

Bibal

Application de gestion de bibliothèque

Guillaume Denis* et Geoffrey Gaillard†

Université de Lorraine, UFR Mathématiques et Informatique

6 novembre 2016

*guillaume.denis3@etu.univ-lorraine.fr

†geoffrey.gaillard3@etu.univ-lorraine.fr

Table des matières

1	Analyse du cahier des charges	3
1.1	Rappel	3
1.2	L'analyse de la demande	3
2	Modélisation de la solution	4
2.1	Les usages	4
2.1.1	Réserver une œuvre	5
2.1.2	Emprunter un exemplaire d'une œuvre réservée	6
2.2	Logique applicative	6
2.2.1	Diagramme de classes	7
2.2.2	Diagramme d'états	8
2.2.3	Diagrammes de séquences	8
2.3	Architecture de la solution	9
2.3.1	Diagramme de composants	9
2.3.2	Diagramme de déploiement	9
3	Implémentation	10
3.1	Particularités	10
3.2	Captures d'écran de l'application	12
3.3	Résumé	13
4	Annexes	14
4.1	Diagrammes d'activité	14
4.1.1	Créer un exemplaire	14
4.1.2	Ajouter un usager	15
4.1.3	Annuler une réservation	16
4.1.4	Emprunter un exemplaire	17
4.1.5	Modifier un exemplaire	18
4.1.6	Supprimer un exemplaire	19
4.1.7	Vérifier l'état d'un exemplaire	20
4.1.8	Rendre un exemplaire	21

1 Analyse du cahier des charges

1.1 Rappel

Bibal est une application de gestion de bibliothèque destinée à assurer le suivi des ouvrages, des exemplaires, des auteurs, des usagers ainsi que leurs emprunts et réservations. La bibliothèque possède deux types d'ouvrage : les livres et les magazines. Les usagers n'ont accès qu'à la page d'accueil de l'application, où ils peuvent consulter la liste des ouvrages de la bibliothèque. Ils doivent passer par l'intermédiaire d'un bibliothécaire pour effectuer des réservations, des emprunts ou consulter la disponibilité des exemplaires.

1.2 L'analyse de la demande

Nous avons discerné trois cas d'usage :

- gérer les usagers, c'est-à-dire les personnes consultant la bibliothèque dans le but d'utiliser ses services,
- gérer les ouvrages et les exemplaires, c'est-à-dire toutes les ressources bibliographiques,
- gérer les emprunts et les réservations, qui concerne les services proposés par la bibliothèque aux usagers.

Une œuvre représente soit un livre soit un magazine et est immatérielle. Elle désigne bien l'œuvre et non un exemplaire d'un livre ou d'un magazine. Un exemplaire est un livre physique ou un magazine physique. Un exemplaire peut être emprunté, être réservé, être rendu, être abîmé. Un usager est un utilisateur de l'application Bibal, qu'il soit employé de la bibliothèque ou non. Une réservation concerne une œuvre et un usager. L'usager peut donc réserver une œuvre, dès qu'un exemplaire de cette œuvre est disponible, l'usager peut l'emprunter. Sa réservation devient alors un emprunt.

La gestion des usagers inclus :

- laisser la possibilité aux usagers sans droits de consulter les œuvres disponibles,
- l'ajout, la modification et la suppression d'un usager,
- l'affectation d'un usager à une réservation ou à un emprunt.

La gestion des ouvrages inclus :

- l'ajout, la modification et la suppression des œuvres et des exemplaires,
- le suivi des nouvelles œuvres des auteurs,
- le suivi des parutions des magazines.

La gestion des exemplaires et des réservations concerne :

- le suivi de l'état (en état, abîmé, etc.) des exemplaires,
- le suivi du statut des exemplaires (emprunté, réservé, disponible).

2 Modélisation de la solution

2.1 Les usages

Voici, sous forme schématique, les trois cas d'usage que nous avons discerné :

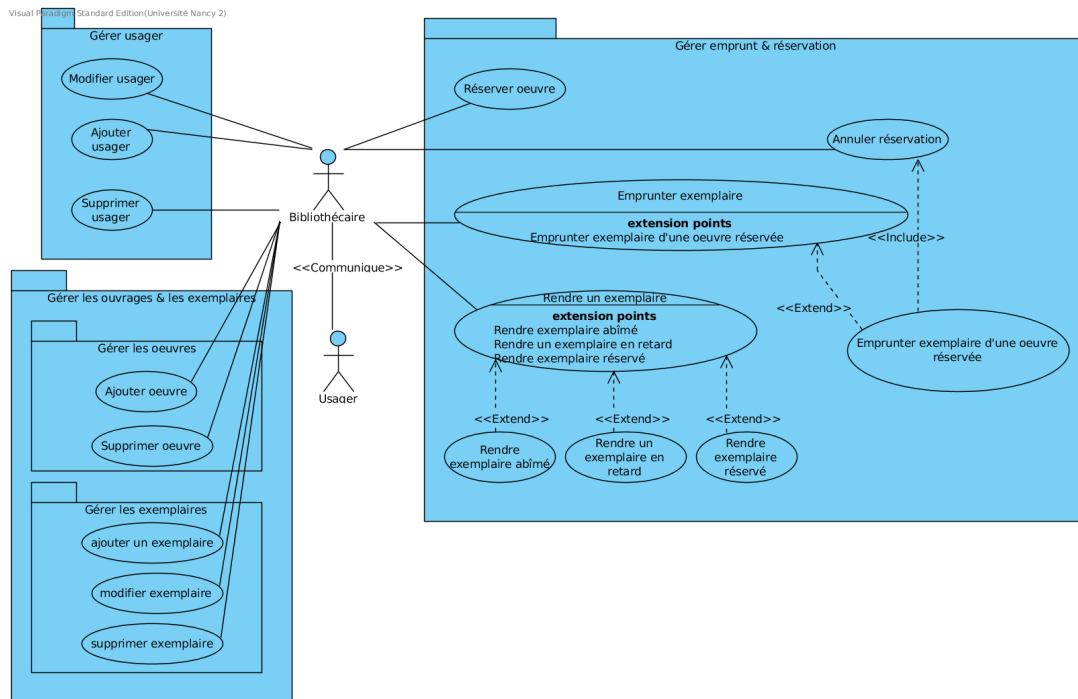


FIGURE 1 – Cas d'utilisation

On retrouve les trois pôles que qui ont été extraits suite à notre analyse du cahier des charges :

- gérer les usagers,
- gérer les ouvrages et les exemplaires,
- gérer les emprunts et les réservations.

Chaque pôle doit permettre d'effectuer les opérations basiques de lecture, création, modification et suppression (on utilisera le l'acronyme CRUD pour désigner ces opérations).

Dans le cas des emprunts et des réservations un cas d'usage s'ajoute à ces opérations basiques, rendre un exemplaire qui a été emprunté. Il est lui-même décliné en trois cas d'utilisation en fonction de différents paramètres :

- rendu d'un exemplaire abîmé,
- rendu d'un exemplaire après la date limite,
- rendu d'un exemplaire réservé par un autre utilisateur.

Lors de l'implémentation de l'application, ces cas d'utilisation devront tous être réalisables.

Pour décrire plus finement les cas d'utilisation modélisés précédemment nous avons utilisé des diagrammes d'activité. Cela nous a permis, pour chaque cas d'utilisation, de définir les étapes à réaliser pour mener à bien ce cas. Nous allons nous concentrer sur deux cas d'utilisation : réserver une oeuvre et emprunter un exemplaire d'une oeuvre réservée.

2.1.1 Réserver une œuvre

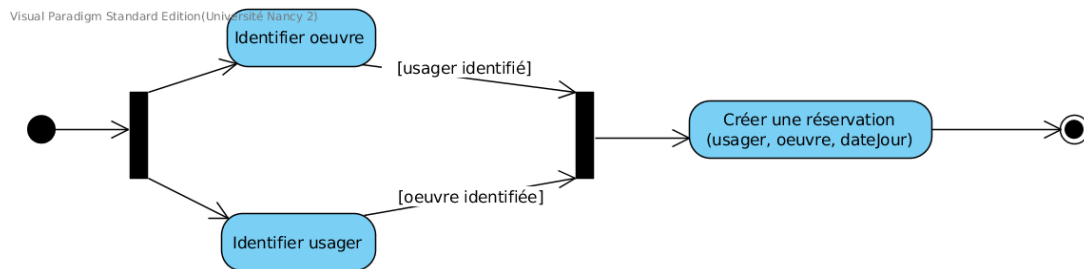


FIGURE 2 – Réserver une œuvre

Le but de ce cas d'utilisation est la création d'une réservation pour une œuvre donnée. Les étapes nécessaires et obligatoires à cette création sont l'identification de l'œuvre en question et l'identification de l'utilisateur qui désire réserver l'œuvre. Il s'agit de la partie lecture des opérations CRUD d'une œuvre et d'un utilisateur. Une fois les deux entités identifiées la bibliothécaire peut réserver une œuvre au nom de l'utilisateur datée du jour.

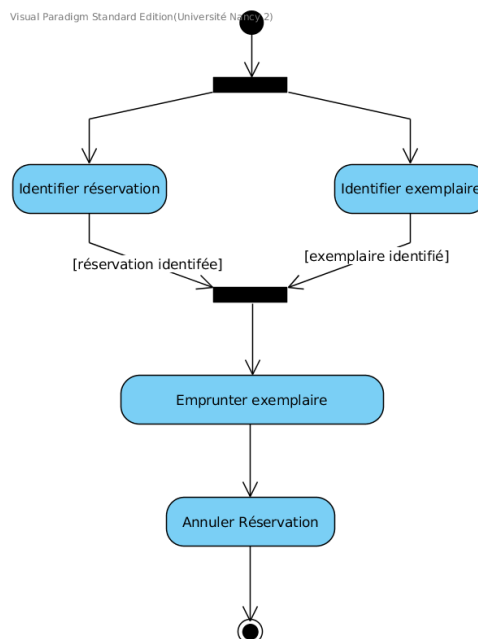
2.1.2 Emprunter un exemplaire d'une œuvre réservée

Ce cas d'utilisation annule une réservation effectuée par un usager lorsque celui-ci parvient à emprunter un exemplaire de l'œuvre réservée.

Afin de parvenir à l'annulation effective de la réservation (équivalent à une suppression dans les opérations CRUD), la bibliothécaire doit passer par plusieurs étapes :

- Trouver la réservation émise sur l'œuvre concernée par l'utilisateur concerné,
- Trouver un exemplaire de cette œuvre qui n'est pas emprunté,
- Créer un emprunt pour l'exemplaire par l'utilisateur.

Les diagrammes pour les autres cas d'utilisation sont disponibles en annexe.



2.2 Logique applicative

Une fois la logique métier bien définie grâce aux cas d'utilisation et aux diagrammes d'activité nous avons modélisé la logique applicative de la solution au besoin exprimé dans le cahier des charges.

Dans un premier temps, un diagramme des classes nous a permis de définir les objets métiers qui sont nécessaires à la réalisation de l'application. Pour les cas induisant un changement d'état d'un objet métier, une modélisation à l'aide d'un diagramme d'états a été choisie. Enfin, nous avons modélisé les enchaînements de traitements effectués par l'application à l'aide de diagrammes de séquence.

2.2.1 Diagramme de classes

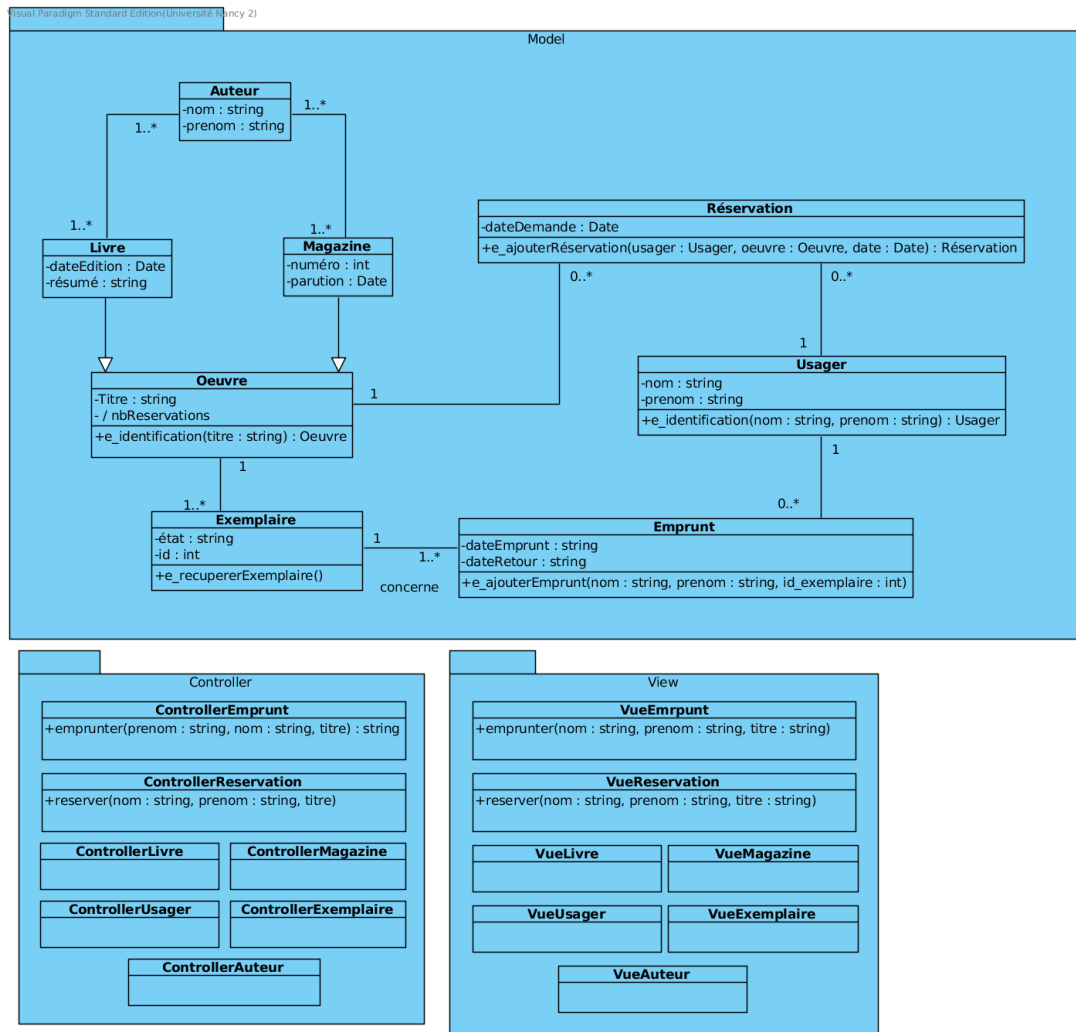


FIGURE 3 – Diagramme de classes

Nos objets métiers sont regroupés dans le paquetage **Model**. Il représente les liaisons entre chaque entité. On peut par exemple voir qu'un Exempleire est associé à 1 ou plusieurs Emprunt(s).

Le paquetage **Controller** regroupe tout ce qui concerne la manipulation de ces objets métiers. Les opérations basiques CRUD ne sont pas représentées car elles sont présentes dans chaque entité.

Le paquetage **View** regroupe tout ce qui concerne l'Interface Homme Machine de l'application. C'est avec les fonctions de ces entités que la bibliothécaire va interagir.

2.2.2 Diagramme d'états

L'entité Œuvre a demandé l'utilisation d'un diagramme d'états pour décrire comment le nombre de réservations d'une œuvre donnée va évoluer pendant l'utilisation de l'application.

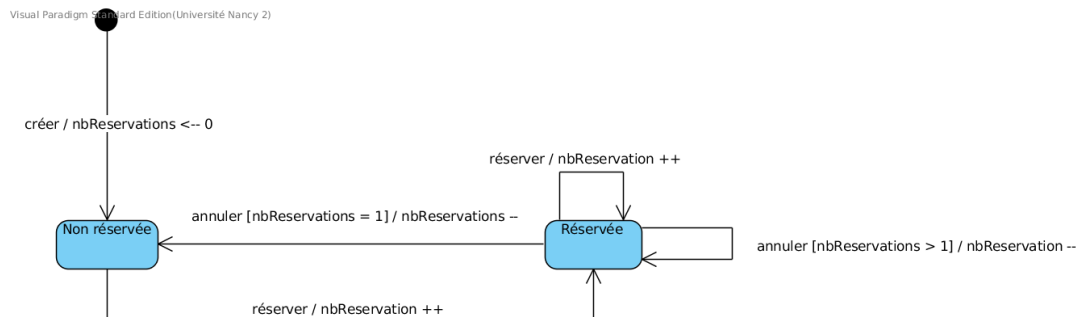


FIGURE 4 – État d'une œuvre

Une œuvre va passer par 2 états : **Non réservée** et **Réservée**. Lorsqu'une nouvelle œuvre est ajoutée par la bibliothécaire, elle n'est pas réservée par défaut (son nombre de réservations et donc de 0). Un utilisateur réservant une œuvre induit une incrémentation du nombre de réservations de 1. L'œuvre passe donc dans l'état **Réservée**. Une annulation de réservation induit une décrémentation du nombre de réservations de 1. Une fois le nombre de réservations à 0, l'œuvre repasse dans l'état **Non réservée**.

2.2.3 Diagrammes de séquences

Seuls les diagrammes de séquence des deux cas d'utilisation présentés vont être décrit ici.

1. Réserver une œuvre

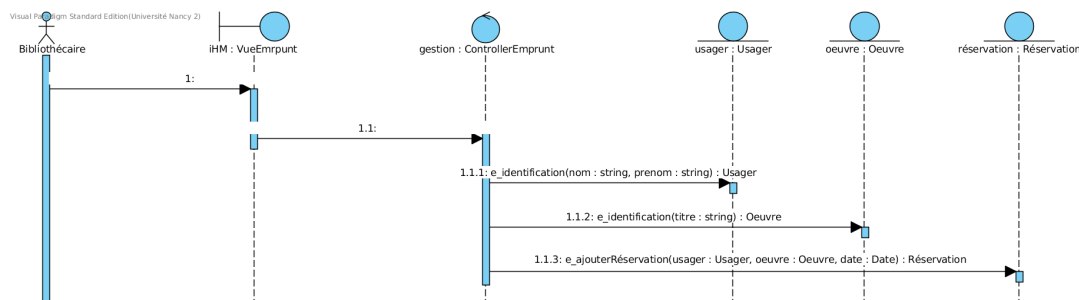


FIGURE 5 – Réserver une œuvre

Afin de créer une réservation pour une œuvre, la bibliothécaire va dans un premier temps interagir avec l'IHM (une entité de notre paquetage **View** dans le diagramme des classes) en renseignant les informations nécessaires (celles pour permettre l'identification de l'Usager et de l'Œuvre). Cette IHM interagit avec le contrôleur qui lui est associé (une entité du paquetage **Controller** du diagramme des classes). Ainsi, il permet d'identifier l'utilisateur grâce aux informations remplies

par la bibliothécaire. L'application va ensuite identifier l'œuvre concernée. Pour finir, elle va créer une réservation concernant l'œuvre identifiée pour l'utilisateur identifié, en ajoutant la date du jour).

2. Emprunter un exemplaire d'une œuvre réservée

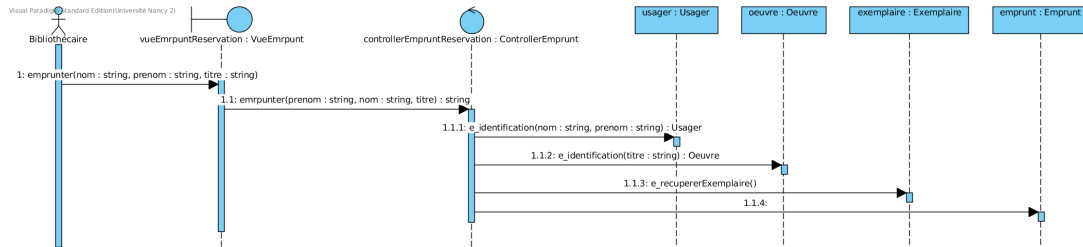


FIGURE 6 – Emprunter un exemplaire

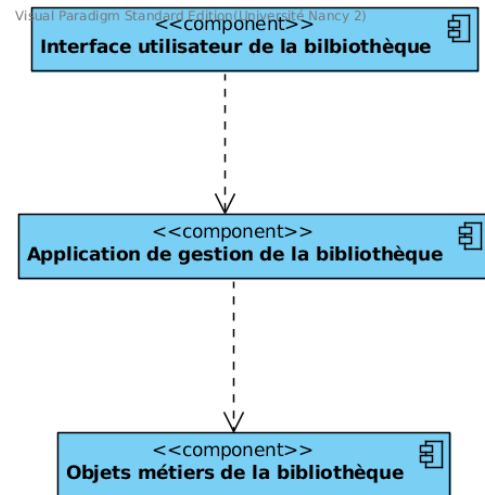
Le principe est similaire au précédent diagramme. La bibliothécaire interagit avec l'IHM pour renseigner les informations nécessaires à l'identification de l'œuvre, de l'utilisateur et de l'exemplaire. Le contrôleur associé va interagir avec les objets métiers Usager, Œuvre, Exemplaire et Réservation. Il va récupérer l'utilisateur concerné par l'emprunt, identifier l'œuvre, récupérer un exemplaire de cette œuvre. La réservation de cet utilisateur pour l'œuvre concernée est supprimée, et un nouvel emprunt est créé.

2.3 Architecture de la solution

2.3.1 Diagramme de composants

Comme on a pu le voir dans le diagramme des classes, l'application se découpe en trois couches qui interagissent entre elles. L'interface utilisateur de la bibliothèque qui va permettre à la bibliothécaire d'effectuer les actions exprimées dans le cahier des charges. L'application de gestion, contient les entités qui vont interagir avec les objets métiers. Les objets métiers de la bibliothèque qui sont la représentation physique des données.

De cette description succincte des composants de l'application découle le diagramme de déploiement, qui va permettre de définir l'architecture applicative de la solution.



2.3.2 Diagramme de déploiement

Ici dans un cadre d'application en production, les données physiques sont stockées dans un SGBD. Il communique par protocole TCP/IP à l'application de gestion de la bibliothèque contenue dans un serveur d'application sur une machine avec une JVM.

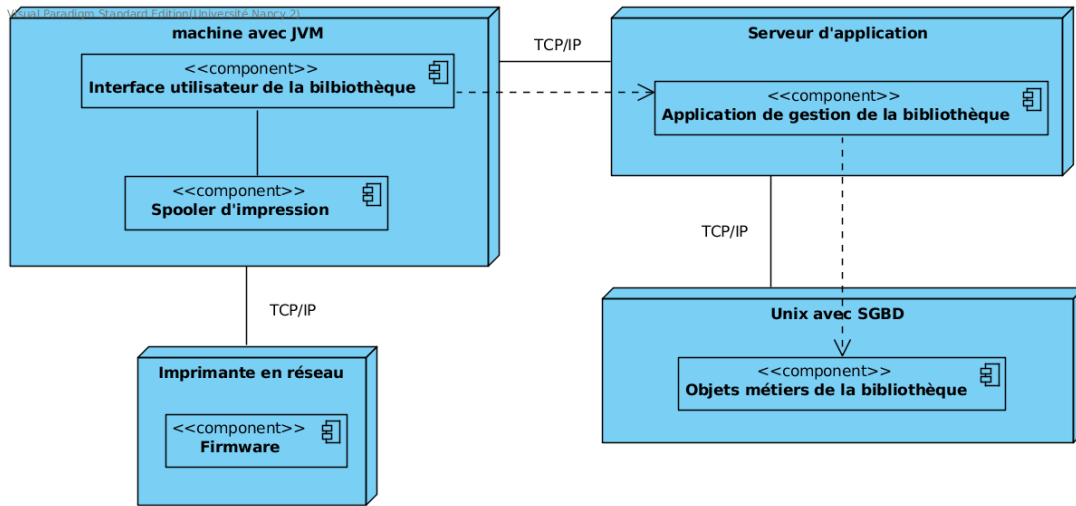


FIGURE 7 – Diagramme de déploiement

3 Implémentation

3.1 Particularités

Comme vu sur le diagramme de composants, Bibal est une application distribuée composée de plusieurs tiers :

- une application web côté serveur,
- une application web côté navigateur (faisant office d'IHM),
- une base de données.

La base de données est une base relationnelle, aussi elle ne supporte pas l'héritage. Livre et Magazine sont des spécialisations d'Œuvre. Nous avons donc construit un modèle physique des données pour modéliser les relations entre Livre, Magazine et les entités associées.

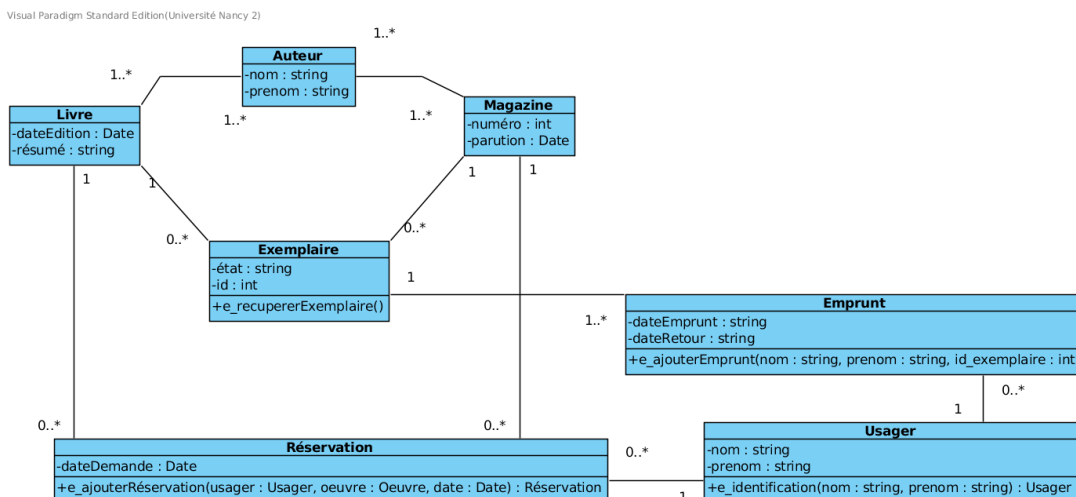


FIGURE 8 – Modèle physique des données

Nous avons également ajouté une fonctionnalité d'authentification. La bibliothécaire étant la seule personne utilisant l'application, il nous paraissait important que personne d'autre ne puisse gérer les usagers, les réservations, etc.

Notre implémentation a été réalisée avec :

- MariaDB pour la base de données,
- Hibernate pour gérer la persistance des données,
- Spring Boot pour la logique applicative,
- AngularJS pour l'interface homme-machine.

3.2 Captures d'écran de l'application

La page d'accueil de l'application permet plusieurs choses :

- enregistrer un utilisateur,
- se connecter à l'application,
- visualiser les livres et magazines présents dans la bibliothèque.



FIGURE 9 – Page d'accueil

En tant que bibliothécaire connecté, d'autres fonctionnalités de gestion sont débloquées pour les différents objets métiers. Les opérations CRUD sur tous les objets sont possibles si l'on est identifié en tant que bibliothécaire.



FIGURE 10 – Page d'accueil en étant authentifié

La vue des emprunts permet à la bibliothécaire, en plus des opérations CRUD, de rendre l'exemplaire associé à cet emprunt.

Emprunts

+ NOUVEL EMPRUNT

En retard:

ID	Date Emprunt	Date Retour	Usager	Exemplaire	
2	22 Oct. 2016	29 Oct. 2016	John DOE	Les joies de la Miage	
5	27 Oct. 2016	27 Oct. 2016	toto TOTO	test	

Dans les temps:

ID	Date Emprunt	Date Retour	Usager	Exemplaire	
1	21 Oct. 2016	19 Nov. 2016	John DOE	Stack Overflow Copy-Paste	

FIGURE 11 – Emprunts

3.3 Résumé

Toute la difficulté de cette réalisation réside dans la compréhension du besoin et des contraintes, de leurs raisons d'être et de comment réaliser la solution en en tenant compte. L'adoption d'une démarche scientifique est essentielle pour éviter les écueils classiques en matière de communication, d'analyse, d'implémentation, de gestion des ressources (humaine, temporelles, financières, etc...). Nous avons décortiqué la problématique autant que possible en étapes simples et tenté d'écarter au maximum toute ambiguïté. Le résultat est simple mais répond au besoin : une application de gestion de bibliothèque fournissant toutes les fonctionnalités demandées de façon cohérente, simple d'accès, sécurisée, simple d'utilisation et facilement évolutive.

4 Annexes

4.1 Diagrammes d'activité

4.1.1 Créer un exemplaire

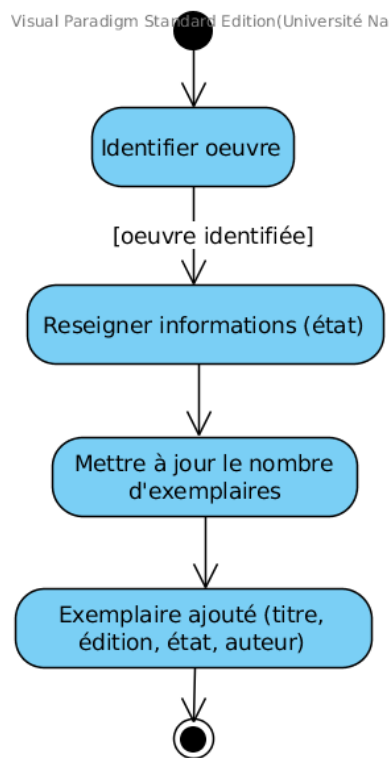


FIGURE 12 – Créer un exemplaire

4.1.2 Ajouter un usager

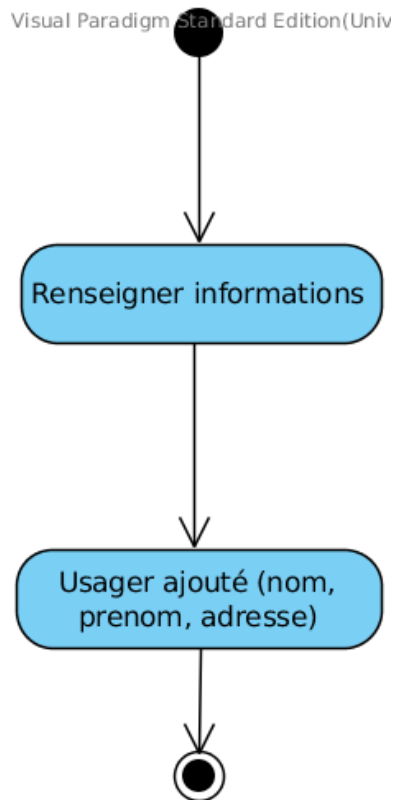


FIGURE 13 – Ajouter un usager

4.1.3 Annuler une réservation

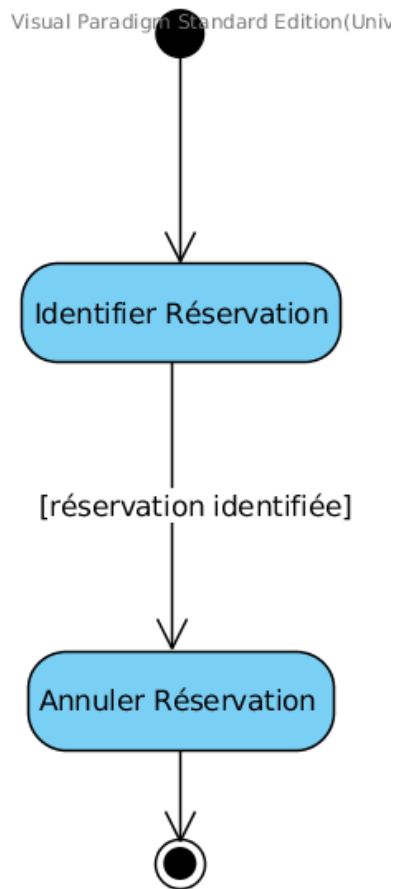


FIGURE 14 – Annuler une réservation

4.1.4 Emprunter un exemplaire

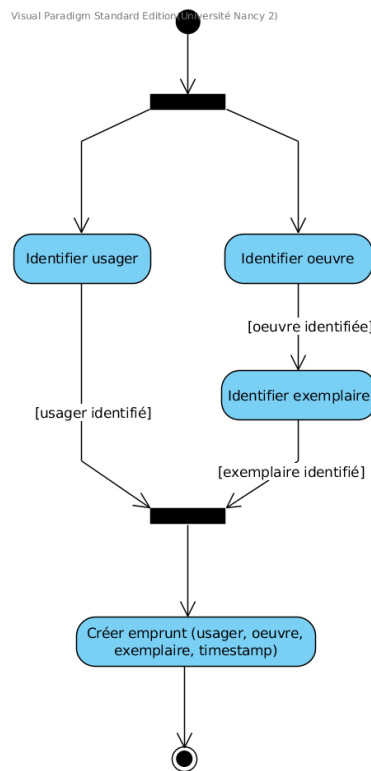


FIGURE 15 – Emprunter un exemplaire

4.1.5 Modifier un exemplaire

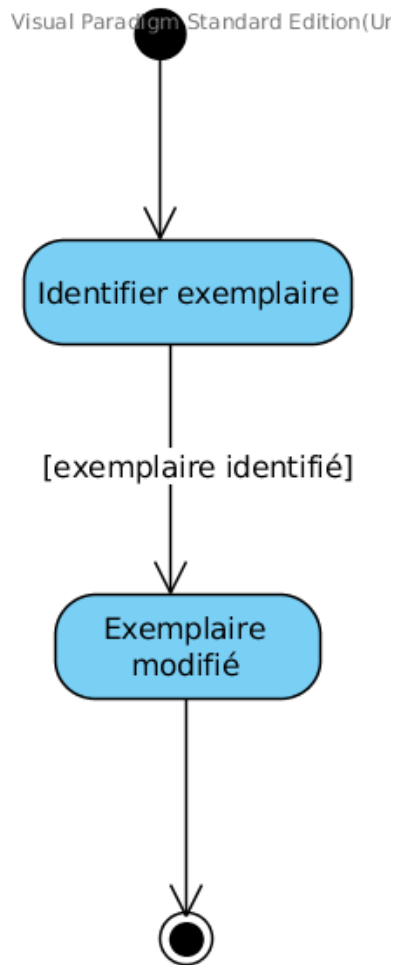


FIGURE 16 – Modifier un exemplaire

4.1.6 Supprimer un exemplaire

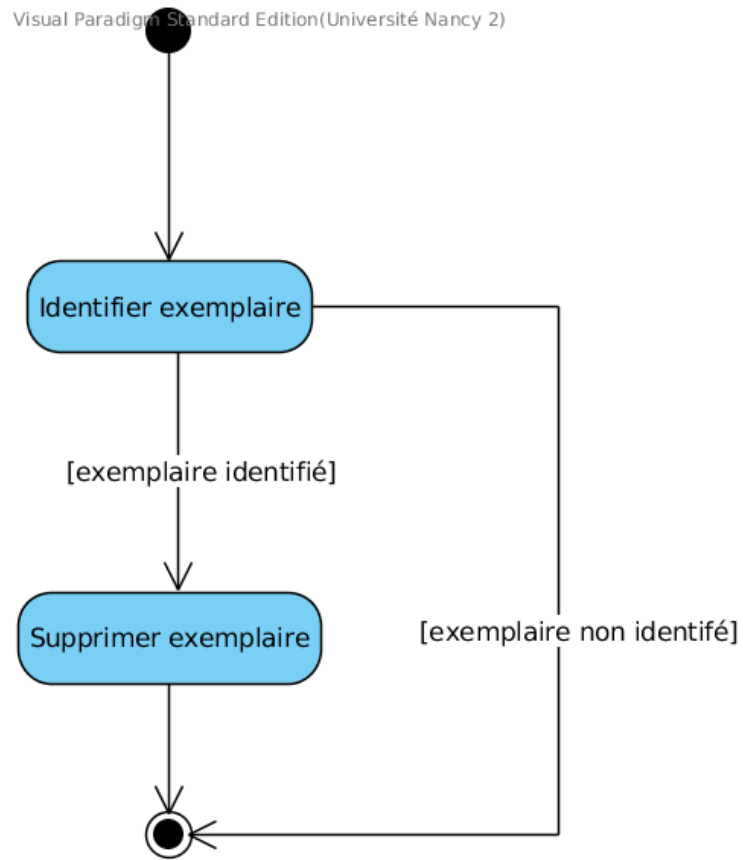


FIGURE 17 – Supprimer un exemplaire

4.1.7 Vérifier l'état d'un exemplaire

Visual Paradigm Standard Edition(Université Nancy 2)

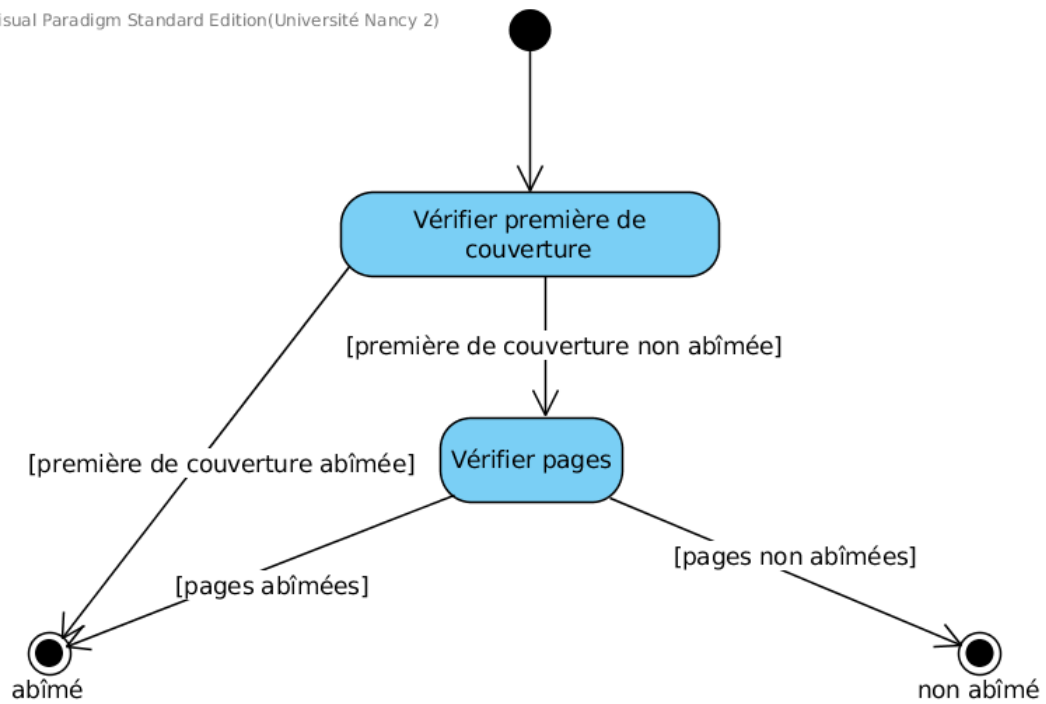


FIGURE 18 – Vérifier l'état d'un exemplaire

4.1.8 Rendre un exemplaire

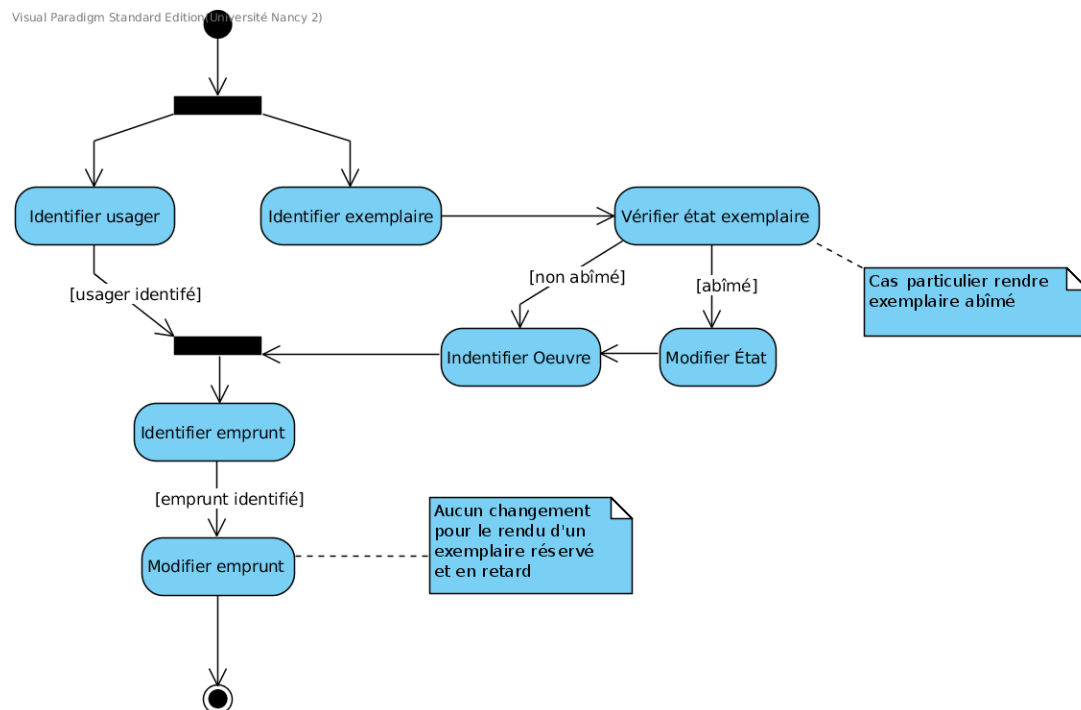


FIGURE 19 – Rendre un exemplaire