

# ALGORITHME Gestion d'une Bibliothèque

Dans ce fichier, nous n'avons pas utilisé le langage de l'algorithme littéralement car dans le script, nous avons utilisé le langage sql, donc nous avons remplacé la majorité du script par des commentaires pour expliquer le rôle de chaque fonction. Nous espérons que vous comprenez la situation actuelle, merci.

Structure Database

Debut

```
    QSqlDatabase db
    QFile file
    QString path
```

Fin structure

*Procedure Database()*

Debut

```
    path = "/home/hamma/Project/qt/Gestion_Bibliotheque/Gestion";
```

```
    //Rq: path contient le chemin de la base Gestion
```

Fin

*Fonction dbConnect(): bool* //verifier la connection au base

Debut

```
    db = QSqlDatabase::addDatabase("QSQLITE") //c'est des langages sql il n'y a pas
                                              d'algorithme pour cette langage
```

```
    db.setDatabaseName(path)
```

```
    Si(!db.open()) alors
```

```
        Ecrire("error connection ")
```

```
        return faux;
```

```
    sinon
```

```
        Ecrire("connected ")
```

```
        return vrai;
```

```
    Fin si
```

Fin

*Procedure dbclose()* //Pour deconnecter, aussi c'est une langage sql

Debut

```
    QString connection
```

```
    connection = db.connectionName() //des fonctions prés definie
```

```
    db.close()
```

```
    db = QSqlDatabase()
```

```
    db.removeDatabase(connection)
```

Fin

Structure adduser //cette classe hérite de la classe Database

Debut

```
    adduser *ui
```

Fin structure

*Procedure adduser(QWidget \*parent)\_* //constructeur contient l'initialisation  
et une fonction prés definie pour fixer  
la date courante.

Debut

```
    ui->setUpUi(this)
```

```
    ui->date_ins->setDate(QDate::currentDate())
```

Fin

**Procedure** ~adduser() //distructeur

Debut

delete ui

Fin

**Procedure** on\_cancel\_clicked() //Pour annuler L'operation

Debut

close()

Fin

**Procedure** on\_save\_clicked()

Debut

//cette méthode utiliser La langage sql en utiisant Le requête **INSERT** pour inserer dans un tableau sous Le nom **UserInfo** Les informations de utilisateur

**INSERT INTO TABLE\_NAME [(column1, column2, column3,...columnN)]**

**VALUES (value1, value2, value3,...valueN);**

Fin

Structure addbook //cette classe hérite de La classe **Database**

Debut

adduser \*ui

QString imgPath

Fin structure

**Procedure** addbook(QWidget \*parent) //constructeur

Debut

ui->setupUi(this)

Fin

**Procedure** ~addbook() //distructeur

Debut

delete ui

Fin

**Procedure** addbook::on\_save\_clicked()

Debut

//cette méthode utiliser La langage sql en utiisant Le requête **INSERT** pour inserer dans un tableau sous Le nom **Books** Les informations de livre

**INSERT INTO TABLE\_NAME [(column1, column2, column3,...columnN)]**

**VALUES (value1, value2, value3,...valueN);**

Fin

**Procedure** on\_cancel\_clicked() //Pour annuler L'operation

Debut

close()

Fin

**Procedure** on\_cover\_clicked()

Debut

imgPath = QFileDialog::getOpenFileName(this,tr("open file"),"/home",tr("Images (\*.png \*.jpeg \*.jpg)"))

//Pour inserer La path d'image dans La base

Fin

Structure MainWindow //cette classe hérite de La classe **Database**

Debut

```

    MainWindow *ui;
    QString user //Rq: QString == type chaine
    QString titre,soustitre,domaine,date,auteur
    QPixmap cover
    QString id11
    QString userName
    QString id
    int id_book,id_taked

```

**Fin structure**

**MainWindow(QWidget \*parent) //constructeur**

**Debut**

**ui->setupUi(this)**

**Fin**

**Procedure ~MainWindow() //distructeur**

**Debut**

**delete ui**

**Fin**

**Procedure on\_pushButton\_clicked() // fonction Login, il y a deux interfaces, user et admin qui est capable de gerer Le contenu du bibliotheque**

**Debut**

**Si(!dbConnect()) alors //verifier si l'utilisateur est connecté à la BD**

**Ecrire(" not Connected to database")**

**return**

**Fin si**

**//cette méthode utiliser la langage sql en utilisant la requête SELECT pour selectionnee a partir d'un tableau sous le nom UserInfo Les informations de user**

**Si(le user et un super-user) alors**

**//on va afficher l'interface de l'admine**

**//Selectionner from Books tous les livres existent dans la bib**

**sinon //L'utilisateur est un etudiant ou un enseignant**

**//afficher tous les livres**

**//verifier si il y a un retard pour envoyer un alert**

**Fin si**

**Fin**

**Procedure on\_Log\_out2\_clicked()**

**Debut**

**//quitter la biblio**

**//Retourner a la page Login**

**Fin**

**Procedure on\_BookView\_clicked(const QModelIndex &index)**

**Debut**

**//dans cette fonction on vas afficher les info d'un livre dans la partie utilisateur**

**//selectionnet a partir du tableau Books les information correspondant a la livre clicker par l'utilisateur**

**Fin**

**Procedure on\_recherche\_clicked()**

**Debut**

**//Fonction gestion de recherche**

**//Selon le type de recherche(Simple/avancer)**

**Fin**

**Procedure** on\_ListWidget\_itemClicked(QListWidgetItem \*item)

Debut // pour la gestion de l'interface admin

Selon (partie)

0: //afficher les livres

//selectionner from la tab **Books** le liste des livres en utilisant le requête  
**SELECT**

1: //liste des livres emprunté:

//selectionner from la tab **Taked** le liste des livres emprunté en utilisant le requête **SELECT**

2: //liste des users"

//selectionner from la tab **UserInfo** le liste des utilisateurs en utilisant le requête **SELECT**

Fin

**Procedure** on\_chek\_avancer\_stateChanged(int arg1)

Debut //pour choisir entre recherche avancer et recherche simple

**Si**(le cas de recherche avancer et cocher) **alors**

//on va faire un rech avancer selon le type(date, domaine, auteur)

**sinon**

//une recherche simple selon le titre de livre

**Fin si**

Fin

**Procedure** on\_emp\_clicked()

Debut

Fin

**Procedure** on\_TakeBook\_clicked(const QModelIndex &index)

Debut

Fin

**Procedure** on\_addUser\_clicked()

Debut

Fin

**Procedure** on\_DeUser\_clicked()

Debut

Fin

**Procedure** on\_AdminUsers\_clicked(const &index : QModelIndex)

### Debut

*//cette méthode utilise la langage sql en utilisant la requette SELECT pour faire la selection de user\_id du*

*//tableau UserInfo où user\_id= id de la ligne où le curseur est placé*

*// la valeur de user\_id selecter est ainsi affectée à la variable globale id*

### Fin

**Procedure** *on\_DelUser\_clicked()*

### Debut

*//cette méthode utilise la langage sql en utilisant la requette DELETE pour effacer du*

*//tableau UserInfo où user\_id= id de la ligne où le curseur est placé*

### Fin

**Procedure** *on\_alert\_clicked()*

### Debut

*//cette méthode utilise la langage sql en utilisant la requette SELECT pour faire la selection de date et user\_id du*

*//tableau UserInfo où date > date limite*

### Fin

**Procedure** *on\_Book\_clicked(const &index : QModelIndex )*

### Debut

*//cette méthode utilise la langage sql en utilisant la requette SELECT pour faire la selection de tous les informations*

*//tableau Books où id\_book=id de la ligne où le curseur est placé*

### Fin

**Procedure** *on\_chek\_avancer\_2\_stateChanged(int)* *//pour choisir entre recherche avancé et recherche simple*

### debut

*si ui->chek\_avancer\_2->isChecked() == vrai*

*{*

*ui->Domaine->setEnabled(1);*

*ui->Date\_2->setEnabled(1);*

```
    ui->auteur_2->setEnabled(1);  
}
```

si non

```
{  
    ui->Domaine->setDisabled(1);  
    ui->Date_2->setDisabled(1);  
    ui->auteur_2->setDisabled(1);  
}
```

fin si

fin

**Procedure** *on\_recherche\_2\_clicked()*

debut

si ui->chek\_avancer\_2->isChecked() = vrai //pour une recherche avancée

si ui->Date\_2->text()==" et ui->auteur\_2->text()=="

*//SELECT Titre,SousTitre,Domaine,Nombre\_copie,Auteur du tableau Books où*

*//titre =ui->titre\_livre\_2->text()*

*//domaine = ui->Domaine->currentText());*

si non

si ui->Date\_2->text()==" et ui->auteur\_2->text()!="")

*//SELECT Titre,SousTitre,Domaine,Nombre\_copie,Auteur du tableau Books*

*//titre=ui->titre\_livre\_2->text()*

*//domaine=ui->Domaine->currentText()*

```
//auteur=ui->auteur_2->text()$
```

**si non**

```
si ui->Date_2->text()!=" et ui->auteur_2->text()=="
```

Books

```
//SELECT Titre,SousTitre,Domaine,Nombre_copie,Date,Auteur du tableau
```

```
//titre=ui->titre_livre_2->text()+"%
```

```
//domaine=ui->Domaine->currentText()
```

```
//auteur= ui->Date_2->text()
```

**si non**

tableau Books où

```
//SELECT Titre,SousTitre,Domaine,Nombre_copie,Date,Auteur du
```

```
//titre=ui->titre_livre_2->text()+"%
```

```
//domaine=ui->Domaine->currentText()
```

```
//auteur= ui->Date_2->text()
```

```
//date=ui->Date_2->text()
```

**fin si**

**fin si**

**si non //pour une recherche simple selon le titre de livre**

```
//SELECT Titre,SousTitre,Domaine,Nombre_copie,Date,Auteur du tableau Books où
```

```
//titre=ui->titre_livre_2->text()+"%";
```

**fin si**

**fin**

**Procedure** *on\_AdminBook\_clicked*(const &index :QModelIndex)

**Debut**

*//cette méthode utilise la langage sql en utilisant la requette SELECT pour faire la selection de id et titre du*

*//tableau Books où id= id de la ligne où le curseur est placé*

*// la valeur de id et titre selecter sont ainsi affectées au variables globales id et titre*

**Fin**

**Procedure** *on\_addBook\_clicked*()

**debut**

*addbook add;*

*add.setModal(true);*

*//cette méthode utilise la langage sql en utilisant la requette SELECT pour faire la selection de tous les informations du tableau books*

*model->setQuery(qry);*

*ui->AdminBook->setModel(model);*

*ui->AdminBook->horizontalHeader()->setSectionResizeMode( QHeaderView::Stretch);*

**fin**

**Procedure** *on\_deleteBook\_clicked*()

*//supprimer un livre*

**Debut**

*//cette méthode utilise la langage sql en utilisant la requette DELETE pour effacer du*

*//tableau Books où book\_id= id de la ligne où le curseur est placé*

**Fin**

**Procedure** *on\_rendreBook\_clicked*()

**Debut**

*//cette méthode utilise la langage sql en utilisant la requette DELETE pour effacer du*



*//tableau taked où book\_id= id de la ligne où le curseur est placé*

*// et update Le nombred'exemplaire du book*

**Fin**

**Procedure** *on\_mod\_book\_clicked()*

**Debut**

*//cette méthode utilise la langage sql en utilisant la requette update pour  
mettre a jour Le nombred'exemplaire du book*

**Fin**