

Projet :
Gestion de livres
Dans une école



-Nom des participants :
Yassine AITALLA
Sofiane ZAION

TABLE DES MATIERES

Section 1 Expression des besoins

1.1 Liste des versions.....	3
1.2 But du document.....	3
1.3 Charte graphique.....	4
1.4 Cahier des charges.....	5
1.5 Planning.....	6

Section 2 Spécification fonctionnelle

1.6 Maquettes.....	8-12
--------------------	------

Section 3 Spécification mécanique

1.7 Conception	12-22
----------------------	-------

Section 5 Test

1.8 Tests unitaires.....	22
--------------------------	----

Section 3 Mise en production

1.9 Conversion en exe.....	23-24
----------------------------	-------

Section 1

Expression des besoins

1.1 Liste des versions

Version	Date de la version	Libellé
1.3	26/06/2022	Cahier des charges
1.4	25/07/2022	Diagramme UML (use case)
1.6	29/07/2022	Diagramme de classe

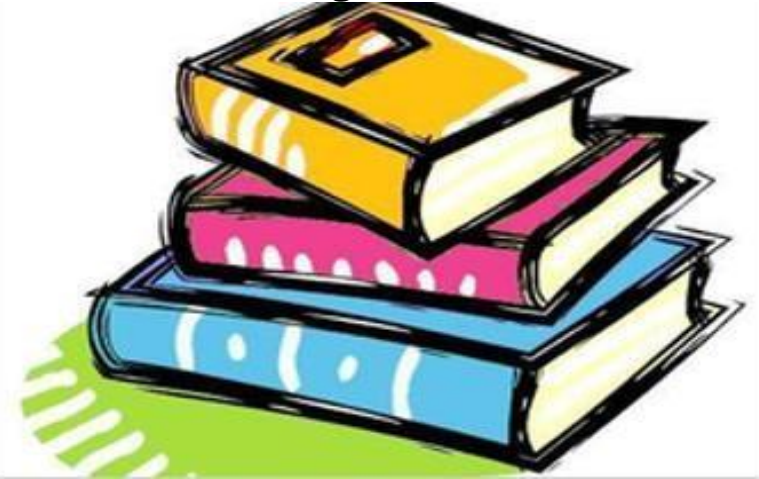
1.2 But du document



- Le projet sur lequel nous avons décidé de nous baser mon camarade ainsi que moi est le suivant :
 - **Gestion de livres dans une école.**
- L'application Bibyaso va permettre aux bibliothèques de gérer les emprunts de leurs ouvrages. Elle sera, dans un premier temps, destinée aux écoles de type primaire, collège, lycée, facultés et dans l'avenir, à plus grande échelle, pour tout type d'école.

1.3 Charte graphique

- Logo :



- Nb pages : 7 ;

- Langue : Française ;

- Couleur principale :

Aperçu de l'orange	Code HEXA
	#ff7f00

- Couleur secondaire :

Aperçu du vert	Code HEXA
	#00FF00

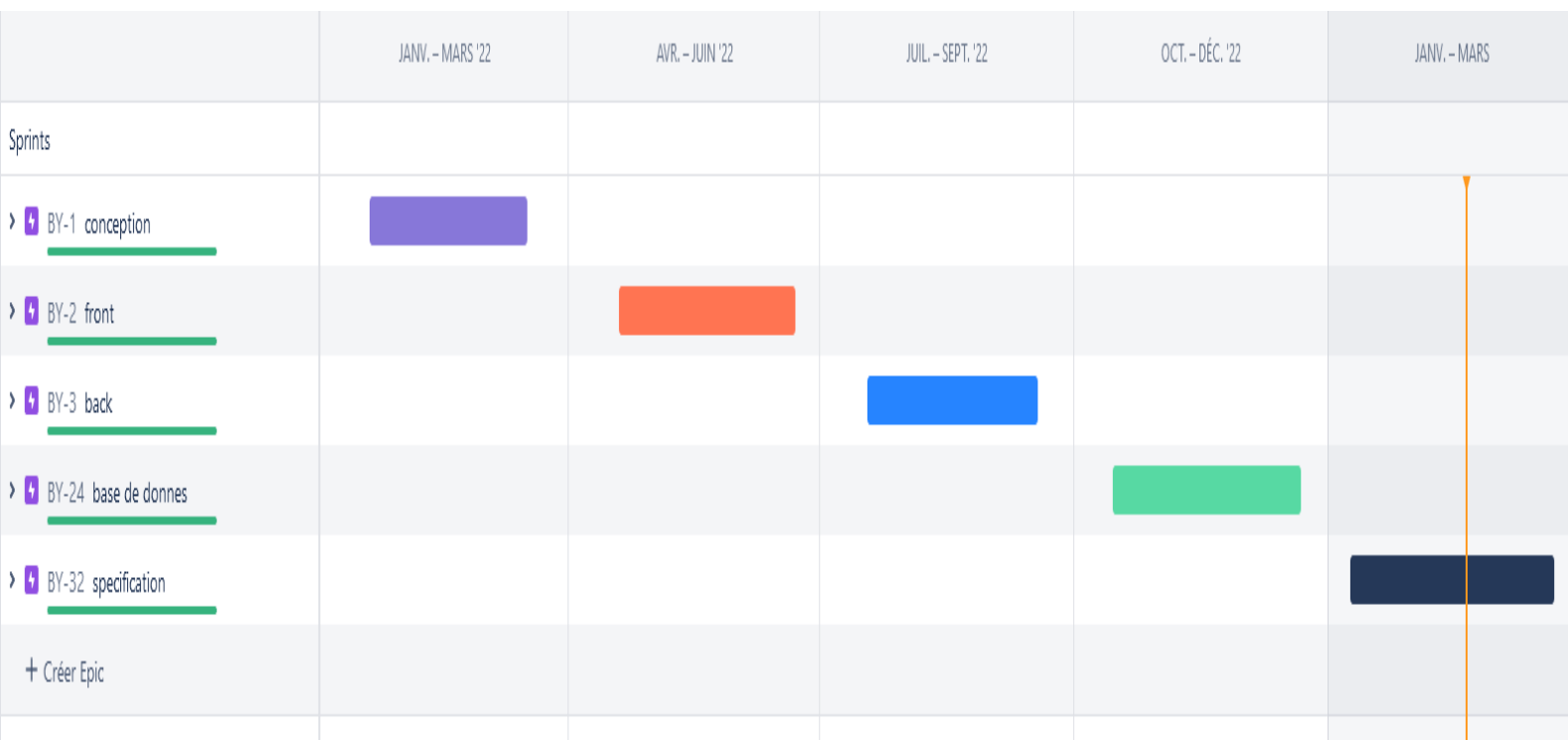
1.4 Cahier des charges

<i>Quel est notre objectif ?</i>	<i>Qui sont nos utilisateurs ?</i>	<i>Qu'attend t'en-t-il de nous ?</i>	<i>Les objectifs du projet</i>	<i>Demande du client (NOUS)</i>	<i>Conditions</i>
<ul style="list-style-type: none"> - Notre objectif est de satisfaire les besoins du client (Nous). - Nous Permettre une bonne gestion de notre application (BibYaso). 	<ul style="list-style-type: none"> -les clients qui utiliseront l'application sont : - Nous <p><u>-Avantage :</u></p> <p>Ne plus utilisé de feuille pour gérer les emprunts mais plutôt une application simple à l'utilisation.</p>	<ul style="list-style-type: none"> -Une application monotone pratique à l'utilisation -Plusieurs choix pour un maximum de plaisir concernant vos lectures. 	<ul style="list-style-type: none"> -Construire de manière collective ce projet pour pouvoir atteindre le résultat final ; -Notre application est qualifiée de plateforme bien structurer et fluide à l'utilisation. Permettre une bonne gestion de notre plateforme 	<ul style="list-style-type: none"> - Il faut s'identifier pour accéder aux fonctions de l'outil. - Qui contient 7 pages <ul style="list-style-type: none"> - Connexion - Gestion livres - Nos emprunteurs - Ajout d'emprunteurs - Gestion prêts (qui affiche un tableau des prêts effectués par nos emprunteurs) - Gestion des emprunts (qui permet de récupérer nos emprunteurs inscrits dans notre outil de gestion pour ensuite effectuer un emprunt) -Qui permet de référencer des nouveaux livres, auteurs, collections, éditeurs. -Qui permet de référencer des emprunts pour l'emprunt de Livre. -Qui permet de gérer l'emprunt, la restitution d'un livre. -Qui montre la liste des livres empruntés. <ul style="list-style-type: none"> - Qui affiche la date de l'emprunt de livre. - Qui affiche 4 Bouton Image -Gestion de livres -Emprunteur -Gestion des prêts -Deconnexion 	<p>.</p> <p><u>-Emprunteur :</u></p> <p>Un emprunteur ne peut pas emprunter plusieurs livres à la fois pour la gestion de notre bibliothèque et aussi éviter que l'emprunteur dispose de plusieurs livres sans restituer celui qu'il à emprunter.</p>

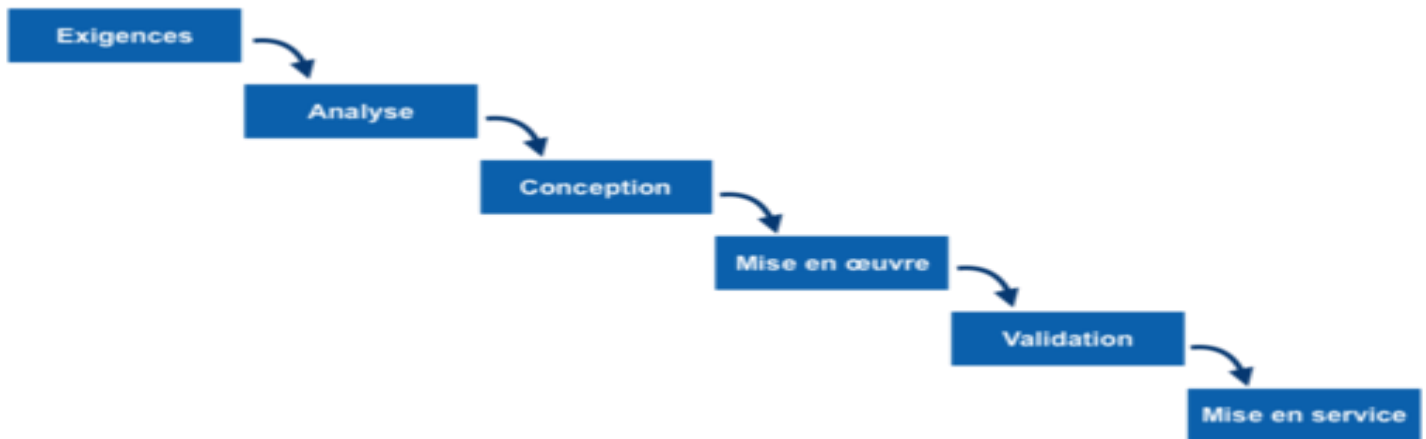
1.5 Planning

Jira

(Répartition des tâches)



Méthode cascade



- Utilisation du modèle cascade (organisation des activités d'un projet sous forme de phases linéaires et séquentielles) pour effectuer notre projet.

- Le modèle en cascade comprend les phases et les livrables suivants :

Exigences : les exigences font l'objet d'une expression des besoins ;

Analyse : les exigences sont analysées pour établir un cahier des charges fonctionnel ;

Conception : le produit est conçu et spécifié de sorte à pouvoir être réalisé ;

Mise en œuvre : le produit est réalisé sur la base des spécifications ;

Validation : le produit est testé et vérifié et sa conformité aux exigences est validée ;

Mise en service : le produit est installé, les préparatifs pour sa mise en service sont organisés, puis le produit est utilisé.

Chaque phase ne commence qu'une fois les résultats de la phase précédente validés. Le point fort de cette approche est de garantir l'existence d'une documentation bien structurée

Section 2 Spécification fonctionnelle

1.6 Spécification

Maquettes :

- Voici notre parti conception ou figure dans un premier temps la maquette (la structure de l'application) de notre projet.

Partie crée un compte :

La maquette de l'application BibYaso est présentée sur un fond orange. Elle est divisée en trois sections principales :

- Connexion** (à gauche) : Affiche une image de rayonnages remplis de livres et le titre "BibYaso".
- Crée un compte** (au centre) : Un formulaire intitulé "Créer votre compte BibYaso" avec les champs suivants :
 - Nom :
 - Prénom :
 - Email :
 - Téléphone :
 - Sélectionnez une question : Prénom (menu déroulant)
 - Répondre :
 - Mot de passe :
 - Confirmez votre mot de passe :Il y a également une case à cocher "J'accepte les conditions et les termes" et un bouton "Suivant".
- S'identifier** (à droite) : Affiche le titre "S'identifier" et deux champs pour l'identification, avec un bouton "Se connecter".

Description :

- Tout d'abord, avant de se connecter puis de lancer l'application, la création du compte est requise pour pouvoir entrer dans celle-ci.

Page de connexion :



Description :

- Ensuite, voici la page de connexion de notre application pour pouvoir y avoir accès. Le principe est tout à fait simple :
- Insérer son Email ainsi que son mot de passe après avoir créé son compte pour pouvoir notamment faire partie des membres (emprunteur qui s'enregistre) de chez BibYaso.

Page gestion livres :



Gestion des livres

Rechercher les livres disponible par: titreLivre

Rechercher Actualiser

idLivre	titreLivre	idAuteur	idCollection	etatLivre
1	Les misérables	1	1	Disponible
2	Le corbeau et le renard	2	2	Disponible

Cliquer Pour Ajouter un livre

Supprimer livre

Description :

- Bienvenue dans la page Gestion livres (la page principale), qui, une fois la connexion de l'utilisateur à l'application avec l'aide de son E-mail ainsi que de son mot de passe sur la page de connexion a été établi avec succès, celle-ci s'affiche directement.
- Elle permet d'ajouter un livre en ligne chez BibYaso pour qu'il puisse être visible auprès des différentes écoles par exemple.

Page Emprunteur:



Gestion Adhérents

Rechercher les adhérents par : idAdherent

idAdherent	nomAdherent	prenomAdherent	codepostalAdherent	villeAdherent
1	Ait Alla	Yassine	92700	Colombes
2	Zaion	Sofiane	92230	Gennevilliers
3	M	Younes	92700	Colombes
4	AIT	Yacine	92700	Colombes

Description :

- Ensuite, voici la page Emprunteur qui permet de gérer l'emprunt des livres en ayant accès à la liste des emprunteurs de l'application. L'emprunteur sur le tableau lorsque on insère ses coordonnées. Cette page permet donc de référencer des emprunteurs pour l'emprunt de Livre.

Page Gestion Prêts :



Gestion Prêts

Rechercher les emprunts par :

	idEmprunt	numAdherent	numLivre	dateEmprunt	dateRetour
1		1	2	2023-02-14	2023-02-22
2		2	1	2023-02-02	2023-02-23

Description :

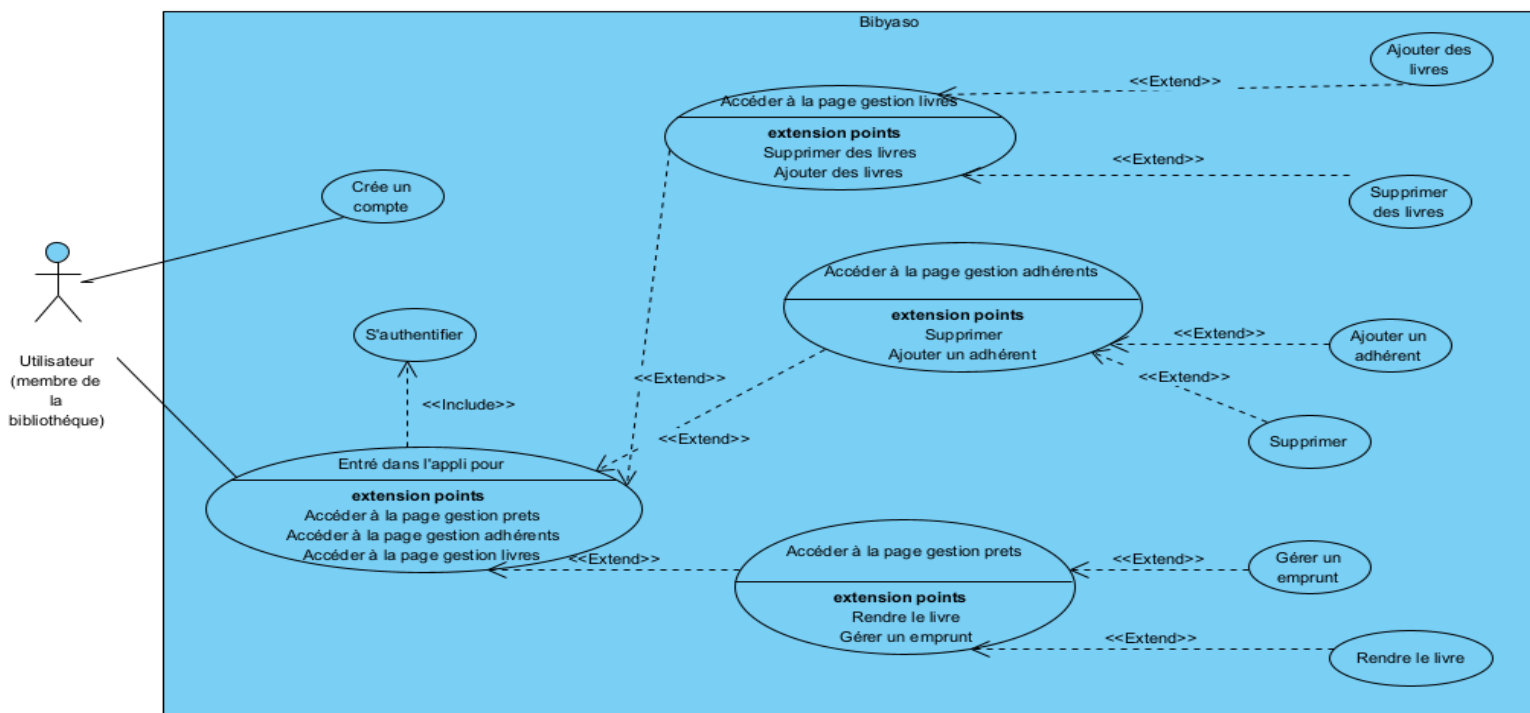
- Enfin, la page de prêts quant à elle s'occupe de la gestion de l'emprunt des livres que l'utilisateur choisi. Celle-ci montre notamment la liste des livres empruntés avec l'emprunteur concernée avec la date d'emprunt et la date de retour sur le tableaulorsque on insère les informations requis dans les cases.

Section 3 Spécification mécanique

1.8 Conception

Diagrammes UML :

Diagramme Use Case:



**Adhèrent qui correspond à emprunteur.*

Pre-requis :

- Pour pouvoir réaliser le graphe UML de notre application, nous avons commencé par installer le logiciel Visual paradigm, un outil UML CASE.

Description :

- Ci-dessus figure une capture d'écran qui caractérise le diagramme de cas d'utilisation, plus précisément le diagramme UML (use case) de notre application. Celui-ci est utilisé pour pouvoir visualiser le comportement fonctionnel d'un système logiciel en général.

Traitement :

- Lorsque l'acteur, plus précisément l'utilisateur externe rentre dans l'application BibYaso (relation entre actor et use case "Entre dans appli pour"), il doit impérativement("includ") créé un compte (use case « crée un compte") et par la suite obligatoirement("includ") se connecter (use case "s'authentifier").
- Ensuite, après avoir effectué l'authentification à l'aide de son E-mail et mot de passe, il a accès à l'appli avec ses trois différents pages. Celui-ci a le choix, c'est bien ce que nous montre les extend qui indiquent qu'il peut en sélectionner une d'entre elles a chaque fois puis changer (use case "accéder à page emprunteur, Gestion livres, Gestion prêts").
- Enfin, dans chaque page figure deux fonctionnalités qui permettent d'effectuer obligatoirement un ajout par exemple si l'utilisateur le souhaite ou bien supprimer son ajout si le désire. (Use case "Accéder à page Gestion livres"include(obliger)->"Ajout des livres "extend(choix)->"supprimer des livres").



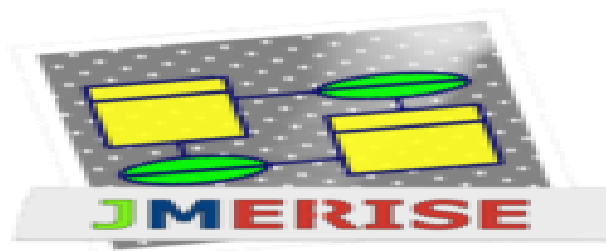
- Bienvenue à la fabrication de notre application intitulé BibYaSo, spécialement dédié à la gestion de livres. Celle-ci permet la lecture à chacun en empruntant des livres ainsi que toutes autres fonctionnalités.
- Nous allons vous présenter notre proposition destinée à notre client qui souhaiterait une application simple et efficace à l'utilisation.
- Pour concevoir BibYaSo, mon camarade ainsi que moi avons utilisé plusieurs outils pour chaque partie abordée (conception, base de données, langage de programmation).
- Le langage de programmation de notre application BibYaso choisi est le langage python. Pour ce faire, nous nous sommes appuyés sur l'outil Tkinter pour pouvoir créer et générer les fenêtres de l'application ainsi que l'outil Visual Studio Code qui permet de coder celle-ci.

Lancement de l'application:

-1ère étape :

- Récupérer le code de l'application (**fichier zip “bibliothequeBIBYASO” situé dans le dossier main du lien github**) puis l'ouvrir sur l'outil Visual studio code. Ensuite, tester le code en faisant ctrl f5 puis une fois le contenu de l'appli afficher, cliquez sur le bouton créé un compte pour pouvoir s'authentifier. Vous allez maintenant créer un compte en insérant les coordonnées demandées, Validez puis re basculer sur la page de connexion pour pouvoir se connecter avec l'E-mail et MDP mis durant la création du compte.
- Pour consulter la base de données de l'application, il faut utiliser l'outil WampServer et faire un clique-droit sur son icone, une fois que celle-ci apparaîtra en couleur verte. Ensuite, cliquer sur phpMyAdmin puis se connecter avec le MDP root et importer le fichier sql du projet Bibyaso

Conception du diagramme Merise

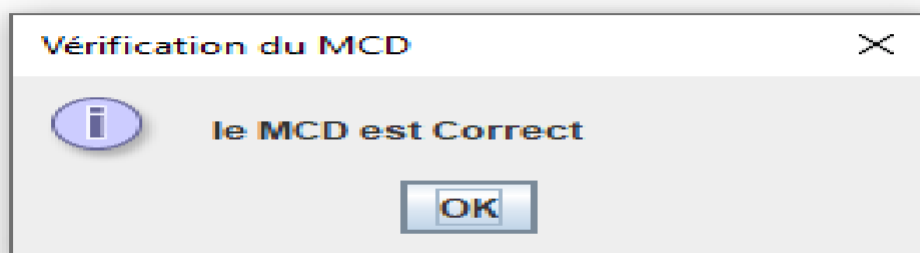


- **Capture d'écran 3(MCD) :**

-

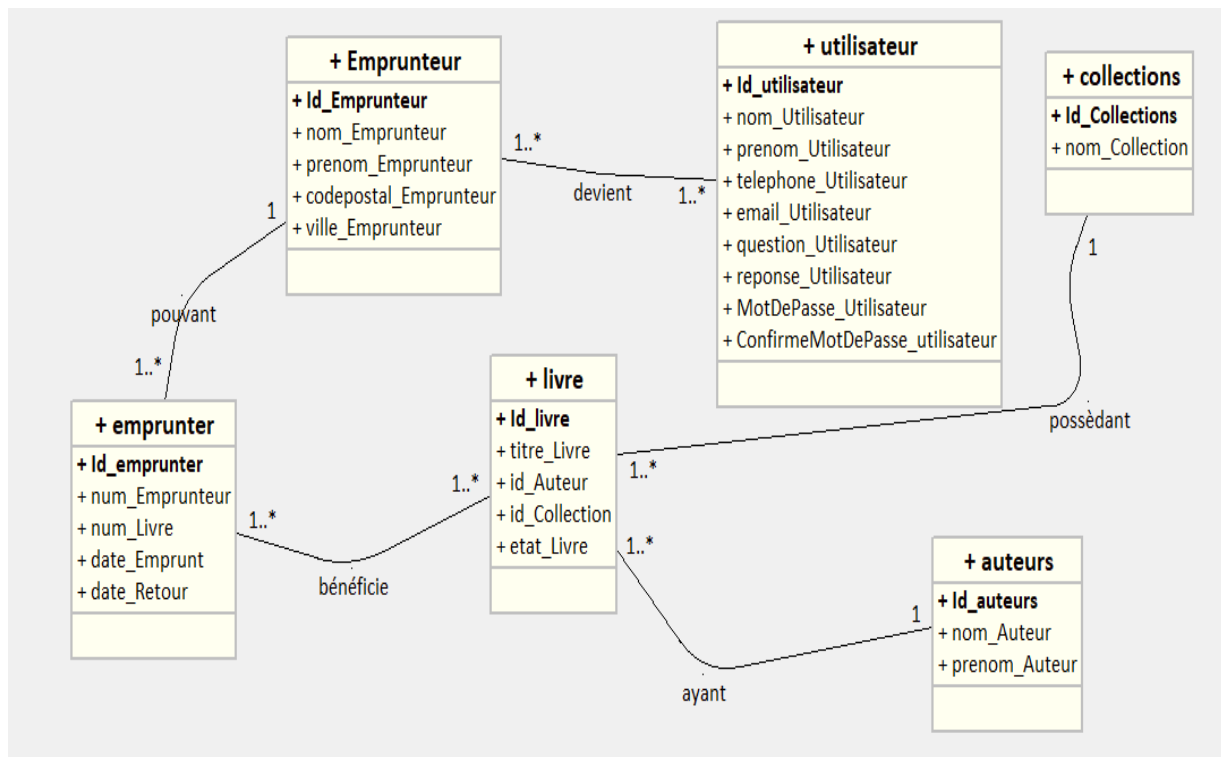
Description :

- Ici, nous avons pu réaliser la partie MCD du graphe de dépendance fonctionnel (GDF) dont le but est d'écrire de façon formelle les données qui seront utilisés par notre système d'information.



-
- La capture ci-dessus nous montre la vérification et validation du MCD qui est correct.

Diagramme de classe :

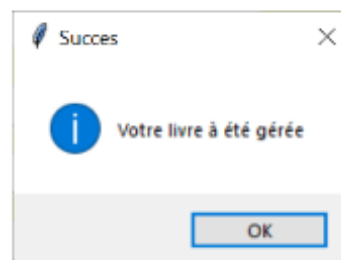
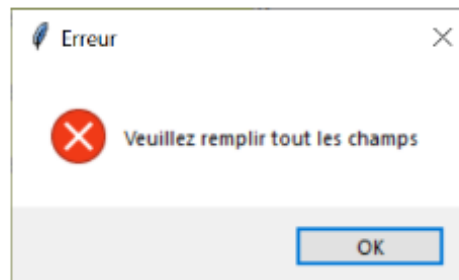


- Sur la capture ci-dessus figure le diagramme de classe de l'application. Celui-ci est reparti avec différentes classes qui Correspond aux 4 pages de Bibyaso.
- En effet, on peut retrouver à la fois les attributs qui correspondent aux fonctionnalités de la page (insertion des coordonnées) dans la 1ère partie d'une classe ainsi que les méthodes (reprise des attributs Dédié au code) dans la 2ème partie.
- Enfin, figure des relations (association ci-dessus) qui consiste à relier les classes avec notamment des cardinalités qui signifient que les Fonctionnalités de l'appli peut se faire par un ou plusieurs utilisateurs.

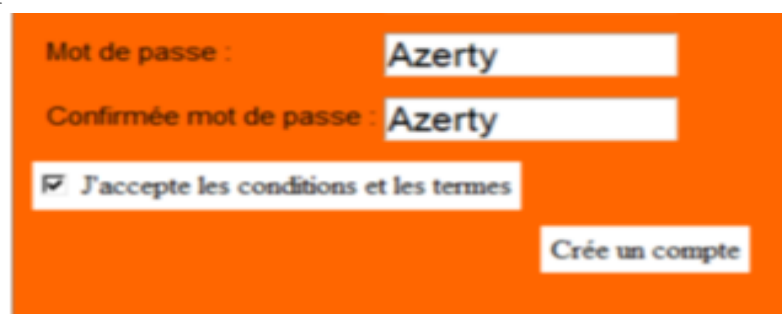
Section 4 Test

1.9 Tests unitaire

Attention à l'orthographe dans les messages



Nous pouvons constater sur la capture d'écran un message d'erreur concernant les champs à remplir entièrement pour pouvoir créé un compte bibyaso comme nous l'indique la capture d'écran ci-dessous.

A screenshot of a user registration form with an orange background. It has two text input fields: "Mot de passe :" with the value "Azerty" and "Confirmée mot de passe :" with the value "Azerty". Below these is a checkbox labeled "J'accepte les conditions et les termes" which is checked. At the bottom right is a button labeled "Crée un compte".

Il aurait été bien que le mot de passe soit caché avec un contrôle graphique marquant des étoiles : comme dans la page de connexion.

Section 5 Mise en production

1.10 Conversion en exe :

-Premièrement ont converti notre logo png en ico.

Ensuite dans l'invite de commande on met pip install auto-py-exe.

```
C:\Users\yass>pip install auto-py-to-exe
```

-On lance notre auto-py-to-exe

```
C:\Users\yass>auto-py-to-exe
```



Auto Py to Exe

Message

Langage: French (Français)

Emplacement des scripts

C:/Users/yass/Desktop/bibliothequefinale/bibliothequefinal/Connexion.py

Navig

un fichier (--onedir / --onefile)

un répertoire

Un fichier

Console Windows (--console / --windowed)

Présence de la console

Basé sur windows (la console n'est pas visible)

↑ Icône (--icon)

C:/Users/yass/Desktop/bibliothequefinale/bibliothequefinal/Images/bib.ico

Naviga

↑ Fichiers additionnels (--add-data)

Ajout de fichiers

Ajout de dossiers

Ajout de pages

C:/Users/yass/Desktop/bibliothequefinale/bibliothequefinal/Cc

.

Si vous souhaitez placer des fichiers dans le répertoire racine, mettez un point (.) dans la destination.

✓ Paramètres avancées

↑ Paramètres

Options spécifiques

Repertoire de sortie ? C:\Users\yass\output

Naviga

Augmenter la limite de récursivité ?

Autorisé

Fournir manuellement des options

Saisie manuelle des arguments ? ARGUMENTS

Paramétrage

Importation de la configuration d'un fichier JSON

Exportation de la configuration vers un fichier JSON

Commande

```
pyinstaller --noconfirm --onedir --console --icon
"C:/Users/yass/Desktop/bibliothequefinale/bibliothequefinal/Images/bib.ico" --add-data
```

CONVERT. .PY VERS .EXE

On insère nos fichier python et en converti en exécutable

Commande

```
pyinstaller --noconfirm --onedir --console --icon
"C:/Users/yass/Desktop/bibliothequefinale/bibliothequefinal/Images/bib.ico" --add-data
```

Sortie

```
Running auto-py-to-exe v2.26.0
Building directory: C:\Users\yass\AppData\Local\Temp\tmprlut9x3u
Provided command: pyinstaller --noconfirm --onedir --console --icon "C:/Users/yass/Desktop/bibliothequefinale
Recursion Limit is set to 5000
Executing: pyinstaller --noconfirm --onedir --console --icon C:/Users/yass/Desktop/bibliothequefinale/bibliot

709013 INFO: PyInstaller: 5.6.2
709023 INFO: Python: 3.10.4
709053 INFO: Platform: Windows-10-10.0.22000-SP0
709071 INFO: wrote C:\Users\yass\AppData\Local\Temp\tmprlut9x3u\Connexion.spec
709088 INFO: UPX is not available.
709108 INFO: Extending PYTHONPATH with paths
['C:\\Users\\yass\\Desktop\\bibliothequefinale\\bibliothequefinal']
710118 INFO: checking Analysis
710128 INFO: Building Analysis because Analysis-00.toc is non existent
710144 INFO: Initializing module dependency graph...
710149 INFO: Caching module graph hooks...
710177 WARNING: Several hooks defined for module 'numpy'. Please take care they do not conflict.
710211 INFO: Analyzing base_library.zip ...
712066 INFO: Loading module hook 'hook-heapq.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Python\\Pyt
712192 INFO: Loading module hook 'hook-encodings.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Python\\
713994 INFO: Loading module hook 'hook-pickle.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Python\\Py
716220 INFO: Caching module dependency graph...
716333 INFO: running Analysis Analysis-00.toc
716362 INFO: Adding Microsoft.Windows.Common-Controls to dependent assemblies of final executable
required by C:\Users\yass\AppData\Local\Programs\Python\Python310\python.exe
716885 INFO: Analyzing C:\Users\yass\Desktop\bibliothequefinale\bibliothequefinal\Connexion.py
716897 INFO: Processing pre-find module path hook distutils from 'C:\\Users\\yass\\AppData\\Local\\Programs\\
717328 INFO: Loading module hook 'hook-distutils.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Python\\
717364 INFO: Loading module hook 'hook-distutils.util.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Py
717450 INFO: Loading module hook 'hook-sysconfig.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Python\\
718119 INFO: Processing module hooks...
718158 INFO: Loading module hook 'hook-setuptools.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Pythor
720243 INFO: Loading module hook 'hook-pkg_resources.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Pyt
721234 INFO: Loading module hook 'hook-platform.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Python\\
721337 INFO: Loading module hook 'hook-xml.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\Python\\Pytho
722380 INFO: Processing pre-safe import module hook win32com from 'C:\\Users\\yass\\AppData\\Local\\Programs\\
723929 INFO: Loading module hook 'hook-multiprocessing.util.py' from 'C:\\Users\\yass\\AppData\\Local\\Progre
724365 INFO: Loading module hook 'hook-setuptools.msvc.py' from 'C:\\Users\\yass\\AppData\\Local\\Programs\\F
```

CONVERSION...

Voici notre dossier dans lequel se trouve notre application convertie en exécutable

