



Université du Québec en Outaouais Automne 2025
Département d'informatique et d'ingénierie
INF1163 – Modélisation et conception orientée objet
Prof. Karim El Guemhioui

Titre du projet : Système de contrôle d'une borne de prêt en libre-service

Membres : Claude Raphael Coly

Mouhamadou Lamine Dial

Cheikh Bamba Gueye

Bilal Fall Niang



Table des matières

Titre du projet : Système de contrôle d'une borne de prêt en libre-service	1
1. Introduction	3
2. Description générale du système	4
3. Cas d'utilisation du système.....	4
3.1 Cas d'utilisation – Emprunter documents	4
3.2 Cas d'utilisation – Démarrer le système	5
4. Diagramme de séquence système (DSS).....	5
5. Diagramme de classes – Couche d'affaires	6
6. Catalogue des documents	7
6.1 Catalogue complet des livres.....	7
7. Dépendances et librairies	8
7.1 Librairies utilisées	8
7.2 Version Java requise	9
7.3 Absence de dépendances externes	9
8. Annexe — Contribution des membres de l'équipe.....	9
8.1 Répartition des rôles.....	9
8.2 Travail collaboratif	10
8.3 Organisation du travail	10
Conclusion	11

1. Introduction

Ce projet a pour objectif de concevoir et de modéliser un système de contrôle d'une borne de prêt en libre-service destinée à une bibliothèque. Cette borne permet aux usagers d'effectuer eux-mêmes l'emprunt de documents sans passer par un comptoir traditionnel. Le système repose sur une identification sécurisée des usagers ainsi que sur l'utilisation de codes RFID permettant d'identifier chaque exemplaire de livre de manière unique.

Le projet vise principalement à appliquer les concepts de la modélisation orientée objet vus dans le cours INF1163, notamment les cas d'utilisation, les diagrammes UML et la structuration logique d'un système logiciel cohérent. Une implémentation en Java permet également de valider la logique métier modélisée et d'illustrer le fonctionnement global du système.

2. Description générale du système

Le système prend la forme d'une borne de prêt automatisée accessible aux usagers de la bibliothèque. Pour utiliser la borne, l'utilisateur doit d'abord s'authentifier à l'aide de son numéro de compte et de son NIP personnel. Cette étape garantit un accès sécurisé au système.

Une fois l'authentification réussie, l'utilisateur peut emprunter un ou plusieurs documents. Chaque document physique est identifié par un code RFID unique composé de six chiffres. Le système vérifie l'existence et la disponibilité de chaque exemplaire avant de l'ajouter à l'emprunt. À la fin de la transaction, le système calcule automatiquement la date de retour prévue, fixée à 21 jours après la date d'emprunt, et propose à l'utilisateur de recevoir ou non un reçu.

3. Cas d'utilisation du système

3.1 Cas d'utilisation – Emprunter documents

Acteur principal :

- Usager

Description détaillée (format acteur / système)

Usager	Système
Se présente devant la borne	Affiche l'écran d'accueil
Entre son numéro de compte	Demande le NIP
Entre son NIP	Vérifie l'authentification
Saisit le code RFID d'un exemplaire	Vérifie l'existence du code RFID
—	Vérifie la disponibilité de l'exemplaire
Saisit d'autres codes RFID	Ajoute les exemplaires à l'emprunt
Confirme la fin de la saisie	Calcule les dates de retour
Choisit s'il souhaite un reçu	Génère ou non le reçu
Quitte la borne	Termine la transaction et déconnecte l'utilisateur

Préconditions :

- Le système est démarré et fonctionnel
- Le catalogue de livres est chargé
- L'utilisateur possède un compte valide

Postconditions :

- Les exemplaires sont marqués comme empruntés
- Les dates de retour sont définies (21 jours)
- L'utilisateur est déconnecté du système

Exceptions :

- Authentification échouée
- Code RFID invalide
- Exemplaire non disponible
- Tentative de finalisation sans exemplaire

3.2 Cas d'utilisation – Démarrer le système

Lorsque l'application est lancée, le système s'initialise automatiquement. Il charge le catalogue de livres et d'exemplaires en mémoire, crée les objets correspondants et prépare les données nécessaires à l'exécution du système. Une fois cette étape complétée, l'interface d'authentification est affichée et la borne devient prête à recevoir des demandes d'emprunt.

Postconditions :

- Le catalogue est chargé
- Les usagers sont disponibles
- Le système est opérationnel

Exception :

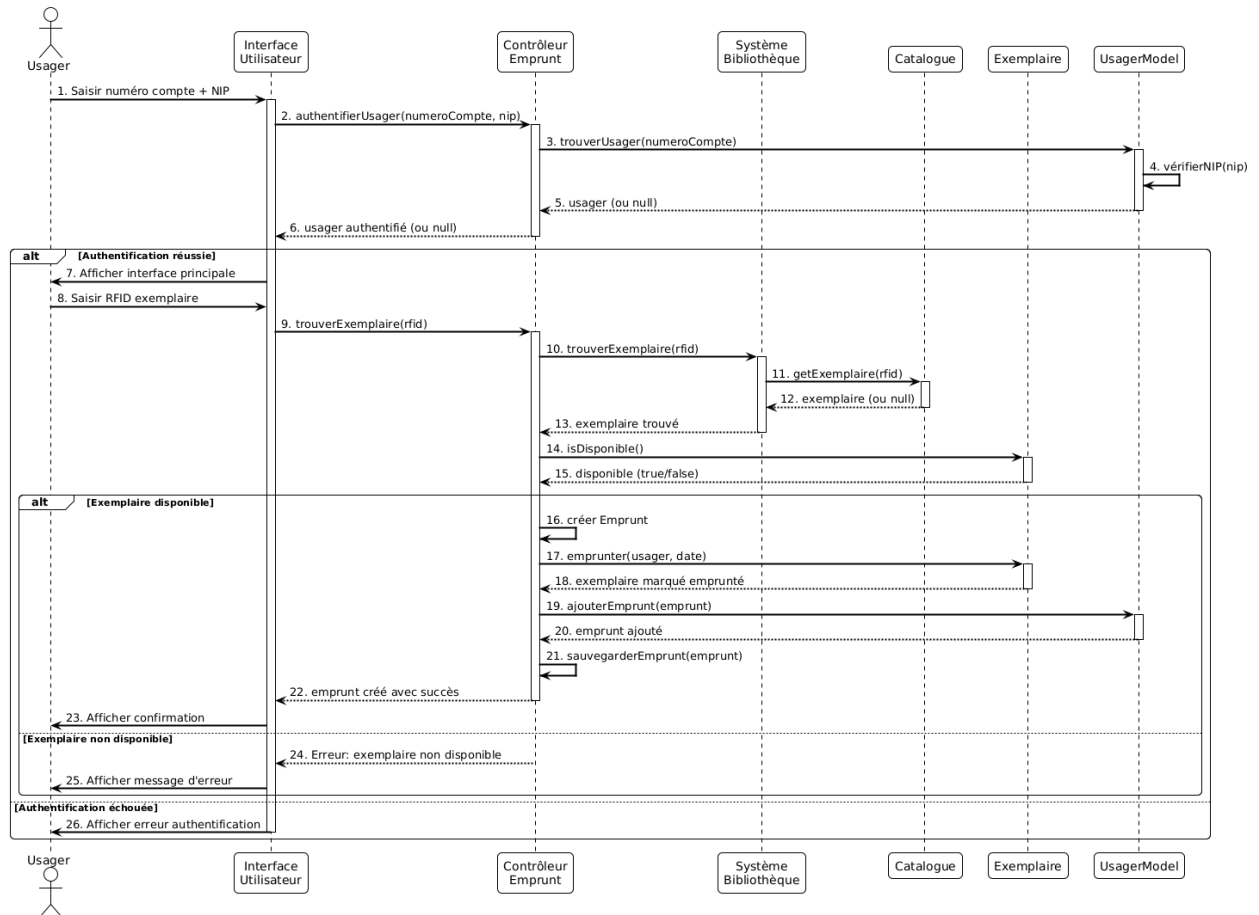
- En cas d'échec du chargement, un message d'erreur est affiché et le système se ferme

4. Diagramme de séquence système (DSS)

Le diagramme de séquence système (DSS) illustre les interactions entre l'acteur Usager et le Système lors du cas d'utilisation « Emprunter documents ». Il met en évidence les événements-système principaux tels que l'authentification, la saisie des codes RFID, la vérification des exemplaires et la finalisation de l'emprunt.

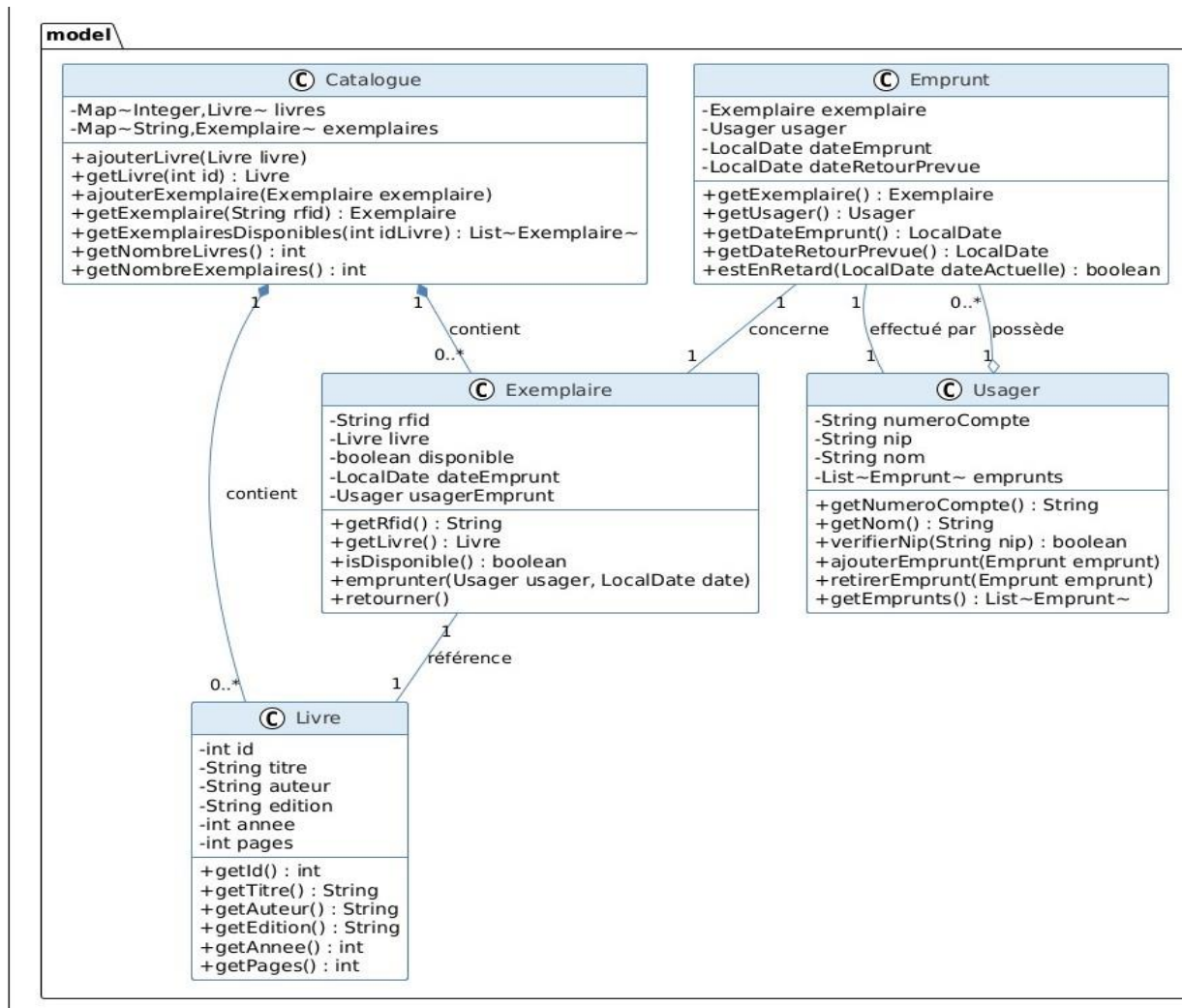
Événements-système identifiés :

- demarrerSession()
- saisirIdentifiants(numeroCompte, nip)
- saisirRFID(codeRFID)
- ajouterExemplaire()
- terminerSaisie()
- choisirTypeRecu(type)



5. Diagramme de classes – Couche d'affaires

Le diagramme de classes de la couche d'affaires modélise les principales entités métier du système. Les classes Usager, Livre, Exemplaire et Emprunt représentent les concepts centraux de la bibliothèque. Un usager peut effectuer plusieurs emprunts. Chaque emprunt concerne un exemplaire et chaque exemplaire est associé à un livre. Ce diagramme formalise les règles métier indépendamment de l'interface utilisateur.



6. Catalogue des documents

Le catalogue utilisé dans le projet comprend dix livres distincts pour un total de quinze exemplaires, chacun identifié par un code RFID unique.

6.1 Catalogue complet des livres

RFID	Titre	Auteur	Édition	Date de parution	Pages
123456	Introduction à la programmation	Robert Bouchard	3e	2020	450
234567	Introduction à la programmation	Robert Bouchard	3e	2020	450
345678	Analyse et conception orientée objet	Sylvie Moreau	2e	2019	580

RFID	Titre	Auteur	Édition	Date de parution	Pages
456789	Bases de données relationnelles	Pierre Desjardins	4e	2021	620
567890	Bases de données relationnelles	Pierre Desjardins	4e	2021	620
678901	Algorithmes et structures de données	Marie-Claire Lévesque	1re	2020	520
789012	Architecture des ordinateurs	Jean-François Dufour	5e	2019	480
890123	Intelligence artificielle	Lucie Beaulieu	1re	2022	680
901234	Intelligence artificielle	Lucie Beaulieu	1re	2022	680
012345	Sécurité informatique	André Côté	2e	2021	550
111111	Réseaux et télécommunications	Isabelle Parent	3e	2020	600
222222	Ingénierie logicielle	Thomas Gagnon	2e	2021	590
333333	Gestion de projets informatiques	Valérie Dubois	1re	2022	510
444444	Analyse et conception orientée objet	Sylvie Moreau	2e	2019	580
555555	Algorithmes et structures de données	Marie-Claire Lévesque	1re	2020	520

7. Dépendances et librairies

7.1 Librairies utilisées

Le projet a été développé en utilisant **uniquement les librairies standard fournies avec le Java Development Kit (JDK)**. Aucune librairie externe (Maven, Gradle ou autre framework) n'est requise pour la compilation ou l'exécution du système.

Cette contrainte permet de garantir la portabilité du projet et de faciliter son exécution sur tout environnement disposant d'un JDK conforme.

Les principales librairies Java utilisées sont :

- **java.lang** : classes de base du langage (String, Object, Exception, etc.)
- **java.util** : structures de données (List, Map, ArrayList, HashMap, etc.)
- **java.io** : lecture et écriture des fichiers (BufferedReader, IOException, etc.)
- **java.nio.file** : gestion moderne des fichiers et des chemins
- **java.time** : gestion des dates (LocalDate, DateTimeFormatter)
- **javax.swing** : composants de l'interface graphique (JFrame, JPanel, JButton, JTextField, JLabel, JOptionPane, etc.)
- **java.awt** : éléments graphiques de base (Color, Font, Layouts)
- **java.awt.event** : gestion des événements utilisateur

7.2 Version Java requise

Le projet nécessite **Java JDK 8 ou une version ultérieure**. Les fonctionnalités utilisées (API de dates, collections avancées, gestion des fichiers) sont entièrement compatibles avec Java 8 et les versions plus récentes.

7.3 Absence de dépendances externes

Aucune dépendance externe n'a été utilisée dans le projet. L'ensemble des fonctionnalités repose exclusivement sur les bibliothèques standard de Java, ce qui simplifie la compilation, l'exécution et l'évaluation du projet.

8. Annexe — Contribution des membres de l'équipe

8.1 Répartition des rôles

Le projet a été réalisé de manière collaborative avec une répartition claire des responsabilités. Chaque membre de l'équipe a contribué de façon équitable aux différentes étapes du projet, soit la modélisation, le développement, les tests et la documentation.

Les rôles attribués visaient à structurer le travail et à assurer une progression efficace du projet, tout en maintenant une collaboration constante entre les membres.

Membre	Rôle	Contributions
Claude Raphael Coly	Chef d'équipe / Coordination	Organisation du projet, coordination de l'équipe, gestion du dépôt GitHub, intégration des différentes parties du projet, validation finale du livrable et préparation de la soumission
Mouhamadou Lamine Dial	Conception UML et modélisation	Analyse de l'énoncé, rédaction des cas d'utilisation (format détaillé à deux colonnes), conception du diagramme de

Membre	Rôle	Contributions
		classes de la couche d'affaires, création du diagramme de séquence système (DSS), vérification de la cohérence entre la modélisation et le système
Bilal Fall Niang	Développement logiciel	Implémentation des classes métier (Livre, Exemple, Usager, Emprunt, Catalogue), développement du contrôleur d'emprunt, gestion des fichiers CSV, développement du système principal et de l'interface utilisateur, création du fichier JAR
Cheikh Bamba Gueye	Tests et validation	Réalisation des tests unitaires et d'intégration, validation des cas d'utilisation, tests des scénarios alternatifs, vérification de la conformité aux exigences du projet, tests de compilation et d'exécution

8.2 Travail collaboratif

En plus de leurs rôles principaux, tous les membres de l'équipe ont participé activement aux activités suivantes :

- réunions de planification et de suivi ;
- révision croisée du code et des diagrammes UML ;
- correction des erreurs détectées lors des tests ;
- amélioration continue de la documentation ;
- validation finale du rapport et du système.

Cette approche collaborative a permis d'assurer une cohérence entre la modélisation, l'implémentation Java et la documentation finale.

8.3 Organisation du travail

Le projet a été développé en équipe à l'aide d'un dépôt GitHub partagé. Cette organisation a permis un suivi clair des contributions, une intégration progressive des fonctionnalités et une meilleure coordination entre les membres. Les modifications ont été validées avant leur intégration afin de garantir la stabilité du projet.

Conclusion

Ce projet a permis de concevoir, de modéliser et d'implémenter un système de borne de prêt en libre-service pour une bibliothèque, en appliquant rigoureusement les concepts de la modélisation orientée objet enseignés dans le cours **INF1163**.

Les cas d'utilisation détaillés, les diagrammes UML (diagramme de classes et diagramme de séquence système) ainsi que l'implémentation Java démontrent une compréhension claire des exigences fonctionnelles et des règles métier du système. Le projet obtenu est structuré, cohérent et fonctionnel, et constitue un livrable conforme aux attentes académiques du cours.