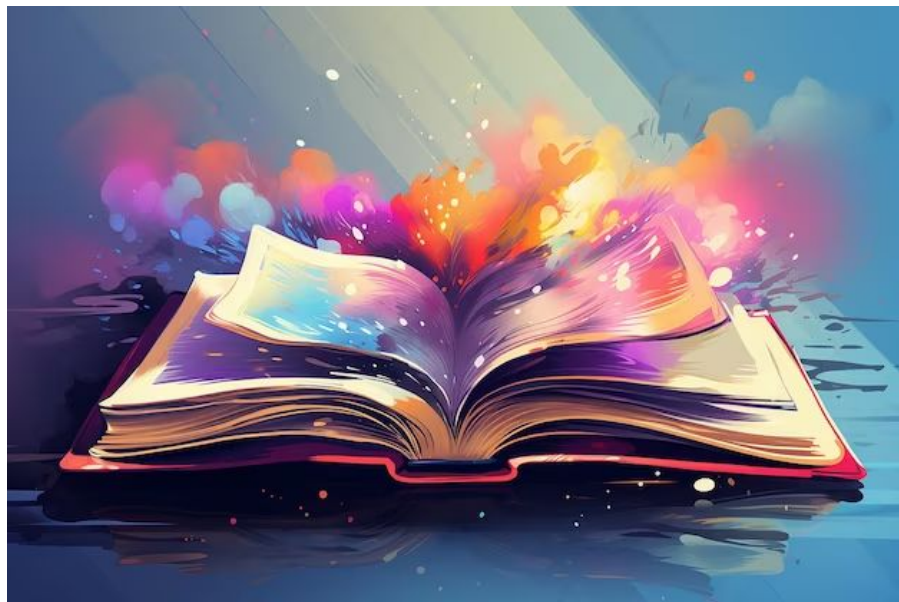


LE LIVRE DONT VOUS ÊTES LE HÉROS

ÉDITEUR



PAR LA SUPERMETALBESSE TEAM

TABLE DES MATIÈRES



01

COMPOSANTS &
INTERFACES

02

DIAGRAMME DE
SEQUENCE

03

DIAGRAMME DE
STRUCTURE INTERNE

04

TEST D'INTEGRATION

05

DIAGRAMME DE
CLASSE

06

SYNCHRONISATION
REVERSE

01.

INTERFACES

«interface» ISection

```
+ getTexte () : String
+ setTexte ( texte : String )
+ getIsDebut () : Boolean
+ getIsFin () : Boolean
+ setFin ( isFin : Boolean )
+ getIsAccessible () : Boolean
+ getObjetsObtenus () : IObjet [*]
+ setListeObjets ( listeObjets : IObjet [*] )
+ supprimerEnchainementsLiés () : Boolean
```

«interface» IEnchainement

```
+ getDescription () : String
+ setDescription ( description : String )
+ getTete () : ISection
+ setTete ( tete : ISection )
+ getQueue () : ISection
+ setQueue ( queue : ISection )
+ getListePrerequis () : IObjet [*]
+ setPrerequis ( prerequis : IObjet [*] )
```

«interface» ILivre

```
+ getTitre () : String
+ getAuteur () : String
+ getListeObjets () : IObjet [*]
+ verifObjetExist ( nomObjet : String ) : Boolean
+ addObjet ( nomObjet : String )
+ removeObjet ( nomObjet : IObjet )
+ getListeSections () : ISection [*]
+ verifPremiereSection () : Boolean
+ addSection ( isDebut : Boolean ) : Integer
+ setSection ( idSection : Integer, texte : String, isFin : Boolean, listeObjet : IObjet [*] ) : Boolean
+ removeSection ( idSection : Integer ) : Boolean
+ getListeEnchainements () : IEnchainement [*]
+ addEnchainement ( idTete : Integer, idQueue : Integer ) : Integer
+ setEnchainement ( idEnchainement : Integer, idTete : Integer, idQueue : Integer, description : String, prerequis : IObjet [*] ) : Boolean
+ removeEnchainement ( idEnchainement : Integer ) : Boolean
```

«interface» IImpression

```
+ genererHTML ( titreLivres : String, nom : String ) : Boolean
+ genererPDF ( titreLivres : String, nom : String ) : Boolean
```

«interface» ILivres

```
+ getListeLivres () : ILivre [*]
+ creerLivres ( titreLivres : String, auteur : String ) : Boolean
+ ouvrirLivres ( titreLivres : String ) : Boolean
+ verifLivresExist ( titreLivres : String ) : Boolean
+ creerObjet ( titreLivres : String, nom : String ) : Boolean
+ supprimerObjet ( titreLivres : String, nom : String ) : Boolean
+ creerSection ( titreLivres : String ) : Integer
+ modifierSection ( titreLivres : String, idSection : Integer, texte : String, isFin : Boolean, listeObjets : IObjet [*] ) : Boolean
+ supprimerSection ( titreLivres : String, idSection : Integer ) : Boolean
+ creerEnchainement ( titreLivres : String, idTete : Integer, idQueue : Integer ) : Integer
+ modifierEnchainement ( titreLivres : String, idEnchainement : Integer, idTete : Integer, idQueue : Integer, description : String, prerequis : IObjet [*] ) : Boolean
+ supprimerEnchainement ( titreLivres : String, idEnchainement : Integer ) : Boolean
+ genererPDF ( titreLivres : String, nom : String ) : Boolean
+ genererHTML ( titreLivres : String, nom : String ) : Boolean
```

«interface» IAnalyse

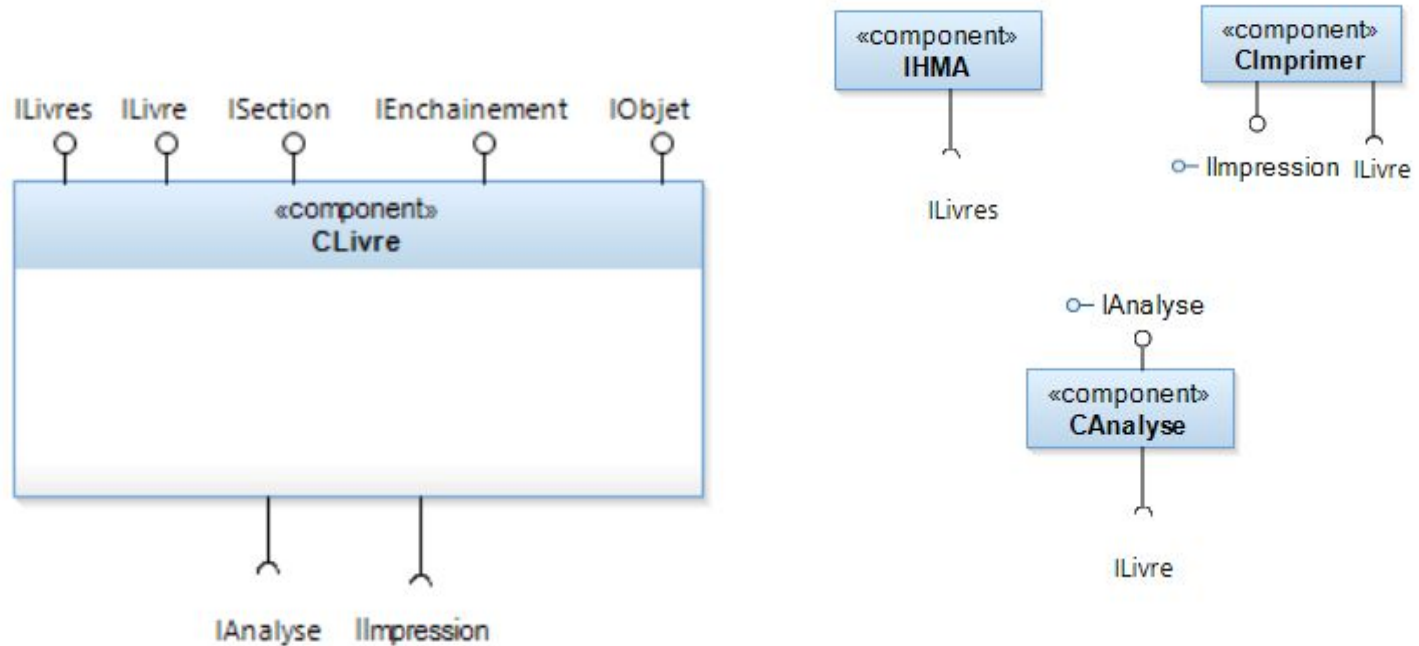
```
+ analyseGraphe ( titreLivres : String )
```

«interface» IObjet

```
+ getNom () : String
```

01.

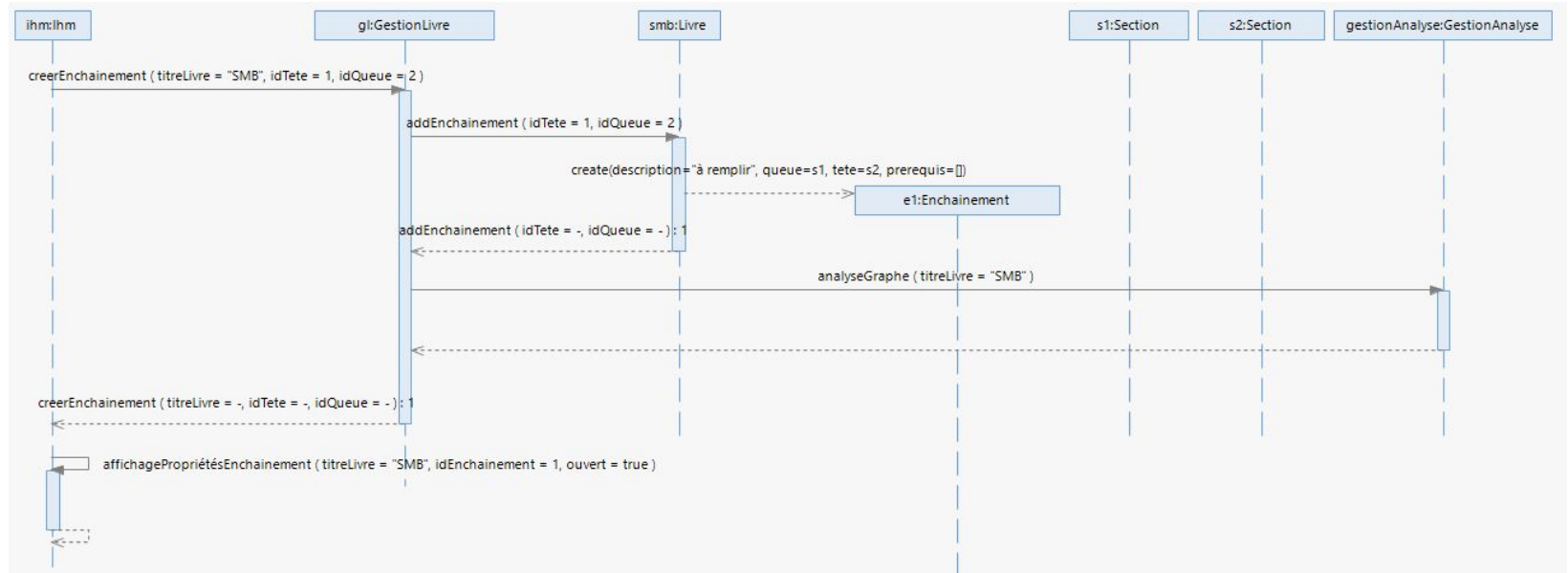
COMPOSANTS



02.

DIAGRAMMES DE SÉQUENCE

UC04 : Gérer un enchaînement : Scénario Nominal : Créer un enchaînement



SN1 : l'auteur choisit "Créer un enchaînement"

SN2 : l'auteur clique sur deux sections pour les lier

SN3 : le système crée l'enchaînement avec une description égale à "à remplir" par défaut

SN4 : le système appelle l'analyseur qui met à jour les sections inatteignables

SN5 : le système affiche l'enchaînement sur le graphe (voir maquette)

SN6 : le système affiche l'interface d'édition mise à jour

02.

DIAGRAMMES DE SÉQUENCE

UC04 : Gérer un enchaînement

Alternative 1 : Modifier un enchaînement

A1.1 : l'auteur clique sur un enchaînement déjà créé

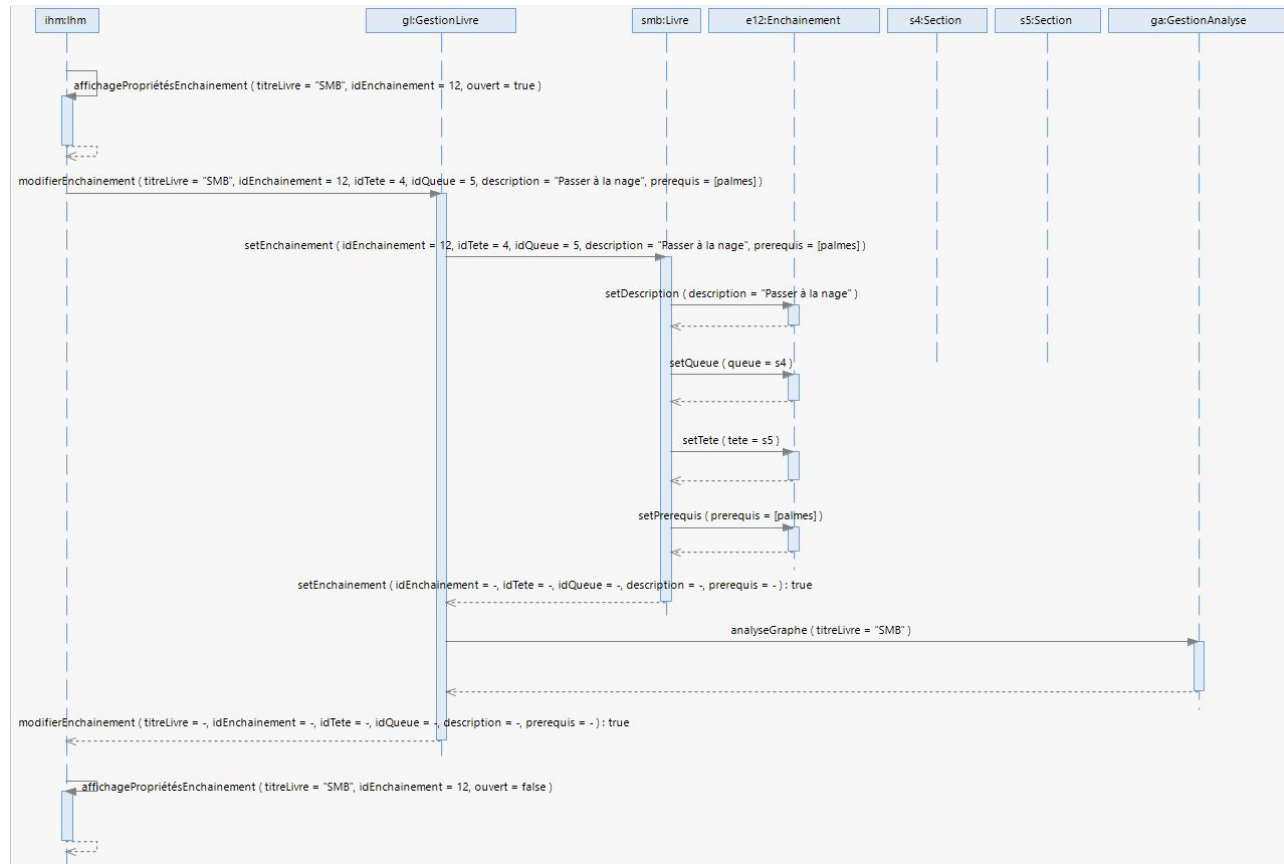
A1.2 : le système affiche les propriétés de l'enchaînement

A1.3 : l'auteur modifie un ou plusieurs champs et valide

A1.4 : le système vérifie la présence de la description et des sections de départ et d'arrivée

A1.5 : le système met à jour les modifications et ferme la fenêtre des propriétés

A1.6 : retour en SN4



02.

DIAGRAMMES DE SÉQUENCE

UC04 : Gérer un enchaînement

Alternative 2 : Supprimer un enchaînement

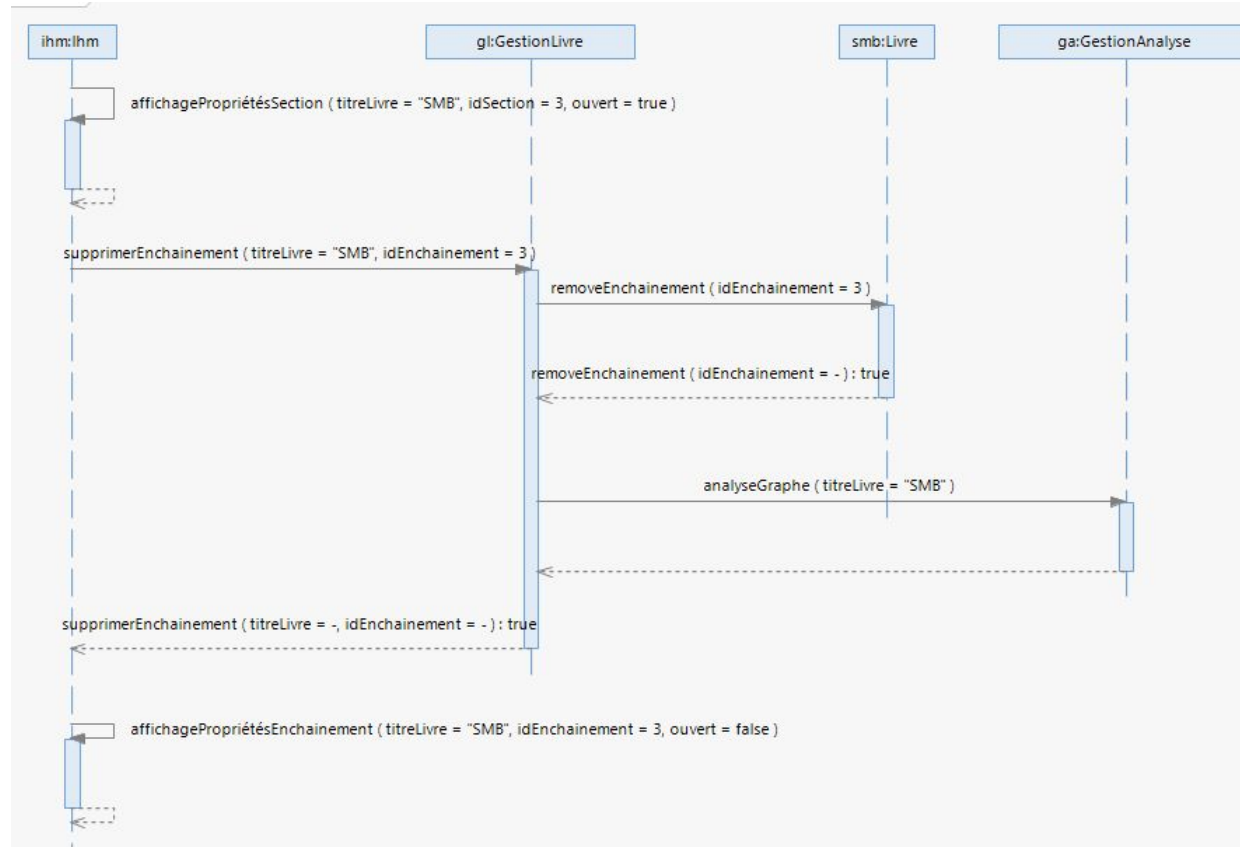
A2.1 : l'auteur clique sur un enchaînement déjà créé

A2.2 : le système affiche les propriétés de l'enchaînement

A2.3 : l'auteur choisit "Supprimer"

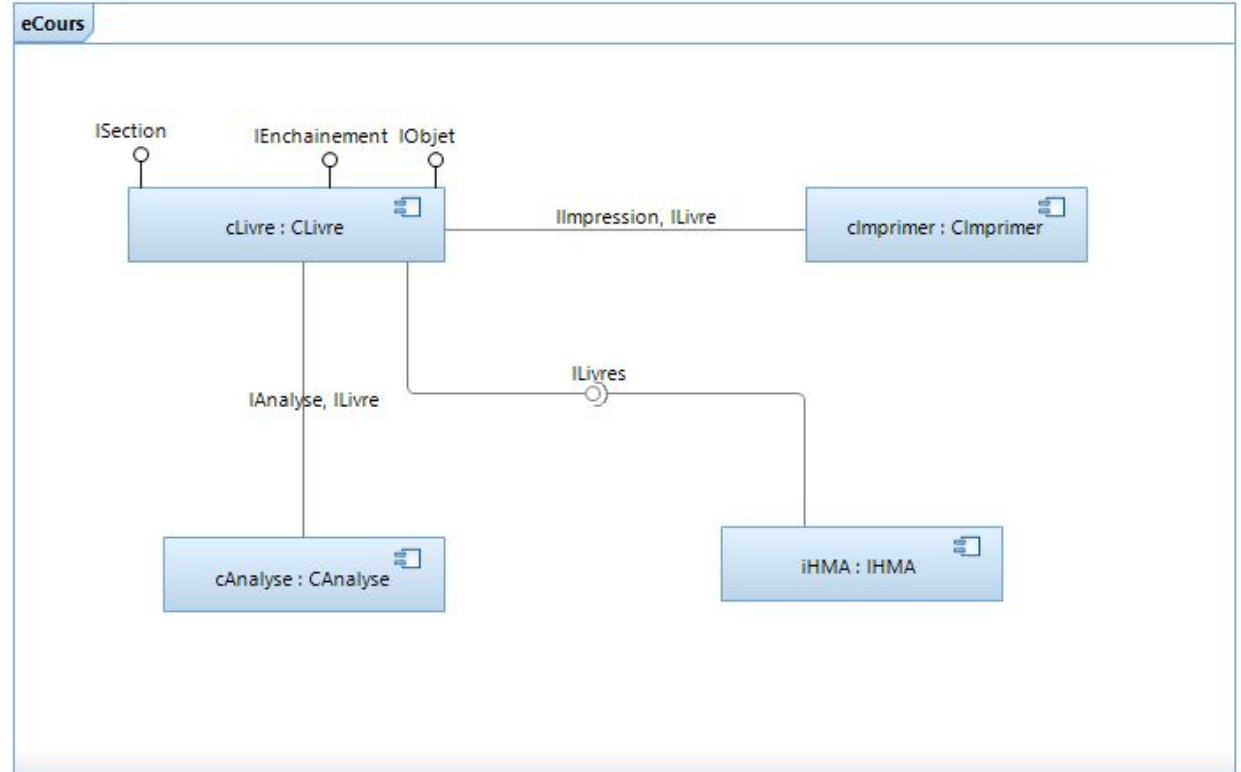
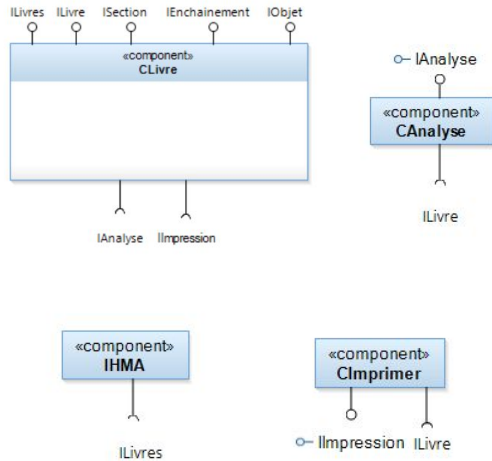
A2.4 : le système supprime l'enchaînement, met à jour la liste des enchaînements et ferme la fenêtre des propriétés

A2.5 : retour en SN4



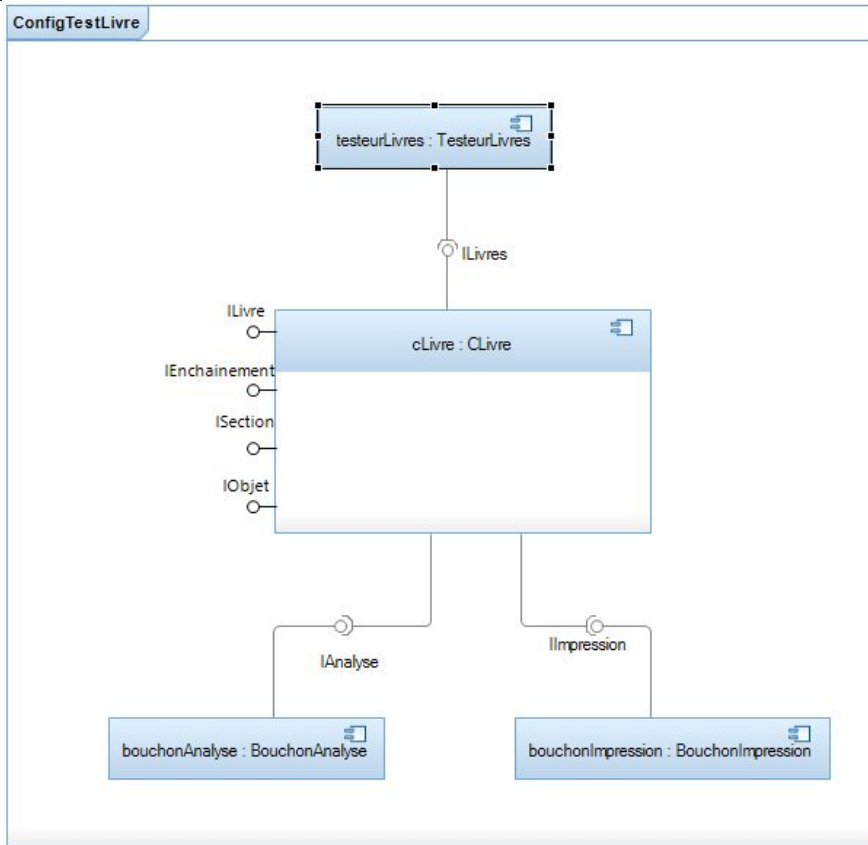
03.

DIAGRAMME DE STRUCTURE INTERNE



04.

TEST D'INTEGRATION



04.

TEST D'INTEGRATION



```
package Tests;
import static org.junit.Assert.*;

import java.util.HashSet;
import java.util.Set;

import org.junit.jupiter.api.Test;

import Clivre.*;
import Factory.Factory;
import Interfaces.*;

public class CTestLivre {

    @Test
    void testLivre() {

        IAnalyse mock_Analyse = Factory.newGestionAnalyse();
        IImpression mock_Impression = Factory.newImpression();
        ILivres s = Factory.newGestionLivre(mock_Analyse, mock_Impression);

        assertFalse(s.verifLivreExist("SMB"));
        assertTrue(s.creerLivre("SMB", "Alex"));
        ILivre l = s.getLivreOuvert();

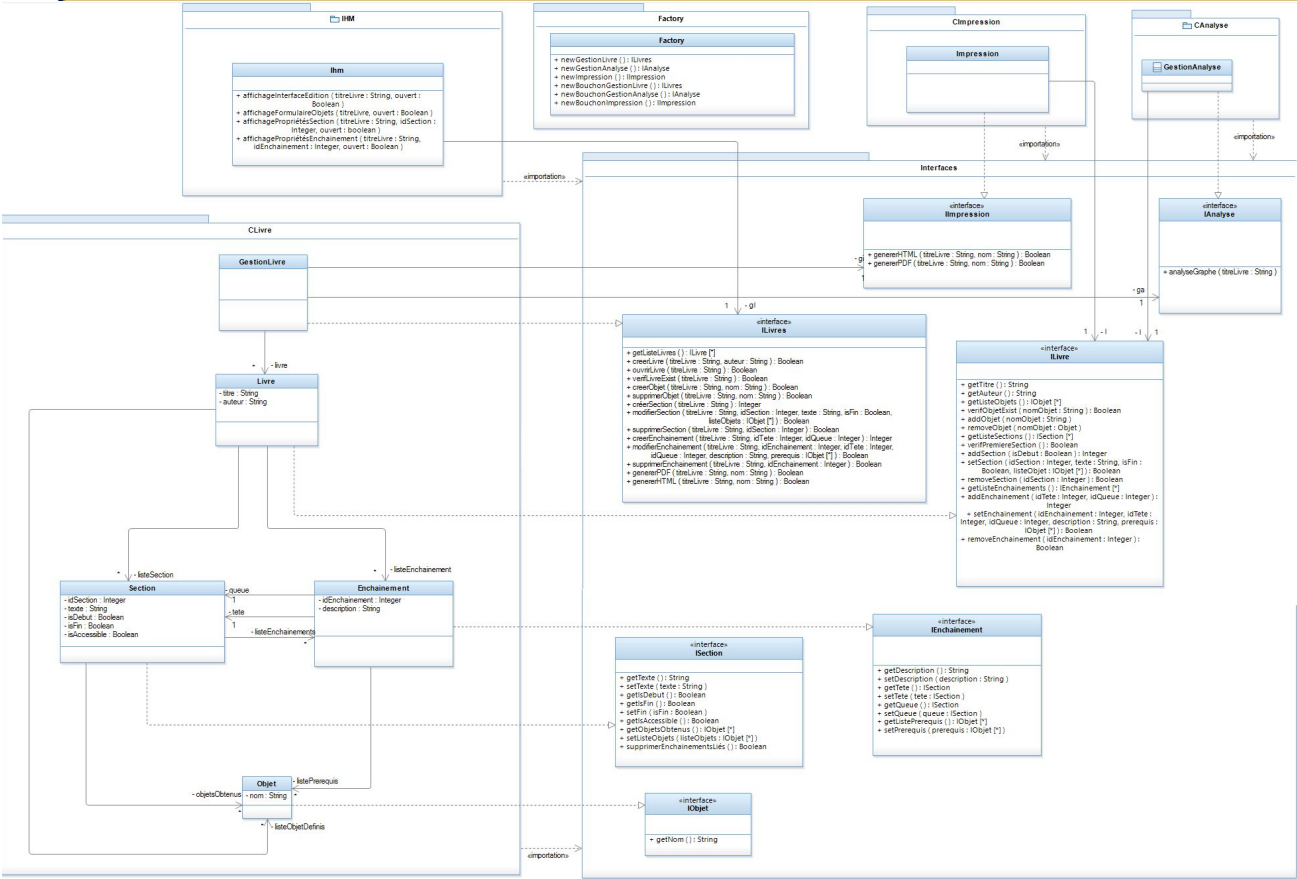
        assertEquals("Alex", l.getAuteur());
        assertFalse(l.verifObjetExist("baton"));
        assertTrue(l.verifPremiereSection());
        assert 0 == l.addSection(true);
        Set<IObjet> objets = new HashSet<>();
        objets.add(new Objet("couteau"));
        assertTrue(l.setSection(0,"entrer",false,objets));
        assert 1 == l.addSection(false);
        Set<IObjet> objets2 = new HashSet<>();
        objets2.add(new Objet("fourchette"));
        assertFalse(l.setSection(99,"manger les plats",true,objets2));
        assert 0 == l.addEnchainement(0,1);

    }

}
```

05.

DIAGRAMME DE CLASSE



06.

SYNCHRONISATION / REVERSE



```
public class GestionLivre implements ILivres {

    private static GestionLivre instance;
    private Set<ILivre> livres;
    private IAnalyse ga;
    private IImpression gi;
    private ILivre livreOuvert;

    private GestionLivre(IAnalyse ga, IImpression gi) {
        this.livres = new HashSet<>();
        this.ga = ga;
        this.gi = gi;
        this.livreOuvert = null;
    }

    // Méthode publique statique pour obtenir l'instance
    public static GestionLivre getInstance() {
        if (instance == null) {
            instance = new GestionLivre(new GestionAnalyse(), new Impression());
        }
        return instance;
    }

    @Override
    public Set<ILivre> getListeLivres() {
        return livres;
    }

    @Override
    public Boolean creerLivre(String titreLivre, String auteur) {
        if (this.verifLivreExist(titreLivre)) return false;
        Livre newLivre = new Livre(titreLivre, auteur);
        livres.add(newLivre);
    }
}
```

**MERCI DE VOTRE
ATTENTION!**

