



université  
virtuelle  
Burkina ★ Faso

# RAPPORT DE FIN DE PROJET

GESTION DE BIBLIOTHEQUE EN LANGAGE C



## GROUPE 1 :

- KOUCHANOU Juan Prince Christel Djidenou
- KABORE Wendgouda Marie Innocent

# **PLAN :**

I. LA CONCEPTION

II. L'IMPLEMENTATION

III. PROBLEMES RENCONTRES

IV. LECONS APPRISES

# **I. LA CONCEPTION**

## **1. L'élaboration du plan de notre projet**

La première étape lors de la conception de notre programme est l'élaboration du plan sur lequel nous allons nous baser pour concevoir notre projet. Ce plan contient les spécifications fonctionnelles de notre programme.

Le Projet doit contenir 3 modules : Le module de Gestion des catégories, celui de Gestion des livres et enfin la Recherche.

### **a) Gestion des catégories**

Pour la gestion des catégories, l'utilisateur doit pouvoir ajouter une catégorie, supprimer une catégorie et enfin pouvoir afficher la liste des catégories. Après de longue réflexion, nous avons décidé de créer un fichier texte lorsque l'on ajoute une catégorie (« nomcategorie.txt »). En effet, si l'on crée une catégorie DRAME, un fichier DRAME.txt sera créé contenant tous les romans de la catégorie. Aussi, le nom de la catégorie sera enregistré automatiquement dans un fichier nommé « catégorie.txt ». Lors de la création aussi, d'autres fichiers texte seront aussi créés comme « nomcategorie\_id.txt », « nomcategorie\_auteur.txt », « nomcategorie\_titre.txt » qui nous servira plus tard dans les autres modules.

Lorsqu'un utilisateur supprime une catégorie, notre programme doit supprimer les fichiers textes associés à cette

catégorie et aussi supprimer le nom de cette catégorie dans le fichier « catégorie.txt ».

Lorsque l'utilisateur décide d'afficher la liste des catégories, on lit le fichier texte « catégorie.txt » et on l'affiche à l'écran tout simplement.

### **b) Gestion des livres**

Pour la gestion des livres, lors de l'ajout, l'utilisateur doit entrer le titre, l'auteur, la date de publication, le genre, le prix et le nombre en stock. Ses informations saisis seront enregistrées dans le fichier correspondant à la catégorie auquel le livre appartient. Aussi, les informations de l'auteur sont enregistrées à la première ligne du fichier de la catégorie\_auteur et les informations du titre est enregistrer à la première ligne du fichier de la catégorie\_titre. Aussi chaque roman ajouter prennent un id et chaque id est conserver dans le fichier de la catégorie correspondant à l'ID(categorie\_ID).

Lors de la suppression d'un roman, la ligne correspondant à l'identifiant du roman est supprimée dans tous les fichiers correspondant à la catégorie. Ensuite, une petite mise à jour sera effectuée au niveau des identifiants.

Lors de la modification des informations d'un livres, on demande la catégorie et l'id du roman à modifier. Ensuite, l'utilisateur re-entre les informations. Ses informations seront modifiées dans tous les fichiers correspondant à la catégorie.

Lorsque l'utilisateur souhaite afficher la liste des romans, on lui demande la catégorie puis il lit les fichiers de la catégorie et affiche tous les romans à l'écran.

### **c) Recherche**

Pour la recherche, l'utilisateur pourra rechercher selon 3 modes, Par l'auteur, par titre des œuvres et par catégories.

Pour la recherche par l'auteur, on compare chaque ligne du fichier `categorie_auteur` avec l'auteur entré par l'utilisateur. Si trouvé, on affiche le roman dans la `nomcategorie.txt`. Pareil également pour la recherche par titre qui lui se fera dans la `categorie_titre`.

Pour la recherche par catégorie, on fait juste la recherche dans le fichier `categorie.txt` puis on affiche tous les romans contenus dans cette catégorie si trouvé.

### **d) La Gestion des Menus**

Pour afficher le menu, on affiche à l'écran les différents choix possibles à l'utilisateur. Il saisit son choix au clavier, puis est redirigé vers le module voulu. On met l'affichage du menu dans une boucle ou 0 permet de quitter la boucle. Un choix différent des choix proposés à l'utilisateur renvoie un message d'erreur au clavier.

## **2. La conception proprement dit du programme**

On commence par coder notre programme. Chaque membre de notre binôme prend des parties des modules à gérer. La

première étape à gérer est un menu fonctionnel ou l'utilisateur pourra communiquer avec la machine.

On crée une boucle do while qui s'exécute tant que l'utilisateur n'entre pas 0 (le choix 0 permet de fermer le programme). On y met des condition if pour chaque cas choisit. S'il rentre 1, on le redirige vers le menu des catégories et ainsi de suite.

Chaque module est codé dans un fichier header (.h) qui sera incorporer dans le fichier principal (.c).

Après un choix, l'écran est nettoyé automatiquement grâce à notre fonction clean.

Une structure est déclarée puis peut être appeler à chaque ajout de livre dans une catégorie.

On fait de sorte à ce qu'il faut forcément une catégorie avant d'ajouter un livre.

Après une suppression de catégorie ou d'un roman correspond à la suppression de la ligne correspondant à sa position. On ouvre le fichier puis un fichier temporaire, on réécrit tous les lignes sauf la ligne correspondant à ce que l'on veut supprimer. Ensuite, on supprime le fichier principal puis on renomme le fichier temporaire en le fichier principale.

Pour modifier c'est à peu près la meme chose sauf qu'arrivé à la ligne concerner, on réécrit avec les nouvelles informations entrer.

### **3. CONCLUSION APRES LA CONCEPTION**

Après la conception, malgré tous les problèmes rencontrés lors du développement du programme, aux finales tout répond correctement. Nous sommes fiers de notre réalisation.

## **II. L'IMPLEMENTATION**

Pour pouvoir exécuter notre programme, il faut avoir les fichiers « catégorie.txt », le fichier Gestion\_bibliotheque.c et les menu.h, choix\_menu.h, gestion\_livre.h, gestion\_categorie.h, et recherche\_livre.h dans le meme répertoire que l'exécutable. Il faut noter que tous ses fichiers sont indispensables pour une bonne exécution du programme.

Si tout ses cas sont remplis, vous pouvez sans soucis exécuter le programme et profiter de ses différentes fonctionnalités.

## **III. PROBLEMES RENCONTRES**

Lors des différentes phases de conceptions de notre programme, nous avons été confrontés à plusieurs probleme.

Le premier souci rencontrer était la suppression car elle était très compliquée à réaliser, puis après de longs essaie et recherche nous avons pu finaliser cette problématique.

Ensuite, le fait de supprimer, une ligne, il fallait mettre à jour toutes les lignes avec de nouveau identifiant. Il fallait

récupérer les lignes et modifier la position correspondante à l'id avant de l'écrire dans un autre fichier temporaire, supprimer le fichier principal et renommer le fichier temporaire. C'est sur ce problème nous avons passé le plus long temps. Mais nous avons pu pallier au problème.

Ensuite, la recherche a été aussi très compliquée car il fallait si la recherche a été trouvée de retourner la catégorie et l'id du roman. Afin d'afficher à la fin que cette ligne.

Aussi, on voulait pouvoir taper des chaînes de caractères avec des espaces. Il nous a fallu apprendre à manipuler la fonction fgets pour pouvoir arriver à notre résultat final.

On aurait voulu avoir moins de fichiers texte, car en plus du fichier categorie.txt on a près de 4 fichiers texte par catégorie. Mais on n'a pas pu trouver une méthode nous permettant d'utiliser moins de fichiers tout en respectant les spécifications voulues par le programme.

## **IV. LECONS APPRISES**

Lors de la réalisation de ce projet, d'abord on a dû apprendre comment manipuler un fichier, c'est-à-dire ouvrir, supprimer, modifier un fichier, les règles permettant d'y parvenir, car avant on n'y avait aucune notion là-dedans.

Ensuite, on a appris à mieux utiliser les pointeurs, et les tableaux et on a découvert pleins de fonctions utiles. On a effleuré aussi l'allocation dynamique qui nous a servi dans une partie de nos modules.



Aussi, on a appris à créer des fonctions externe ou bibliothèque (.h) et à les intégrer dans le fichier (.c).

On a su comment pourvoir gérer les espacements en entrée et comment supprimer un saut de ligne.

Ensuite, on a appris à nettoyer l'invite de commande pour rendre joli notre programme Windows comme les autres systèmes.

On a appris à debugger lors des crashes car il était très difficile de repérer les erreur ou bugs qui se produisait souvent lors de l'exécution du code.

On a appris à réfléchir et à mettre en œuvre un plan de structuration quant à la fonctionnalité de notre programme sur comment parvenir au resultat espérer.

## **CONCLUSION**

Ce projet de gestion d'une bibliothèque nous a permit d'aborder la conception d'un programme fonctionnel. On y a rencontré beaucoup de difficulté mais après de longue recherche et de détermination, on a réussi à tous les corriger. Malgré tout, nous savons que notre programme possède des limites. Ce projet nous a permis d'apprendre énormément plein de nouveau trucs sur le langage C.