



GESTION D'UNE MÉDIATHÈQUE: Rapport de CONCEPTION:

GROUPE H :

**ISSAD Mehdi-Mayas
BEN MIMOUN Iheb
RAGON Clémence**

Sommaire:

1.Langages de programmation.....page 3

1.1Les langages choisis.....page 3

1.2Les avantages du PHP.....page 4

2.Architecture utilisée.....page 5

3.Diagramme de classe.....page 7

4.Diagramme de cas d'utilisation.....page 10

4.1-Créer un compte.....page 10

4.2-Ajouter une ressource..... page 11

4.3-Retirer une ressourcepage 12

4.4-Emprunter une oeuvrepage 13

4.5-Gérer les retourspage 14

5.Diagramme de séquencepage 15

5.1-Créer un comptepage 15

5.2-Enregistrer un adhérentpage 16

5.3-Se connecterpage 17

5.4-Se déconnecterpage 18

5.5-Ajouter une ressourcepage 19

5.6-Retirer une ressourcepage 20

5.7-Emprunter une oeuvre.....page 21

5.8-Acheter une oeuvre.....page 22

5.9-Gérer les retourspage 23

6.Diagramme d'activitépage 24

6.1-Créer un comptepage 24

6.2-Emprunter une oeuvrepage 25

6.3-Etat de l'oeuvrepage 26

1 - Langage de programmation:

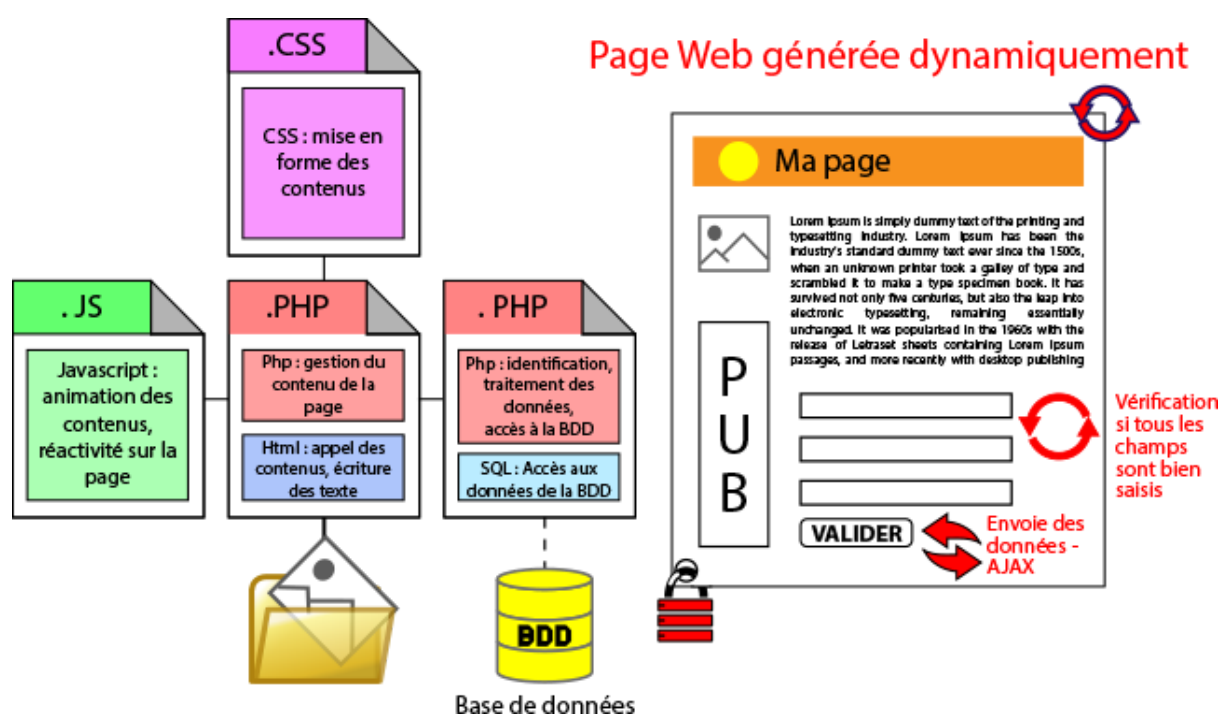
1.1 - Les langages choisis:

- Afin de concevoir le site web de notre médiathèque, nous allons utiliser principalement le langage PHP qui est considéré comme le langage de programmation le plus adapté à la conception d'un site web dynamique.

- Nous allons nous appuyer sur une base de données qu'on va administrer en utilisant des requêtes SQL.

- Nous allons également implémenter quelques fonctions en javascript, notamment lors de la création des boutons.

- Afin de coder l'interface de notre site, nous allons utiliser du HTML ainsi que du CSS.



1.2-Avantages de l'utilisation du langage PHP:

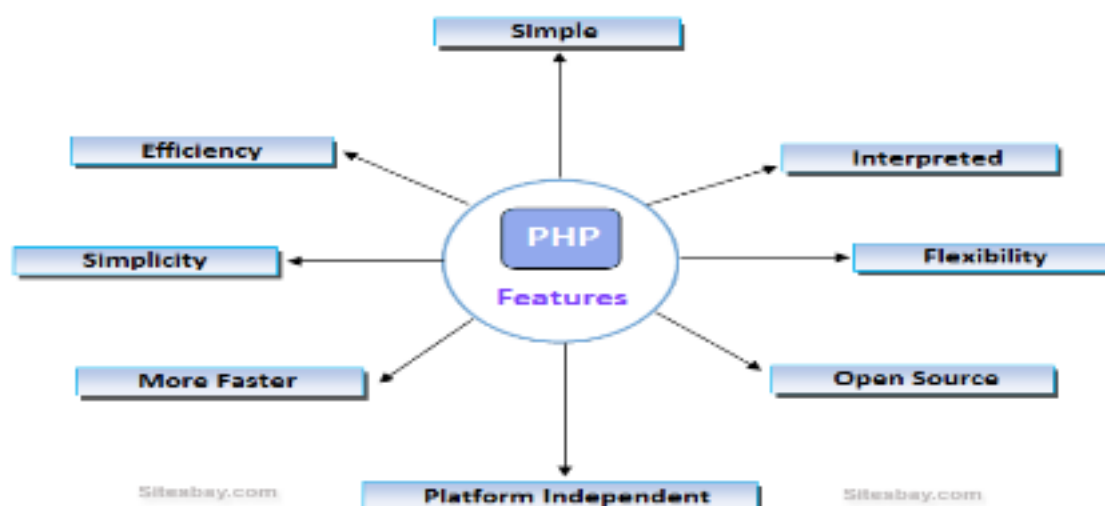
-Le PHP gère très bien les requêtes MySql contrairement à JAVA, nous pourrions donc facilement écrire des programmes qui affichent des données extraites de bases SQL , ou qui stockent des données postées par un formulaire dans une table SQL. S'ajoute à cela le fait que le langage PHP sait communiquer avec presque tous les SGBD contrairement aux autres langages qui sont plus ou moins limités dans ce point la .

Nous pourrions donc utiliser des SGBD tel que Oracle , MySql , Db2 , Informix , Postgresql , SQL Server , Acces etc .. , on voit donc que le choix s'ouvre à nous avec de multiples options dans un niveau d'abstraction concernant les bases de données.

-Il existe un large panel d'offre mettant à la disposition du client un répertoire et un serveur apache qui sera très stable. Le déploiement du site web sera simple et peu coûteux.

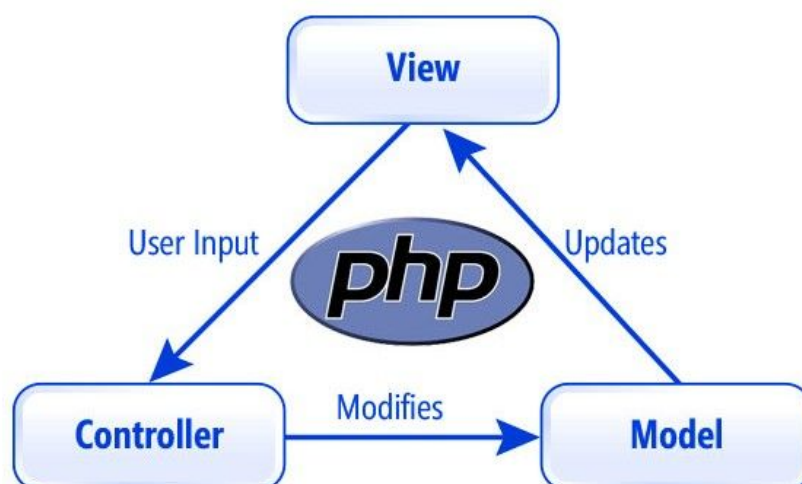
PHP est principalement conçu pour servir de langage de script côté serveur, ce qui fait qu'il est capable de réaliser tout ce qu'un script CGI quelconque peut faire, comme collecter des données de formulaire, générer du contenu dynamique, ou gérer des cookies.

-Le langage Java est compilé avant d'être exécuté ce qui permet d'éviter un grand nombre de bug. Cependant lorsqu'une erreur d'exécution est détectée alors l'application cesse de fonctionner ce qui n'est pas le cas en PHP qui est un langage non compilé.



2 - Architecture utilisée:

- Nous allons utiliser une architecture logicielle MVC (modèle-vue- controller). Elle permet de bien organiser son code source et de le rendre plus clair en définissant un rôle pour chaque fichier. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.



- Modèle :

-Élément qui contient les données ainsi que de la logique en rapport avec les données: validation, lecture et enregistrement. Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application. Par exemple pour une application de banque, le modèle représente des comptes, des clients, ainsi que

les opérations telles que dépôt et retraits, et vérifie que les retraits ne dépassent pas la limite de crédit.

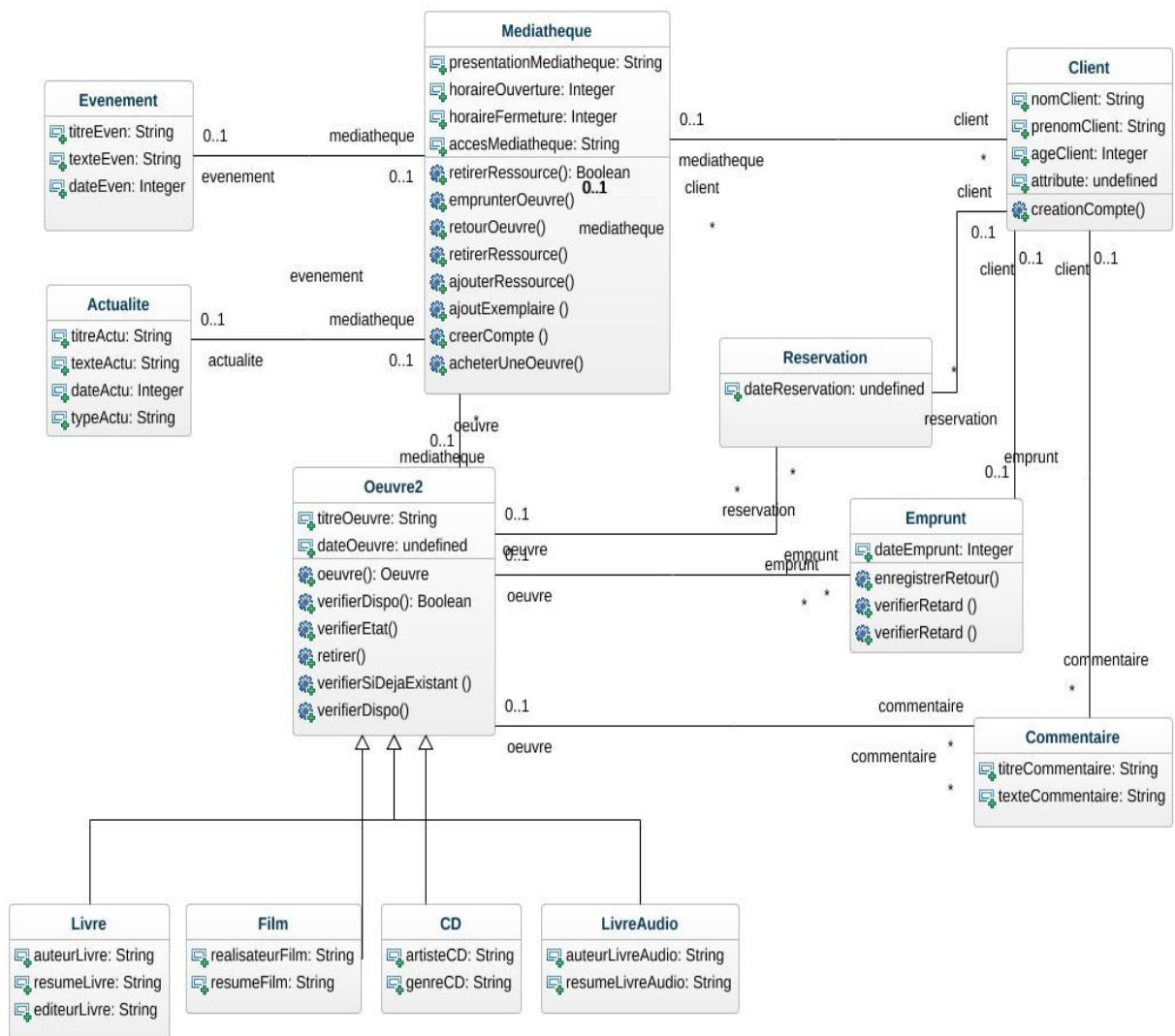
- Vue :

Partie visible d'une interface graphique. La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle. Dans une application de bureau classique, la vue obtient les données nécessaires à la présentation du modèle en posant des questions. Elle peut également mettre à jour le modèle en envoyant des messages appropriés. Dans une application web une vue contient des balises HTML.

- Contrôleur :

Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

3- Diagramme de Classe:



Notre diagramme de classe comporte 12 classes :

- Évènement
- Actualité
- Médiathèque
- Oeuvre
- Réservation
- Emprunt
- Client
- Commentaire
- Livre
- Film
- CD
- LivreAudio

Classe Evenement:

-Attributs:

titreEven, dateEven, texteEven

Classe Actualite:

-Attributs:

titreActu, texteActu, dateActu

Classe Mediathèque:

-Attributs:

presentationMediathèque, horaireOuverture, horaireFermeture,
accesMediathèque

-Opérations:

retirerRessource(), emprunterOeuvre(), retourOeuvre(),
ajouterRessource(), ajoutExemplaire(), creerCompte(),
acheterUneOeuvre()

Classe Reservation:

-Attributs:

dateReservation

Classe Client:

-Attributs:

nomClient, prenomClient, ageClient

-Opérations:

creationCompte()

Classe Emprunt:

-Attributs:

dateEmprunt

-Opérations:

enregistrerRetour(), verifierRetard()

Classe Oeuvre:

-Attributs:

titreOeuvre, dateOeuvre

-Opérations:

VérifierDispo(), verifierEtat(), retirer(), verifierSiDejaExistant(),
verifierDispo()

Classe commentaire:

-Attributs:

titreCommentaire, texteCommentaire

Classe Livre:

-Attributs:

auteurLivre, resumeLivre, editeurLivre

Classe Film:

-Attributs:

realisateurFilm, resumeFilm

Classe CD:

-Attributs:

artisteCD, genreCD

Classe LivreAudio:

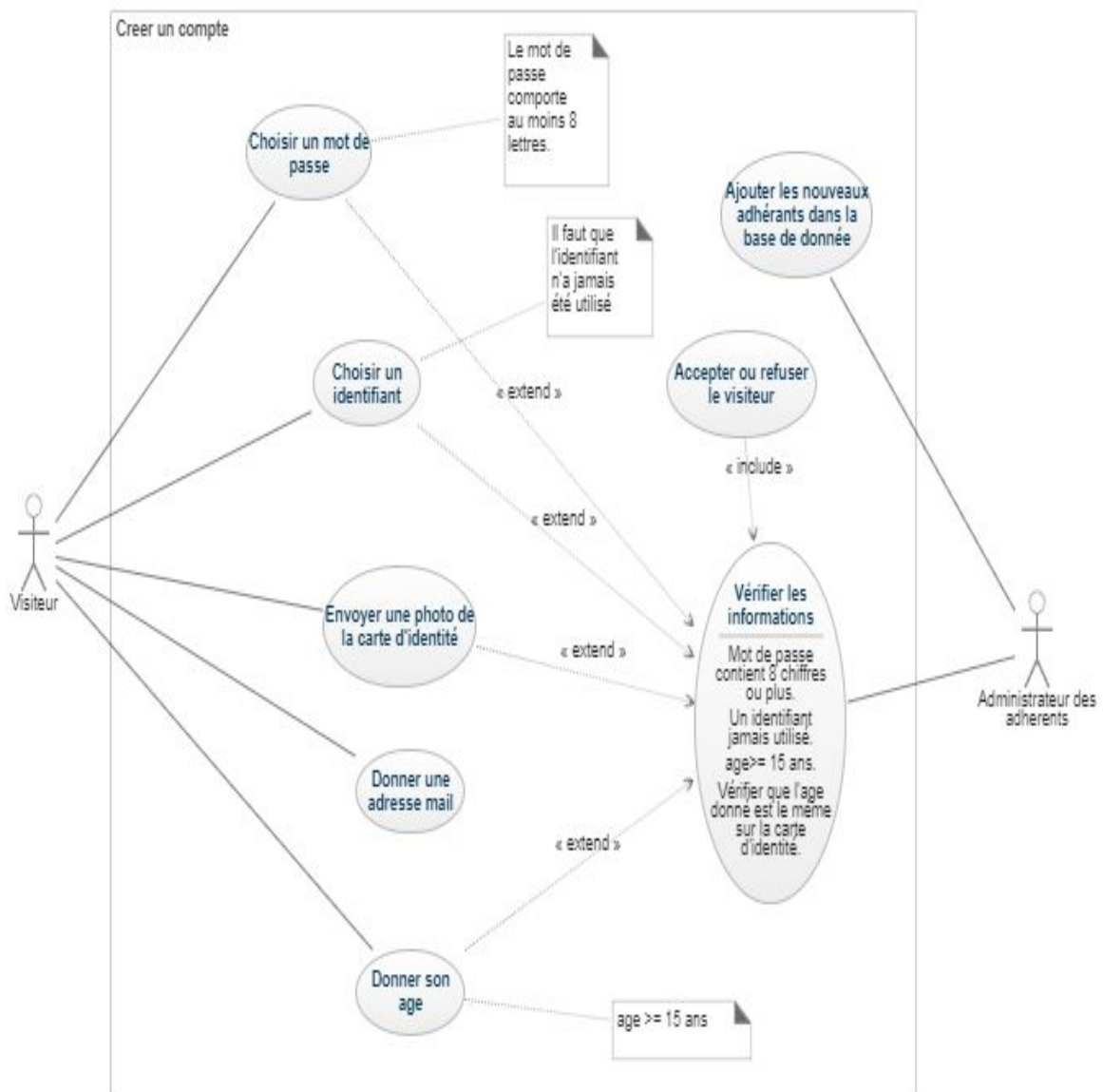
-Attributs:

auteurLivreAudio, resumeLivreAudio

4- Diagrammes de cas d'utilisation:

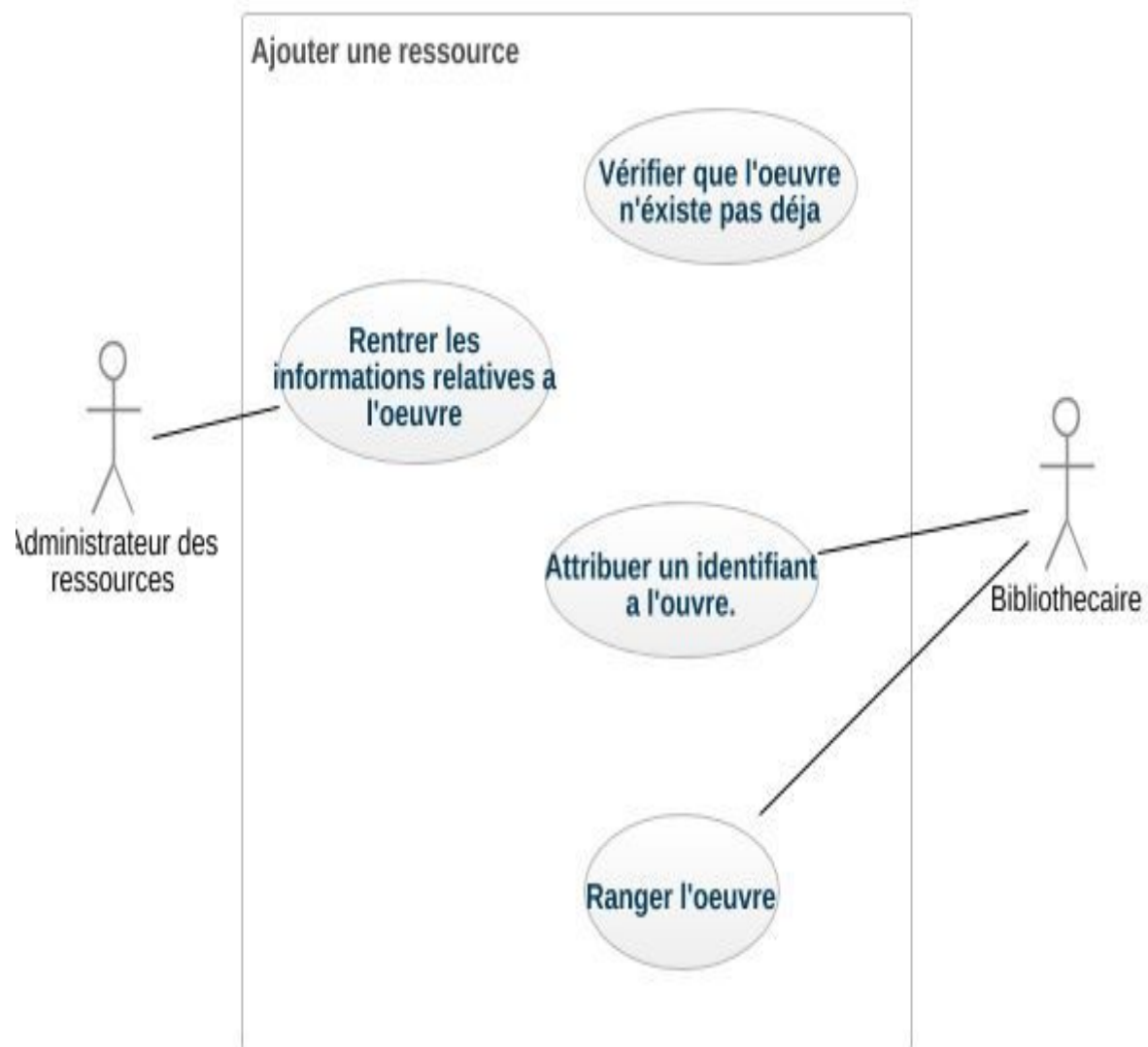
4.1- Créer un compte:

Cela permet au visiteur d'avoir accès à de nouvelles fonctionnalités. Le visiteur saisit son nom, prénom, âge, adresse mail. L'Administrateur des adhérents vérifie que les informations sont correctes et enfin crée le compte.



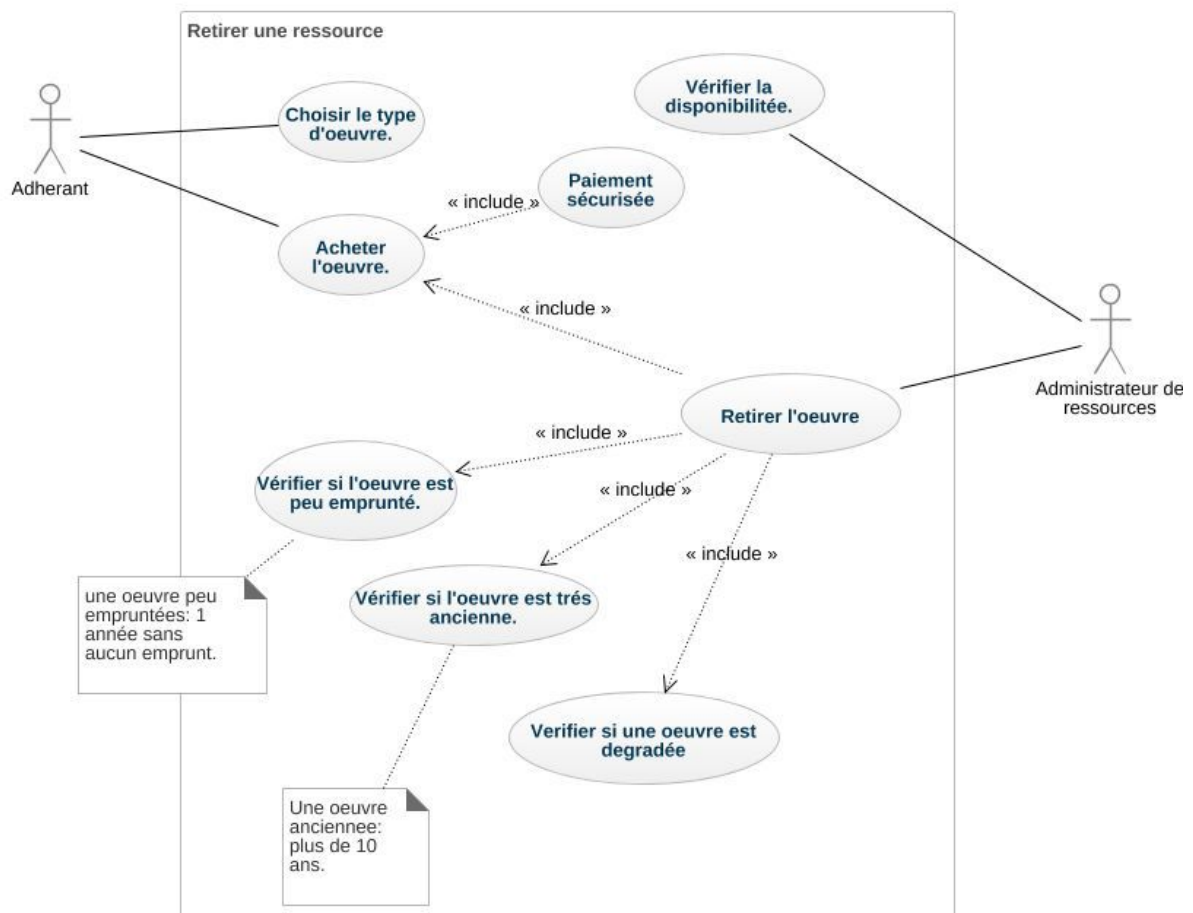
4.2- Ajouter une ressource:

L'administrateur des ressources rentre les informations relatives à l'oeuvre, le bibliothécaire attribue un identifiant à l'oeuvre et la range. Le système vérifie ensuite que l'oeuvre n'existe pas déjà.



4.3- Retirer une ressource:

L'administrateur des ressources vérifie si l'oeuvre est achetée, peu empruntée, très ancienne ou dégradée. Si elle vérifie une de ces conditions, il la retire.

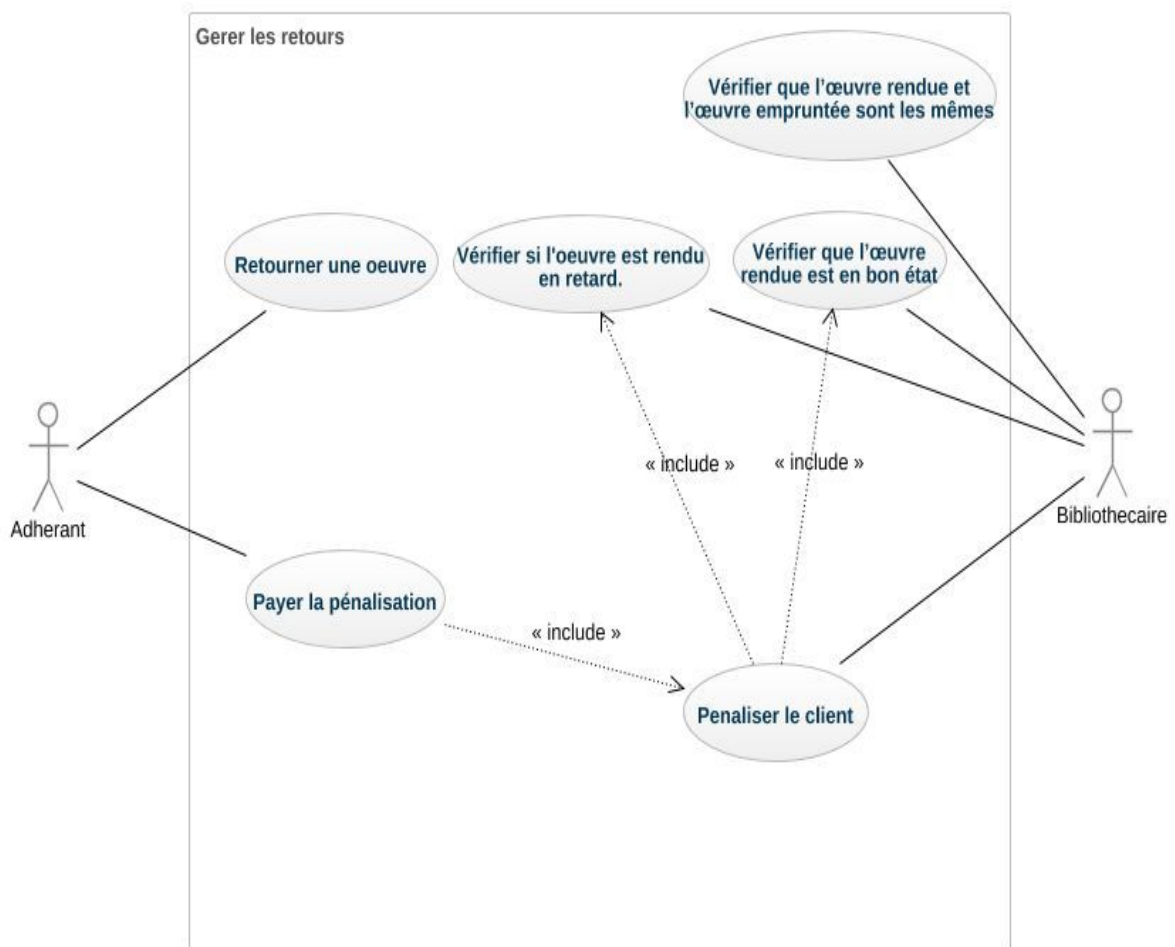


L'adhérent choisit le type d'oeuvre qu'il va emprunter. Le bibliothécaire cherche l'oeuvre, effectue quelques vérifications et ensuite marque la date d'emprunt et de retour et associe chaque oeuvre à l'identifiant de l'emprunteur.



4.5- Gérer les retours :

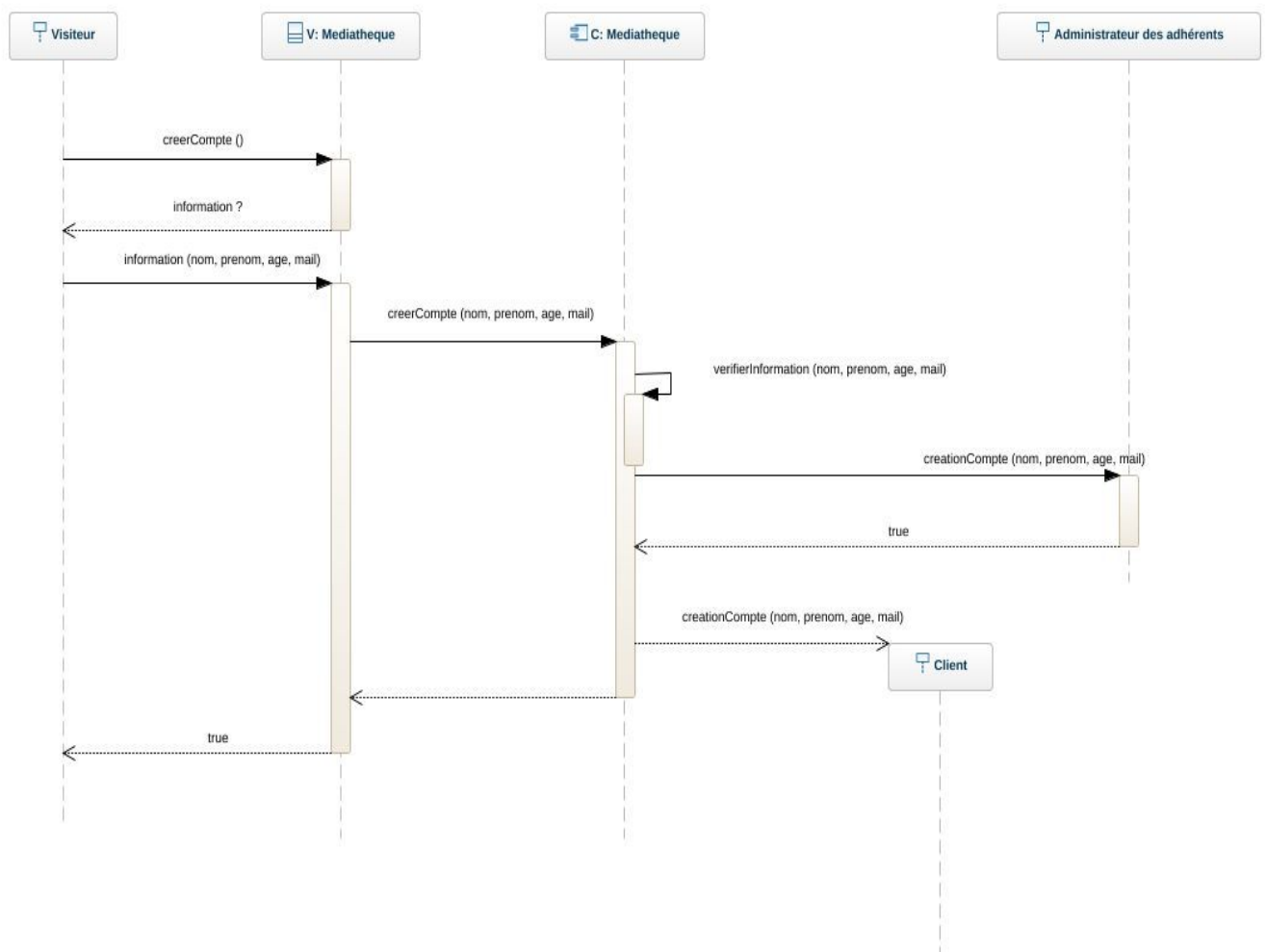
Lorsqu'un adhérent retourne une oeuvre, le bibliothécaire vérifie si elle n'est pas rendue en retard, si elle correspond à l'oeuvre empruntée et si elle est en bon état. Dans le cas contraire l'adhérent reçoit une pénalité qu'il devra payer.



5- DIAGRAMMES DE SÉQUENCES:

5.1- Créer un compte:

- Pour créer un compte le visiteur doit saisir ses informations(nom, age, prenom, mail), le contrôleur de la médiathèque vérifie les informations et les transmet vers l'administrateur des adhérents qui l'enregistre.



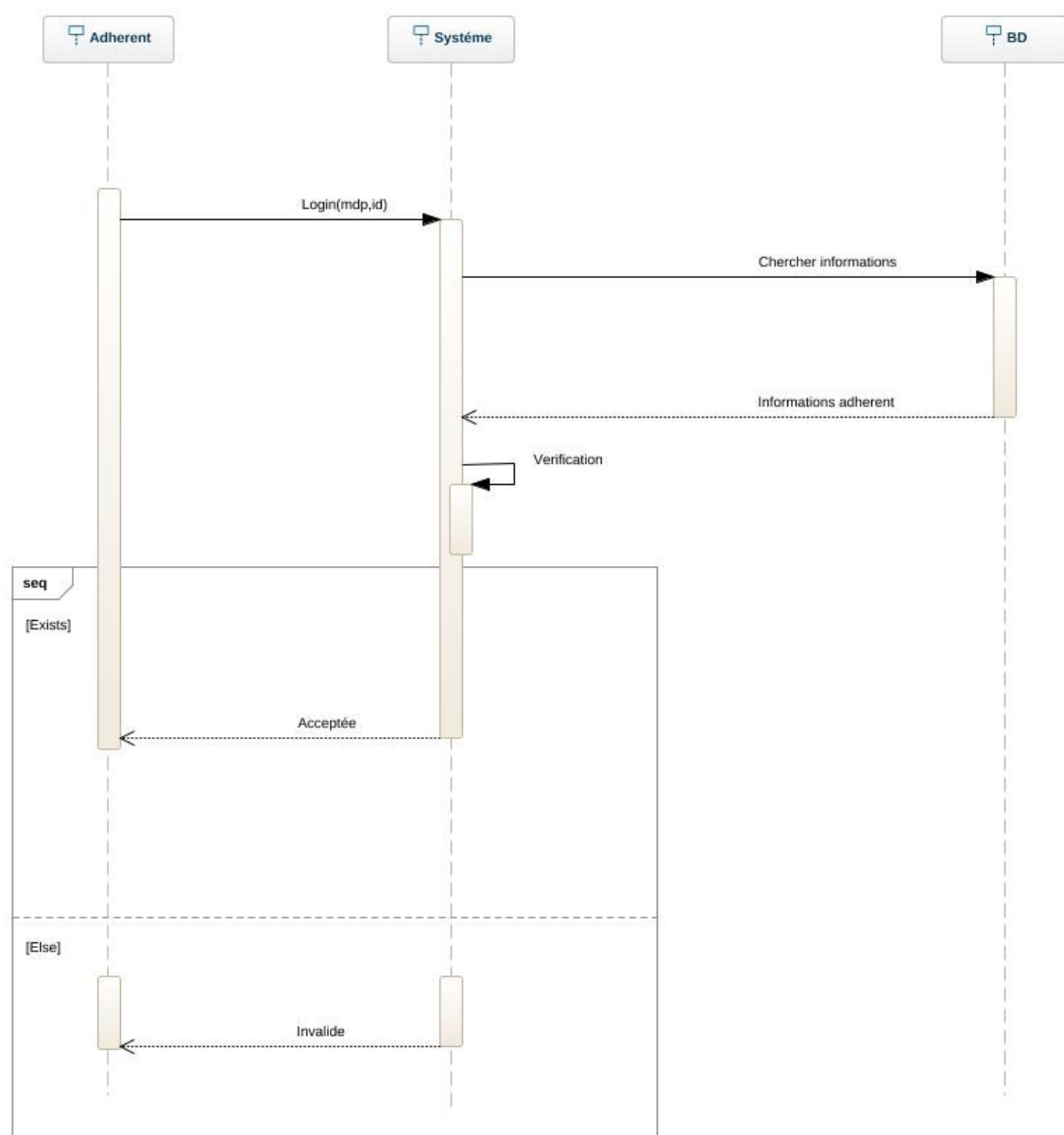
5.2- Enregistrer un adhérent :

- l'administrateur des adhérents vérifie les informations saisies par le futur adhérent avant de confirmer l'enregistrement.



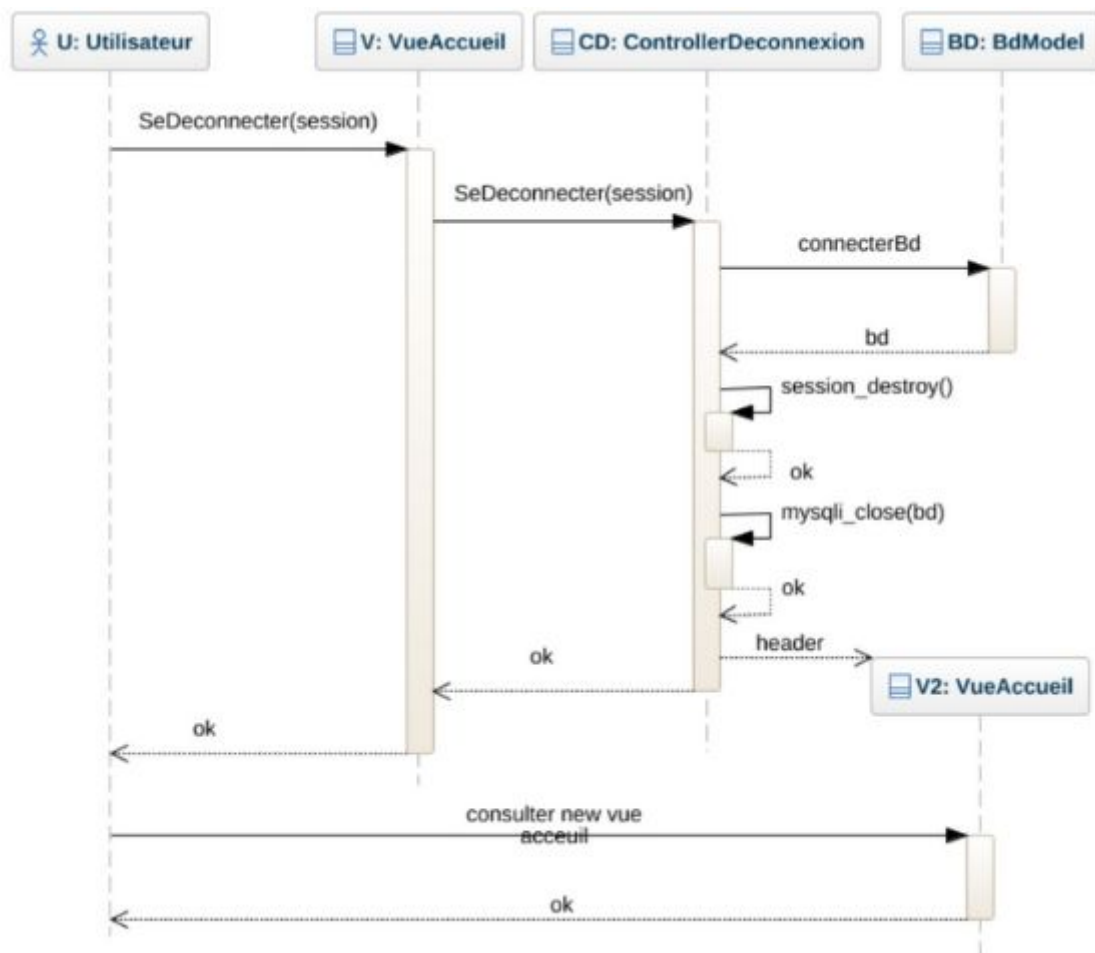
5.3- Se connecter:

- L'adhérent saisit son mot de passe et son login, le système va chercher ses informations dans la base de données et les vérifie, si le mot de passe et le login sont justes il peut s'authentifier sinon il doit ressaisir le mot de passe et le login à nouveau.



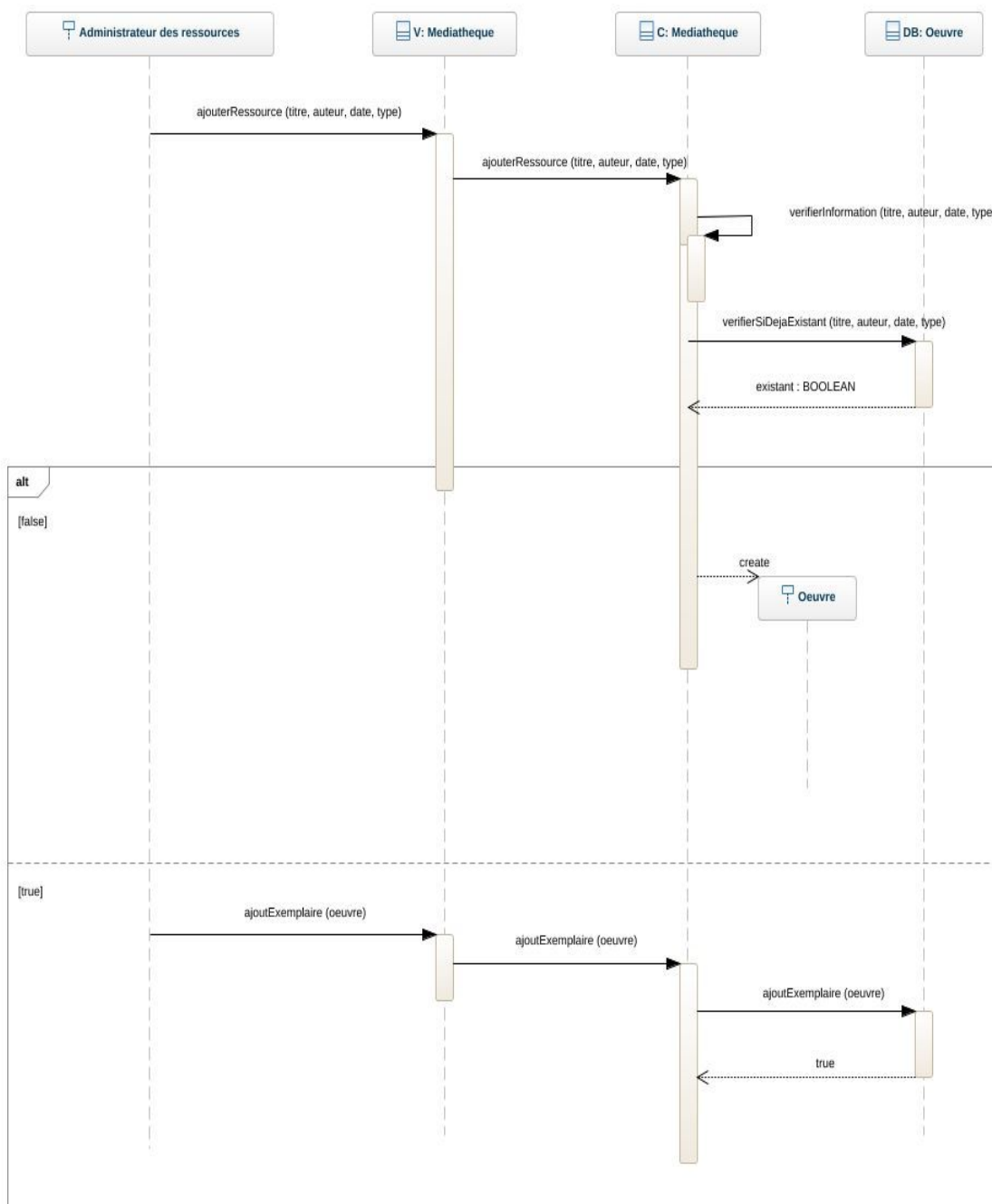
5.4-Se déconnecter:

-L'utilisateur fait une demande de déconnexion, le système va retirer sa session de la base de donnée et faire retourner le client à l'accueil.



5.5- Ajouter une ressource:

-L'administrateur des ressources saisit les informations d'une oeuvre, la médiathèque vérifie les informations, puis vérifie si cette oeuvre existe déjà dans la base de donnée du médiathèque avant de l'ajouter.



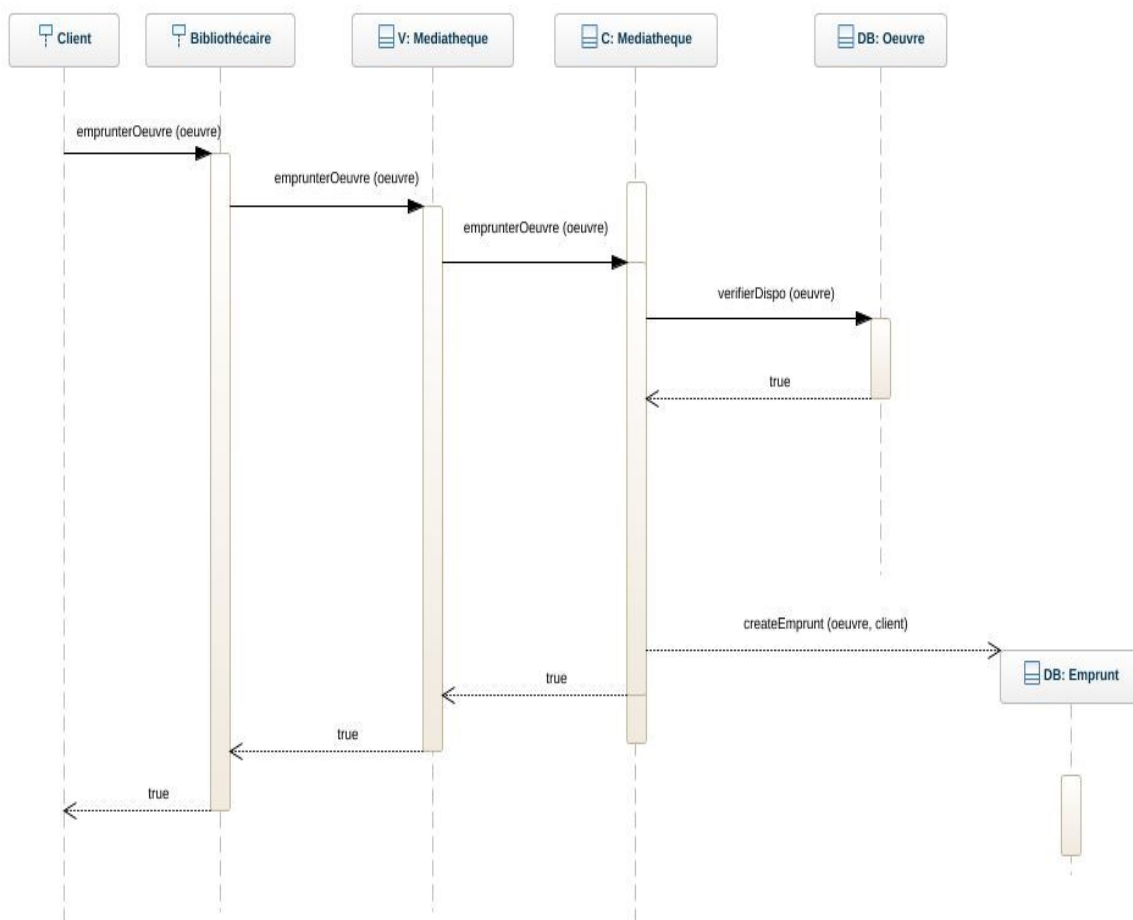
5.6- Retirer une ressource:

-L'administrateur des ressources envoie au médiathèque qu'il veut retirer une ressource, le contrôleur du médiathèque vérifie l'état de l'oeuvre (peu emprunté, durée de vie, ..) avant de le retirer.



5.7- Emprunter une oeuvre:

-L'adhérent donne les informations de l'oeuvre qu'il veut emprunter au bibliothécaire, celui-ci saisit les informations dans la médiathèque qui va chercher la disponibilité dans la base de données, s'il est disponible l'adhérent l'emprunte.



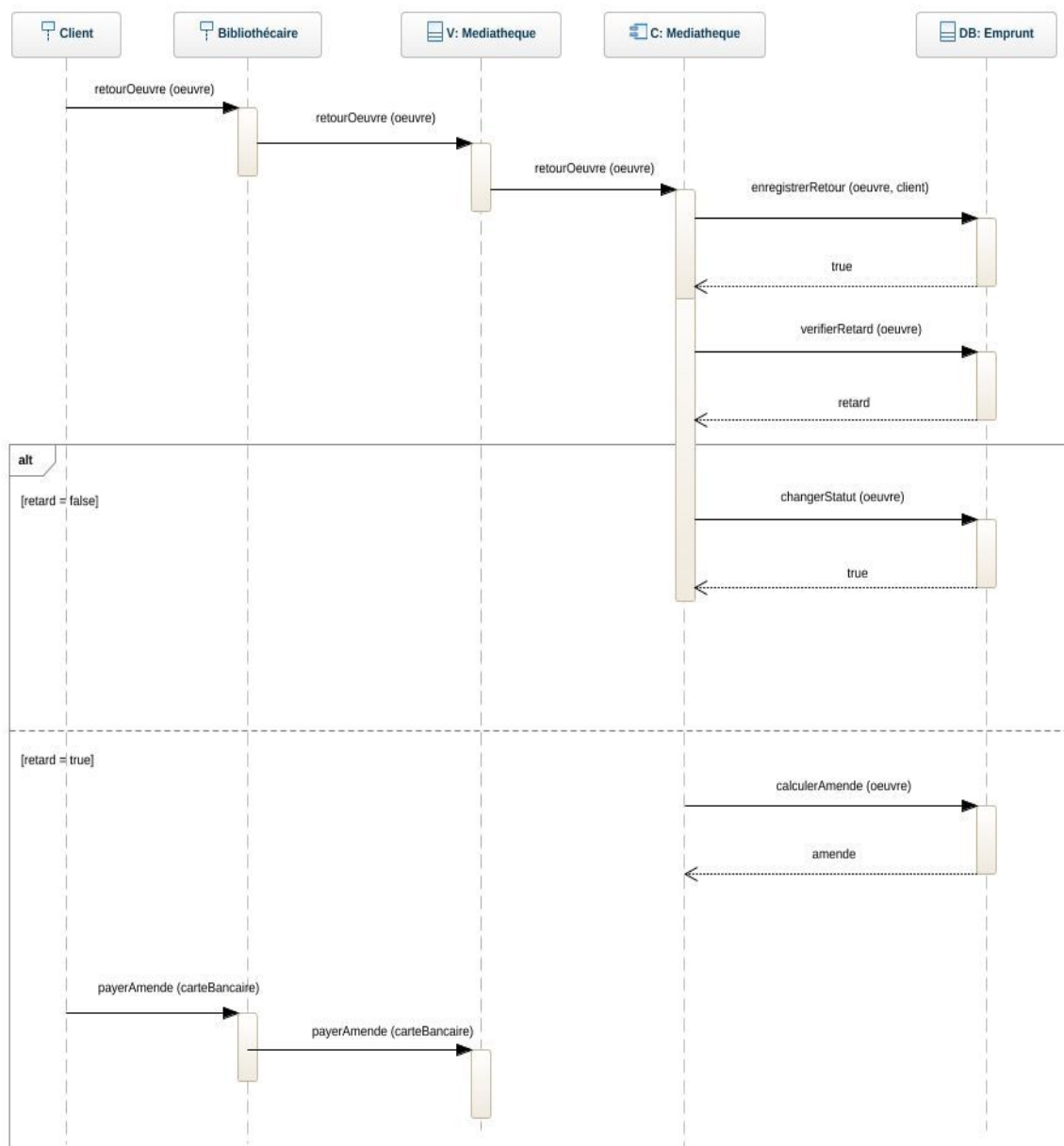
5.8- Acheter une oeuvre:

L'adhérent donne les informations de l'oeuvre qu'il veut acheter au système, la médiathèque vérifie ensuite la disponibilité de l'oeuvre. Si celle ci est avérée, le client saisit ses coordonnées bancaires et procède au paiement.



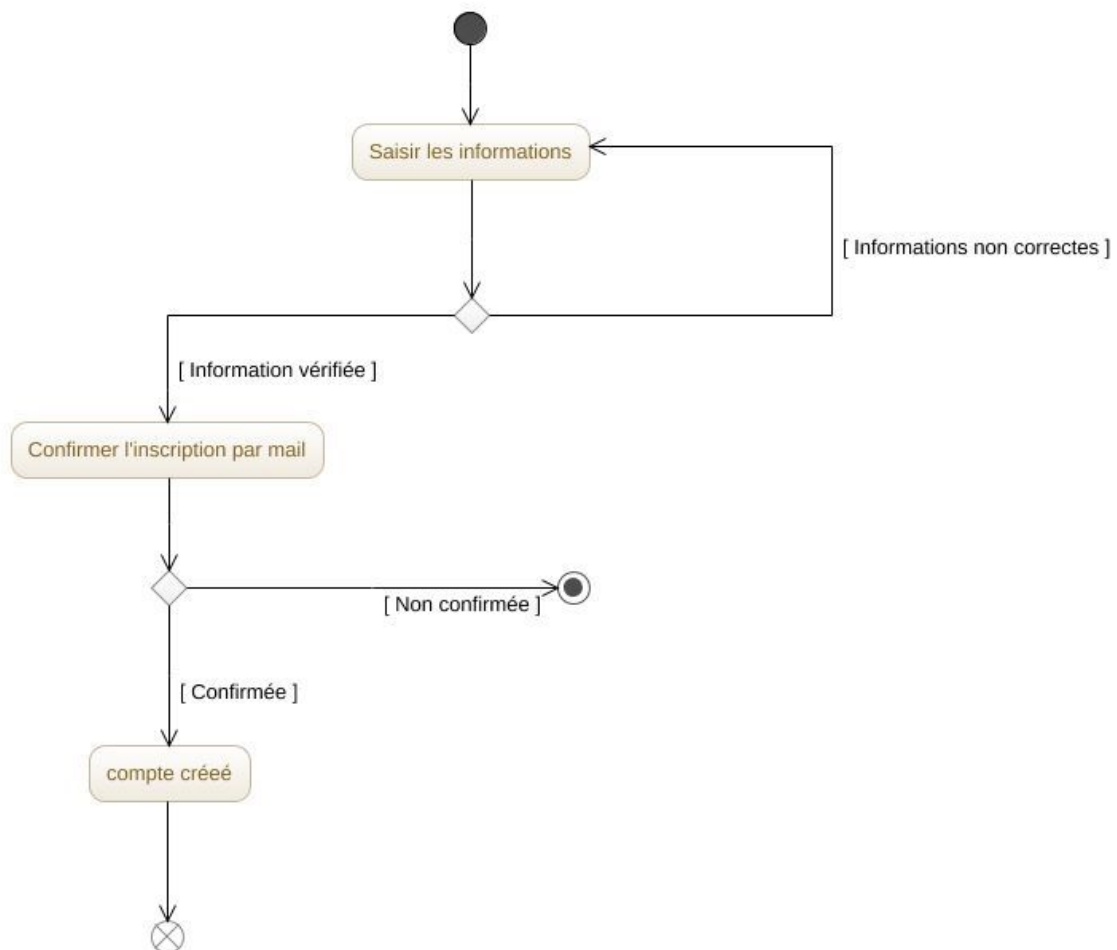
5.9- Gérer les retours :

Lorsque le client retourne une oeuvre auprès du bibliothécaire, celui-ci va (par l'intermédiaire du système) vérifier si l'oeuvre est rendue dans les temps, changer son statut dans ce cas là (elle devient disponible) mais dans le cas contraire infliger une pénalité au client.

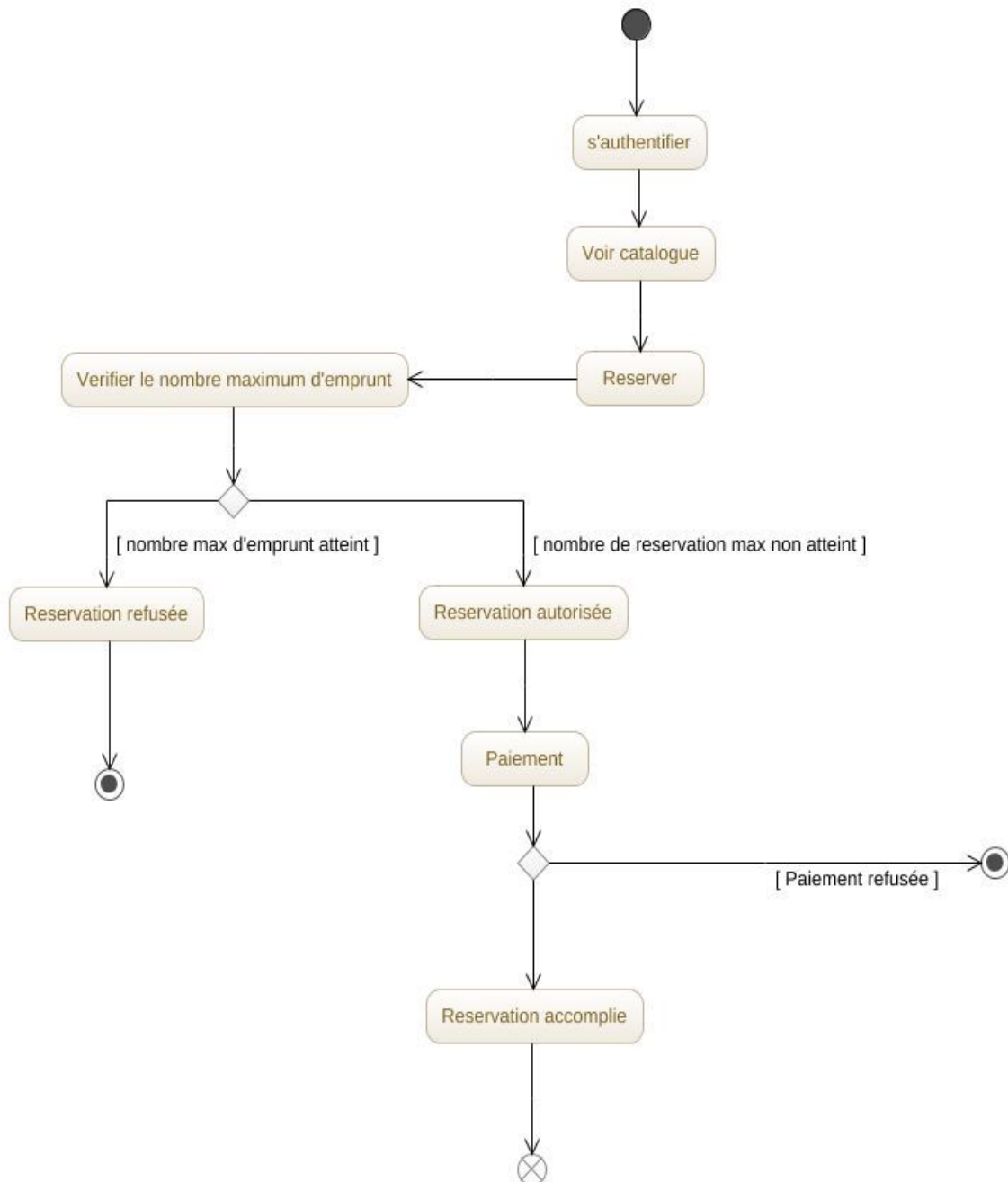


6- Diagramme d'activités:

6.1- Créer un compte:



6.2- Emprunter une oeuvre :



6.3- Etat de l'oeuvre:

