Spécification des Exigences Logicielles (SEL) pour l'Application E-Library

1. Introduction

1.1 But du Document

Ce document de spécification des exigences logicielles (SEL) décrit les fonctionnalités et les contraintes de l'application "E-Library". L'objectif est de fournir un système de gestion de bibliothèque convivial et efficace pour les bibliothécaires, les étudiants et les enseignants. Il détaillera les exigences fonctionnelles et non fonctionnelles nécessaires au développement du logiciel.

1.2 Portée du Produit

L'application E-Library est un système de gestion de bibliothèque numérique qui permettra aux utilisateurs (étudiants, enseignants et bibliothécaires) de gérer les livres, les emprunts, les retours, les réservations et les informations utilisateur. Le système inclura également une fonctionnalité de notification par e-mail simulée pour les mises à jour.

1.3 Glossaire

- **Livre (Book)** : Représente une copie physique individuelle d'un livre dans la bibliothèque, avec un bookId unique.
- **Utilisateur (User)** : Représente un usager de la bibliothèque (étudiant ou enseignant) avec un identifiant unique et des informations personnelles.
- **Bibliothécaire (Librarian)** : Utilisateur ayant des privilèges administratifs pour gérer les livres et les utilisateurs.
- **Étudiant/Enseignant (Student/Teacher)** : Utilisateur de la bibliothèque pouvant emprunter, retourner et réserver des livres.
- **ISBN** : Numéro international normalisé du livre, identifiant une édition/titre spécifique d'un livre.
- **Book ID** : Identifiant unique pour une copie physique spécifique d'un livre.
- **UUID** : Identifiant Universellement Unique, utilisé pour générer des identifiants uniques pour les livres et les utilisateurs.
- **FCFA** : Franc de la Communauté Financière Africaine, utilisé pour les calculs de pénalités.
- **QApplication** : Classe fondamentale pour les applications Qt avec une interface utilisateur graphique.
- **QMainWindow** : Fenêtre principale de l'application.
- **QVector** : Conteneur de données similaire à un tableau dynamique.

- **QString** : Chaîne de caractères Unicode.
- **QDate** : Classe pour la manipulation des dates.
- **QFile, QTextStream** : Classes pour la lecture et l'écriture de fichiers.
- **QMessageBox** : Boîte de dialogue pour afficher des messages.

2. Description Générale

2.1 Perspective du Produit

L'application E-Library est un système autonome, mais elle interagit avec le système de fichiers local pour la persistance des données. Elle est conçue pour fonctionner sur une seule machine et n'est pas une application web ou distribuée.

2.2 Fonctions du Produit

Les principales fonctions de l'application E-Library incluent :

- **Gestion des Livres** : Ajout, suppression, emprunt, retour et réservation de livres.
- **Gestion des Utilisateurs** : Ajout et recherche d'utilisateurs.
- **Persistance des Données** : Chargement et sauvegarde des données des livres et des utilisateurs à partir de fichiers texte.
- **Notifications** : Envoi simulé de mises à jour par e-mail aux utilisateurs enregistrés.
- **Interface Utilisateur** : Interface graphique distincte pour les bibliothécaires et les étudiants/enseignants.

2.3 Caractéristiques des Utilisateurs

- **Bibliothécaire** :
  - Nécessite un mot de passe (admin123) pour se connecter.
  - Peut ajouter de nouvelles copies de livres (titre, auteur, ISBN) et spécifier le nombre de copies.
  - Peut supprimer une copie spécifique d'un livre par son ID.
  - Peut envoyer des mises à jour simulées par e-mail à tous les utilisateurs enregistrés.
  - Peut voir toutes les copies physiques de livres avec leurs détails, y compris le statut d'emprunt/réservation et les pénalités.
- **Étudiant/Enseignant** :
  - Peut se connecter en fournissant son nom, son numéro de téléphone et son adresse Gmail. Si l'utilisateur n'existe pas, un nouveau compte est créé.
  - Peut emprunter une copie spécifique d'un livre par son ID.

- Peut retourner une copie spécifique d'un livre par son ID, avec calcul des pénalités.
- Peut réserver une copie spécifique d'un livre par son ID.
- Peut annuler une réservation par l'ID du livre.
- Peut voir toutes les copies physiques de livres avec leur statut (Emprunté, Réservé, Disponible).

2.4 Contraintes

- **Persistance des Données** : Les données des livres et des utilisateurs sont stockées dans des fichiers texte (books.txt et users.txt).
- **Gestion des ID** : Les bookId et userId sont générés automatiquement et sont uniques.
- **Pénalités de Retard** : Une pénalité de 100 FCFA est appliquée par jour de retard pour les livres retournés en retard.
- **Période d'Emprunt** : La période d'emprunt maximale est de 14 jours.
- **Fonctionnalité d'Email** : L'envoi d'e-mails est simulé et n'envoie pas de véritables e-mails.
- **Authentification** : Le mot de passe du bibliothécaire est codé en dur (admin123).

3. Exigences Fonctionnelles

3.1 Gestion des Livres

- **FR1 : Ajouter un Livre**
  - Le système doit permettre au bibliothécaire d'ajouter un livre en spécifiant son titre, son auteur et son ISBN.
  - Le système doit permettre au bibliothécaire d'ajouter un nombre spécifié de copies du livre. Chaque copie aura un bookId unique.
  - Le système doit sauvegarder les nouvelles copies de livres dans le fichier de données des livres.
- **FR2 : Supprimer un Livre**
  - Le système doit permettre au bibliothécaire de supprimer une copie physique spécifique d'un livre en utilisant son bookId.
  - Le système ne doit pas permettre la suppression d'un livre s'il est actuellement emprunté ou réservé.
  - Le système doit mettre à jour le fichier de données des livres après la suppression.
- **FR3 : Emprunter un Livre**
  - Le système doit permettre à un étudiant/enseignant d'emprunter une copie de livre disponible en utilisant son bookId et son userId.

- Le système doit définir le statut du livre comme emprunté, enregistrer l'ID de l'emprunteur, la date d'emprunt et la date de retour prévue (14 jours après la date d'emprunt).
- Le système ne doit pas permettre l'emprunt d'un livre déjà emprunté ou réservé.
- Le système doit sauvegarder les modifications dans le fichier de données des livres.

- **FR4 : Retourner un Livre**
- Le système doit permettre à un utilisateur de retourner une copie de livre empruntée en utilisant son bookId.
- Le système doit calculer une pénalité si la date de retour dépasse la date d'échéance (100 FCFA/jour de retard).
- Le système doit remettre le statut du livre à "non emprunté", effacer l'ID de l'emprunteur, la date d'emprunt et la date d'échéance.
- Le système doit sauvegarder les modifications dans le fichier de données des livres.

- **FR5 : Réserver un Livre**
- Le système doit permettre à un utilisateur de réserver une copie de livre disponible en utilisant son bookId et son userId.
- Le système doit définir le statut du livre comme réservé et enregistrer l'ID de l'utilisateur qui l'a réservé.
- Le système ne doit pas permettre la réservation d'un livre déjà emprunté ou réservé.
- Le système doit sauvegarder les modifications dans le fichier de données des livres.

- **FR6 : Annuler une Réservation**
- Le système doit permettre à un utilisateur d'annuler une réservation pour une copie de livre en utilisant son bookId.
- Le système doit remettre le statut du livre à "non réservé" et effacer l'ID de l'utilisateur qui l'a réservé.
- Le système doit sauvegarder les modifications dans le fichier de données des livres.

- **FR7 : Afficher les Livres**
- Le système doit afficher une liste de tous les livres disponibles et leur statut (emprunté, réservé, disponible) pour les étudiants/enseignants.
- Le système doit afficher une liste détaillée de tous les livres (copies physiques) avec titre, auteur, ISBN, Book ID, statut d'emprunt, emprunté par, date d'emprunt, date d'échéance, statut de réservation et réservé par, pour le bibliothécaire.

3.2 Gestion des Utilisateurs

- **FR8 : Ajouter un Utilisateur**
  - Le système doit permettre d'ajouter un nouvel utilisateur si les détails fournis (nom, numéro de téléphone, adresse Gmail) ne correspondent pas à un utilisateur existant.
  - Le système doit générer un id unique pour chaque nouvel utilisateur.
  - Le système doit sauvegarder les informations du nouvel utilisateur dans le fichier de données des utilisateurs.
- **FR9 : Rechercher un Utilisateur**
  - Le système doit permettre la recherche d'un utilisateur par son nom, numéro de téléphone et adresse Gmail.

3.3 Persistance des Données

- **FR10 : Charger les Données**
  - Au démarrage, le système doit charger les données des livres depuis books.txt et les données des utilisateurs depuis users.txt.
- **FR11 : Sauvegarder les Données**
  - À la fermeture de l'application ou après chaque modification pertinente (ajout/suppression de livre, emprunt/retour/réservation), le système doit sauvegarder les données des livres et des utilisateurs dans leurs fichiers respectifs.

3.4 Notifications

- **FR12 : Envoyer des Mises à Jour par E-mail**
  - Le système doit permettre au bibliothécaire de simuler l'envoi d'e-mails de mise à jour à toutes les adresses Gmail des utilisateurs enregistrés.

4. Exigences Non Fonctionnelles

4.1 Performances

- **NFR1.1 : Temps de Réponse** : Les opérations courantes (ajout, suppression, emprunt, retour, réservation, annulation de réservation) devraient être complétées en moins de 2 secondes.
- **NFR1.2 : Chargement des Données** : Le chargement des données des fichiers (books.txt, users.txt) doit être rapide, même avec un grand nombre d'enregistrements.

4.2 Sécurité

- **NFR2.1 : Authentification du Bibliothécaire** : L'accès aux fonctionnalités du bibliothécaire doit être protégé par un mot de passe.

- **NFR2.2 : Intégrité des Données** : Le système doit garantir l'intégrité des données stockées dans les fichiers, en évitant la corruption des données.

4.3 Utilisabilité

- **NFR3.1 : Interface Intuitive** : L'interface utilisateur doit être claire et facile à comprendre pour les bibliothécaires et les étudiants/enseignants.
- **NFR3.2 : Messages d'Erreur** : Le système doit fournir des messages d'erreur clairs et utiles en cas d'échec d'une opération.

4.4 Fiabilité

- **NFR4.1 : Persistance** : Toutes les données des livres et des utilisateurs doivent être persistantes entre les sessions de l'application.
- **NFR4.2 : Robustesse des Fichiers** : Le système doit gérer gracieusement les cas où les fichiers de données sont manquants ou corrompus (par exemple, en créant de nouveaux fichiers ou en affichant des avertissements).

4.5 Maintenabilité

- **NFR5.1 : Code Modulaire** : Le code doit être structuré de manière modulaire, permettant des modifications et des extensions faciles (par exemple, séparation de la logique métier, de l'interface utilisateur et de la persistance des données).
- **NFR5.2 : Documentation** : Le code source doit être bien commenté.

5. Exigences Techniques

5.1 Environnement de Développement

- **NFR6.1 : Langage de Programmation** : C++
- **NFR6.2 : Framework GUI** : Qt 5.x (avec modules Widgets, Core, Gui)
- **NFR6.3 : Outil de Construction** : QMake
- **NFR6.4 : Norme C++** : C++17

5.2 Stockage des Données

- **NFR7.1 : Format de Fichier** : Fichiers texte délimités par des pipes pour les données des livres et des utilisateurs.