

Rapport de conception de la Médiathèque

Karine Lemarchand

Conception Architectural	3
Architecture Logicielle	3
Choix Architecture logicielle	4
Frameworks et base de données	6
Introduction aux Frameworks	6
Frameworks Frontend	7
Frameworks Backend	8
Base de données	8
Scénarios et mise à jour des diagrammes	9
Diagramme de classe	9
Diagrammes de séquences (conception)	11
Scénario cas n°1	11
Scénario Cas n°2	13
Scénario Cas n°3	15
Scénario Cas n°4	17
Scénario Cas n°5	19
Diagramme d'Activité :	21
Diagramme de Déploiement :	22

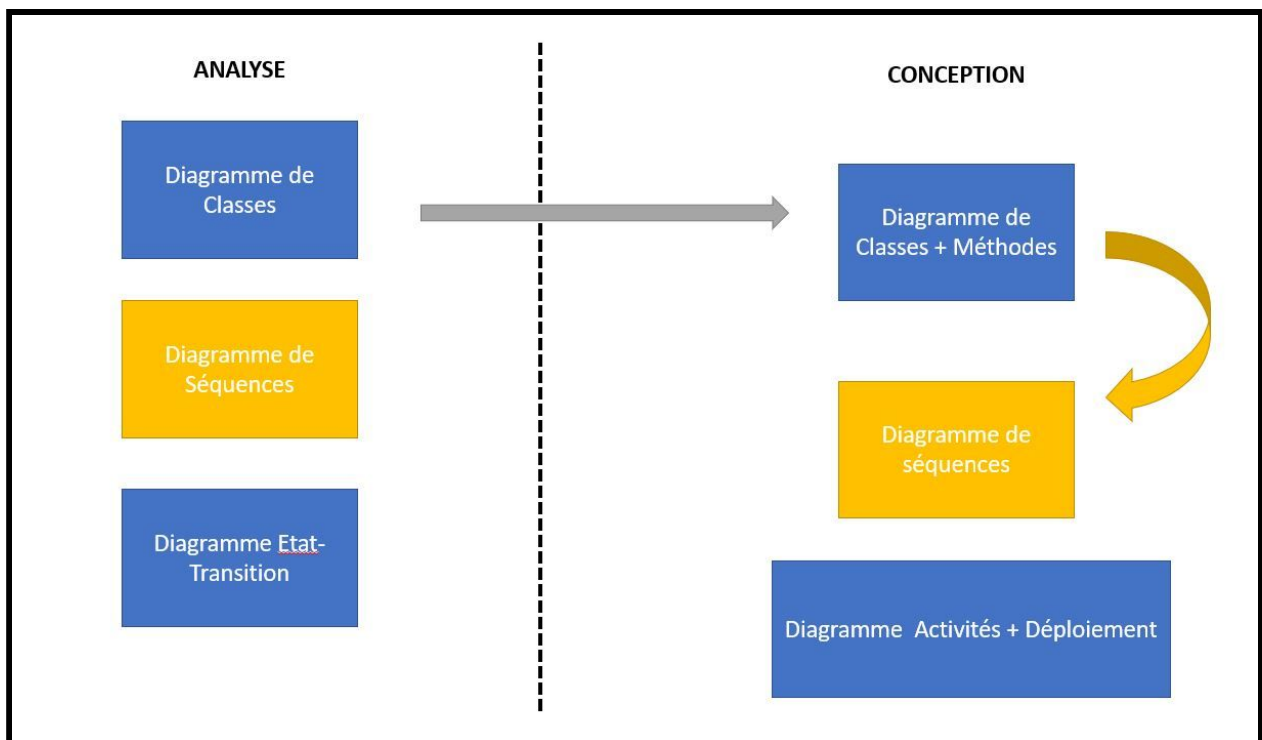
Conception Architectural

Architecture Logicielle

La phase de conception logicielle est l'équivalent, en informatique, à la phase de conception en ingénierie traditionnelle; cette phase consiste à réaliser entièrement le produit sous une forme abstraite avant la production effective. Les deux objectifs principaux de toute architecture logicielle sont la réduction de coûts et l'augmentation de la qualité du logiciel.

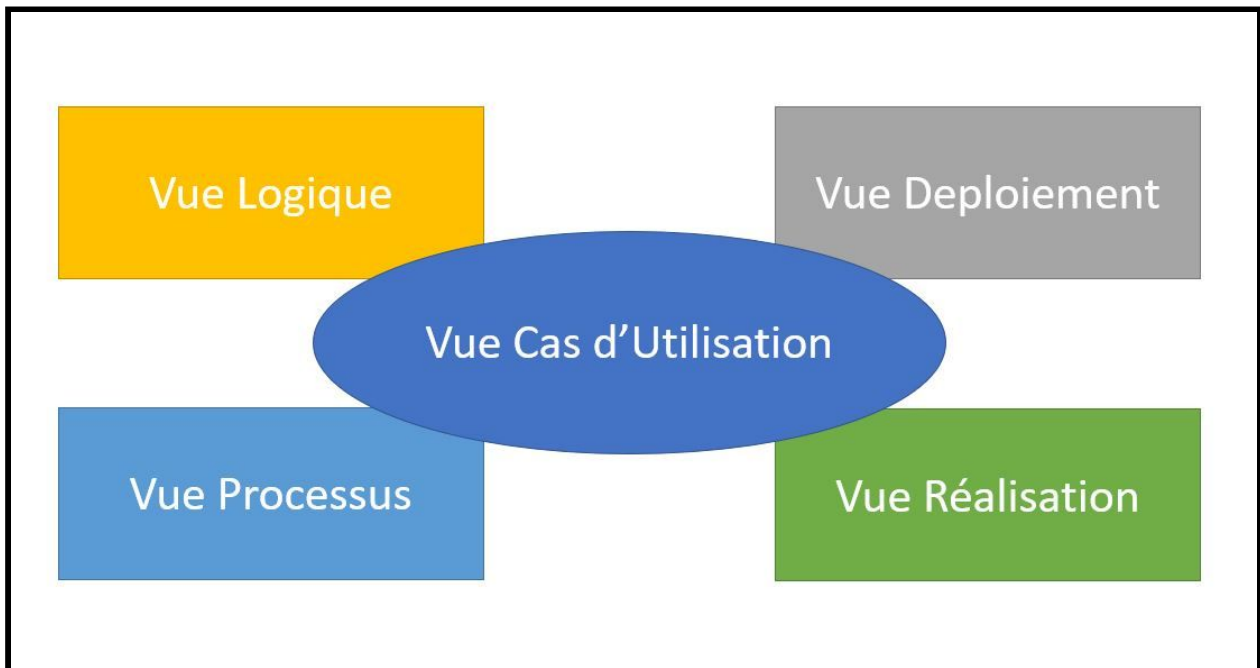
L'architecture logicielle décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatique, leur interrelations et leurs interactions.

Contrairement aux spécifications produites par l'analyse fonctionnelle, le modèle d'architecture, produit lors de la phase de conception, ne décrit pas ce que doit réaliser un système informatique mais plutôt comment il doit être conçu de manière à répondre aux spécifications.



Choix Architecture logicielle

Durant la phase de conception, il est nécessaire d'avoir une organisation de description de notre système. De ce fait, nous avons recours au modèle 4+1 vues dit de Kruchten. Le découpage du système en plusieurs vues complémentaires nous permet de présenter notre système sous différents angles et donc de traiter séparément les différents groupes d'intervenants.



La figure ci dessus représente le modèle “4+1” vues. Ce dernier est composé de 5 vues :

- La vue “logique” qui décrit tout ce qui se rapporte aux aspects statique et dynamique de notre système en termes de classes, d’objets, de connexion et de communication.
- La vue “processus” va permettre de capter les aspects de concurrence et de synchronisation et permet une décomposition en flux d'exécution. Elle se rapporte aux objets actifs et aux interactions.
- La vue “ déploiement ” représente l’organisation statique des modules dans l’environnement de développement.

- La vue “ réalisation ” : décrit les différentes ressources matérielles et l’implantation logicielle tenant compte de ces ressources. Donc, elle se rapporte aux nœuds physiques d’exécution et au placement des objets sur les nœuds.
- La vue “ cas d’utilisation ” : se concentre sur la cohérence en présentant des scénarios d’utilisation qui mettent en œuvre les éléments des quatre premières vues. Les scénarios sont une abstraction des exigences fonctionnelles de l’application.

L'architecture logicielle peut se catégoriser en différents styles. Un système informatique peut utiliser plusieurs styles selon le niveau de granularité ou l'aspect du système décrit. Nous ferons remarquer que, comme en architecture traditionnelle, c'est souvent par le mélange d'anciens styles que les nouveaux apparaissent.

Dans notre cas, nous allons suivre un style que l’on nomme : Architecture objets orientées. Cette architecture permet aux composants du système (objets) d'intégrer des données et les opérations de traitement de ces données. La communication et la coordination entre les objets sont réalisées par un mécanisme de passage de messages. L'architecture orientée objets est souvent décrite par les trois piliers : encapsulation, héritage et polymorphisme

Frameworks et base de données

Introduction aux Frameworks

Afin de faciliter notre projet, il nous a semblé impératif d'utiliser plusieurs frameworks. Lors de nos recherches, nous avons, dans un premier temps, fait une liste détaillée des frameworks connus et conseillés pour pouvoir choisir, dans un second temps, les plus adaptés à nos besoins.

En effet, nous avons besoin d'une framework Backend afin d'améliorer le contenu du site, et d'une framework Frontend afin d'améliorer le design du site. Avant d'expliquer nos choix de Framework, il est capital d'expliquer l'utilité et le fonctionnement d'une Framework.

Une framework est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.

Les frameworks sont donc conçus et utilisés pour modeler l'architecture des logiciels applicatifs, des applications web et des composants logiciels.

Celles-ci sont divisées en deux catégories : le Frontend et le Backend.

Le terme Frontend désigne les éléments visuels d'un site, ceux avec lesquels on peut interagir depuis un navigateur, c'est en général une combinaison de CSS, HTML et JavaScript. Ces langages de programmation sont interprétés par le navigateur pour afficher un résultat visuel, tels que la police, l'affichage de menus déroulants, de boutons, de transitions, de curseurs ou de formulaires. Tout cela représente le design du site.

Le terme Backend désigne la partie non visible par les internautes, et est souvent composée de trois éléments : le serveur, l'application et la base de donnée. Ces trois éléments sont utilisés et modifiés en fonction de l'utilisation de l'internaute et de ses actions, par exemple, réserver un média modifie la base de donnée qui est stockée sur un serveur. L'affichage et l'utilisation de ces données est géré par le Backend. Les langages de programmation les plus utilisés pour le Backend sont le PHP, le Ruby ou le Python.

Frameworks Frontend

Les frameworks initialement choisis pour le Frontend étaient Angular, React, Nuxt et Bootstrap. Chaque Framework possède ses points forts et ses avantages, mais aussi ses points faibles et ses inconvénients.

Le premier de cette liste, Angular JS, ne nous était connu que de nom. Il est compatible avec les modèles MVC et MVVM. Ses avantages principaux sont le fait qu'il permet de réduire la quantité de code nécessaire pour certaines fonctionnalités, et qu'il permette de vérifier les erreurs de lui-même. De plus, c'est une Framework très puissante. Cependant, elle n'est pas facile à prendre en main et nous demanderait trop de temps pour en maîtriser les bases, c'est pourquoi il n'a pas retenu notre attention.

Le second, React JS, est une interface que nous avons déjà utilisée auparavant. En effet, c'est une Framework très populaire, facile à utiliser permettant aussi de réduire la quantité de code nécessaire. Elle offre la possibilité de réutiliser les composants développés dans d'autres applications, en conservant leur fonctionnalité. Cependant, même s'il est facile à utiliser, il nécessite un temps de formation non négligeable dans la mise en place de notre projet, conduisant à l'écartement de cette interface.

Le troisième, Nuxt JS, n'était pas connu de notre groupe avant nos recherches. Nuxt est une interface française avec une communauté très active, permettant d'interagir facilement avec celle-ci lors de notre apprentissage. De plus, c'est une framework très puissante et pas très lourde, avec une template de départ permettant d'avoir un point de départ à notre projet. Enfin, il propose aussi une séparation automatique du code permettant de réduire la taille de chaque fichier javascript. Malgré tous ces avantages indéniables, et étant un possible très bon choix, nous n'avons pas choisi Nuxt comme framework frontend.

Le dernier de cette liste, Bootstrap, est une framework PHP que nous connaissions avant de faire des recherches, cela représente un avantage majeur. Il est aussi compatible avec le modèle MVC. Bootstrap est aussi simple d'utilisation, indépendamment de notre connaissance, est facile à intégrer, contient un support très complet, et est fortement personnalisable, ce sont tous ces avantages qui font que Bootstrap est l'interface Frontend que nous utiliserons dans la mise en place du site.

Frameworks Backend

Les interfaces étudiées pour servir de framework Backend sont Symphony et Laravel. Ces deux interfaces proposent de nombreux avantages similaires, tels que la facilité d'utilisation, la flexibilité, la réduction de la quantité de code nécessaire, la réutilisabilité des composants ou bien l'interopérabilité.

Symphony propose un système de template très développé, est très rapide et possède une philosophie propre à l'interface. Cependant, Laravel possède des avantages correspondants incontestablement à nos besoins, à savoir, une compatibilité MVC, un système d'authentification déjà développé et aussi un système de notification et mailing. Ces deux options sont prévues dans le fonctionnement de notre site et constituent donc des avantages clés lors de notre choix d'interface backend. C'est pourquoi nous utiliserons Laravel et non Symphony.

Base de données

Lors de notre formation, nous avons déjà utilisé, comme système de gestion de base de données, MySQL, et, comme application Web de gestion pour les systèmes de gestion de base de données, PHPMyAdmin.

Or, ces deux entités sont compatibles avec les Frameworks retenues précédemment (Bootstrap et Laravel). Il nous a donc semblé logique de nous

diriger dans l'utilisation de ceux-ci pour notre projet, afin de ne pas nous charger davantage avec l'apprentissage de nouveaux gestionnaires de base de données.

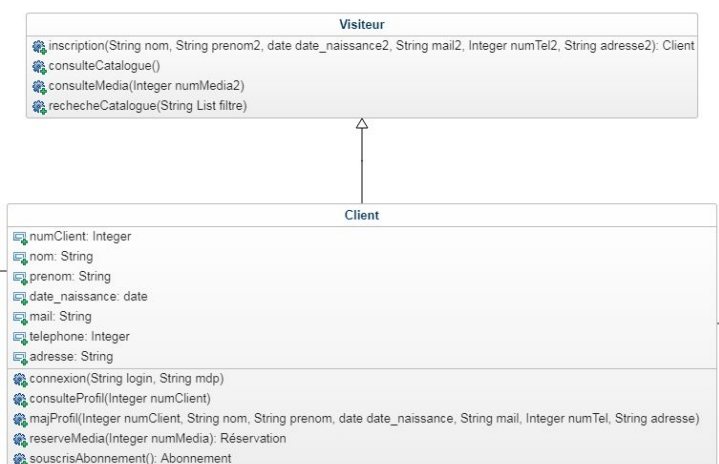
Scénarios et mise à jour des diagrammes

Diagramme de classe



Dans ce diagramme, nous avons ajouté les différentes fonctions que peuvent utiliser nos classes, de nombreuses classes ont par ailleurs été ajoutées afin de faire les liens entre les différentes classes et comme par exemple Réservation et Employé lié par la table RéservationEmployé.

Une classe Visiteur a été créée, elle possède des fonctions lui permettant de naviguer sur le site



ainsi que de s'inscrire et devenir un client.

La classe Client quant à elle possède les fonctions du visiteur mais peut se connecter, consulter son profil, le mettre à jour, prendre un abonnement et bien sûr réserver un média.



La classe Employé peut mettre à jour les informations pratiques des médias de la base de données et noter les retraits et rendu de commande lorsque les clients viennent à la médiathèque chercher ou rendre leur médias.

Les classes Responsable Adhérent et Responsable des oeuvres, en plus de posséder les fonctions de la classe Employé, récupèrent des fonctions qui permettent respectivement d'ajouter et supprimer des comptes client et abonnement ainsi que l'ajout et retrait de médias dans la base de données.



Diagrammes de séquences (conception)

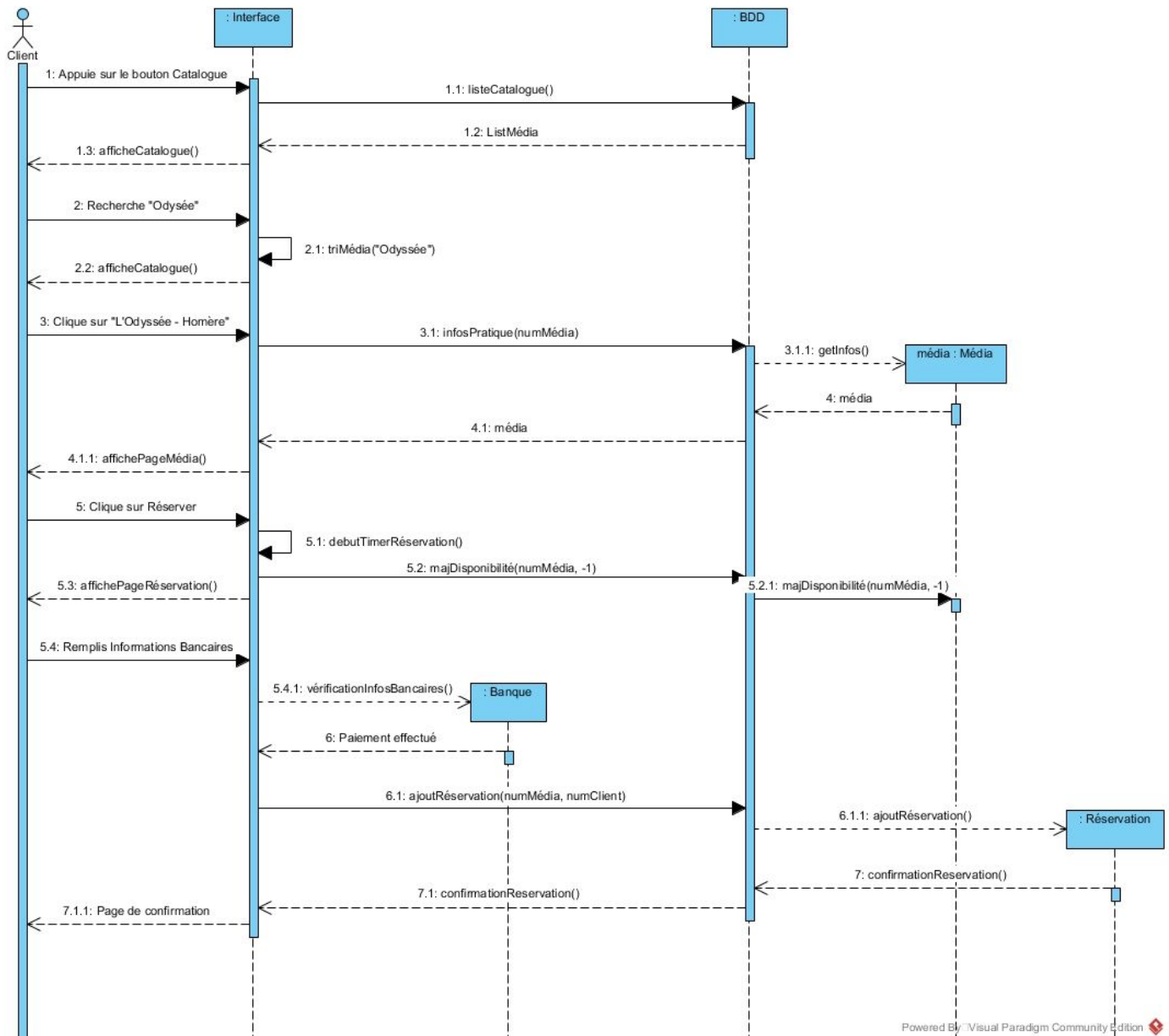
Scénario cas n°1

Cas n°1	
Nom	Réservation d'un média
Acteur(s)	Utilisateur n°1 (Olivier)
Description	Oliver cherche un livre qu'il veut réserver, seul un exemplaire est disponible, il le réserve.
Préconditions	Olivier est connecté avec son compte client
Démarrage	Olivier vient sur la site afin de consulter le catalogue
DESCRIPTION	
Scénario nominal	<ol style="list-style-type: none"> 1) Le système affiche la page d'accueil du site 2) Olivier se rend sur la page du catalogue 3) Le site envoie une requête à la base de données afin d'avoir une liste des médias qu'il conserve dans un tableau 4) Olivier écrit "L'épouvanteur" dans la barre de recherche 5) Le site lui affiche la liste des médias correspondant qui possèdent "L'épouvanteur" dans leur titre 6) Le site tri le tableau avec les critères d'Olivier et affiche le résultat 7) Olivier clique sur l'un des résultats 8) Le site envoie une requête à la base de donnée afin de récupérer les infos pratique de ce résultat 9) Le site lui affiche la page d'infos pratique du livre 10) Olivier appuie "Réserver" afin de réserver le dernier exemplaire 11) La base de données enlève préalablement 1 exemplaire, le site lance un timer de 15 minutes 12) Le site lui affiche la page de réservation 13) Karine souhaite réserver le même livre qu'Olivier, elle se rend sur la page du livre 14) Le site lui affiche la page mais il n'est plus possible de le réserver car il n'y a plus d'exemplaire disponible 15) Olivier entre ses informations de paiements et valide le

paiement

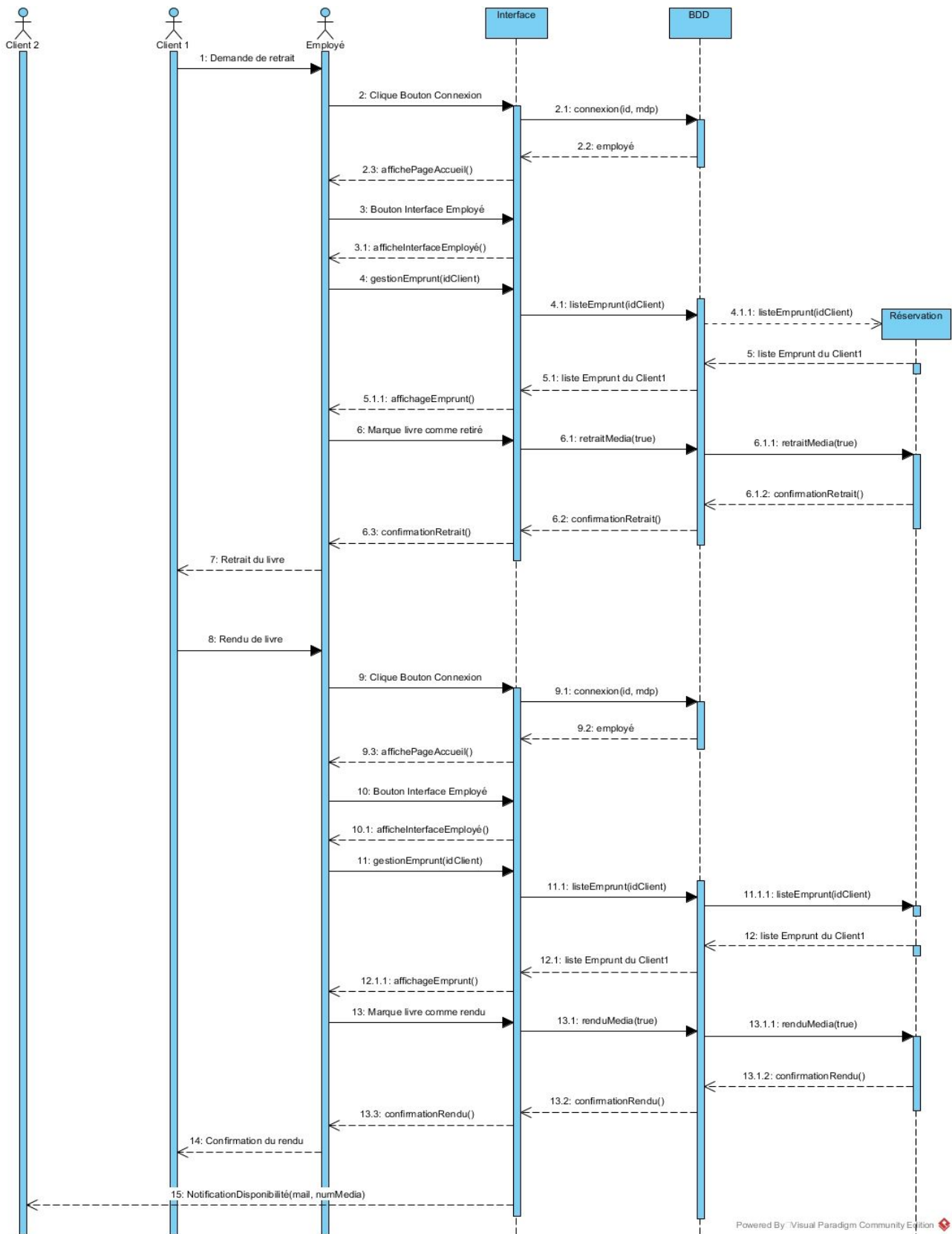
16) Le site lui confirme sa réservation et lui envoie un mail de confirmation avec les détails concernant les dates de retrait et de rendu maximum de son livre

17) Une réservation est créée dans la base de données



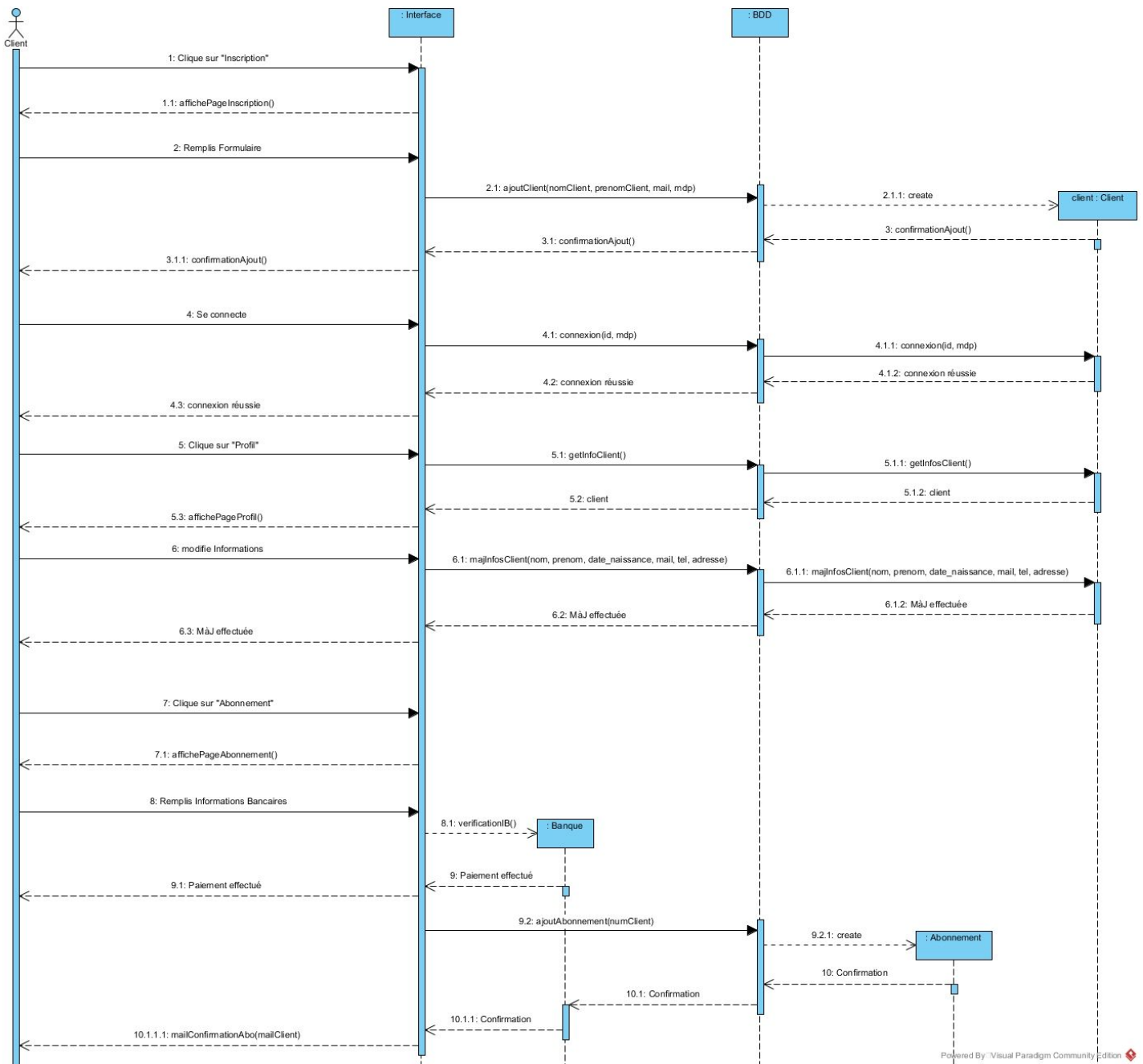
Scénario Cas n°2

Cas n°2	
Nom	Retrait et rendu d'un livre
Acteur(s)	Utilisateur n°1 (Olivier), Utilisateur n°2 (Karine), Employé (Arthur)
Description	Olivier vient retirer son livre puis le rend 3 jours après, Karine reçoit un mail l'informant de la disponibilité du livre rendu par Olivier
Préconditions	Olivier a réservé un livre, Karine a entré son email afin de s'informer de la disponibilité dudit livre
Démarrage	Olivier va à la médiathèque récupérer son livre
DESCRIPTION	
Scénario nominal	<ol style="list-style-type: none"> 1) Olivier vient chercher son livre à la médiathèque 2) Arthur va sur le site de la médiathèque 3) Il se connecte 4) La base de données vérifie que ses identifiants sont correcte 5) Le site l'envoie vers sa page d'accueil 6) Il clique sur la page destiné aux employés 7) Le site lui renvoie l'interface dédié aux employés 8) Arthur va sur la gestion des emprunts 9) Le système lui affiche la page associé aux emprunts 10) Arthur entre le nom d'Olivier 11) La base de données lui renvoie tous les emprunts actuels d'Olivier 12) Il note le livre comme emprunté 13) Le statut du livre est mis à jour dans la BDD 14) Le site lui confirme que son action a été pris en compte 15) Olivier revient 3 jours plus tard et rend son livre 16) Arthur note le livre comme rendu 17) Le statut du livre est mis à jour dans la BDD 18) Le système envoie un mail à Karine



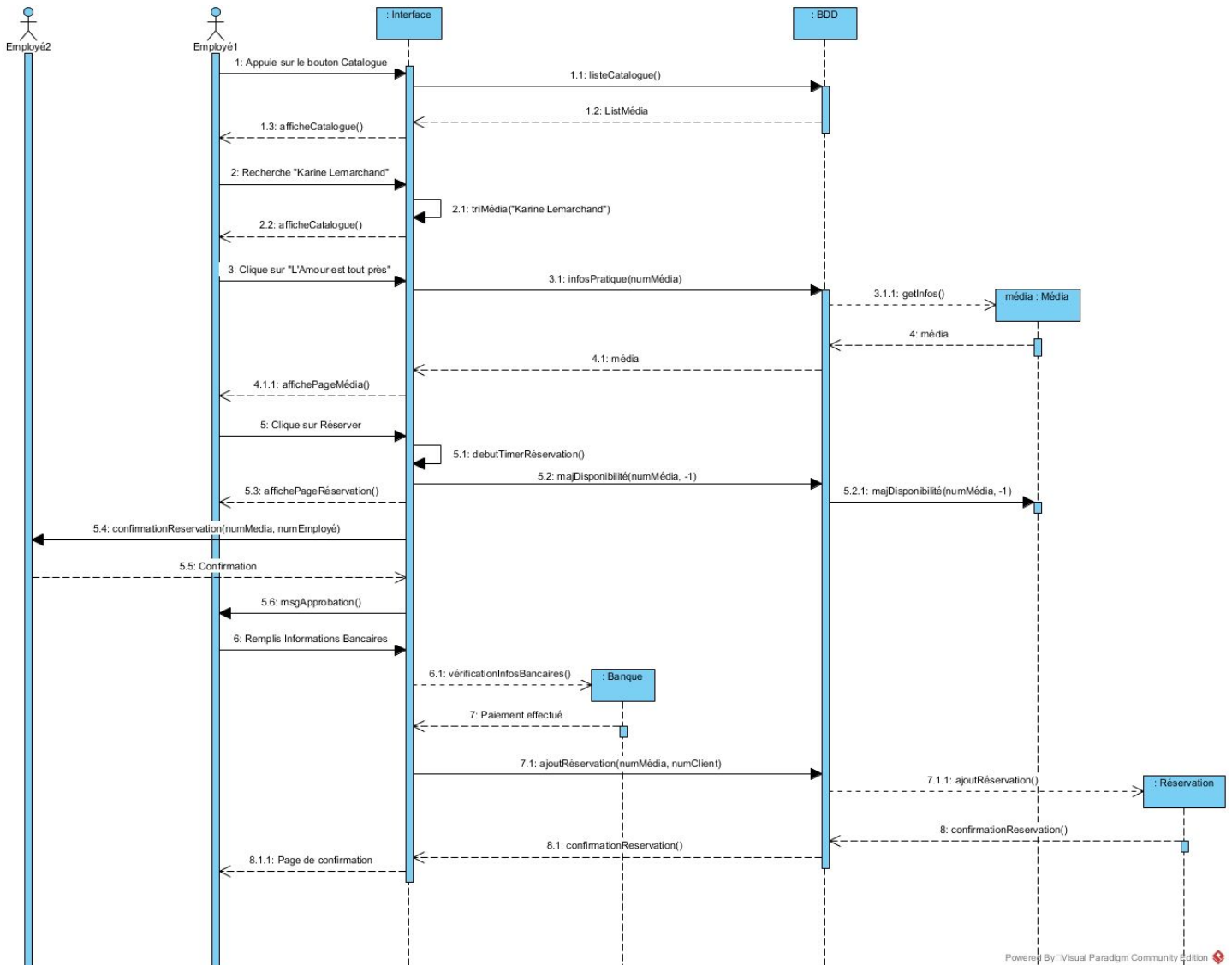
Scénario Cas n°3

Cas n°3	
Nom	Création d'un compte
Acteur(s)	Utilisateur n°1 (Marc)
Description	Marc veut accéder à la partie client du site, il n'est pas encore client donc il se crée un nouveau compte. Ensuite, il modifie ses informations personnelles afin de compléter son profil, et enfin souscrit un abonnement à la médiathèque.
Préconditions	Marc ne possède pas de compte.
Démarrage	Marc va sur l'interface d'inscription
DESCRIPTION	
Scénario nominal	<ol style="list-style-type: none"> 1) Marc va sur l'interface d'inscription 2) Il remplit correctement les champs proposés 3) Il confirme la création de son compte 4) La base de donnée ajoute un nouveau compte Client 5) Le site lui confirme que son compte a bien été créé 6) Il se connecte 7) La base de données vérifie ses identifiants puis accepte la connexion 8) Marc accède à son profil 9) Le site demande à la base de données les informations de la table Client afin de les afficher sur sa page 10) Il complète les informations manquantes 11) Il valide ses changements 12) Les changements sont mis à jour dans la table Client de la base de données 13) Marc se rend à l'interface d'abonnement 14) Il paie un abonnement d'un mois 15) Un abonnement est ajouté dans la base de données lié à Marc 16) Le système envoie un mail à Marc contenant les informations relatives à son abonnement



Scénario Cas n°4

Cas n°4	
Nom	Gestionnaire réserve un média
Acteur(s)	Deux Gestionnaires différents
Description	Un Gestionnaire veut réserver un média et nécessite l'accord d'un second gestionnaire.
Préconditions	Le média en question est libre est peut être réservé par n'importe qui
Démarrage	Un Gestionnaire veut réserver un média pour son usage personnel
DESCRIPTION	
Scénario nominal	<ol style="list-style-type: none"> 1) R1, un gestionnaire, recherche un média sur la médiathèque 2) La base de données envoie une liste du catalogue que le site trie en fonction de la recherche du gestionnaire 3) R1 fait une demande de réservation sur l'interface 4) L'interface envoie une demande de confirmation à un deuxième gestionnaire R2 5) R2 accepte la demande de réservation 6) L'interface demande à R1 un moyen de paiement 7) R1 effectue le paiement de la réservation 8) Une ligne est ajouté dans la table Réservation de la base de données 9) R1 reçoit une confirmation de réservation



Scénario Cas n°5

Cas n°5	
Nom	Ajout, mise à jour et suppression d'un média
Acteur(s)	Gestionnaire des oeuvres
Description	Un gestionnaire souhaite ajouter un CD venant d'être reçue à la médiathèque à la base de données, puis il met à jour les informations d'un autre média et en supprime un autre.
Préconditions	L'employé est un gestionnaire des Oeuvres ou un administrateur afin d'avoir accès aux pages dédié à la modification des stocks.
Démarrage	L'employé se connecte sur le site et va sur la page de Gestion des stocks
DESCRIPTION	
Scénario nominal	<ol style="list-style-type: none"> 1) Le gestionnaire clique sur Ajout d'un média 2) Le système le redirige vers un formulaire d'ajout 3) Le gestionnaire le remplit et appuie sur le bouton de validation 4) Le système vérifie que les champs sont correcte et valide l'ajout à la base de données 5) La base de données ajoute une nouvel ligne dans la table Média 6) Le gestionnaire clique sur le Catalogue et recherche un livre 7) La base de données envoie une liste des médias au site 8) Le système lui renvoie le catalogue puis le résultat de sa recherche 9) Le Gestionnaire clique sur Modifier Informations Pratique 10) Le système met la page en mode édition 11) Il met à jour les informations et valide 12) La base de données met à jour les données dans la table Média 13) Le système met à jour et affiche la nouvelle page 14) Le gestionnaire décide de supprimer des exemplaires de médias qui ne sont plus disponible à la médiathèque 15) Il recherche dans le catalogue le média en question 16) Il clique sur Supprimer puis choisi d'en supprimer 3 17) Le système demande une confirmation 18) Il confirme son action 19) Le système envoie alors une notification aux autres gestionnaire afin de valider la suppression du média 20) Un autre gestionnaire valide la suppression 21) Le système supprime 3 exemplaire du média de la base de données

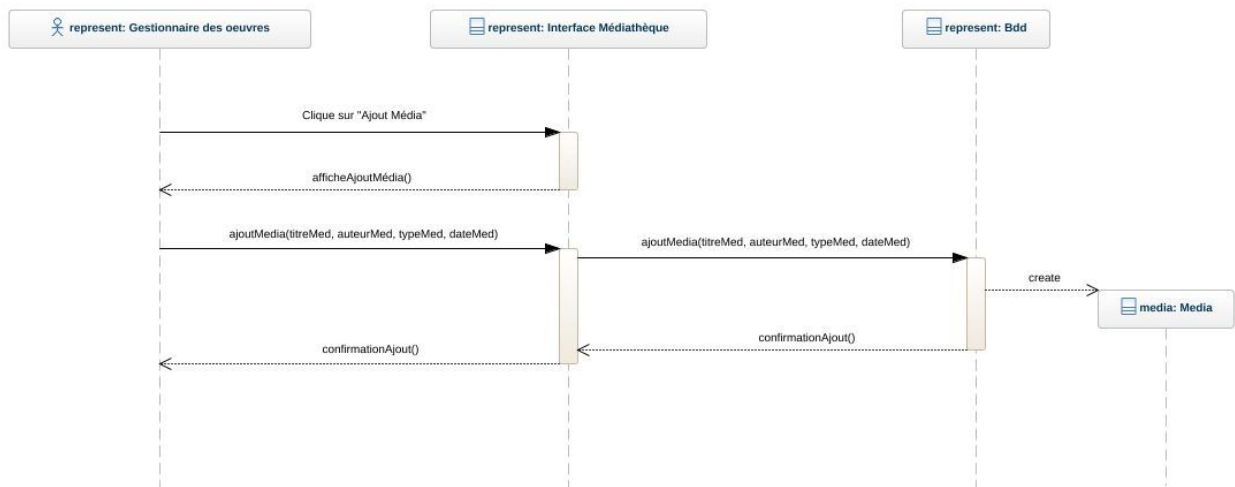
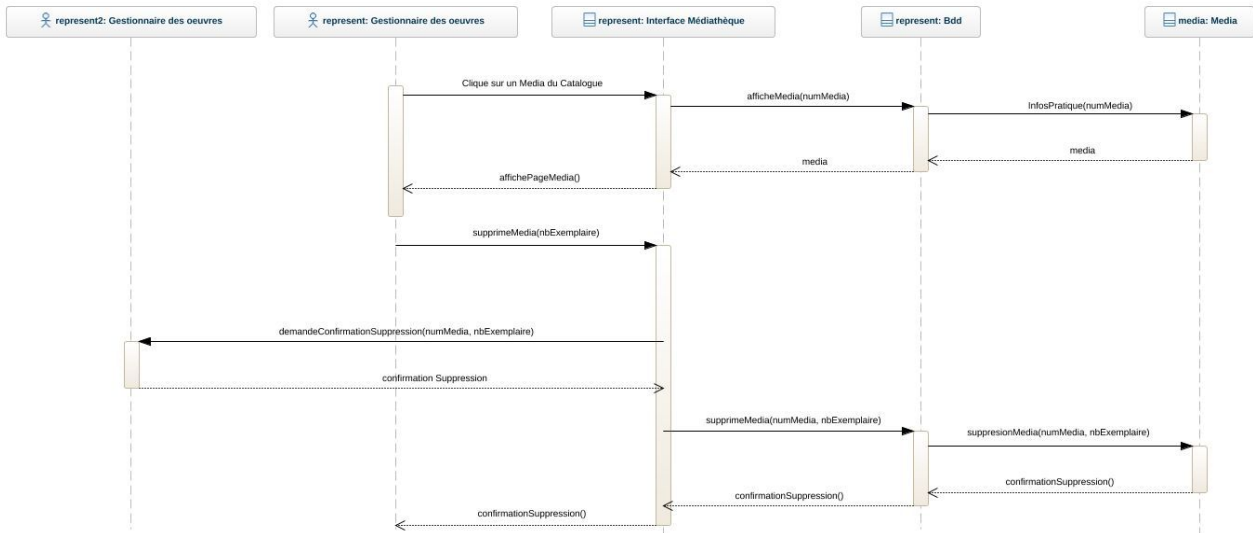
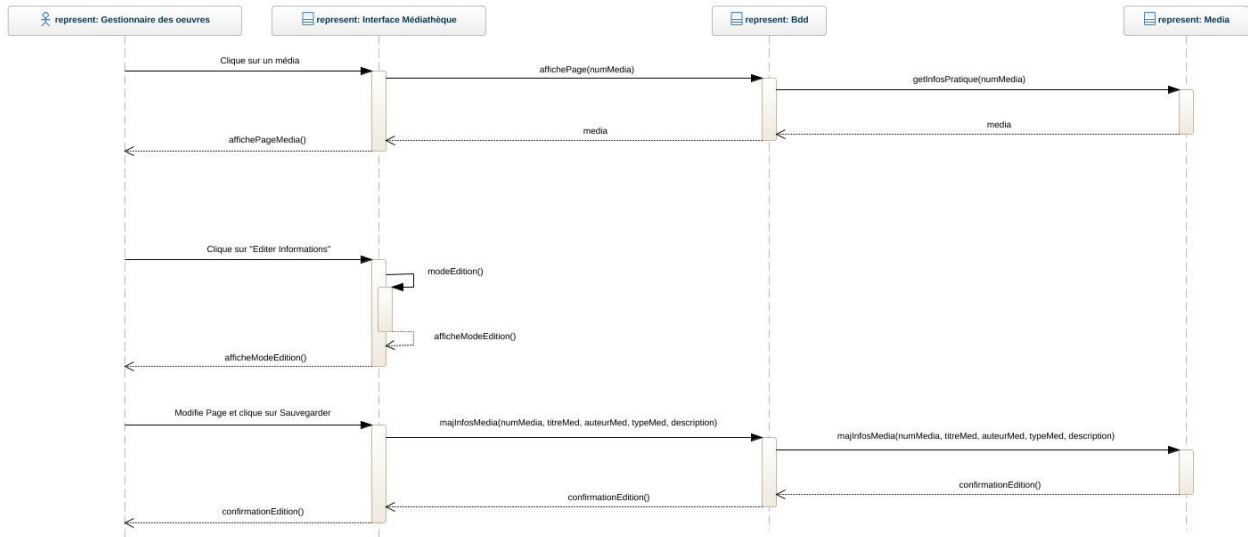


Diagramme d'Activité :

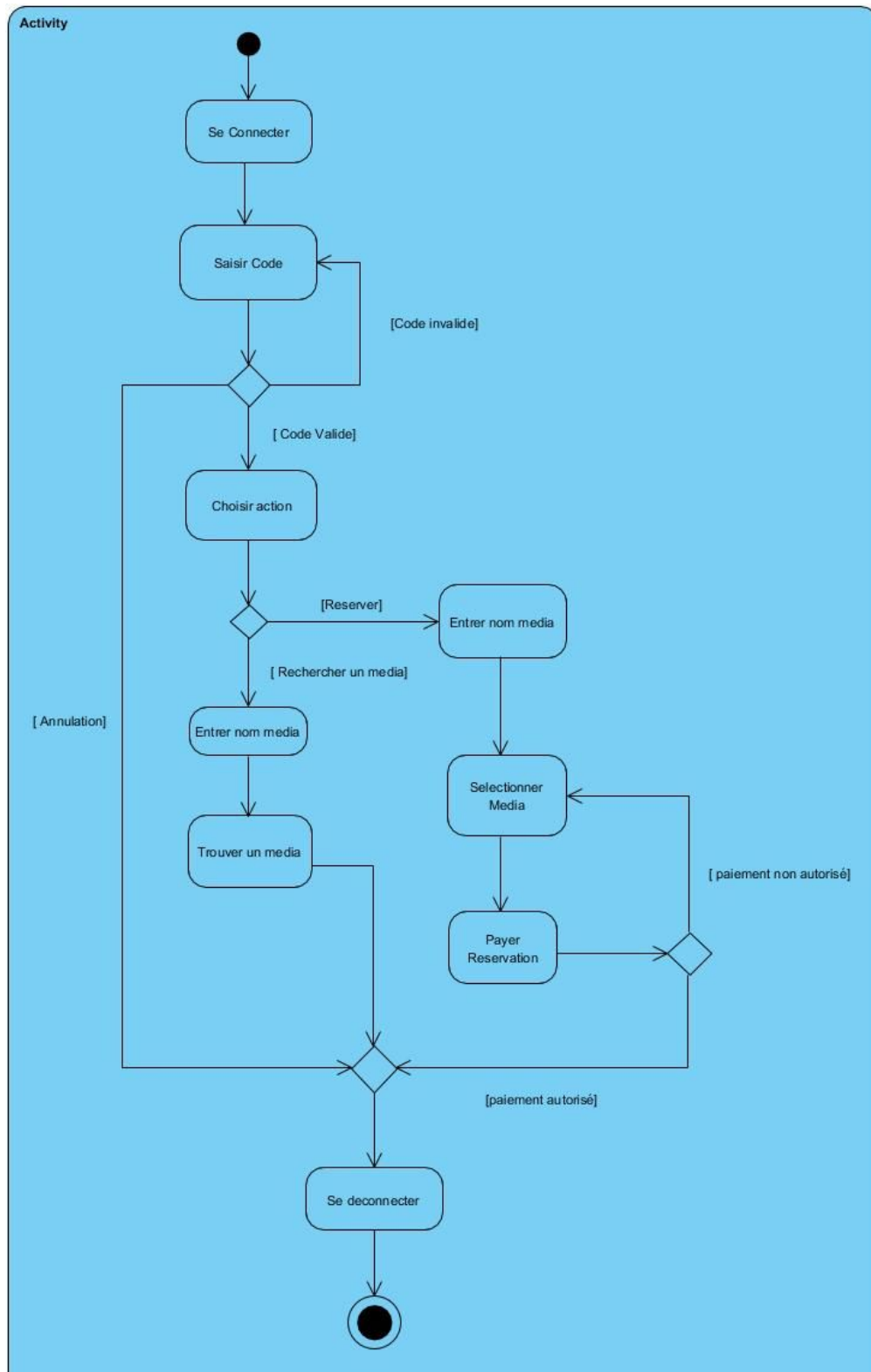


Diagramme de Déploiement :

