

Projet 2020 de génie logiciel

Génie logiciel

Baptiste Pesquet, Sébastien Bertrand

Introduction	2
Répartition et conception	2
1.1. Répartition des tâches	3
1.2. Maquette	3
1.3. Démarche de conception de l'IHM	3
Documentation technique	5
2.1. Modélisation de données	5
2.2. Architecture technique	6
2.3. Processus de génération	6
2.4. Fonctionnalités et exigences respectées	6
2.5. Procédure de tests	13
Documentation utilisateur	15
3.1. Initialisation de la BDD	15
3.2. Fonctionnement de l'application	15
Utilisateur	15
Administrateur	17
Conclusion	17
Annexes	18
Diagramme des classes métiers	18
Modèle relationnel	19
Maquette	19

Introduction

Dans le cadre du projet de génie logiciel, nous avons dû mettre en place un système de gestion de collection de BD. Le but de ce projet est de mettre en application les connaissances acquises en génie logiciel ce semestre. Voici une présentation de ce que nous avons produit.

Nous étant trompées d'emplacement, nous commençons par vous fournir le lien du GitHub où nous avons travaillé si vous souhaitez voir l'évolution du projet :

https://github.com/ndlmariam/Projet_GL_Migeon_Ndiaye

1. Répartition et conception

Afin de répondre à la liste d'exigences posées, nous avons dû extraire et répartir les tâches nécessaires.

1.1. Répartition des tâches

Nous avons décidé de rassembler les tâches dans un tableau et de les prioriser en fonction de leur importance pour l'évolution du projet.

Voici notre répartition :

Tâche	Personne(s)	Priorité
Modéliser les données de la BDD	Mariam et Agathe	1
Réalisation de maquettes	Mariam et Agathe	1
Initialisation de NHibernate	Agathe	1
Initialisation des formulaires	Agathe	2
Design formulaire accueil	Mariam et Agathe	2
Design formulaire utilisateur	Agathe et Mariam	2
Design formulaire administrateur	Agathe	2
Code formulaire accueil	Mariam et Agathe	3
Code formulaire utilisateur	Mariam et Agathe	3
Code formulaire administrateur	Agathe	3
Mise en place des tests unitaires	Mariam	3
Rédiger le rapport	Mariam et Agathe	4

1.2. Maquette

Vous trouverez en annexe les [maquettes](#) réalisées en début de projet.

1.3. Démarche de conception de l'IHM

Afin de proposer une interface adaptée à l'utilisateur, nous avons suivi quelques recommandations. Nous nous sommes notamment appuyées sur les heuristiques de Nielsen.

Tout d'abord, nous avons tenté d'anticiper les erreurs de l'utilisateur, à la fois en prévenant les erreurs avant qu'elles n'arrivent mais aussi en fournissant les moyens de les réparer si elles se produisent.

Par exemple, lors de la création d'un album, nous affichons le boutons valider une fois que tous les champs sont remplis, afin qu'il ne se produise pas d'erreur.

Lorsque l'utilisateur se connecte avec un mauvais compte ou s'il tente de créer un compte avec un login déjà existant, un message d'erreur lui est renvoyé. De même, si par inadvertance l'utilisateur tente de se connecter à son compte utilisateur en utilisant son mot de pass administrateur, l'erreur lui est signifiée :



Figure 1 : Messages d'erreur connexion / création

Lors de la suppression d'un album de la liste de souhaits, un message s'assure de l'action que l'utilisateur a fait pour lui éviter une suppression involontaire (idem pour l'ajout aux albums).

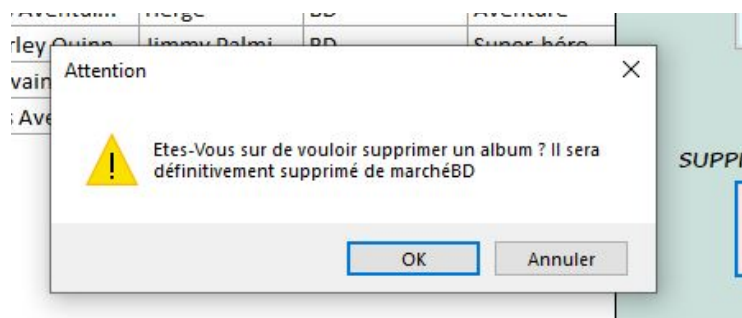


Figure 2 : Message de vérification de suppression

Ensuite, nous avons tenté de rendre le statut du système visible, c'est-à-dire offrir un feedback à l'utilisateur quand il effectue une tâche. Dès lors que l'utilisateur ajoute un album à ses souhaits ou sa liste d'album, un message de confirmation de l'ajout lui est affiché.

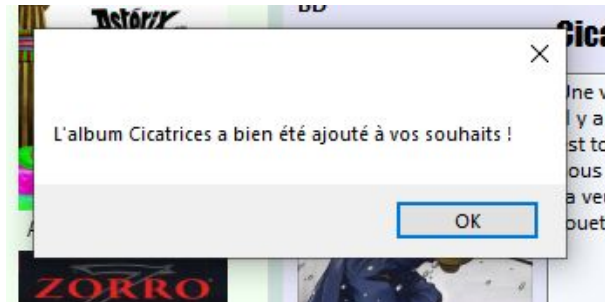


Figure 3 : Message de validation d'un ajout

De plus, lorsqu'un album est ajouté à la liste de souhaits, l'utilisateur en est informé lorsqu'il clique à nouveau sur ajouter aux souhaits et il ne peut plus le rajouter. De même lorsque l'utilisateur ajoute les albums à sa collection, ils s'effacent automatiquement de la liste des souhaits. Afin d'indiquer à l'utilisateur sur quelle page il se trouve nous avons mis en évidence les onglets correspondant (dans le header et le menu déroulant).

Ensuite nous avons décidé de faire correspondre notre système au monde réel afin de favoriser la compréhension de l'interface en s'appuyant sur des contextes et un lexique connus par l'utilisateur. En effet, nous avons utilisé le panier pour l'ajout à la liste des albums (comme un achat), le cœur pour la liste des souhaits, un bâtiment pour le marché et une sortie pour la déconnexion.

Enfin, nous avons favorisé la redondance d'informations concernant les onglets afin de permettre une plus grande accessibilité des pages. En effet, le marché, les albums et les souhaits sont présents à la fois dans le menu déroulant et dans le header.

2. Documentation technique

L'application que nous avons mise en place respecte toutes les exigences techniques. Voici une documentation technique afin de détailler comment fonctionne notre logiciel.

2.1. Modélisation de données

Afin de modéliser le fonctionnement de notre application, nous avons mis en place deux schémas de données : le diagramme des classes métier et le modèle relationnel.

Les modèles initiaux ont été amenés à évoluer car notre vision du problème à quelque peu changé durant le projet, les schémas finaux sont en annexe ([Diagramme des classes métier](#), [Modèle relationnel](#)).

Nous avons décidé d'utiliser 3 classes, Album, Personne et Action. Album représente les bd, comics et manga que l'on retrouve dans notre application. Un album est identifié par plusieurs attributs : son titre, sa série, sa couverture, sa catégorie, son genre, son éditeur et son ou ses auteur(s).

Personne est une classe abstraite dont les classes utilisateurs et administrateurs héritent. Un administrateur peut effectuer deux types d'actions : ajouter un album ou supprimer un album du marché. Un utilisateur peut lui effectuer 3 actions : ajouter un album ou supprimer un album de ses souhaits, et ajouter un album à sa collection.

Enfin, Action est une classe association permettant de faire le lien entre Album et Personne. Elle simplifie la modélisation étant donné que, même s'il peut se retrouver dans la collection de plusieurs personnes, un même album n'est présent qu'une seule fois dans la base de donnée. Il est lié par des actions différentes à des personnes différentes. Par ailleurs, une même personne peut effectuer plusieurs actions différentes sur un album. Ainsi, la classe action permet de facilement mettre en relation des albums et des personnes tout en gardant une trace de leurs choix. Elle permet notamment d'afficher un historique des actions pour les souhaits mais également dans la partie administrateur.

2.2. Architecture technique

Nous avons choisi d'utiliser une architecture en couche (App/DAL/Domain) ce qui inclut entre autres l'utilisation de Nhibernate avec le framework 4.6.1.

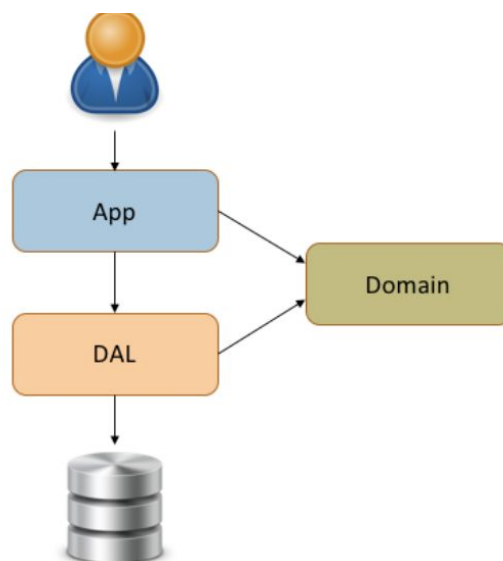


Figure 4 : Architecture

L'utilisateur a accès à l'application divisée en 3 formulaires : accueil, formulaire utilisateur et formulaire administrateur.

Les données rentrées par l'utilisateur sont enregistrées dans une base de données par l'intermédiaire de DAL qui stocke les informations dans les Repository.

Enfin la structure des classes métiers qui contiennent tous les attributs nécessaires sont gérées par la partie Domain. Domain échange à la fois avec App et DAL.

2.3. Processus de génération

Afin d'être en mesure de générer le projet, il faut d'abord initialiser la base de données, les explications correspondantes sont apportées dans la partie [documentation utilisateur](#).

2.4. Fonctionnalités et exigences respectées

L'application mise en place respecte les onze exigences métier données à l'initialisation du projet.

Nous avons décidé d'apporter quelques modalités en plus :

EF_12 : en tant qu'administrateur je peux supprimer un album du marché s'il est erroné.

EF_13 : en tant qu'utilisateur je peux ajouter manuellement un album qui n'est pas présent dans le marché à ma liste

Voici l'ensemble des exigences respectées par l'application :

Page de connexion

Première page lancée où la personne peut se créer un compte (utilisateur ou administrateur) ou se connecter à son compte. Nous avons choisi d'afficher le bouton de validation que lorsque tous les champs sont remplis.

Cette page permet de mettre en place les exigences, EF_01 et EF_10 , de connexion à l'application avec login et mot de passe.

The figure displays two side-by-side screenshots of a web application interface for user management. Both screens have a light blue header with the text 'Veuillez vous connecter ou créer un compte'.

The left screenshot, titled 'Informations de connexion', features a green 'Se connecter' button and a grey 'Créer un compte' button. Below the buttons, there are radio buttons for 'Utilisateur' and 'Administrateur'. The 'Login' field is a text input, and the 'Mot de passe' field is a password input.

The right screenshot, titled 'Création de compte', features a grey 'Se connecter' button and an orange 'Créer un compte' button. Below the buttons, there are radio buttons for 'Utilisateur' and 'Administrateur'. The 'Login' field is a text input, the 'Mot de passe' field is a password input, and the 'Pseudo' field is a text input.

Figure 5 : Affichage création et connexion compte

Partie utilisateur

Header

Affiche le pseudo de l'utilisateur et lui permet d'aller sur le marché, d'afficher sa liste de souhaits, sa liste d'albums, d'ajouter l'ensemble de ses souhaits à sa liste d'albums ou de se déconnecter. Cela valide les exigences EF_02, EF_07, EF_09.



Figure 6 : Header utilisateur

Page d'ajout manuel

Cette page correspond à la page d'ajout manuel d'un album à sa collection. Dès lors que l'utilisateur a rempli tous les champs, ci-dessous, le bouton de validation apparaît. L'album ajouté à la liste n'est présent que pour l'utilisateur et ne s'ajoute pas au marché. Nous avons ajouté cette fonctionnalité, EF_13, dans le cas où un album possédé ne serait pas présent sur le marché.



Mes Albums

Mes Albums

Albums
Marché
Souhaits

Bienvenu(e) dans votre gestionnaire de BD. Vous pouvez déclarer manuellement ci dessous les albums appartenant à votre collection. Vous pouvez aussi en ajouter rapidement à partir de la liste situer dans le marché. Si vous souhaitez consulter la liste des albums que vous possédez rendez-vous dans l'onglet Albums du menu déroulant !

Veuillez entrer les caractéristiques de l'album

Titre *

Auteur(s) *

Série *

Editeur *

Catégorie *

Genre *

Couverture

Résumé *

* champs obligatoires

Figure 7 : Affichage page d'ajout manuel

Page d'accueil - Page du marché

Cette page permet à l'utilisateur de consulter l'ensemble des albums du marché et peut effectuer des recherches à partir d'un mot clé (titre, auteur, éditeur). Il peut cliquer sur le titre d'un album pour en obtenir les informations (titre, auteur, éditeur, résumé ...), l'ajouter à ses souhaits ou encore l'ajouter à la liste de ses albums.

Cette page permet de mettre en place les exigences, EF_03, EF_04, EF_05 et EF_06.



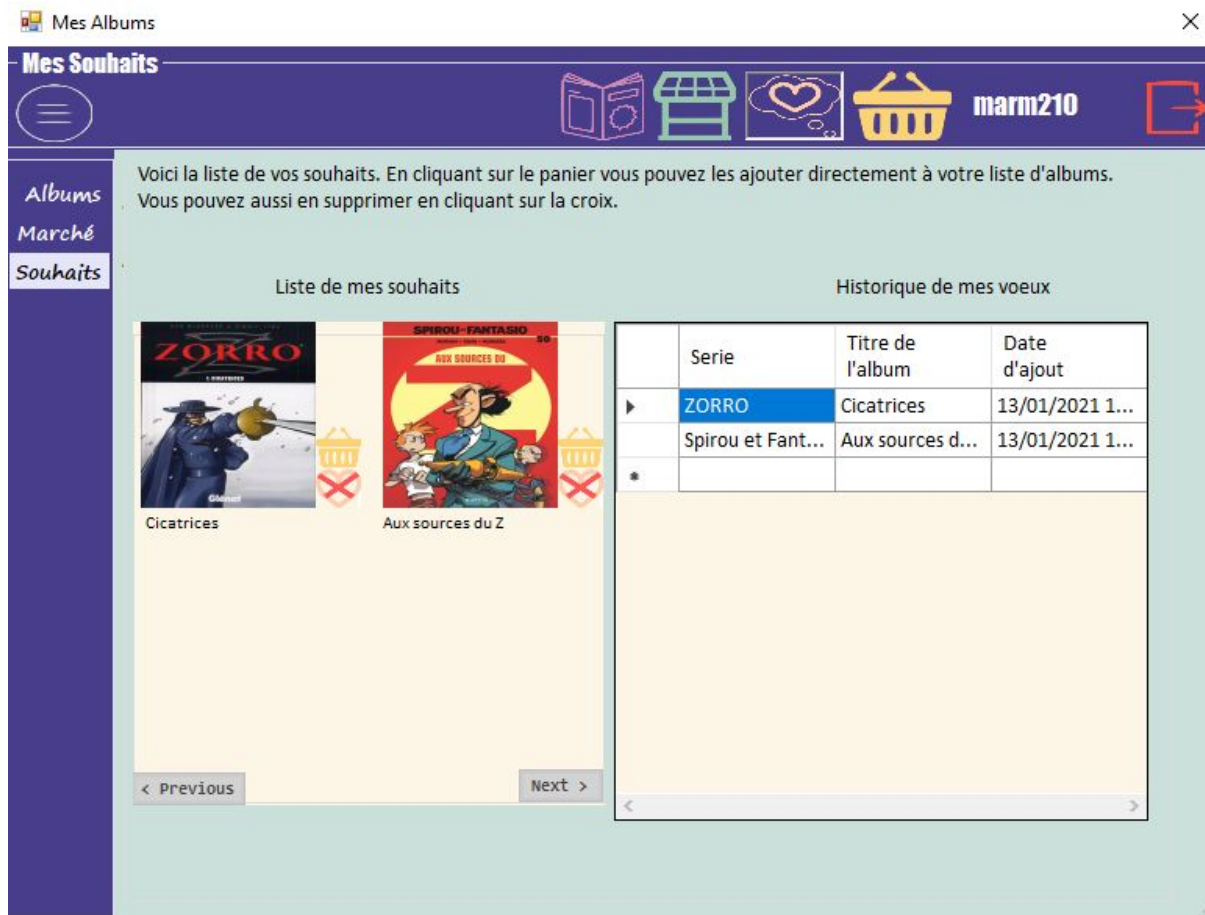
Figure 8 : Affichage page du marché



Figure 9 : Affichage page du marché avec recherche "dupuis" (éditeur) et informations

Page liste des souhaits

Cette page permet à l'utilisateur de consulter la liste de ses souhaits. Il peut enlever des albums de sa liste ou les ajouter à la liste de ses albums. Il peut à tout moment ajouter tous ses souhaits à sa liste d'album en cliquant sur le panier présent dans le header. Cette page permet de compléter les exigences EF_07 et EF_08.



Mes Albums

Mes Souhaits

Albums
Marché
Souhaits

Voici la liste de vos souhaits. En cliquant sur le panier vous pouvez les ajouter directement à votre liste d'albums. Vous pouvez aussi en supprimer en cliquant sur la croix.

Liste de mes souhaits

Historique de mes vœux

	Serie	Titre de l'album	Date d'ajout
▶	ZORRO	Cicatrices	13/01/2021 1...
	Spirou et Fant...	Aux sources d...	13/01/2021 1...
*			

< Previous

Next >

Figure 10 : Affichage page liste de souhaits

Page liste des albums

Cette page permet à l'utilisateur de consulter la liste de ses albums. Il peut aussi, en cliquant sur le titre de l'album, lire les informations correspondantes. Elle met en place l'exigence EF_02.

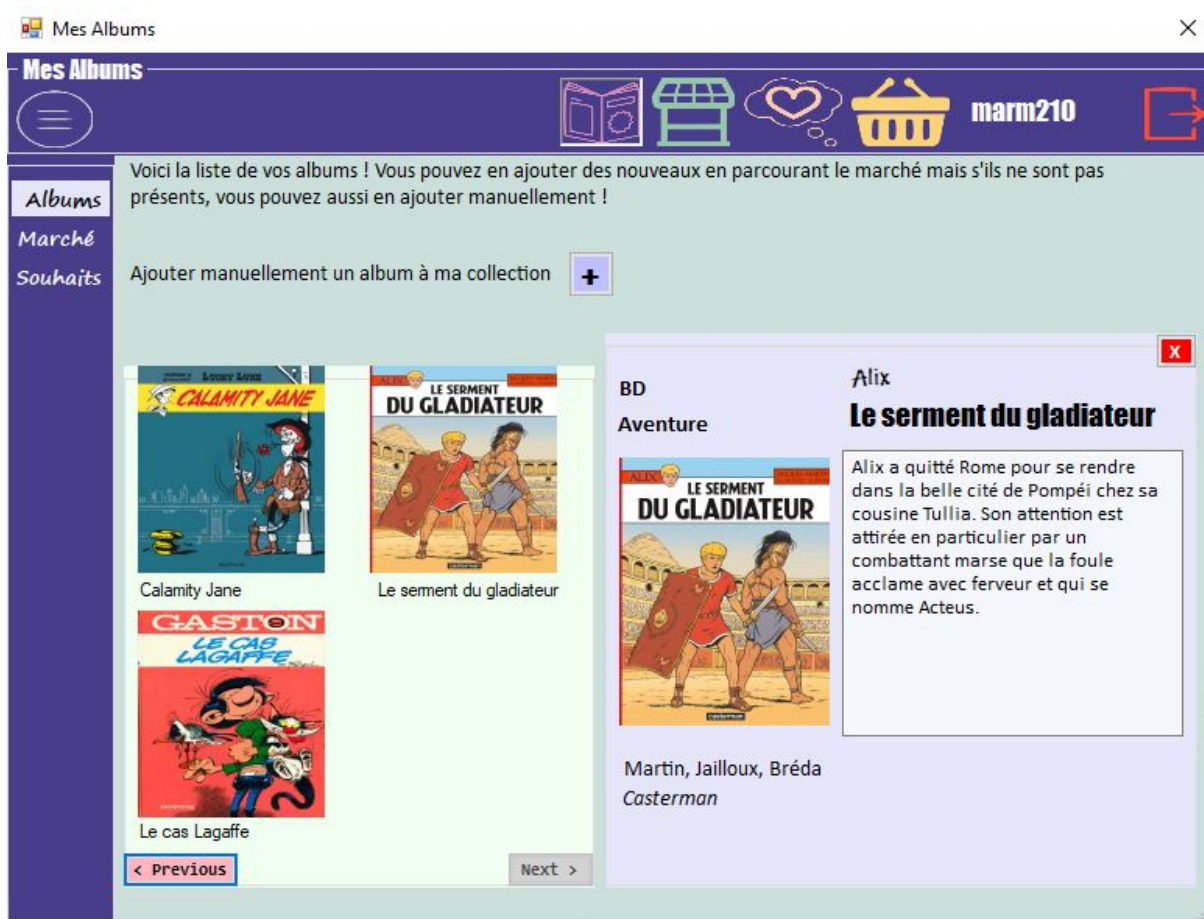


Figure 11 : Affichage page liste des albums et informations

Partie administrateur

Header

Affiche le pseudo de l'administrateur et lui permet de revenir sur la page d'accueil et de se déconnecter. Cela valide l'exigence EF_09.



Figure 12 : Header administrateur




Page d'accueil administrateur

Cette page détaille l'ensemble des albums présents dans le marché. L'administrateur a la possibilité d'ajouter un nouvel album au marché ou d'en supprimer un s'il considère que des informations sont erronées par exemple. Cette page permet d'appliquer les exigences EF_11 et EF_12.

CircuitBD espace Administrateur

Espace Administrateur

Bienvenu(e) sur MarchéBD, voici l'ensemble des albums du marché.

Couverture	Titre	Série	Auteur	Editeur	Catégorie	Genre	Résumé
	Tintin au Tibet	Les Aventures de Tintin	Hergé	Casterman	BD	Aventure	Un avion de ligne duquel le jeune Tchang se rend s'est écrasé dans le Tibet. Tintin, son ami, va à sa recherche.
	Folle à lier	Harley Quinn	Jimmy Palmiotti, Amanda Conner, Chad Hardin	Urban Comics	BD	Super-héros	Un jour Power Girl s'écrase aux côtés de Harley Quinn. Frappée par une puissante Kryptonienne, elle retrouve malgré tout sa mission dans les missions abracadabrantes de sa coéquipière au sein de la Justice.
	La ferme abandonnée	Sylvain et Sylvette	Jean-Louis Pesch	Dargaud	BD	Humour	Ces récits pour enfants relatent les aventures de Sylvain et Sylvette, deux enfants d'environ 10 ans, le frère et la sœur.

AJOUTER un album

SUPPRIMER un album

Figure 13 : Affichage accueil administrateur




Page d'ajout d'album

Cette page est équivalente à celle de l'ajout manuel dans la partie utilisateur. Elle permet de valider l'exigence EF_11 qui est d'ajouter des albums au marché.

Page de suppression d'album

Cette page permet de supprimer un album du marché et de vérifier l'exigence EF_12.

Bienvenu(e) sur MarchéBD, voici l'ensemble des albums du marché.

Couverture	Titre	Série	Auteur	Editeur	Catégorie	Genre	Résumé
	Tintin au Tibet	Les Aventures de Tintin	Hergé	Casterman	BD	Aventure	Un avion de ligne duquel le jeune Tchang se rend s'est écrasé dans le Tibet. Tintin, son ami, va à sa recherche.
	Folle à lier	Harley Quinn	Jimmy Palmiotti, Amanda Conner, Chad Hardin	Urban Comics	BD	Super-héros	Un jour Power Girl s'écrase aux côtés de Harley Quinn. Frappée par une puissante Kryptonienne, elle retrouve malgré tout sa mission dans les missions abracadabrantes de sa coéquipière au sein de la Justice.
	La ferme abandonnée	Sylvain et Sylvette	Jean-Louis Pesch	Dargaud	BD	Humour	Ces récits pour enfants relatent les aventures de Sylvain et Sylvette, deux enfants d'environ 10 ans, le frère et la sœur.

AJOUTER un album

SUPPRIMER un album

Entrez le titre de la BD à supprimer

titre

Suppression

Figure 14 : Affichage suppression

2.5. Procédure de tests

Nous avons mis en place différents tests automatisés, couvrant les parties “accès aux données” (DAL) et “classes métier” (Domain), qui ne provoquent pas d’erreurs.

Attention l’exécution des tests effacera les nouvelles données que vous avez pu entrer dans la base de données, cette dernière sera remise à l’état initial (qui comprend 18 albums).

Fonction test / fichier	Utilité
TestRepository.cs	Permet d’initialiser les données de tests et la BDD (insertion de différentes valeurs dans les tables album, action et personne).
Fonctions de AlbumRepository	
TestAlbumRepo_GetAll()	Vérifie que la fonction GetAll() récupère bien tous les albums du test, dans le cas présent il y en a 24.
TestAlbumRepo_SaveNew()	Vérifie que la fonction Save(Album album) enregistre bien un nouvel album créé. Dans notre cas, on vérifie qu’il y a bien un seul album intitulé “A l’aube d’une grande aventure” qui est celui qu’on vient de créer et qui n’est pas encore présent.
TestAlbumRepo_SaveExisting()	Vérifie que la fonction Save(Album album) enregistre bien les modifications apportées sur album existant.
TestAlbumRepo_GetAlbumByActionID()	Vérifie que la fonction GetAlbumByActionID(int id) retrouve bien le bon album en le sélectionnant avec son identifiant.
TestAlbumRepo_GetAlbumByTitle()	Vérifie que la fonction GetAlbumByTitle(string title) retrouve bien le bon album en le sélectionnant avec son titre.
TestAlbumRepo_GetAlbumByRecherche()	Vérifie que la fonction GetAlbumByRecherche(string motcle) renvoie bien une liste d’Albums qui contient “motcle” soit dans son titre, nom d’auteur, éditeur, genre, série ou catégorie. Dans l’exemple, on vérifie qu’il y a bien deux albums liés au mot clé “Hergé”.
TestAlbumRepo_DeleteAlbum()	Vérifie que la fonction DeleteAlbum(Album album) supprime bien l’album du repository.
Fonctions de PersonneRepository	
TestPersonneRepo_GetAll()	Vérifie que la fonction GetAll() récupère bien toutes les personnes du test, dans le cas présent il y en a 4.
TestPersonneRepo_SaveNew()	Vérifie que la fonction Save(Personne pers) enregistre bien une nouvelle personne créée. Dans notre cas, on vérifie qu’il y a bien une seule personne dont le nom est Paul qui est celle qu’on vient de créer et qui n’est pas encore présente.
TestPersonneRepo_SaveExisting()	Vérifie que la fonction Save(Personne pers) enregistre bien les modifications

	apportées sur une personne déjà présente.
TestPersonneRepo_CompareMdp()	Vérifie que la fonction CompareMdp(string login, string mdp) permet de renvoyer si le mot de passe et le login entrés correspondent bien. Ici on teste avec des valeurs présentes dans les tests.
TestPersonneRepo_TrouverPersonne()	Vérifie que la fonction TrouverPersonne(string login, string mdp, string type) permet bien de renvoyer une personne en l'identifiant par son login, son mot de passe et son type.
TestPersonneRepo_PresentBDD()	Vérifie que la fonction PresentBDD(string login) repère bien si un login a déjà été utilisé ou non.
Fonctions de ActionRepository	
TestActionRepo_GetAll()	Vérifie que la fonction GetAll() récupère bien toutes les actions du test, dans le cas présent il y en a 27.
TestActionRepo_SaveNew()	Vérifie que la fonction Save(Action action) enregistre bien une nouvelle action créée. Dans notre cas, on vérifie qu'il y a bien une seule action intitulée "AjouterSouhait" qui est celle qu'on vient de créer et qui n'est pas encore présente.
TestActionRepo_SaveExisting()	Vérifie que la fonction Save(Action action) enregistre bien les modifications apportées sur une action existante.
TestActionRepo_DeleteAction()	Vérifie que la fonction DeleteAlbum(Action action) supprime bien l'action du repository.
TestActionRepo_GetActionByNameAndAlbum()	Vérifie que la fonction GetActionByNameAndAlbum(Album album, string nom) retrouve bien l'action en fonction de son nom et d'un album.

Dans chacun des RepositoryTest.cs il y a un test d'initialisation permettant d'affecter les premières variables nécessaires.

Les tests de Domain sont simplement des vérifications du fonctionnement des différents constructeurs.

3. Documentation utilisateur

3.1. Initialisation de la BDD

Nous vous proposons plusieurs méthodes pour initialiser la base de données permettant de contenir à la fois les informations sur les albums, mais aussi les comptes et les actions des utilisateurs.

Pour initialiser la bdd , vous devez d’abord :

1. **Lancer MySQL sur xampp**
2. Créer une base de données appelée ***bdtheque***.
3. Ajouter un privilège, avec les informations suivantes : **login = AdminBD ; mot de passe = BDmdp ; nom d’hôte = localhost.**

Vous pouvez ensuite **choisir la méthode qui vous convient le mieux parmi celles-ci :**

Méthode 1 : initialisation grâce aux tests

4. A partir de visual studio, exécutez tous les tests. Ce lancement des tests va automatiquement remplir la base grâce au fichier TestRepository.cs

Méthode 2 : initialisation grâce au script sql

4. Dans le dossier contenant le projet vous trouverez un fichier nommé bdtheque.sql : Vous pouvez soit l’ouvrir dans un éditeur de texte et copier son contenu dans l’onglet sql de votre base bdtheque, soit importer directement ce fichier votre base bdtheque avec l’onglet importer.

Méthode 3 : initialisation grâce aux scripts sql de la partie DAL

4. Vous pouvez autrement importer, directement dans la base, les fichiers présents dans le dossier DB de DAL.

Une fois ces 4 étapes effectuées, votre base de données est initialisée et prête à être utilisée. L’application peut fonctionner.

3.2. Fonctionnement de l’application

Voici une documentation vous expliquant le fonctionnement de notre application de gestion des bandes dessinées. Vous avez la possibilité de vous créer un compte utilisateur ou administrateur.


Utilisateur

Connexion ou création de compte

A partir de la page d’accueil vous pouvez créer votre compte, ou vous connecter, si vous en avez déjà un, en tant qu’utilisateur. Veillez à rentrer tous les champs.

Accès au marché

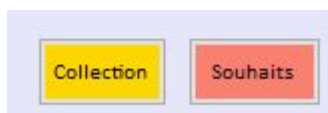
Une fois connecté(e), vous êtes directement dirigé(e) sur le marché.

Vous pouvez aussi accéder aux albums du marché en cliquant sur l'icône  ou à partir du menu déroulant en cliquant sur "marché".


Vous pouvez consulter l'ensemble des albums et affiner vos recherches avec un mot clé, par exemple un nom d'auteur ou un éditeur.



Cliquez sur le titre d'un album pour en connaître les informations, l'ajouter à votre liste de souhaits ou l'ajouter directement à vos albums (collection).



Accès à la liste des souhaits


Vous pouvez accéder à votre liste de souhaits en cliquant sur l'icône  ou sur "Souhaits" dans le menu déroulant.

Vous pouvez consulter l'ensemble de vos souhaits, en supprimer ou en ajouter (à l'unité) à votre liste d'albums.

Suppression  Ajout 

A partir du panier présent dans le header vous pouvez ajouter directement tous vos souhaits à votre liste d'albums.

Accès à la liste des albums

Vous pouvez accéder à votre liste d'albums en cliquant sur l'icône  ou sur "Albums" dans le menu déroulant.

Comme dans le marché vous pouvez obtenir les informations d'un album en cliquant sur son titre. Vous pouvez aussi ajouter un album manuellement s'il n'est pas présent dans le marché en cliquant sur le bouton plus.

Ajout manuel d'un album

Cette fonctionnalité peut être utile si votre album n'est pas présent sur le marché. Il vous suffit simplement de remplir tous les champs indiqués et de valider.

Administrateur

Connexion ou création de compte


A partir de la page d'accueil vous pouvez créer votre compte, ou vous connecter, si vous en avez déjà un, en tant qu'administrateur. Veillez à rentrer tous les champs.

Accès aux albums du marché

Une fois connecté(e) vous êtes dirigé(e) sur la liste des albums du marché. Cette page est accessible en cliquant sur l'icône suivante.



Ajout d'un album

Cliquez sur le bouton  si vous souhaitez créer un nouvel album pour le marché. Ensuite remplissez tous les champs du formulaire présenté et validez.

Suppression d'un album

Si un album possède des informations incorrectes vous pouvez le supprimer du marché en cliquant sur le bouton de suppression.



Il vous suffit ensuite d'entrer le titre de l'album, à le supprimer et valider.

Conclusion

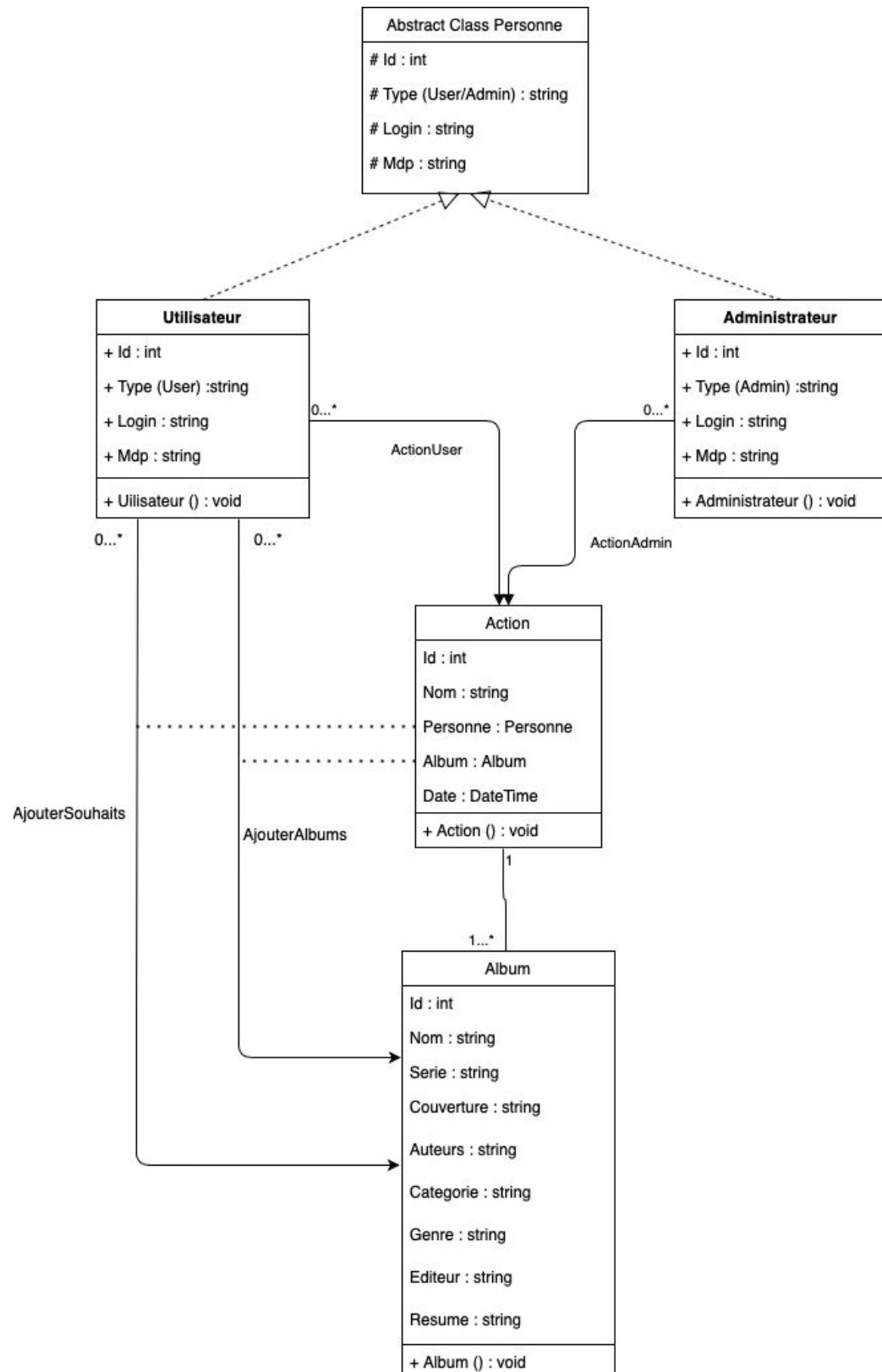
Malgré les conditions de travail actuelles dues à la crise sanitaire, nous avons tout de même réussi à maintenir une bonne communication pour nous permettre d'avancer correctement dans le projet.

Nous considérons ce projet réussi car nous avons pu mettre en pratique les notions vues en cours et les appliquer pour répondre aux exigences du problème.

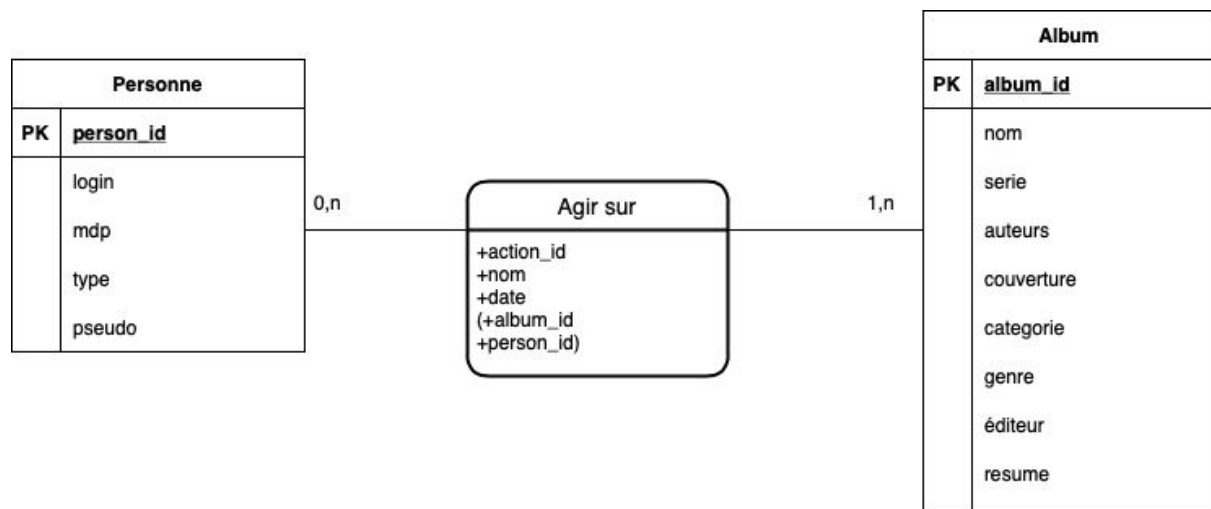
Néanmoins, dans un projet il y a toujours des points d'amélioration. Notamment, nous aurions pu augmenter le nombre de vérifications lors de l'ajout d'un album (si déjà existant ou non) ou lors d'une suppression (pas seulement le titre). L'affichage peut être amélioré, par exemple pour les titres des albums qui pour l'instant chevauchent les autres titres s'ils sont trop longs. Enfin, il est possible d'améliorer la partie administrateur, qui a été moins développée que celle utilisateur, pour la rendre plus agréable d'utilisation.

Annexes

Diagramme des classes métiers



Modèle relationnel



Maquette

Accueil de marchéBD

Bienvenu(e) sur MarchéBD, votre gestionnaire d'albums

C'est ici que vous pouvez acheter et consulter toutes les bandes dessinées que vous désirez. C'est très simple il vous suffit de vous créer un compte utilisateur et de vous faire plaisir !

Si jamais vous souhaitez, à l'inverse, ajouter un album dans le marché, créez vous un compte administrateur et le tour est joué !

Se connecter

Créer un compte

Informations de connexion

Vous êtes : ☐ Utilisateur ☐ Administrateur

Login :

Mot de passe :

Valider

Création de compte

Vous êtes : ☐ Utilisateur ☐ Administrateur

Login :

Mot de passe :

Valider

 Mes Albums

 Nom 

Mes albums

	<u>Titre, auteur</u>
	<u>Titre, auteur</u>
	<u>Titre, auteur</u>
	<u>Titre, auteur</u>

Voici une liste de toutes les bandes dessinées que vous avez achetées.





Vous pouvez aussi consulter les albums de votre liste de souhait en cliquant sur le cœur ou en rechercher des nouveaux en accédant au marché via la maison !

 **Marché**   **Nom** 



Bienvenue dans marchéBD

Amoureux des bandes dessinées voici notre liste d'albums. Un clic et la BD est à vous ! Vous hésitez ? Pas de soucis, ajoutez l'album à votre liste de souhaits ! Pour plus d'informations sur l'album cliquez sur le titre.

	<u>Titre, auteur, prix</u>	<input type="button" value="Acheter"/>	 Ajouter aux souhaits
	<u>Titre, auteur, prix</u>	<input type="button" value="Acheter"/>	 Ajouter aux souhaits
	<u>Titre, auteur, prix</u>	<input type="button" value="Acheter"/>	 Ajouter aux souhaits

 **Mes Souhaits**   **Nom** 

Mes souhaits

	<u>Titre, auteur, prix</u>	<input type="button" value="Acheter"/>
	<u>Titre, auteur, prix</u>	<input type="button" value="Acheter"/>
	<u>Titre, auteur, prix</u>	<input type="button" value="Acheter"/>
	<u>Titre, auteur, prix</u>	<input type="button" value="Acheter"/>

Voici une liste de toutes les bandes dessinées qui vous ont tapé dans l'oeil. Pour découvrir ce qu'elles renferment cliquez simplement sur acheter !

Vous pouvez également continuer vos recherches !