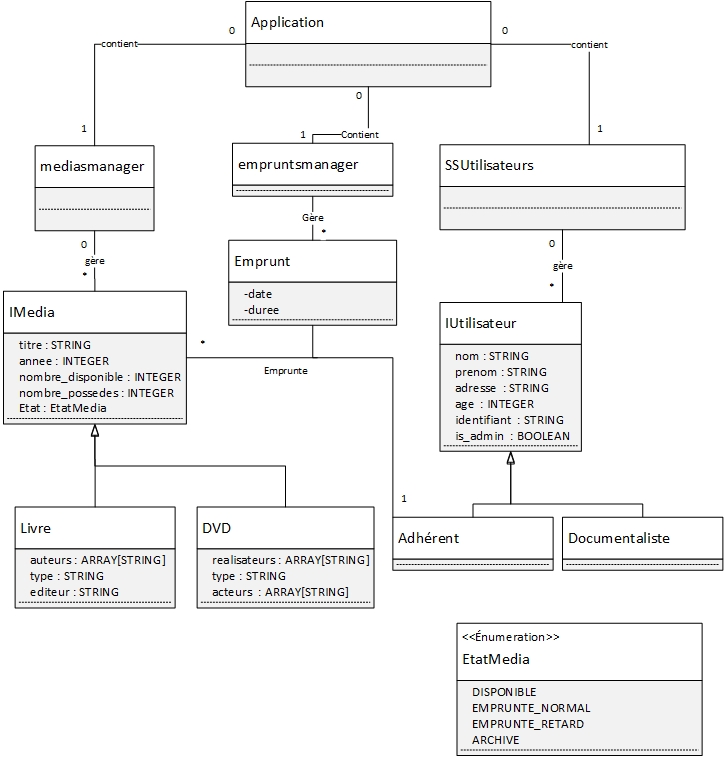


Figure 21 - Diagramme de classes affiné

# Conclusion générale

## Etat des lieux des développements

Le logiciel produit permet de répondre au besoin exprimé, à savoir gérer une médiathèque ainsi que ses adhérent. Cependant, les tests fonctionnels ne couvrent pas tout le périmètre du logiciel dans des cas limites. Ainsi, lors de la démonstration, Benoît Delahaye nous a facilement démontré que dans certains cas le logiciel plantait, montrant ainsi des lacunes de robustesse.

Pour exécuter le programme existant, il suffit de se placer dans le dossier contenant les binaires et de lancer la commande « ./a.out », le menu de l’application devrait s’afficher et proposer les quelques fonctionnalités basiques énoncées précédemment.

## Bilan

Ce projet nous a permis de mettre en place les enseignements reçus pendant les cours, notamment en termes de cycle de vie d’un projet : application des différentes phases, livraisons intermédiaires, etc. mais également en terme d’UML et de pratiques de développement.

Au niveau du cycle de développement, il aurait été intéressant de pouvoir mettre en application l’une des méthodes agiles que nous avions étudiées en première partie de semestre afin de mieux les comprendre et assimiler.

En outre, nous tenions à remercier Benoît Delahaye pour son encadrement, convivial, souple et efficace, qui nous a permis d’assimiler les préceptes du génie logiciel dans une bonne ambiance.