



v1.0 **Projet CY-Books**

CLASSE ING1-GI • 2023-2024

AUTEURS Eva ANSERMIN – Romuald GRIGNON

E-MAILS eva.ansermin@cyu.fr – romuald.grignon@cyu.fr

DESCRIPTION DU PROJET

Créer une application permettant de gérer une bibliothèque, avec toutes les fonctionnalités telles que l'inscription des usagers, la gestion des livres, le stock, et les différents emprunts.

Le but principal est de créer une application graphique utilisée par les bibliothécaires, utilisable à la souris et au clavier. Il devra être possible de parcourir les utilisateurs, les livres, et toutes leurs propriétés en filtrant les résultats avec différents critères.

Ce projet va stocker uniquement les données utilisateurs, les identifiants des livres, et la liste des emprunts. Toutes les informations des livres à proprement parler (titres, auteur, année, thème, format, éditeur, ...) seront récupérées depuis un service externe : l'API de la Bibliothèque Nationale de France (BNF).

Ce projet mettra donc l'accent sur la connexion du programme avec une base de données locale, ainsi qu'une API distante.

INFORMATIONS GENERALES

Taille de l'équipe

Ce projet est un travail d'équipe. Il est autorisé de se réunir en groupe de 5 personnes. Si le nombre d'étudiants n'est pas un multiple de 5 et/ou si des étudiants n'arrivent pas à constituer des groupes, c'est aux chargés de projet de statuer sur la formation des groupes. Pensez donc à anticiper la constitution de vos groupes pour éviter des décisions malheureuses.

Démarrage du projet et jalons

Vous obtiendrez de plus amples informations quant aux dates précises de rendu, de soutenance, les critères d'évaluation, le contenu du livrable, ..., quand le projet démarrera officiellement.

Quel que soit le planning initial du projet, vous veillerez à planifier plusieurs rendez-vous d'avancement avec votre chargé(e) de projet. C'est à votre groupe de prendre cette initiative. L'idéal étant de faire un point 1 ou 2 fois par semaine mais cette fréquence est laissée libre en fonction des besoins identifiés avec le tuteur de projet.

Versions de l'application

Pour éviter d'avoir un projet non fonctionnel à la fin, il vous est demandé d'avoir une version en ligne de commande fonctionnelle. Ceci vous permettra de tester votre modèle de données indépendamment de l'interface graphique. C'est la version avec l'interface graphique qui sera bien entendu évaluée mais dans le cas où certaines fonctionnalités ne seraient pas visibles avec cette interface, vous devez pouvoir présenter toutes les fonctionnalités en ligne de commande.

Dépôt de code

Vous devrez déposer la totalité des fichiers de votre projet sur un dépôt central Git. Il en existe plusieurs disponibles gratuitement sur des sites web comme gi-

	<p>thub.com ou gitlab.com. La fréquence des commits sur ce dépôt doit être au minimum de 1 commit / 2 jours.</p> <p>Rapport du projet</p> <p>Un rapport écrit est requis, contenant une brève description de l'équipe et du sujet. Il décrira les différents problèmes rencontrés, les solutions apportées et les résultats. L'idée principale est de montrer comment l'équipe s'est organisée, et quel était le flux de travail appliqué pour atteindre les objectifs du cahier des charges. Le rapport du projet peut être rédigé en français.</p> <p>Ce rapport contiendra en plus les éléments techniques suivants : un document de conception UML (diagramme de classe) de votre application, un document montrant les cas d'utilisations.</p> <p>Démonstration</p> <p>Le jour de la présentation de votre projet, votre code sera exécuté sur l'une de vos machines. La version utilisée sera la dernière fournie sur le dépôt Git avant la date de rendu. Même si vous avez une nouvelle version qui corrige des erreurs ou implémente de nouvelles fonctionnalités le jour de la démonstration, c'est bien la version du rendu qui sera utilisée.</p> <p>En parallèle, il vous faudra obligatoirement une deuxième machine avec votre application fonctionnelle car une partie de votre groupe aura des modifications de code à faire pendant que l'autre partie fera la présentation/démonstration de votre projet.</p> <p>Organisation de l'équipe</p> <p>Votre projet sera stocké sur un dépôt git (ou un outil similaire) tout au long du projet pour au moins trois raisons : éviter de perdre du travail tout au long du développement de votre application, être capable de travailler en équipe efficacement, et partager vos progrès de développement facilement avec votre chargé(e) de projet. De plus il est recommandé de mettre en place un environnement de travail en équipe en utilisant divers outils pour cela (Slack, Trello, Discord, ...).</p>

CRITERES GENERAUX	<ul style="list-style-type: none">• Le but principal du projet est de fournir une application fonctionnelle pour l'utilisateur. Il doit implémenter toutes les fonctionnalités de ce document.• Votre code sera généreusement commenté.• Tous les éléments de votre code (variables, fonctions, commentaires) seront écrits en anglais obligatoirement.• Votre code devra être commenté de telle manière que l'on puisse utiliser un outil de génération de documentation automatique de type JavaDoc. Un dossier contenant la doc générée par vos soins sera présent dans votre dépôt de code lors de la livraison.• Votre projet doit être utilisable au clavier ou à la souris en fonction des fonctionnalités nécessaires.• Votre application ne doit jamais s'interrompre de manière intempestive (crash), ou tourner en boucle indéfiniment, quelle que soit la raison. Toutes les erreurs doivent être gérées correctement. Il est préférable de d'avoir une application stable avec moins de fonctionnalités plutôt qu'une application contenant toutes les exigences du cahier des charges mais qui plante trop souvent. Une application qui se stoppe de manière imprévue à cause d'une exception, par exemple, sera un événement très pénalisant.• Votre application devra être modulée afin de ne pas avoir l'ensemble du code dans un seul et même fichier par exemple. Apportez du soin à la conception de votre projet avant de vous lancer dans le code.

	<ul style="list-style-type: none"> Le livrable fourni à votre chargé(e) de TD sera simplement l'URL de votre dépôt Git accessible publiquement.
FONCTIONNALITES DU PROJET	<ul style="list-style-type: none"> Cette application doit être utilisée uniquement par le bibliothécaire, lui permettant de gérer les usagers (notamment les inscriptions, mais aussi les modifications d'informations, ou la recherche d'un usager bien précis), gérer les livres (recherche multi-critères), et les emprunts. Cette application doit inclure un temps maximal d'emprunt par livre, ainsi qu'un nombre maximal de livres empruntés par le même usager à la fois. Des alertes, ou des affichages contrastés doivent permettre aux bibliothécaires de voir les problèmes d'emprunts actuels. Une liste des retards doit également être affichée à la demande. Un historique des emprunts, dates, durées et retards, doit être possible lors de l'affichage du profil d'un usager spécifique. La recherche de livres doit se faire avec différents critères de filtrage : ces critères doivent ensuite être envoyés dans l'API de la BNF afin de récupérer la liste des livres possibles. Les résultats des recherches doivent être affichés page par page si le nombre est trop élevé. Le stockage des emprunts doit se faire en local, soit par l'intermédiaire d'une base de données, soit via des fichiers : ce choix technique est laissé libre. Il est possible aussi d'afficher également une liste des livres les plus empruntés pendant les 30 derniers jours. Votre application doit être logique et ne pas autoriser des actions telles qu'assigner un emprunt à une date passée, permettre l'emprunt d'un livre si il est déjà emprunté (on part du principe qu'il n'y a que X exemplaires de chaque livre, X étant une constante à définir), ... Si une base de données locale est utilisée, c'est à votre programme de lancer le serveur si il ne l'est pas déjà en faisant un appel système.
RESSOURCES UTILES	<p>Github</p> <ul style="list-style-type: none"> www.github.com https://docs.github.com/en/get-started/quickstart/hello-world <p>Patrons de conception</p> <ul style="list-style-type: none"> https://fr.wikipedia.org/wiki/Patron_de_conception Singleton / Strategy / Composite / Observer / Visitor <p>Sérialisation</p> <ul style="list-style-type: none"> https://fr.wikipedia.org/wiki/Sérialisation <p>Appels système depuis Java</p> <ul style="list-style-type: none"> https://docs.oracle.com/javase/8/docs/api/java/lang/Runtime.html <p>Connexion à une base de données</p> <ul style="list-style-type: none"> https://docs.oracle.com/javase/tutorial/jdbc/basics/connecting.html <p>Connexion à une API Web</p> <ul style="list-style-type: none"> https://docs.oracle.com/javase/8/docs/api/java/net/URLConnection.html https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple <p>API Bibliothèque Nationale de France</p> <ul style="list-style-type: none"> https://api.bnf.fr
Eva ANSERMIN – Romuald GRIGNON • 2023-2024 • ING1-GI • eva.ansermin@cyu.fr – romuald.grignon@cyu.fr	