

PROJET : API BIBLIOTHEQUE

Cahier des charges et de
conception partielle



NOTES D'ÉDITION

Première édition @2025

Création de l'œuvre : Patrick Simard

Collaboration(s) : Steven Duquette (2025)

Actualisation(s) : Aucun

CONDITIONS D'UTILISATION

L'auteur autorise les enseignants du département d'informatique du Cégep de Chicoutimi à utiliser cette œuvre exclusivement dans le cadre de leur cours de la formation régulière en présentielle au même collège.

En cas d'appropriation d'une portion mineure afin de contribuer à la création d'une nouvelle œuvre ou en ajout à une œuvre existante, l'auteur demande que sa participation soit reconnue et qu'il soit mentionné dans la nouvelle œuvre à titre de collaborateur. La consultation de l'auteur pour cet acte est fortement suggérée.

L'appropriation d'une portion majeure afin de contribuer à la création d'une nouvelle œuvre ou en ajout à une œuvre existante n'est pas autorisée par l'auteur. L'auteur demande à être consulté au préalable afin d'obtenir son autorisation.

En cas de modification mineure (ajout mineur, retrait ou modification d'élément), l'auteur invite l'enseignant concerné à le mentionner à la section *Actualisation(s)* en ajoutant son nom et la ou les année(s) où cette action a été posée.

RECONNAISSANCE ET PROTECTION DU DROIT D'AUTEUR

Selon l'article 8-1.03 de la convention collective des enseignants, les cahiers de notes ou notes de cours ne peuvent être utilisés sans le consentement explicite de l'auteur. De plus, selon l'annexe V-4 la reproduction sous une forme quelconque, la présentation en public et la publication d'un œuvre est un droit exclusif appartenant à l'auteur.

TABLE DES MATIERES

PRESENTATION DU PROJET	1
Contexte	1
Besoin	1
DESCRIPTION DE LA DEMANDE	2
Objectifs	2
Fonctionnalités du système.....	2
Critères d'acceptabilité	3
SOLUTION A APPLIQUER	4
Application	4
Spécifications techniques.....	4
Langage de développement.....	4
Patron de conception.....	4
Sauvegarde des données	4
MODELISATION (PARTIELLE) DE LA SOLUTION	5
Diagramme de cas d'utilisation.....	5
Maquettes	6
Fenêtre principale	6
Vue d'authentification.....	7
Vue de la modification des livres.....	8
POURSUITE DE LA CONCEPTION.....	9
Maquettes	9
Diagramme de classes	9
Plan de tests.....	9
REMISE DU TRAVAIL	10
Date de remise	10
Format de la remise.....	10

PRESENTATION DU PROJET

CONTEXTE

Dans le cadre du cours 420-D52-CH - Intro aux services de données, vous avez appris à développer une application pour communiquer avec une API et vous avez même développé votre propre API permettant de gérer les données d'une collection de livres et d'auteurs à l'aide d'une de base de données.

BESOIN

Dans la poursuite de la réalisation de cette API, il vous est maintenant demandé de développer une application simple de gestion d'une collection de livres pour exploiter cette API et valider vos connaissances dans cet aspect.

DESCRIPTION DE LA DEMANDE

OBJECTIFS

L'objectif principal du projet consiste à développer une application graphique de gestion d'une collection de livres. Cette application devra permettre de réaliser les tâches suivantes :

- *Gérer une collection de livre ;*
- *Sauvegarder et récupérer les données à l'aide d'une API exploitant une base de données.*

FONCTIONNALITES DU SYSTEME

Afin d'effectuer adéquatement les tâches précédentes, voici les différentes fonctionnalités qui devront être offertes par le système :

- *Accéder sécuritairement à la base de données*
 - *Saisir le nom d'utilisateur*
 - *Saisir le mot de passe*
- *Afficher la liste des auteurs*
- *Afficher la liste des livres (titre et ISBN)*
- *Sélectionner un livre*
- *Afficher les informations d'un livre (titre, ISBN, nombre de pages et auteur)*
- *Modifier les informations d'un livre*
 - *Saisir le titre*
 - *Saisir l'ISBN*
 - *Saisir le nombre de pages*
 - *Sélectionner l'auteur*
- *Afficher un message d'information*

CRITERES D'ACCEPTABILITE

Il est entendu avec le client que le livrable final de l'application respectera les critères d'acceptabilité suivants :

- *L'application permet de réaliser adéquatement les objectifs nommés ci-haut.*
- *La conception de l'interface est ergonomique, professionnelle, simple à utiliser et s'accorde aux standards des applications que l'on rencontre habituellement dans l'environnement Windows.*
- *L'interface devra permettre un ajustement de la taille de la fenêtre principale selon le souhait de l'utilisateur tout en conservant une disposition fonctionnelle des composants.*
- *L'application exploitera adéquatement et sécuritairement l'API développée précédemment à l'aide d'une clé API.*
 - *Notamment en validant les données saisies avant d'effectuer l'envoi d'une requête à l'API.*
- *Affichage adéquat et clair de tous les messages d'erreur et de réussite des requêtes à l'API.*
 - *Par exemple : un utilisateur ayant atteint son nombre de requêtes maximal.*

SOLUTION A APPLIQUER

APPLICATION

L'application développée dans ce projet sera une ***application native avec une interface graphique.***

SPECIFICATIONS TECHNIQUES

Langage de développement

L'application sera développée en C# selon le modèle WPF .Net et en employant le cadriciel (*framework*) .Net 8.0. Les interfaces seront réalisées en XAML.

L'apparence visuelle de l'application sera réalisée à l'aide du *NuGet MaterialDesignThemes*.

Patron de conception

L'application sera développée selon l'architecture MVVM.

Sauvegarde des données

Les données seront récupérées et stockées dans une base de données via l'utilisation de l'API développée précédemment.

MODELISATION (PARTIELLE) DE LA SOLUTION

DIAGRAMME DE CAS D'UTILISATION

Le diagramme de cas suivant représente les fonctionnalités et actions principales de l'application à développer.

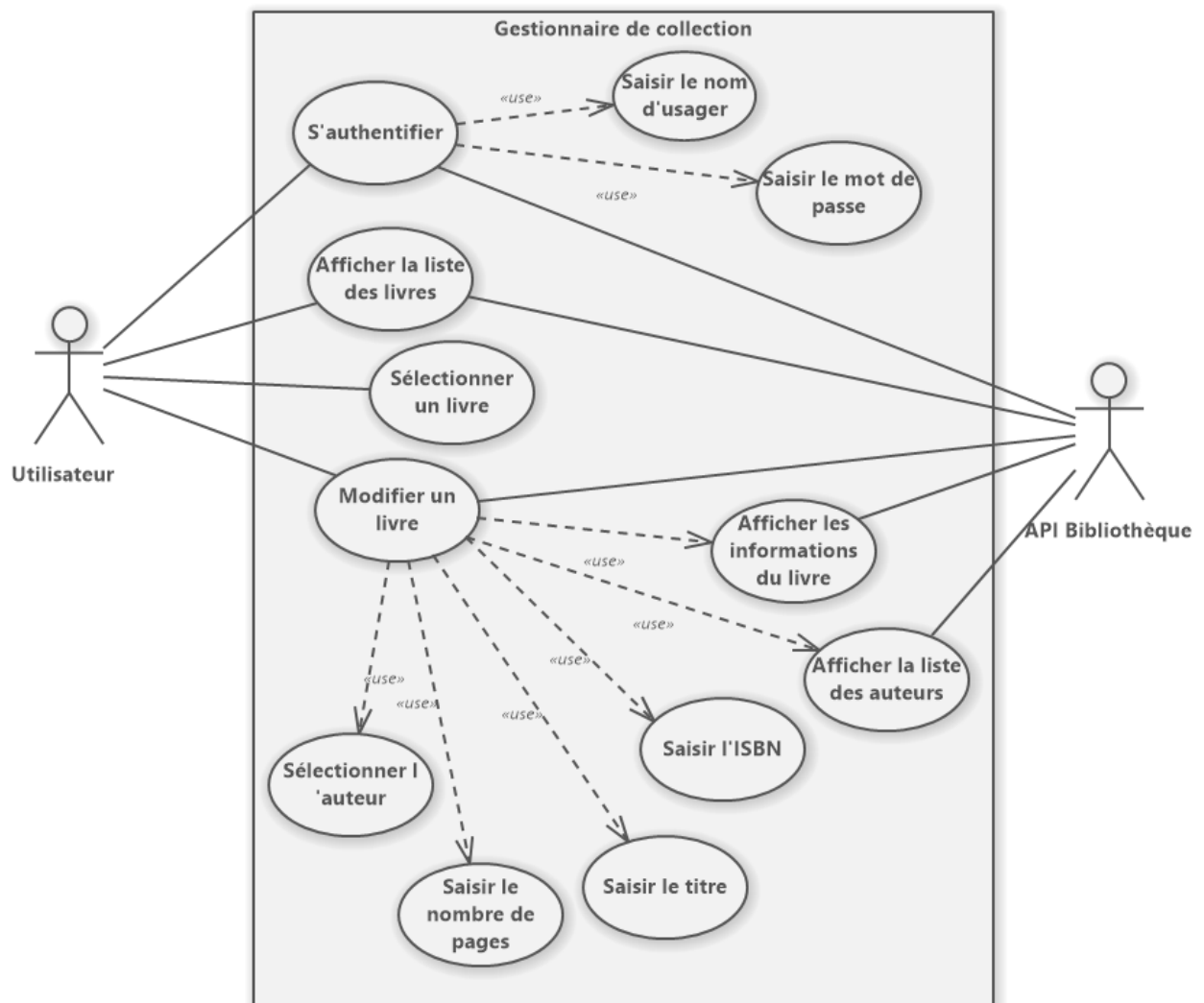


Figure 1 Diagramme de cas d'utilisation

MAQUETTES

Fenêtre principale

La fenêtre principale est dédiée à l'affichage des vues selon l'état du système et des messages à l'utilisateur à l'aide la barre de notification au bas de celle-ci.

L'état initial de l'application consiste à afficher la vue demandant à l'utilisateur de s'authentifier. Une fois l'authentification effectuée avec succès, la vue de modification des livres est affichée.

Bonus

Des points bonus seront attribués pour l'implémentation du choix de langue (français ou anglais) et de l'affichage en mode clair ou obscur de l'interface.

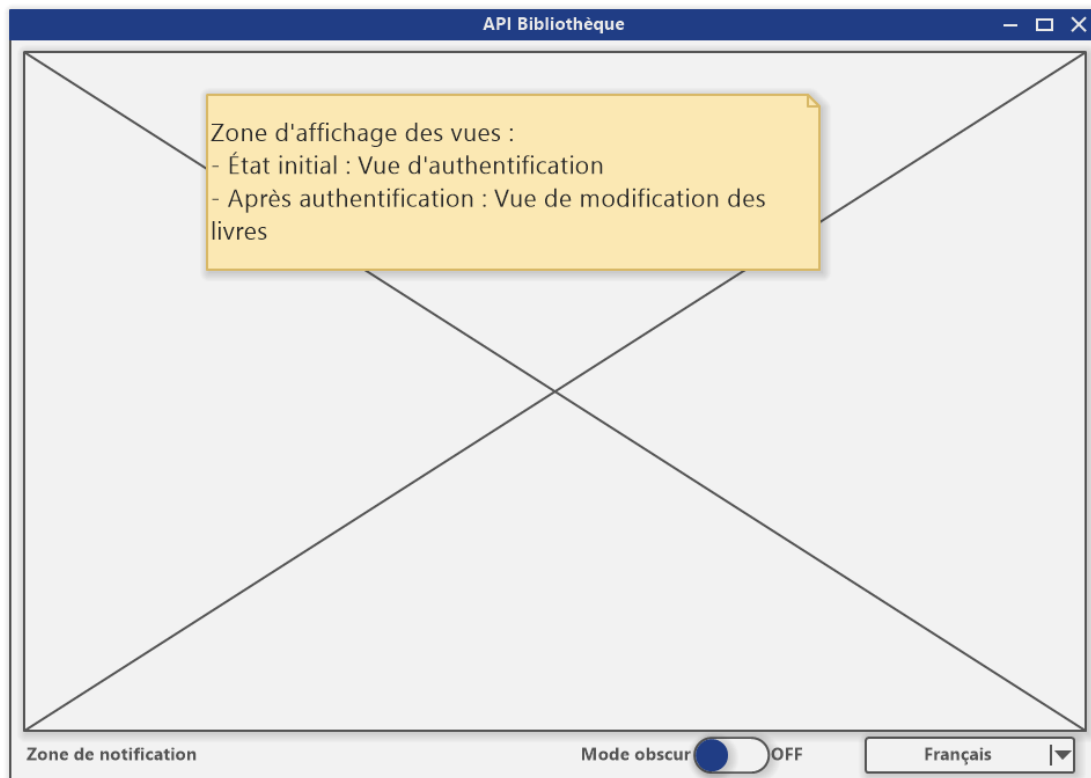


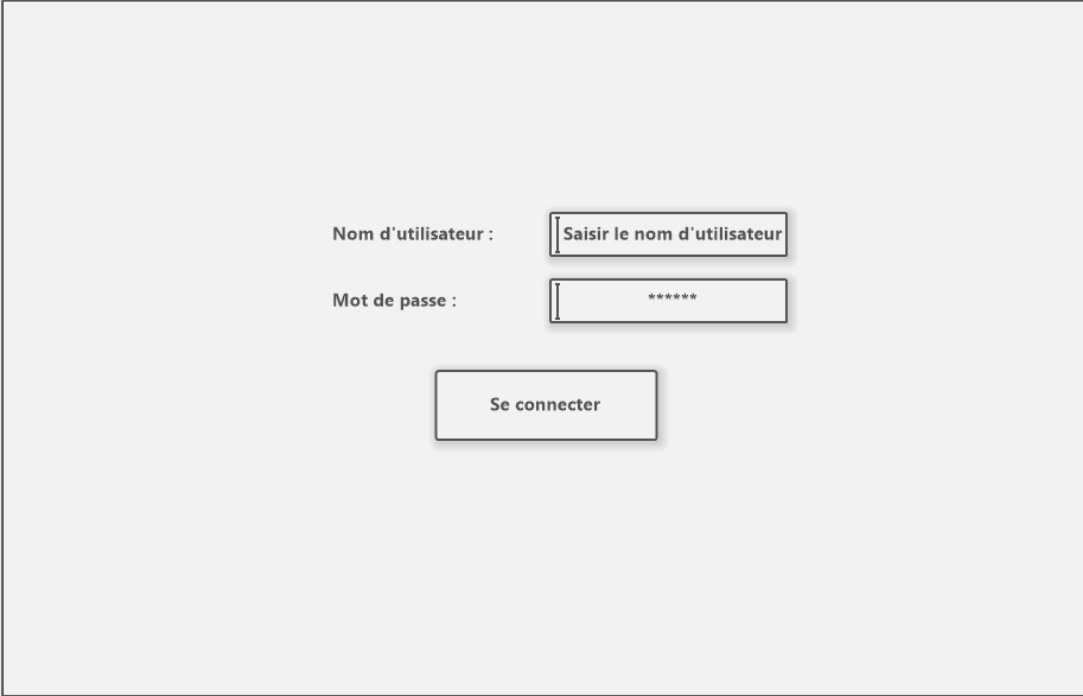
Figure 2 Maquette de la fenêtre principale

Vue d'authentification

La vue d'authentification doit permettre au client de s'authentifier et d'obtenir sa clé d'API.

Pour cette partie, vous devez créer un petit système de connexion avec un nom d'utilisateur et un mot de passe. Pour vous aider, du matériel utile est fourni dans un dossier joint nommé « Ressources » :

- *Le script de création de la table login membre avec des données ;*
- *La factory permettant de récupérer la clé d'API via le nom d'utilisateur et le mot de passe ;*
- *Le contrôleur LoginController permettant de retourner la clé d'API en utilisant la factory.*



The image shows a mockup of a user authentication form. It is centered on a light gray background. The form consists of three main elements: a label 'Nom d'utilisateur :' followed by a text input field containing the placeholder text 'Saisir le nom d'utilisateur'; a label 'Mot de passe :' followed by a password input field containing six asterisks '*****'; and a 'Se connecter' button located below the password field.

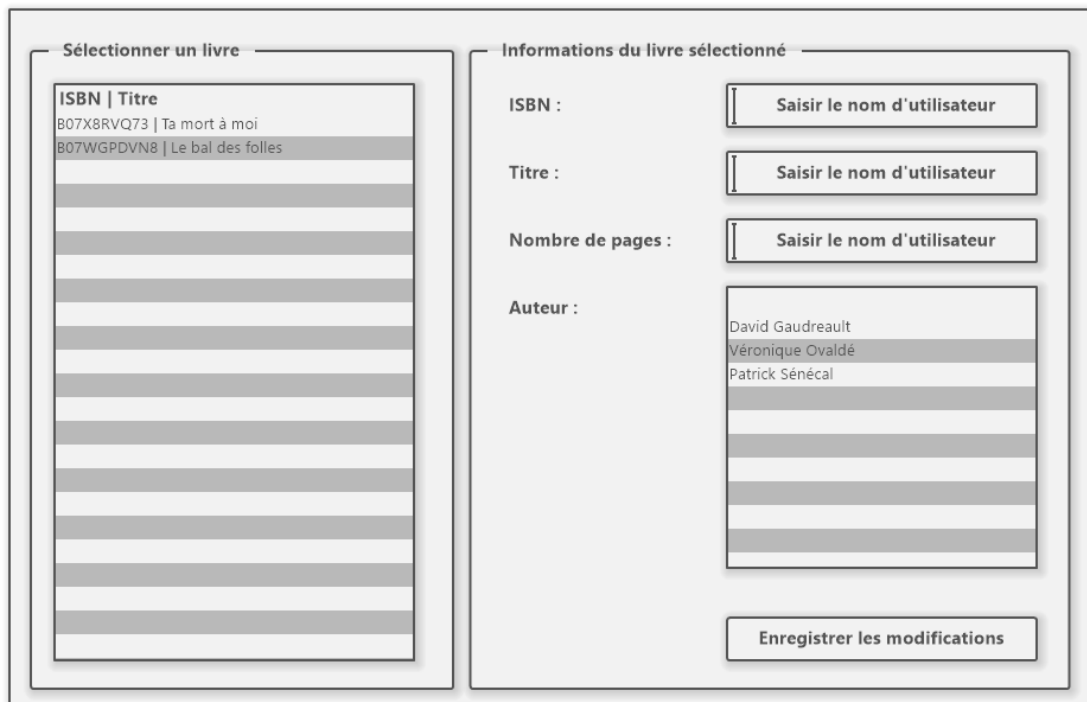
Figure 3 Vue de l'authentification de l'utilisateur

Vue de la modification des livres

Dans cette vue, l'utilisateur doit sélectionner un livre pour permettre l'accès à la partie d'édition des informations du livre sélectionné et du bouton d'enregistrement des données. Sinon, les éléments sont grisés.

Une fois l'enregistrement effectué, les données de la liste des livres doivent être mises à jour (en particulier si l'ISBN et le titre sont modifiés).

La liste doit afficher la liste des livres selon l'ordre alphanumérique de l'ISBN dans le sens ascendant.



Maquette de la modification des livres. L'interface est divisée en deux sections principales :

- Sélectionner un livre :** Une liste de livres avec les colonnes **ISBN** et **Titre**. Les premiers éléments sont :
 - B07X8RVQ73 | Ta mort à moi
 - B07WGPVDVN8 | Le bal des follesLes autres éléments de la liste sont grisés.
- Informations du livre sélectionné :** Des champs pour saisir les données du livre sélectionné :
 - ISBN :** Champ de saisie.
 - Titre :** Champ de saisie.
 - Nombre de pages :** Champ de saisie.
 - Auteur :** Liste déroulante avec les auteurs : David Gaudreault, Véronique Ovaldé, Patrick Sénécal. Les autres options sont grisées.

Un bouton **Enregistrer les modifications** est situé en bas à droite.

Figure 4 Maquette de la modification des livres

POURSUITE DE LA CONCEPTION

MAQUETTES

L'étudiant est *invité* à produire ses propres maquettes s'il désire apporter une approche différente sur le plan de l'interface. Toutefois, sa proposition doit respecter la demande du point de vue des fonctionnalités.

DIAGRAMME DE CLASSES

L'étudiant est *invité* à *produire* un diagramme de classes et à le remettre en même temps que l'application afin de l'aider à structurer sa pensée en vue de l'implémentation et de faciliter la compréhension du projet remis pour l'enseignant.

De plus, réaliser ce diagramme sera une excellente façon d'avoir une vue d'ensemble permettant de bien valider que les classes à développer seront cohérentes avec l'implémentation de votre API.

PLAN DE TESTS FONCTIONNELS

L'étudiant **doit** produire un plan de tests pour valider le bon fonctionnement de son application *avant* l'implémentation. Le plan de tests doit prévoir les scénarios nominaux (où tout fonctionne comme c'est prévu) et des scénarios pour tester comment l'application réagit à des situations problématiques. Quelques exemples : nom d'utilisateur ou de mot de passe invalide ; le serveur API ne fonctionne pas ; la BD n'est pas accessible ; la saisie de données invalides ; etc.

Prévoir l'anormal et valider que l'application récupère bien de ces situations.

REMISE DU TRAVAIL

DATE DE REMISE

Le projet doit être remis au plus tard le 2 mai 2025.

FORMAT DE LA REMISE

Sur LÉA, remettre un dossier compressé au format ZIP contenant :

- *Plan de tests fonctionnels (PDF)*
- *Solution nettoyée de l'application client*
- *Solution nettoyée du serveur*

Une seule solution contenant les projets client et serveur peut être remise plutôt que 2 solutions différentes.