



ECOLE NATIONALE SUPERIEURE
D'INFORMATIQUE ET D'ANALYSE
DES SYSTEMES
Rabat



Projet C

Elevés ingénieurs premiere année

Gestion de prêts de livres à la bibliothèque ENSIAS

Réalisé par :

**YAHYAOUI MOHAMED
ZEMMOURI SALAH EDDINE**

Encadré par :

Mdm K. MOUMANE

**ANNEE UNIVERSITAIRE 2020-
2021**

Remerciements

Nous tenons à remercier Monsieur **Moumane Karima**, pour son soutien et sa générosité considérable quant à l'offre de l'information.

Nous tenons également à adresser nos sincères remerciements à l'ensemble du corps enseignant de l'ENSIAS, pour avoir porté un vif intérêt à notre parcours, et pour avoir accordé leur temps, leur attention et leur énergie et ce dans un cadre agréable de complicité et de respect.

Enfin, nous tenons également à adresser nos plus sincères remerciements à toutes les personnes qui ont contribué, de près ou de loin à la réalisation de ce projet.

Table des matières

Remerciements	i
Introduction	v
1 Présentation et analyse du projet	vi
1.1 Introduction	vi
1.2 Présentation du projet :	vi
1.2.1 Objet :	vi
1.2.2 Définition du projet :	vi
1.2.3 Eléments de contexte :	vi
1.2.4 Objectifs :	vi
1.3 Cahier de charge :	vii
1.3.1 Etude des besoins :	vii
1.4 Conclusion	viii
2 Analyse et conception	ix
2.1 Introduction	ix
2.2 Structure du code :	ix
2.2.1 Choix de la structure de données :	ix
2.2.2 Les différentes structures et fonctions utilisées :	ix
2.3 Hiérarchie des fichiers :	xi
2.3.1 Les différents fichiers/dossiers du projet :	xi
2.4 Conclusion :	xii
3 Réalisation	xiv
3.1 Introduction	xiv
3.2 Outils et techniques de développement	xiv
3.2.1 Langage C :	xiv
3.2.2 Interface graphique GTK :	xiv
3.2.3 Visual code	xv
3.2.4 Glade :	xv
3.2.5 Github :	xvi
3.3 Les di@cultés :	xvi
3.3.1 Compréhension du sujet	xvi
3.3.2 L'interface gtk :	xvi
3.4 Les interfaces :	xvi
3.4.1 Interface d'accueil :	xvi

3.4.2	La gestion des adhérents :.....	xvii
3.4.3	La gestion des livres :	xx
3.4.4	la gestion des emprunts :	xxii
3.5	Conclusion :.....	xxiii
Conclusion générale		xxiv
Bibliographie		xxv

Table des figures

2.1	Hiérarchie des fichiers.....	.xiii
3.1	Langage C.....	.xiv
3.2	GTK.....	.xv
3.3	Visual code studio.....	.xv
3.4	Glade.....	.xv
3.5	Github.....	.xvi
3.6	Interface principale.....	.xvii
3.7	Gestion des adhérents.....	.xvii
3.8	Ajouter un adhérent.....	.xviii
3.9	Modifier un adhérent.....	.xviii
3.10	Supprimer un adhérent.....	.xix
3.11	Cas nominal : adherent supprimé.....	.xix
3.12	Recherche et a@chage d'un adherent :.....	.xx
3.13	Liste des adhérents.....	.xx
3.14	Gestion des livres.....	.xxi
3.15	Tri des livres.....	.xxi
3.16	Liste des livres.....	.xxii
3.17	Gestion des emprunts.....	.xxii
3.18	Emprunter livre.....	.xxiii
3.19	Rendre un livre.....	.xxiii

Introduction

Ce projet entre dans le cadre de la concrétisation des concepts et des notions acquises lors des séances de programmation . Il a pour objectif de consolider et d'améliorer nos compétences en tout ce concerne le développement en partant de la conception du problème a sa realisation.

La bibliothèque occupe une place déterminante dans la vie d'un élève ingénieur , c'est un outil d'apprentissage et d'éducation . Elle contribue à former des nouvelles idées et perspec- tives à la base des sociétés créatives et innovantes.

Avant l'informatisation des bibliothèques , toutes les tâches étaient réalisées manuelle- ment et indépendamment les unes des autres . Par conséquent , l'administrateur d'une biblio- thèque avait du mal à gérer les nouveaux livres , les nouveaux adhérents inscrits , ainsi que les emprunts .

Ce projet : Gestion de prêts de livres à la biblio ENSIAS a pour but l'automatisation de la gestion des emprunts au sein d'une bibliothèque. Pour cela, nous avons passé par 3 étapes : la description du contexte générale du projet, l'analyse et la conception, pour aboutir enfin à la réalisation.

Chapitre 1

Présentation et analyse du projet

1.1 Introduction

Ce chapitre décrit le projet afin d'avoir une meilleure compréhension de ce dernier, en exposant sa problématique ainsi que ses objectifs.

1.2 Présentation du projet :

1.2.1 Objet :

Gestion d'une bibliothèque

1.2.2 Définition du projet :

Développement d'un système permettant l'automatisation de la gestion des adhérents, emprunts et retours des livres en utilisant le langage C.

1.2.3 Eléments de contexte :

Les bibliothèques au Maroc sont de plus en plus courantes et en tant qu'étudiants, nous utilisons souvent ces ressources d'information, il est donc nécessaire d'envisager des solutions pratiques aux problèmes rencontrés dans le processus de gestion des bibliothèques.

1.2.4 Objectifs :

Notre objectif principal est d'automatiser la gestion de la bibliothèque qu'on organisera de la façon suivante :

- Un service s'occupe des nouvelles demandes d'adhésions. Ces derniers sont identifiés par le nom, prénom, adresse, email.
- Un service s'occupe de l'enregistrement des prêts d'exemplaires. Les prêts sont mémorisés afin de savoir à tout moment, quels sont les exem- plaires sortis. Pour chaque prêt ou emprunt on mémorise donc le N° de livre , le N° de l'adhérent emprunteur . Un adhérent peut emprunter au maximum 3 exemplaires.
- Un service s'occupe du retour des exemplaires e Lorsque l'exemplaire est retourné , le nombre

de livres empruntés est décrémenté et le livre n'est plus associé à l'adhérent.

1.3 Cahier de charge :

1.3.1 Etude des besoins :

l'acteur :

Notre acteur principal est l'administrateur du systeme.

Besoins fonctionnels :

Le système doit permettre à l'administrateur :

Gérer les adherents :

- Ajouter un adherent.
- Supprimer un adherent.
- Modifier un adherent.
- Rechercher un adherent par nom.
- A@cher la liste des adherents.

Gérer les livres :

- Ajouter un livre.
- Supprimer un livre.
- Modifier un livre.
- Modifier un livre.
- Ordonner les livres selon la catégorie.
- Rechercher un livre par catégorie et titre du livre.
- A@cher la liste des livres.

Gérer les emprunts :

- Emprunter un livre.
- A@cher la liste des livres empruntés.
- A@cher la liste des adherents ayant emprunté des livres.
- Rendre un livre.

Quitter l'application

Besoins non fonctionnels :

Adaptation des interfaces : Non seulement améliorer les interactions homme-machine et la facilité d'utilisation. SeQuorTalib est défini par des interfaces simples bien conçues savoir ergonomiques et intuitives pour que l'utilisateur la comprenne sans aucune complication.

La performance : Le délai de traitement(demande et réponse) est censé être logique et tolérable par l'utilisateur vu que c'est une notion qui comporte des aspects techniques, mais aussi des aspects liés à l'ergonomie de l'interface entre l'utilisateur et le système. L'application doit garantir l'intégrité et la confidentialité des données.

La capacité : L'application doit être capable de stocker et gérer un grand nombre de données dans des fichiers.

1.4 Conclusion

Cette étape a permis de mieux définir le périmètre du projet puisque on a décrit la problématique puis on a établi le cahier des charges. Le chapitre suivant est consacré à l'analyse et la conception.

Chapitre 2

Analyse et conception

2.1 Introduction

La conception demeure importante et indispensable dans le cycle de vie d'une application. Elle consiste à donner une image claire du déroulement d'un Framework. Cette phase a pour but donc de définir les objectifs d'après sa réalisation et assurer une bonne compréhension des besoins des utilisateurs.

2.2 Structure du code :

2.2.1 Choix de la structure de données :

Nous avons choisie de manipuler nos données dans des fichiers binaires, vu que nous avons travaillé avec des enregistrements, ainsi, l'utilisation de ce type de fichiers nous permet de traiter les informations sous forme de blocs. De plus, nous pouvons accéder à un enregistrement d'une manière aléatoire comme l'accès à une donnée dans un tableau, sans oublier la rapidité de la lecture et l'écriture des données, puisqu'ils sont lus/écrits sous forme binaire directement dans disque.

Les deux fichiers contenant les données sont : adherent.dat et livre.dat.

2.2.2 Les différentes structures et fonctions utilisées :

Les Structures :

Nous avons utilisé 3 structures :
Adherent :

```
1 typedef struct_  
2     adherent { in  
3         tnum_adh ;  
4         char nom_adh [ 1 0 0 ] ;  
5         char prenom_adh [ 1 0  
6         0 ] ; char email_adh [ 1  
7         0 0 ] ; char address_  
8         adh [ 1 0 0 ] ;  
9         in tnbre_emprunts adh : // bitfield
```

Auteur :

```
1 typedef struct _auteur {  
2     char nom_aut [ 100 ]  
3     ;  
4     char prenom_aut [ 100 ] ;  
5 }
```

Livre :

```
1 typedef struct _liv  
2     re { int num_liv  
3     ;  
4     char titre_liv [ 10  
5     0 ] ; char categ_liv  
6     [ 21 ] ; Auteur aute  
7     ur_liv ; int empr  
     unteur_liv ;
```

Les fontions :

Fonction ajouter un adhérent :

```
1 void ajouter_adherent_algo ( adherent ) ;
```

Cette fonction permet l'ajout d'un adhérent dans le fichier adherent.dat.

Fonction modifier un adhérent :

```
1 void modifier_adherent_algo ( adherent ) ;
```

Cette fonction permet la modification d'un adhérent dans le fichier adherent.dat.

Fonction suprmier un adhérent :

```
1 void supprimer_adherent_algo ( int ) ;
```

Cette fonction permet la suppression d'un adhérent dans le fichier adherent.dat.

Fonction rechercher un adhérent :

```
1 void rechercher_adherent_algo ( const char* ) ;
```

Cette fonction permet la recherche d'un adhérent dans le fichier adherent.dat.

Fonction ajouter un livre :

```
1 void ajouter_livre_algo ( Livre ) ;
```

Cette fonction permet l'ajout d'un livre dans le fichier livre.dat.

Fonction rechercher un livre :

```
1 void rechercher_livre_algo ( const char* , const char* ) ;
```

Cette fonction permet la recherche d'un livre dans le fichier livre.dat.

Fonction ordonner la liste des livres :

```
1 void ordonner_livre_algo();
```

Cette fonction permet le tri d'une liste de livre par titre ou par catégorie.

Fonction emprunter un livre :

```
1 void emprunter_livre_algo(int, int);
```

Cette fonction permet l'emprunt d'un livre par un adhérent existant.

Fonction rendre un livre :

```
1 void rendre_livre_algo(int, int);
```

Cette fonction permet de rendre un livre.

Fonction gtk init :

```
1 gtk_init(&argc, &argv);
```

Cette fonction permet d'initialiser le toolkit nécessaire afin de réaliser l'interface graphique.

Fonction load css :

```
1 load_css();
```

Cette fonction permet l'inclusion du fichier css dans le corps du projet.

2.3 Hiérarchie des fichiers :

2.3.1 Les différents fichiers/dossiers du projet:

Ce projet est séparé en 5 sous dossier, contenu dans le dossier Modules. Chaque dossier contient deux fichiers .c et .h. De plus, un dossier "glade" contenant les différentes interfaces de l'application.

Partie Back-end :

Le fichier algorithm.c contient les fonction nécessaire pour réaliser les différentes opérations du cahier de charge.

Partie Front-end :

Les fichiers accueil.c, adherent.c et livre .c contiennent l'appel des fonctions du back-end, ainsi la partie des interfaces(code gtk).

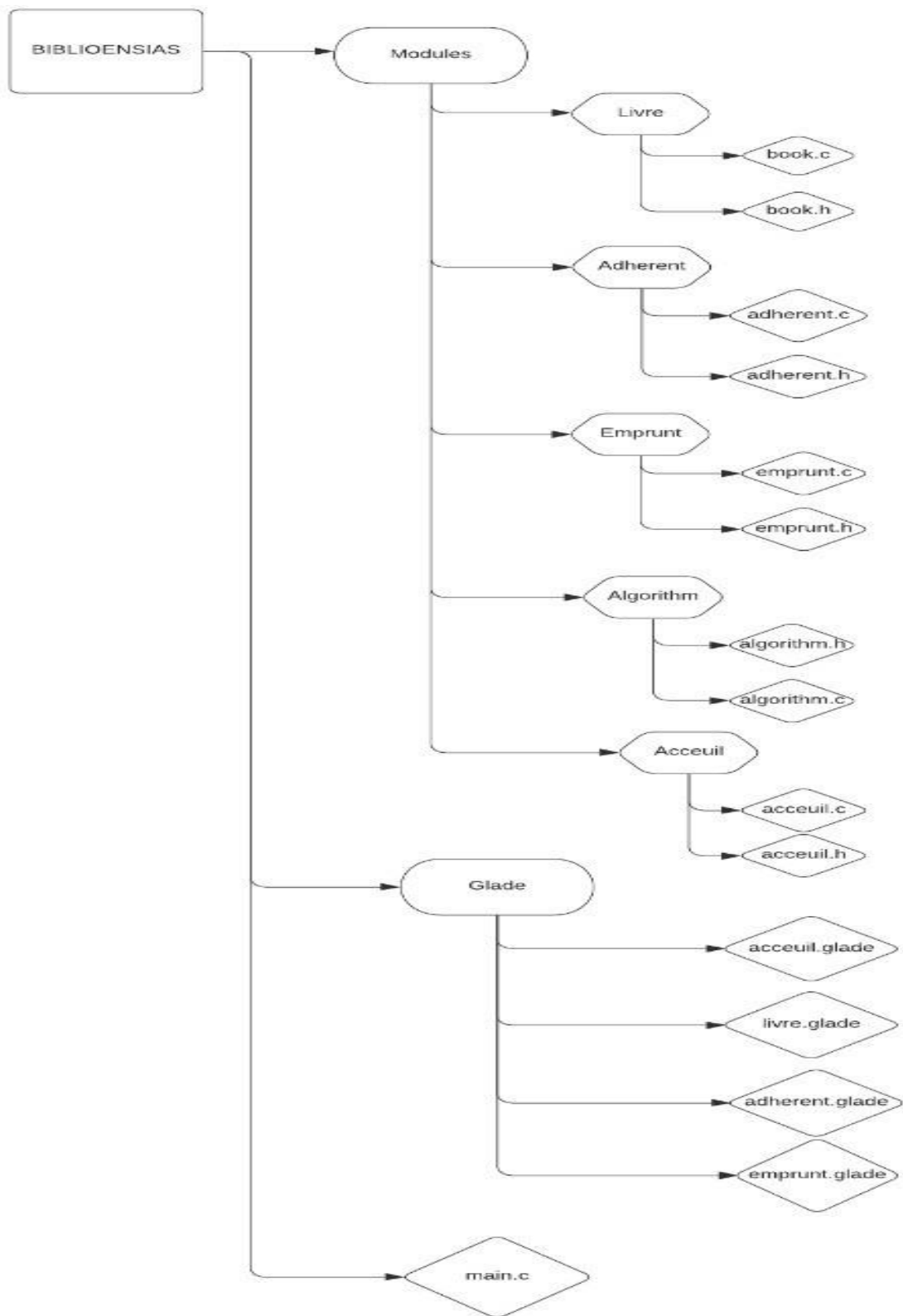


FIGURE 2.1 – Hiérarchie des fichiers

2.4 Conclusion :

Ce chapitre résume le travail effectué lors des étapes d'analyse et de conception de la solution mise en place. D'abord, on a justifié le choix de la structure de données utilisée. Ensuite, on a décrit les différentes structures et les principales fonctions élaborées dans la réalisation du code du projet, pour aboutir enfin à la présentation de la hiérarchie des dossiers et fichiers, afin de faciliter la compréhension du code.

Le chapitre suivant décrira les différents outils utilisés, ainsi, nous allons présenter les interfaces graphiques

Chapitre 3

Réalisation

3.1 Introduction

Il est évident que pour réussir le développement d'une application, il est nécessaire de choisir les bons outils et technologies. De même les environnements qui garantissent une excellente implémentation. Dans ce chapitre, nous présentons la partie réalisation et mise en oeuvre de notre travail, ainsi les différents outils utilisés.

3.2 Outils et techniques de développement

3.2.1 Langage C :

Le C est un langage de programmation de bas niveau très populaire, crée dans les années 1970 par D.Ritchie et B.W.Kernighan. Il est portable, libre, faiblement typé (peu de types de variables différents : son fonctionnement est donc proche de l'ordinateur (gain en rapidité), mais un brin plus difficile à manipuler pour le programmeur). Le C n'est sans doute pas le langage le plus facile à apprendre (notamment à cause de l'adoption du concept parfois un peu obscure des pointeurs), ni le plus récent, mais ses qualités font de lui un langage incontournable en matière de programmation.



FIGURE 3.1 – Langage C

3.2.2 Interface graphique GTK :

GTK, Gimp ToolKit, boîte à outil utilisée d'abord par GIMP (General Image Manipulation Program), puis étendue pour pouvoir être utilisée pour n'importe quelle application. Il est sous

Licence LGPL , ce qui permet à tous de l'utiliser, de la faire évoluer, sans avoir à payer le moindre droit d'auteur et ce, même dans un contexte professionnel.



FIGURE 3.2 – GTK

3.2.3 Visual code :

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS². Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.



FIGURE 3.3 – Visual code studio

3.2.4 Glade :

Glade est un outil interactif de conception d'interface graphique GTK+. Il prend en charge toute la partie de gestion/génération de l'interface pour permettre au développeur de se concentrer sur le code « utile ». Glade enregistre les interfaces graphiques en générant des fichiers XML.



FIGURE 3.4 – Glade

3.2.5 Github :

GitHub est une plateforme open source de gestion de versions et de collaboration destinée aux développeurs de logiciels. Elle repose sur Git, un système de gestion de code open source créé par Linus Torvalds dans le but d'accélérer le développement logiciel.



FIGURE 3.5 – Github

3.3 Les difficultés :

Certaines difficultés ont été rencontrées lors de la mise en œuvre du projet. On a vu C'est la première expérience, et aussi le premier contact avec la programmation. Nous Citant les problèmes suivants :

3.3.1 Compréhension du sujet

En effet, les consignes avaient plusieurs ambiguïtés. D'où la difficulté de poser le plan et la stratégie de résolution

3.3.2 L'interface gtk :

L'utilisation de GTK nécessite beaucoup de documentation et beaucoup de travail, ce qui nous a permis de transformer nos idées en réalité et créer des fenêtres.

3.4 Les interfaces :

Les figures suivantes illustrent les différentes étapes de l'application.

3.4.1 Interface d'accueil :

La figure 3.6 présente la première interface affichée lors du lancement de l'application.



FIGURE 3.6 – Interface principale

3.4.2 La gestion des adhérents :

Interface d'accueil :



FIGURE 3.7 – Gestion des adhérents

La figure 3.7 présente l'interface a@chée lorsque l'administrateur clique sur le bouton "Gestion des adhérents"

Ajouter un adhérent :

La figure 3.8 présente l'interface a@chée lorsque l'administrateur clique sur le bouton "Ajouter un adherent".

Book's Info

SAISIR LES INFORMATIONS DE L'ADHERANT

Numero: 3

Nom:

Prénom:

Adherent est ajoute avec succes

Adresse mail : Email

Adresse personnelle : Adresse

Save

FIGURE 3.8 – Ajouter un adhérent

Modifier un adhérent :

BIBLIOENSIAS

Gestion des adherents

Ajouter un adherent

Entrez le numéro de l'adherent

Modifier

Chercher un adherent

Afficher la liste des adherents

Menu principal

FIGURE 3.9 – Modifier un adhérent

La figure 3.9 présente l'interface a@chée lorsque l'administrateur clique sur le bouton "Modifier un adhérent".

Supprimer un adhérent :

La figure 3.10 présente l'interface a@chée lorsque l'administrateur clique sur le bouton "Supprimer un adhérent".



FIGURE 3.10 – Supprimer un adhérent

Cas nominal : adherent supprimé



FIGURE 3.11 – Cas nominal : adherent supprimé

La figure 3.11 présente l'interface affichée si la suppression est bien faite.

Recherche et affichage d'un adherent :

La figure 3.12 présente l'interface affichant l'adhérent cherché.

BIBLIOENSIAS

INFORMATIONS DE L'ADHERENT

Numero : 2

Nom : Yahyaoui

Prénom : Mohamed

Adresse mail : Yahyaoui.Mohamed@gmail.com

Adresse personnelle : Rabat, Maroc

Nombre de livre empr : 0

FIGURE 3.12 – Recherche et affichage d'un adherent :

Liste des adhérents :

Numero	Nom	Prenom	Email	Adresse	Nombre de livre empruntes
1	Zemmouri	SalahEddine	zemm.salaheddine@gmail.com	Rabat, Maroc	0
2	Yahyaoui	Mohamed	Yahyaoui.Mohamed@gmail.com	Rabat, Maroc	0
3	Nom	Prenom	Email	Adresse	0

FIGURE 3.13 – Liste des

adhérents La figure 3.13 permet l'affichage des

différents adhérents.

3.4.3 La gestion des livres :

Interface d'accueil :

La figure 3.7 présente l'interface affichée lorsque l'administrateur clique sur le bouton "gestion des livres".



FIGURE 3.14 – Gestion des livres

Ajout, suppression, modification et recherche :

Toute ces opérations se font de la meme manière que celle de l’adhérent.

Le tri des livres :

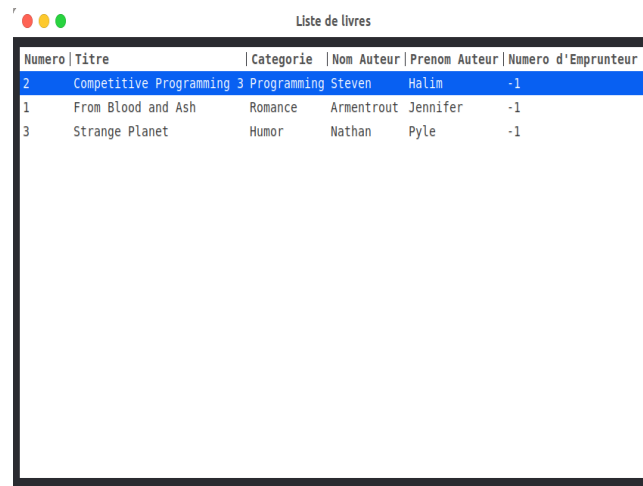


FIGURE 3.15 – Tri des livres

La figure 3.15 présente l’interface a@chée lorsque l’administrateur cherche à trier les livres par titre ou catégorie.

Liste des livres :

La figure 3.16 permet d’a@cher les différents livres.



Numero	Titre	Categorie	Nom Auteur	Prenom Auteur	Numero d'Emprunteur
2	Competitive Programming 3	Programming	Steven	Halim	-1
1	From Blood and Ash	Romance	Armentrout	Jennifer	-1
3	Strange Planet	Humor	Nathan	Pyle	-1

FIGURE 3.16 – Liste des livres

3.4.4 la gestion des emprunts :

Interface d'accueil :



FIGURE 3.17 – Gestion des emprunts

La figure 3.17 présente l'interface affichée lorsque l'administrateur clique sur le bouton "gestion des emprunts".

Emprunter un livre :

La figure 3.18 présente l'interface décrivant le processus de l'emprunt d'un livre.



FIGURE 3.18 – Emprunter livre

Rendre un livre :



FIGURE 3.19 – Rendre un livre

La figure 3.19 présente l'interface décrivant le processus de rendre un livre.

3.5 Conclusion :

Durant ce chapitre, on a décrit la dernière étape du développement qui est celle de la réalisation et la mise en œuvre du projet. D'abord on a présenté les outils utilisés, puis l'architecture logicielle de l'application en présentant le modèle en couches. Et pour finir on a donné quelques captures d'écran de l'application réalisée.

Conclusion générale

Notre projet a consisté à la réalisation d'une application qui sert à faciliter le flux de gestion des livres , adhérents ainsi que les emprunts.

Ce projet nous a permis de consolider et d'approfondir nos connaissances théoriques par la pratique des nouvelles technologies tout au long du processus du développement .

C'était une occasion de nous encourager à l'autoformation et à l'apprentissage de nouveaux atouts , notons également la réalisation et la fusion de l' interface graphique avec la logique du traitement du projet, ainsi, nous avons pu maîtriser le langage C et la bibliothèque GTK utilisée.

Bibliographie

- [1]<https://fr.coursera.org/>
- [2]<https://www.udemy.com/>
- [3]<https://www.pluralsight.com/>
- [4]<https://openclassrooms.com/fr/>
- [5]<https://stackoverflow.com/>
- [6]<https://github.com/>
- [7]<https://www.xmlmath.net/doculatem/>
- [8]<https://www.freecodecamp.org>