**République Algérienne Démocratique Et Populaire Ministère De l’Enseignement Supérieur Et De La Recherche Scientifique**

**Université A/Mira De Béjaia**

****

**Faculté Des Sciences Exactes Département d’Informatique**

**Rapport du projet MC**

**Option : Génie logiciel**

**Thème :**

**Réalisation d'une application pour la gestion d'une bibliothèque avec la méthode UP**

Réalisé par :

ALKAMA Salas.

AYADENE Meriem.

ARAR Manel.

BENKHALIFA Lamia.

Examinatrice :

Mme DJEBBAR Nacera.

**Année Universitaire 2022/2023**

**Table des matières**

[**1.** **Le processus unifié** 1](#_Toc134915349)

[**2.** **Comment on va procéder pour créer l'application au siens de notre groupe** 1](#_Toc134915350)

[3. **Phase d'Inception :** 2](#_Toc134915351)

[**2.1** **Comment diviser les taches de travaille sur ce projet avec la méthode UP dans notre groupe** 2](#_Toc134915352)

[**2.2** **Les AGL et outils utiliser** 3](#_Toc134915353)

[4. **Phase de Elaboration :** 3](#_Toc134915354)

[**4.1** **Identification du besoin** 3](#_Toc134915355)

[**4.2** **Spécification des besoins** 4](#_Toc134915356)

[**4.3** **Analyse** 5](#_Toc134915357)

[**4.1.1** **la description détaillée de chaque cas d'utilisation :** 5](#_Toc134915358)

[**4.1.2** **Diagrammes de séquences systèmes :** 7](#_Toc134915359)

[**4.1.3** **Diagrammes de classes :** 8](#_Toc134915360)

[**5.** **Phase de Construction :** 9](#_Toc134915361)

[**5.1** **Conception** 9](#_Toc134915362)

[**5.1.1** **Diagrammes de séquences détaillés** 9](#_Toc134915363)

[**5.1.2** **Diagramme d’état transition** 11](#_Toc134915364)

[**5.1.3** **Diagramme de classe** 11](#_Toc134915365)

[**5.1.4** **Principales Interfaces Implémenté** 12](#_Toc134915366)

[**6.** **Phase de Transition :** 14](#_Toc134915367)

[**7.** **Conclusion Générale** 14](#_Toc134915368)

**Table d’illustrations**

[Figure 1Diagramme de cas d'utilisation 4](#_Toc134915330)

[Figure 2Diagramme cas d’utilisation après incrémentation 5](#_Toc134915331)

[Figure 3Diagramme de séquence système Inscription d'un nouvel abonné 7](#_Toc134915332)

[Figure 4Diagramme de séquence système Emprunt d'un ouvrage 7](#_Toc134915333)

[Figure 5Diagramme de séquence système Acquisition d'un nouvel ouvrage 8](#_Toc134915334)

[Figure 6Diagramme de classe Inscription d'un nouvel abonné 8](#_Toc134915335)

[Figure 7Diagramme de classe Emprunt d'un ouvrage 8](#_Toc134915336)

[Figure 8Diagramme de classe Acquisition d'un nouvel ouvrage 9](#_Toc134915337)

[Figure 9Diagramme de séquence détaillé Inscription d'un nouvel abonné 9](#_Toc134915338)

[Figure 10Diagramme de séquence détaillé Emprunt d'un ouvrage 10](#_Toc134915339)

[Figure 11Diagramme de séquence détaille Acquisition d'un nouvel ouvrage 10](#_Toc134915340)

[Figure 12Diagramme d’état transition 11](file:///C:\Users\admin\Documents\projMC.docx#_Toc134915341)

[Figure 13Diagramme de classe 11](#_Toc134915342)

[Figure 14La Page D’accueil 12](#_Toc134915343)

[Figure 15L’interface de l’authentification 12](#_Toc134915344)

[Figure 16L’interface de la connexion du gestionnaire 12](#_Toc134915345)

[Figure 18L’interface de la fonction « inscription des abonnes » 13](#_Toc134915346)

[Figure 19L’interface de la connexion du bibliothécaire et son espace 13](#_Toc134915347)

[Figure 20Les interfaces des fonctions de la bibliothécaire 13](#_Toc134915348)

1. **Le processus unifié**

Le Processus Unifié (en anglais, Unified Process, UP) est une méthode de développement de logiciel qui met l'accent sur une approche itérative et incrémentale. Il est considéré comme un modèle de processus de développement de logiciel flexible et adaptable, qui peut être adapté en fonction des besoins de l'équipe de développement et du projet en cours. Le Processus Unifié est basé sur les principes de l'ingénierie logicielle orientée objet et est largement utilisé dans l'industrie pour le développement de grands projets de logiciels.

La méthode UP est composé de plusieurs éléments clés, notamment :

Les phases du processus : Le Processus Unifié comprend quatre phases principales : la phase d'initialisation, la phase d'élaboration, la phase de construction et la phase de transition. Chacune de ces phases est conçue pour atteindre des objectifs spécifiques et pour fournir des résultats mesurables.

Les disciplines du processus : Le Processus Unifié comprend des disciplines, qui couvrent l'ensemble du cycle de vie du développement de logiciel, tels que la gestion de projet, l'analyse des besoins, la conception, la mise en œuvre, les tests et l'assurance qualité.

Les itérations : Le Processus Unifié repose sur une approche itérative, dans laquelle le développement de logiciel est divisé en itérations ou cycles de développement courts. Chaque itération comprend toutes les disciplines du processus, mais se concentre sur une partie spécifique du développement.

Les artefacts : Le Processus Unifié utilise un certain nombre d'artefacts pour documenter et suivre le développement de logiciel. Ces artefacts incluent des modèles de cas d'utilisation, des diagrammes de séquence, des diagrammes de classes et des documents de spécification des exigences.

Le Processus Unifié est conçu pour être personnalisable et adaptable en fonction des besoins spécifiques de chaque projet. Les équipes de développement peuvent choisir les phases, les disciplines et les itérations appropriées pour répondre à leurs objectifs et aux besoins de leur projet.

1. **Comment on va procéder pour créer l'application au siens de notre groupe**

Voici comment on va utiliser le processus UP pour créer cette application de gestion de bibliothèque :

**Phase d'Inception :** La première étape consiste à définir le contexte et les objectifs du projet. Dans cette phase, on doit définir les exigences et les fonctionnalités de base de l'application de gestion de bibliothèque. On peut utiliser les besoins spécifiés ci-dessus pour définir les exigences.

**Phase de Elaboration :** La phase d'élaboration consiste à définir une architecture logicielle pour l'application. Dans cette phase, on doit définir les diagrammes de classes et de séquences ainsi que les cas d'utilisation détaillés pour l'application. On doit également définir les itérations du projet et les besoins pour chaque itération.

**Phase de Construction :** La phase de construction consiste à développer et à tester le code de l'application. On doit créer les composants logiciels tels que le système de gestion de la bibliothèque, le système de gestion des emprunts, le système de gestion des inscriptions, etc.

**Phase de Transition :** La phase de transition consiste à installer et à tester l'application dans l'environnement de production. On doit tester l'application avec des utilisateurs finaux et résoudre les éventuels problèmes ou bogues.

**Itérations :** Chaque phase consiste en plusieurs itérations. Les itérations doivent être planifiées en fonction des besoins et des objectifs du projet. Lors de chaque itération, on doit développer des fonctionnalités spécifiques de l'application et tester leur fonctionnement.

En suivant ce processus, on peut créer cette application de gestion de bibliothèque de manière itérative et incrémentale. Cela nous permettra d’assurer que l'application est bien adaptée aux besoins des utilisateurs et qu'elle fonctionne correctement.

1. **Phase d'Inception :**
2. **Comment diviser les taches de travaille sur ce projet avec la méthode UP dans notre groupe**

Voici comment on a envisagé la division des tâches pour notre groupe de 4 personnes utilisant la méthode UP pour la réalisation du ce projet :

**Responsable de projet :**

* Superviser et coordonner le travail du groupe
* Assurer la communication avec le client
* Planifier les différentes étapes du projet
* S'assurer que les délais sont respectés
* Gérer les ressources du projet

**Analyste fonctionnel :**

* Définir les besoins fonctionnels et non fonctionnels de l'application
* Élaborer les cas d'utilisation et les scénarios
* Concevoir les diagrammes UML (cas d'utilisation, classes, séquences, etc.)
* Participer à la rédaction des spécifications fonctionnelles

**Développeurs :**

* Concevoir et développer l'interface graphique de l'application
* Concevoir et développer la base de données de l'application
* Mettre en place les différentes fonctionnalités de l'application
* Réaliser les tests unitaires et les tests d'intégration

Il est important de noter que la répartition des tâches doit être flexible et évolutive en fonction de l'avancement du projet et des compétences et disponibilités de chacun avec une organisation régulière de réunions d'avancement pour assurer une bonne coordination entre les différents membres de l'équipe.

1. **Les AGL et outils utiliser**

NetBeans : un environnement de développement intégré (IDE) open source qui prend en charge plusieurs langages de programmation, tels que Java (celui avec qui on va travailler). Il offre des fonctionnalités telles que la coloration syntaxique, le débogage et la complétion automatique de code pour améliorer l'efficacité de notre développement du logiciel.

StarUML : un outil de modélisation UML open source et gratuit. Il permet de créer des diagrammes de classes, d'activités, de séquences et d'autres types de diagrammes UML. Avec lui on pourra générer le code java depuis le diagramme de classe qu’on va élaborer un peu plus tard.

MySQL Workbench : un outil de gestion de base de données qui permet de concevoir, de modéliser et d'administrer des bases de données MySQL de manière visuelle et intuitive mais pour nous on va l’utiliser pour la génération du script SQL depuis le diagramme de classe.

1. **Phase de Elaboration :**

**1ere itération**

1. **Identification du besoin**

Après avoir analysé les besoins de la bibliothèque universitaire, voici les besoins identifiés pour l'application de gestion de bibliothèque :

1. Gestion des abonnés : L'application doit permettre de gérer les abonnés selon les catégories d'abonnés identifiées (étudiants, enseignants, abonnés externes). Elle doit permettre d'enregistrer leurs coordonnées, leur statut d'abonnement, leur historique d'emprunts et de pénalités.
2. Gestion des ouvrages : L'application doit permettre de gérer les ouvrages selon leur référence, leur nombre d'exemplaires et leur emplacement dans les rayons. Elle doit permettre de connaître l'état de chaque exemplaire (emprunté, disponible, en réparation).
3. Gestion des emprunts : L'application doit permettre de gérer les emprunts d'ouvrages par les abonnés, en respectant les règles de gestion énoncées (nombre d'ouvrages empruntables, durée d'emprunt, pénalités en cas de retard). Elle doit permettre de générer des reçus d'emprunt et des rappels en cas de retard.
4. Gestion des restitutions : L'application doit permettre de gérer les restitutions d'ouvrages par les abonnés, en enregistrant la date de restitution et l'état de l'ouvrage restitué (en bon état ou endommagé). Elle doit permettre de mettre à jour l'état des exemplaires empruntés.
5. Gestion des pénalités : L'application doit permettre de gérer les pénalités imposées aux abonnés en cas de retard ou de non-restitution d'un ouvrage. Elle doit permettre de bloquer les emprunts des abonnés sanctionnés jusqu'à régularisation de leur situation.
6. Gestion des rapports : L'application doit permettre de générer des rapports d'activité de la bibliothèque, tels que le rapport quotidien des emprunts, le rapport quotidien détaillé et le rapport des ouvrages non restitués.

En résumé, l'application doit permettre une gestion efficace et automatisée de tous les aspects de la bibliothèque, depuis la gestion des abonnés jusqu'à la gestion des ouvrages et des emprunts, en passant par la gestion des pénalités et la génération de rapports d'activité.

1. **Spécification des besoins**

L’ensemble des Acteurs dans ce projet :

* Gestionnaire de la bibliothèque.
* Bibliothécaires.
* Étudiants abonnés.
* Enseignants.
* Étudiants externes abonnés.

Cas d'utilisation :

* S'inscrire à la bibliothèque.
* Effectuer une recherche d'un ouvrage.
* Emprunter un ou maximum trois ouvrages.
* Restituer un ou plusieurs ouvrages.
* Payer une pénalité pour un ou plusieurs ouvrages non restitués dans les délais.
* Consulter l'historique des emprunts et des pénalités.
* Prolonger exceptionnellement la durée d'emprunt d'un ou plusieurs ouvrages.
* Gérer les inscriptions et les pénalités des lecteurs qui ne restituent pas les ouvrages dans les délais.
* Limiter le nombre d'abonnés externes chaque année.

Diagramme de cas d’utilisation :

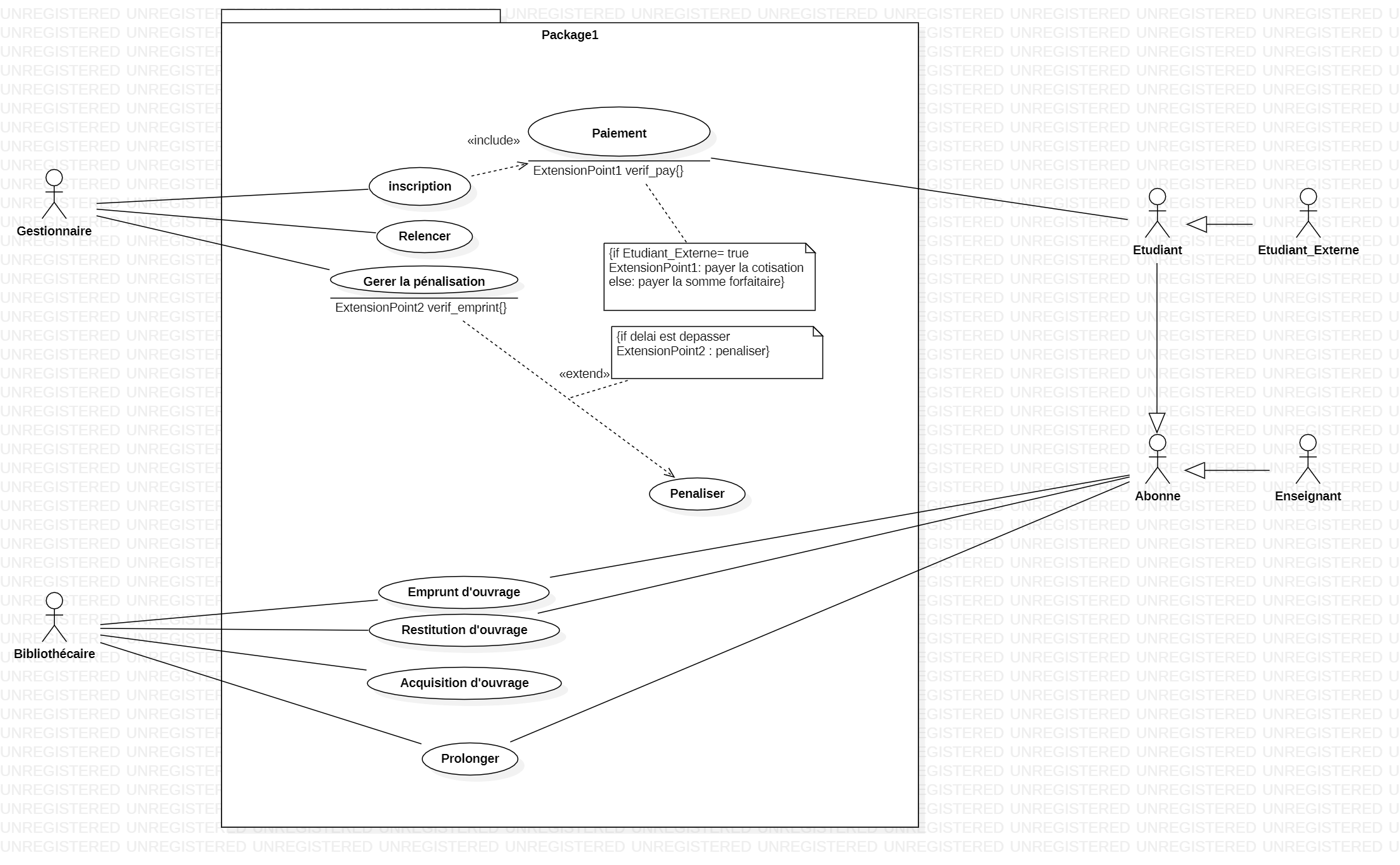


Figure 1Diagramme de cas d'utilisation

**2nd itération**

Après une réunion avec le client on a pu clarifier quelques détaille qui nous ont échapper lors de la première itération. Donc, après coup on a rajouter et raffiner ces cas d’utilisations :

* Éditer un reçu d'emprunt pour chaque ouvrage emprunté.
* Éditer un rapport quotidien de l'ensemble des emprunts de la journée.
* Éditer un rapport quotidien détaillé de l'ensemble des emprunts de la journée.
* Gérer l'acquisition de nouveaux ouvrages.

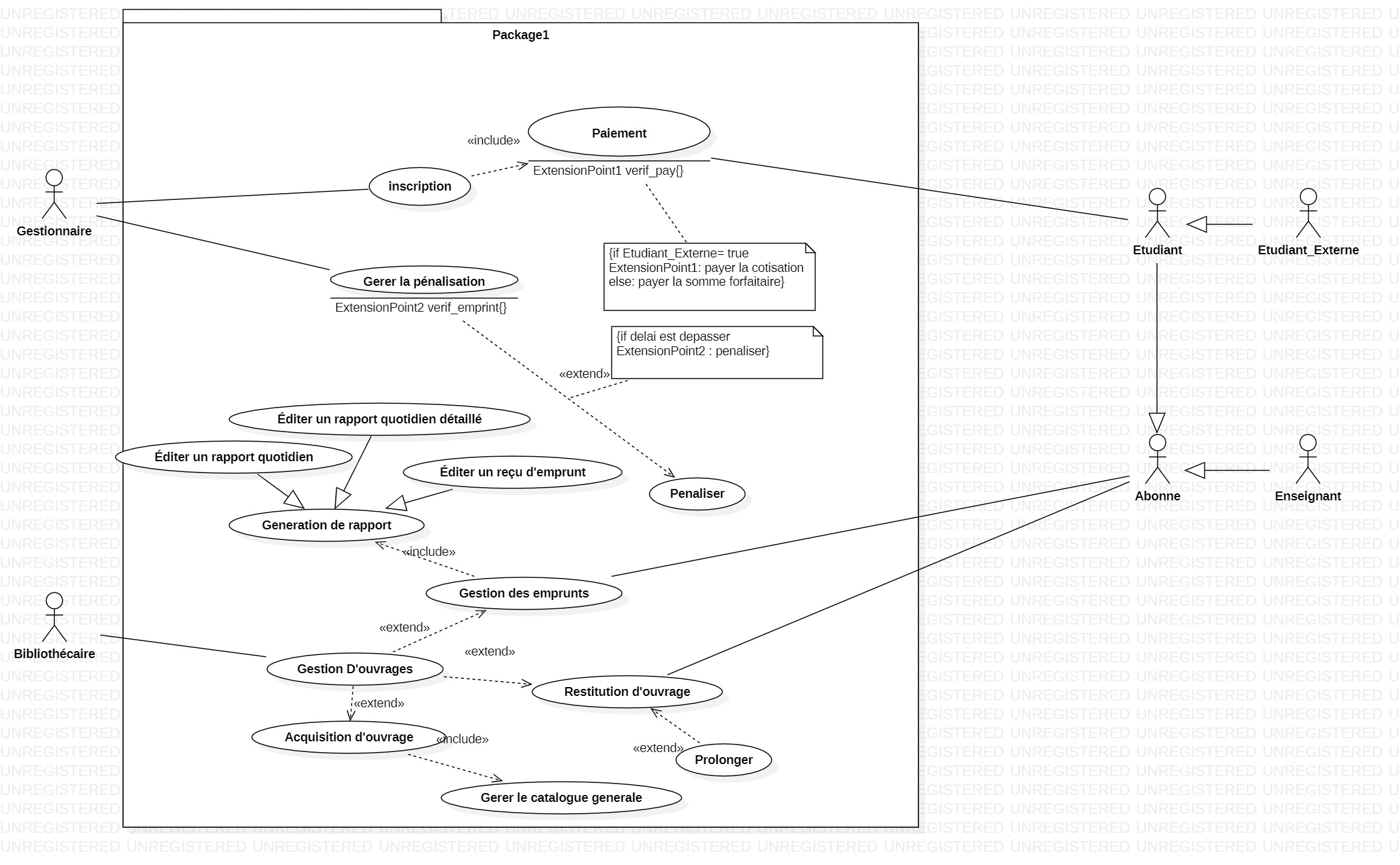


Figure 2Diagramme cas d’utilisation après incrémentation

1. **Analyse**

**3eme itération**

1. **la description détaillée de chaque cas d'utilisation :**

**Gestion de l'inscription :**

Scénario nominal :

* Le gestionnaire de la bibliothèque demande à l'étudiant/enseignant de remplir un formulaire d'inscription avec toutes les informations nécessaires (nom, prénom, adresse, numéro de téléphone, email, etc.).
* L'étudiant/enseignant remplit le formulaire et le remet au gestionnaire.
* Le gestionnaire vérifie les informations fournies et enregistre l'inscription de l'étudiant/enseignant.

Préconditions : Aucune.

Post-conditions : L'étudiant/enseignant est inscrit à la bibliothèque.

**Gestion des emprunts :**

Scénario nominal :

* L'étudiant/enseignant se présente à la bibliothèque pour emprunter un ou plusieurs ouvrages.
* Le bibliothécaire enregistre l'emprunt en scannant le code-barres des ouvrages et en associant l'emprunt à l'étudiant/enseignant.
* Le bibliothécaire imprime un reçu pour chaque ouvrage emprunté et le donne à l'étudiant/enseignant.
* Le système met à jour la base de données pour enregistrer les emprunts.

Préconditions : L'étudiant/enseignant doit être inscrit à la bibliothèque et ne doit pas avoir de pénalités en cours.

Post-conditions : Les ouvrages sont empruntés par l'étudiant/enseignant et les informations d'emprunt sont enregistrées dans la base de données.

Scénarios alternatifs :

* Si l'étudiant/enseignant a déjà emprunté le nombre maximum d'ouvrages autorisé, le système affiche un message d'erreur.
* Si un ouvrage est déjà emprunté par quelqu'un d'autre, le système affiche un message d'erreur.
* Si l'étudiant/enseignant n'a pas restitué un ouvrage dans les délais, le système l'empêche d'emprunter d'autres ouvrages jusqu'à la restitution de l'ouvrage en retard.

**Acquisition d'un nouvel ouvrage :**

Scénario nominal :

* Le bibliothécaire se connecte à l'application de gestion de la bibliothèque.
* Il accède à la fonctionnalité "Acquisition d'un nouvel ouvrage".
* Le système affiche le formulaire d'ajout d'un nouvel ouvrage.
* Le bibliothécaire saisit les informations de l'ouvrage (titre, auteur, éditeur, date de publication, nombre d'exemplaires).
* Le bibliothécaire valide l'ajout de l'ouvrage.
* Le système enregistre l'ouvrage dans le catalogue général des ouvrages et met à jour le nombre d'exemplaires dans le rayon correspondant.
* Le système affiche un message de confirmation de l'ajout de l'ouvrage.

Séquences alternatives :

* Erreur de saisie : Si le bibliothécaire saisit des informations incorrectes (ex: date de publication invalide), le système affiche un message d'erreur et demande de corriger les informations saisies avant de pouvoir valider l'ajout de l'ouvrage.
* Doublon : Si le bibliothécaire tente d'ajouter un ouvrage déjà présent dans le catalogue général des ouvrages, le système affiche un message d'erreur et demande de vérifier si l'ouvrage n'a pas déjà été ajouté.
* Problème d'ajout : Si le système rencontre un problème lors de l'ajout de l'ouvrage dans le catalogue général des ouvrages ou lors de la mise à jour du nombre d'exemplaires dans le rayon correspondant, le système affiche un message d'erreur et demande de réessayer ultérieurement.

Préconditions : Le bibliothécaire est connecté à l'application de gestion de la bibliothèque. L'ouvrage à ajouter n'est pas déjà présent dans le catalogue général des ouvrages.

Post-conditions : L'ouvrage est ajouté au catalogue général des ouvrages.

1. **Diagrammes de séquences systèmes :**

Inscription d'un nouvel abonné :

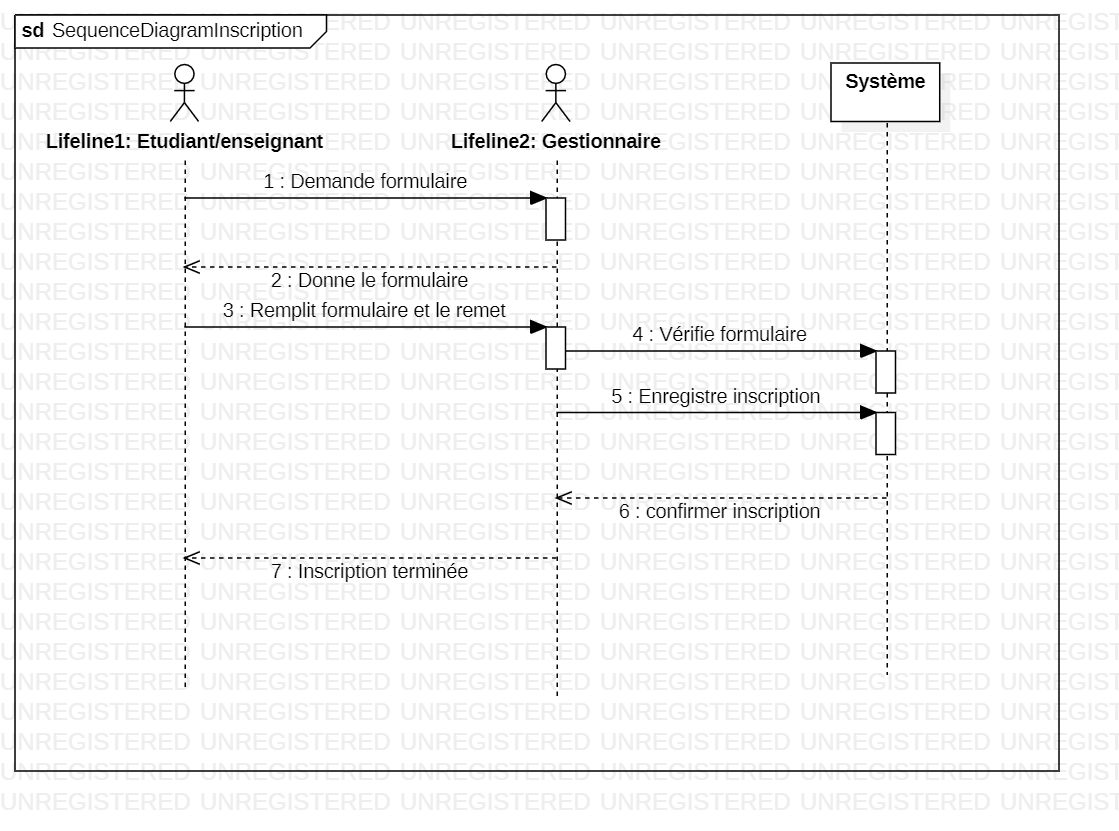


Figure 3Diagramme de séquence système Inscription d'un nouvel abonné

Emprunt d'un ouvrage :

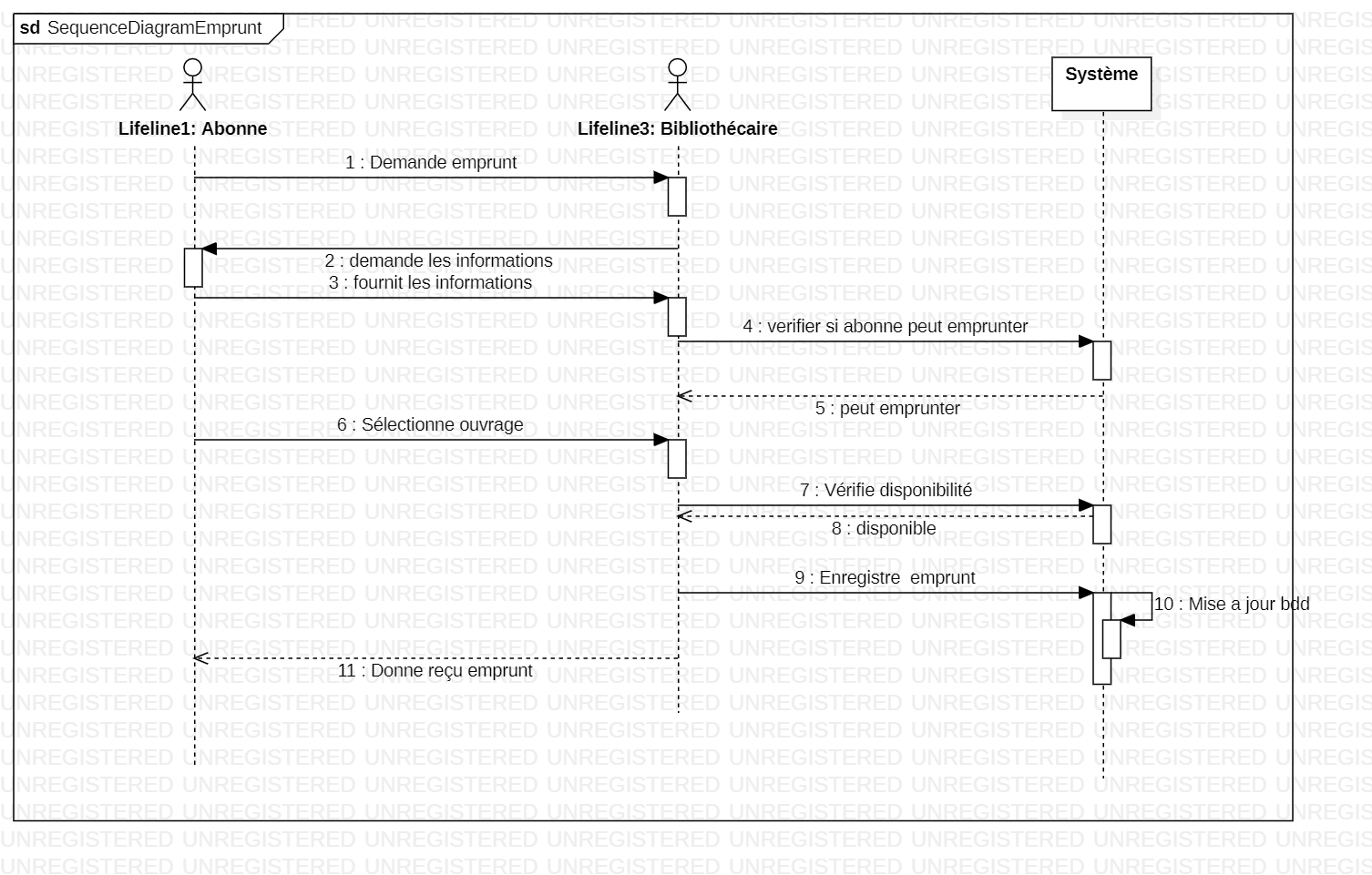


Figure 4Diagramme de séquence système Emprunt d'un ouvrage

Acquisition d'un nouvel ouvrage :

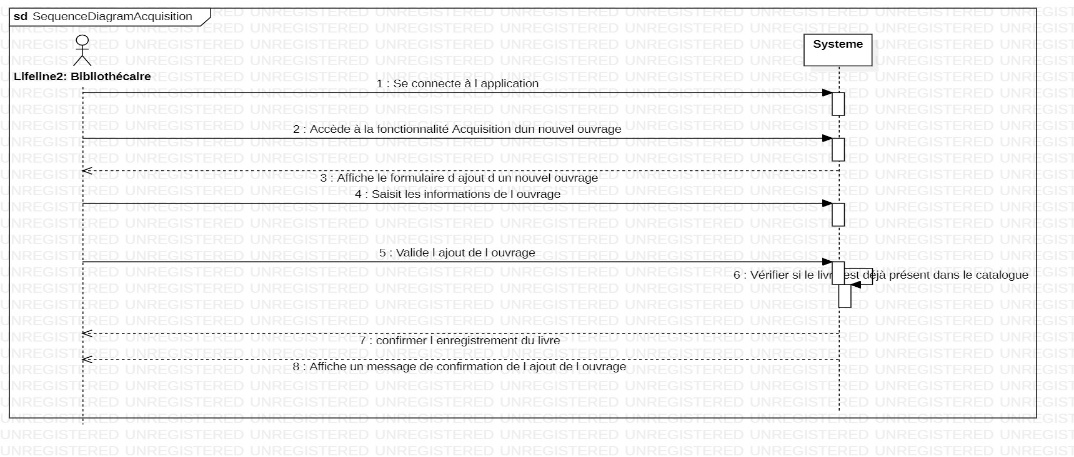
****

Figure 5Diagramme de séquence système Acquisition d'un nouvel ouvrage

1. **Diagrammes de classes :**

Inscription d'un nouvel abonné :

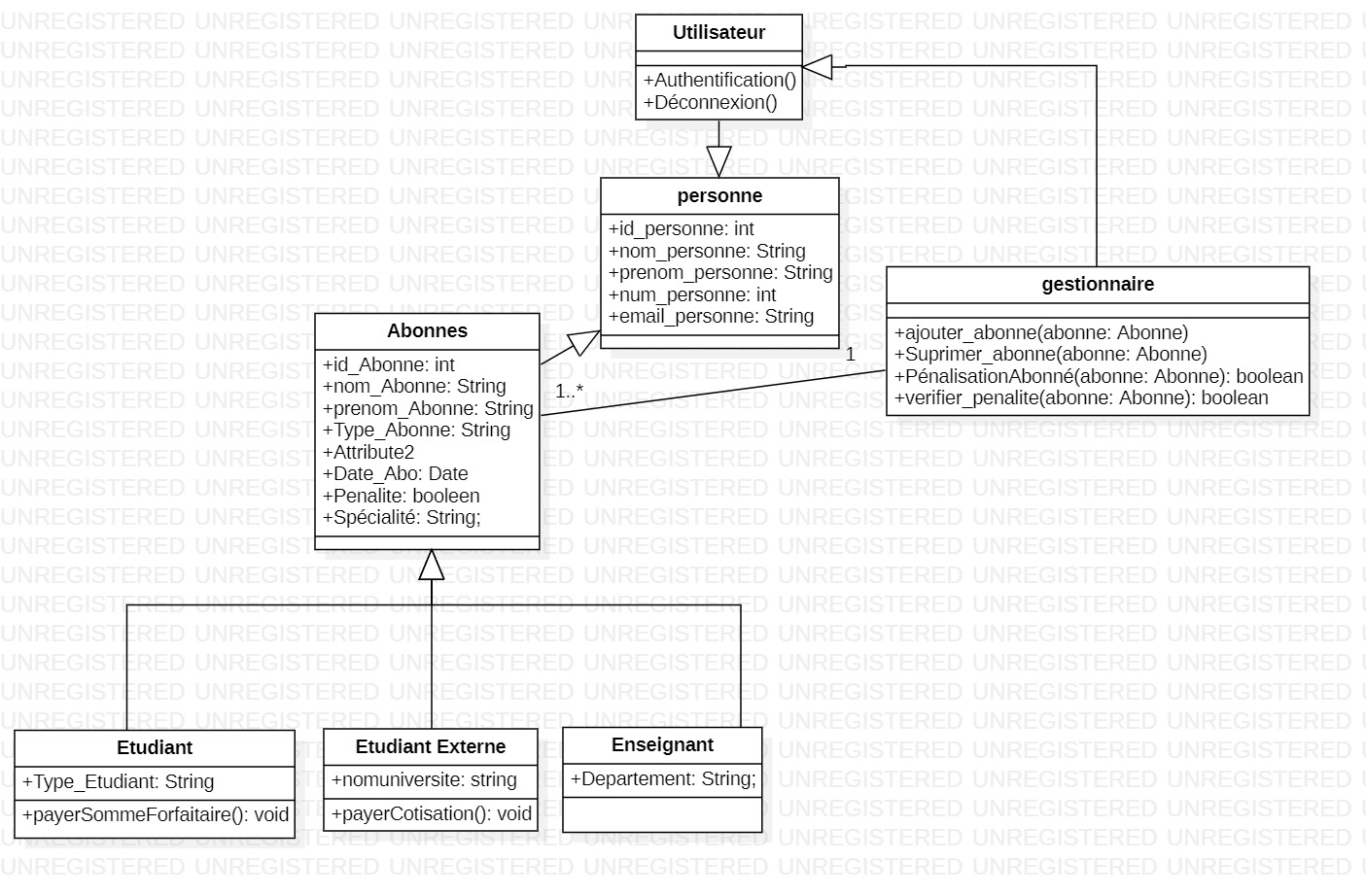


Figure 6Diagramme de classe Inscription d'un nouvel abonné

Empreint d'un ouvrage :

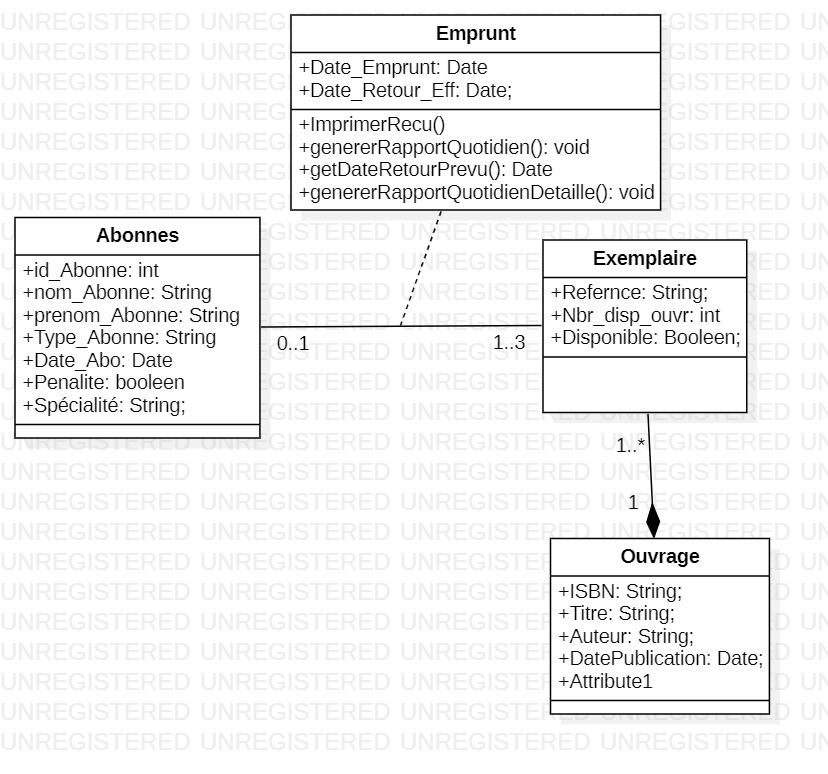


Figure 7Diagramme de classe Emprunt d'un ouvrage

Acquisition d'un nouvel ouvrage :

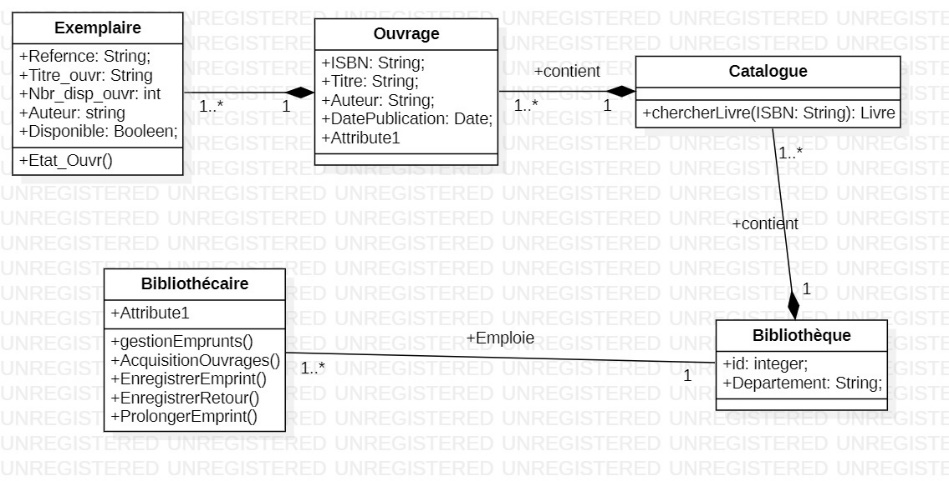
****

Figure 8Diagramme de classe Acquisition d'un nouvel ouvrage

1. **Phase de Construction :**

**4eme itération**

1. **Conception**
2. **Diagrammes de séquences détaillés**

Inscription d'un nouvel abonné :



Figure 9Diagramme de séquence détaillé Inscription d'un nouvel abonné

Dans cette itération on a pris en compte le scénario alternatif : erreur lors de la saisis d’information

Empreint d'un ouvrage :

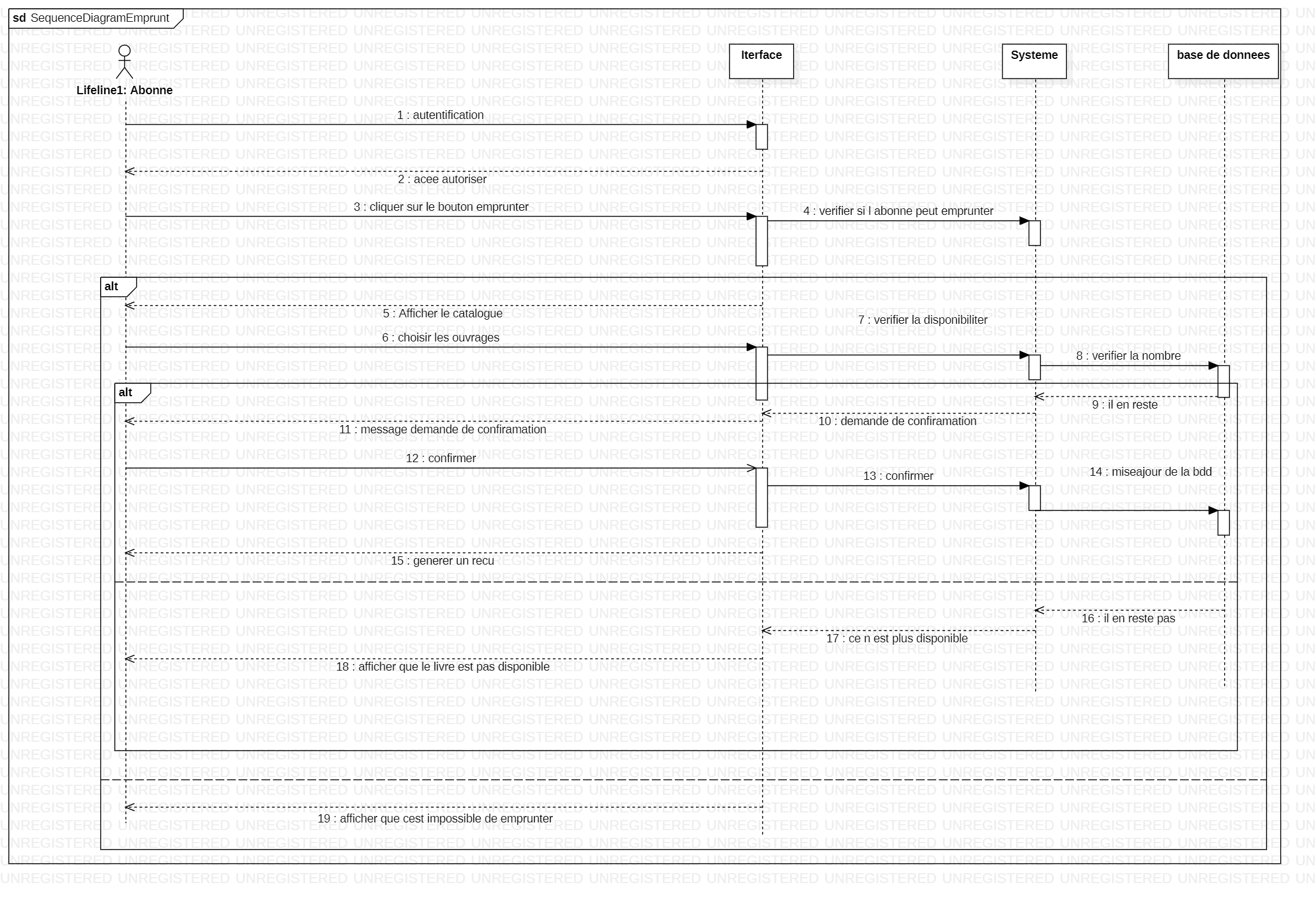


Figure 10Diagramme de séquence détaillé Emprunt d'un ouvrage

Ici aussi lors de cette itération on a pris en compte les scénarios alternatifs déjà exprimer dans la section analyse et on les ajouter.

Acquisition d'un nouvel ouvrage :

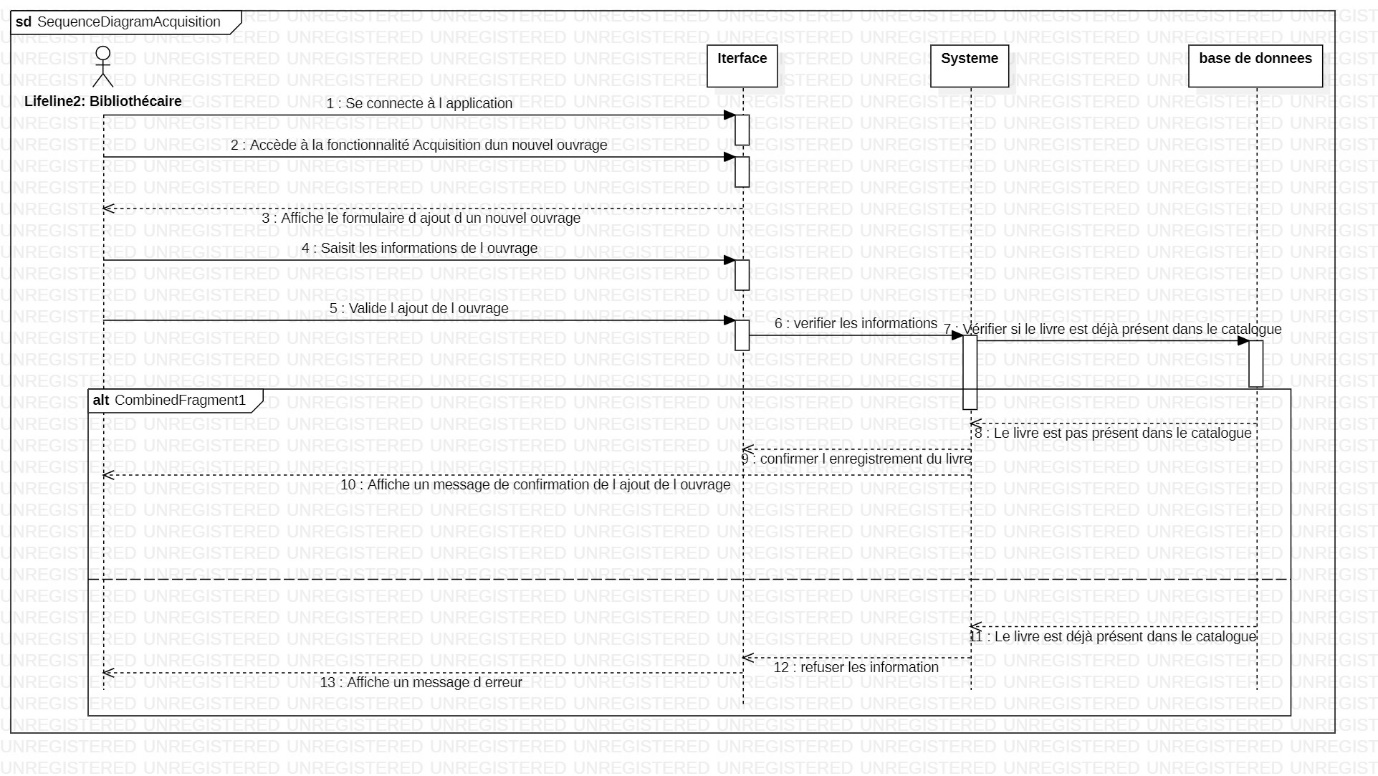


Figure 11Diagramme de séquence détaille Acquisition d'un nouvel ouvrage

Tout comme ses prédécesseurs, lors de cette itération on a pris en compte ses scénarios alternatifs.

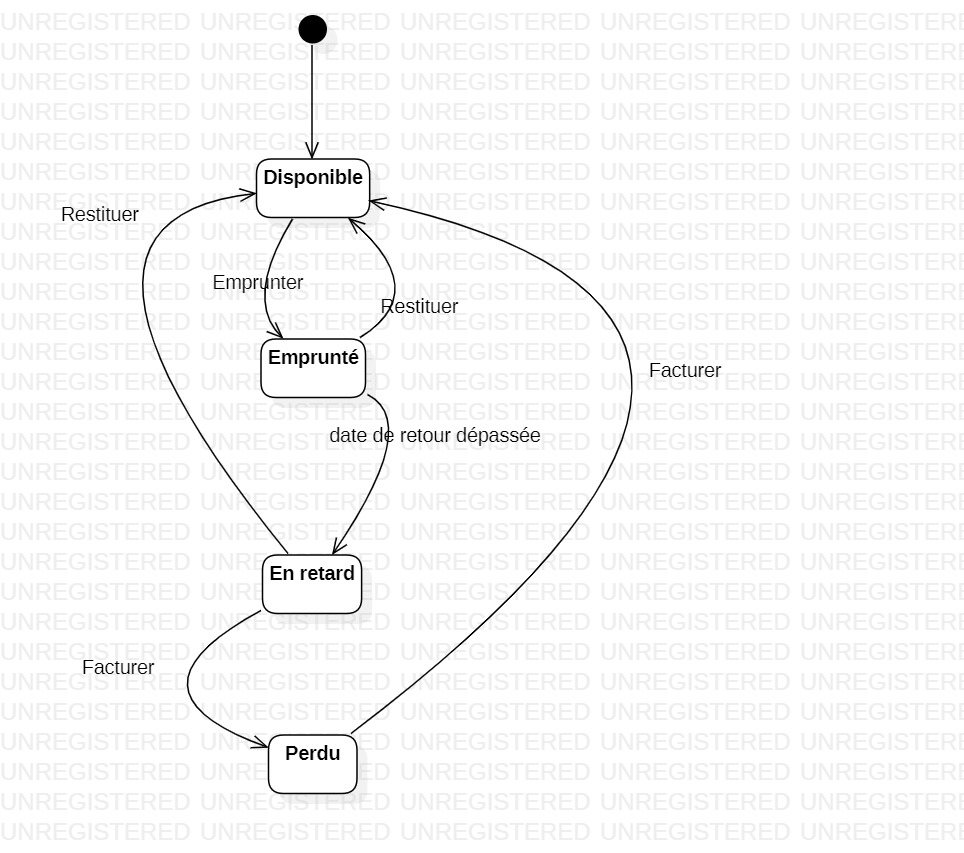
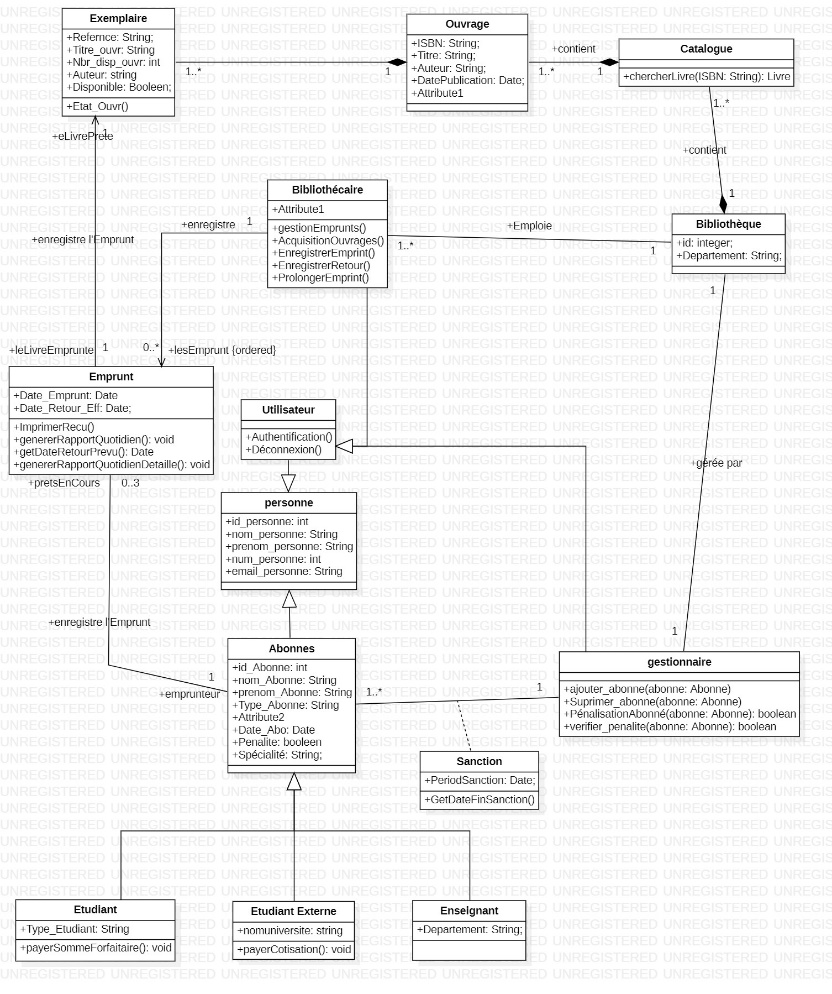
1. **Diagramme d’état transition**

Figure 12Diagramme d’état transition

Pour pouvoir nous aider dans cette étape on a rajouter un diagramme état transition pour bien mettre en évidence les états différents d’un livre pour être encore plus performant sans se perdre sur le chemin pendant le développement.

****

1. **Diagramme de classe**

Grace a toutes les itérations faites auparavant, on a réussi a mettre en place ce diagramme de classe complet qui lui va nous permettre grace au agl cite au paravant pour la generation du code java et celui du code SQL.

Figure 13Diagramme de classe

**5eme itération**

1. **Principales Interfaces Implémenté**

L’objectif de cette partie du chapitre est de donner un aperçu de quelque IHM de l’application :

La Page D’accueil



Figure 14La Page D’accueil

L’interface de l’authentification



Figure 15L’interface de l’authentification

L’interface de la connexion du gestionnaire et son espace



Figure 16L’interface de la connexion du gestionnaire

L’interface de la fonction « inscription des abonnes »



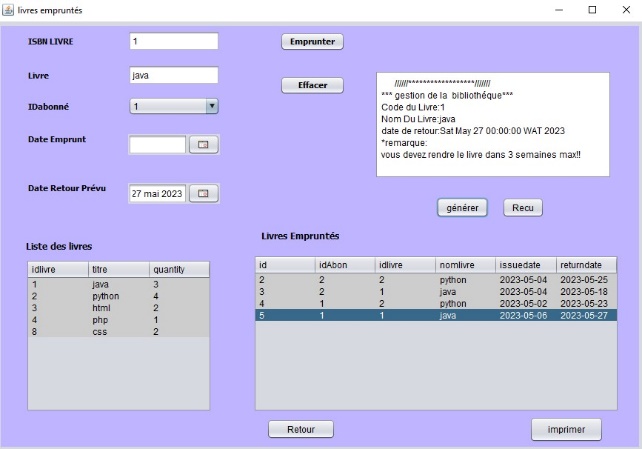
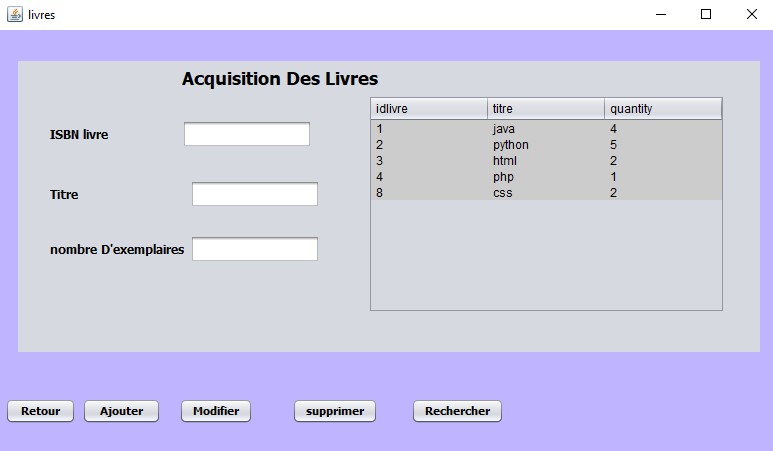
Figure 18L’interface de la fonction « inscription des abonnes »

L’interface de la connexion du bibliothécaire et son espace

Figure 19L’interface de la connexion du bibliothécaire et son espace

Les interfaces des fonctions de la bibliothécaire



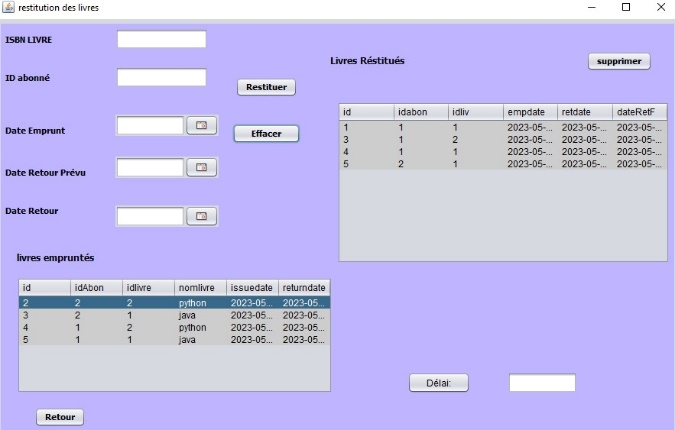


Figure 20Les interfaces des fonctions de la bibliothécaire

1. **Phase de Transition :**

C’est la dernière phase de la méthode UP, et elle consiste à valider et installer le logiciel. Dans cette phase on a effectué les derniers tests afin de garantir que le logiciel est fonctionnel et répond aux exigences.

1. **Conclusion Générale**

Le travail réalisé dans ce rapport consiste à concevoir et mettre en place d'une application pour la gestion d'une bibliothèque avec la méthode UP dédier au gestionnaire et au bibliothécaire.

Ce projet nous a permis de mettre en pratique nos connaissances universitaires et de travailler avec de nouvelles technologies de développement et aussi avec des AGL en suivant la méthode agile « Processus Unifier ». Nous espérons que ce travail pourra être utile à quiconque voudra entreprendre un projet similaire.