

# TP Gestion de bibliothèque

## Auteurs du projet

**Isabel, Théo & Cédric**

Lien Github : <https://github.com/BamiGoreng/Bibliotheque>

## Présentation du projet

Cette application est destinée à une utilisation par des bibliothécaires pour les assister dans la gestion de leur livres ainsi que des emprunts de ces même livres.

Pour cela, l'application comporte deux modules :

1. Module gestion des livres
2. Module gestion des emprunts

Dans la suite de ce document, le bibliothécaire sera appelé « utilisateur » et la personne qui emprunte un livre « emprunteur ».

## Gestion des livres

Ce module permet d'ajouter un livre à la base de donnée des livres de la bibliothèque et d'en afficher le contenu.

Pour ajouter un livre, l'utilisateur devra fournir plusieurs données à l'application :

- Titre
- Année d'édition
- Éditeur
- Prénom de l'auteur
- Nom de l'auteur

Chaque nouveau livre créé se voit attribué un identifiant automatiquement par la base de donnée.

Cela permet de distinguer des livres en tout points identiques au niveau des données entrées par l'utilisateur.

L'utilisateur peut également afficher le contenu de la bibliothèque sous forme d'un tableau avec, pour colonnes, les informations rentrées par l'utilisateur et une colonne supplémentaire affichant l'identifiant de chaque livre.

## Gestion des emprunts

Ce module permet de créer un emprunt, d'afficher l'ensemble des emprunts et d'ajouter une date de restitution d'un livre.

Pour créer un emprunt, l'utilisateur devra fournir plusieurs données à l'application :

- ID du livre emprunté
- Nom de l'emprunteur

Chaque nouvel emprunt créé se voit attribué un identifiant automatiquement par la base de donnée.

L'utilisateur peut également afficher le suivi des emprunts sous forme d'un tableau avec, pour colonnes, les informations rentrées par l'utilisateur et des colonnes supplémentaires affichant l'identifiant de l'emprunt et sa date de restitution.

Enfin, l'utilisateur peut modifier la date de restitution d'un emprunt lorsque le livre est rendu.

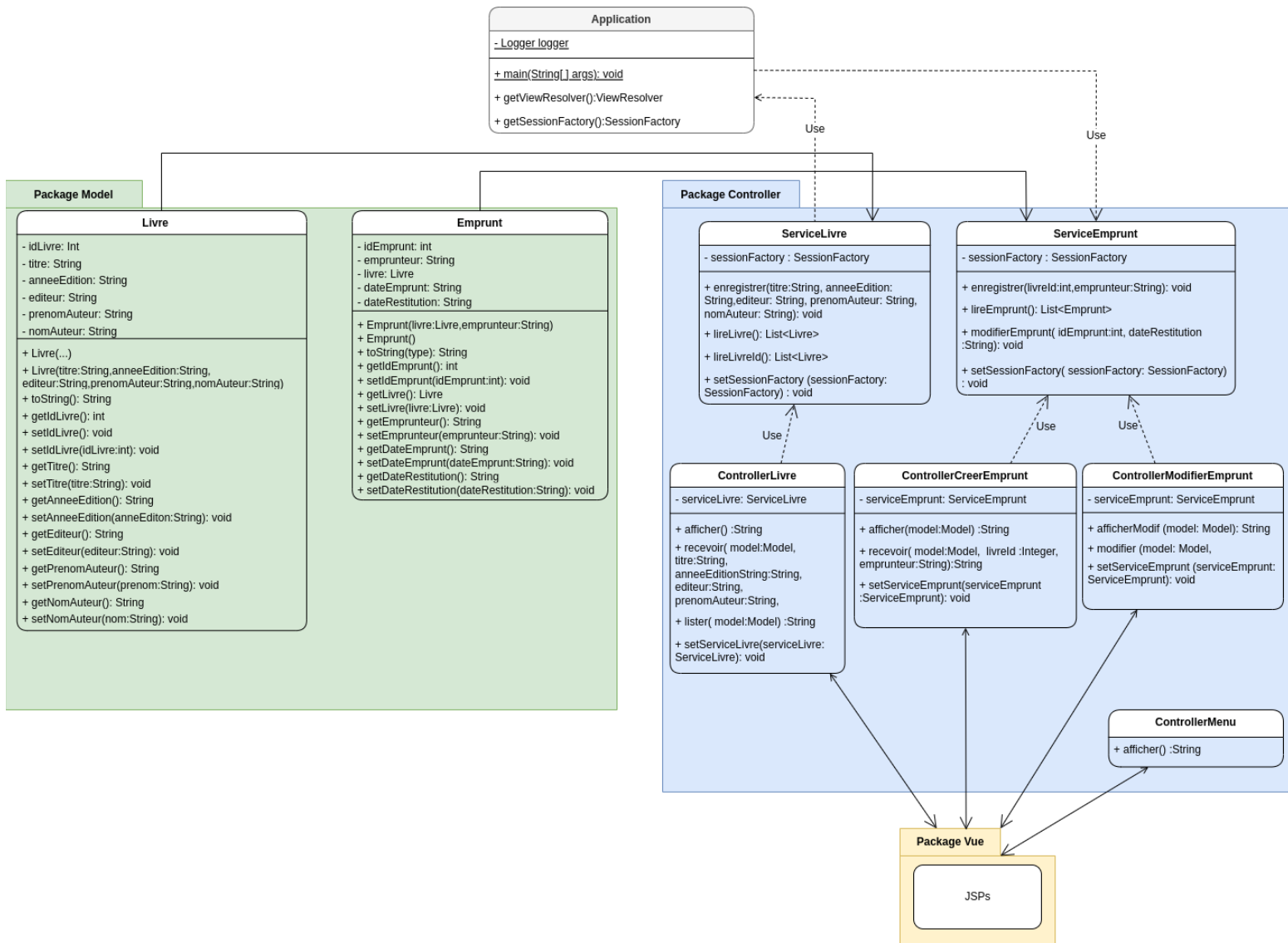
Pour restituer un emprunt, l'utilisateur devra fournir plusieurs données à l'application :

- ID de l'emprunt
- Date de restitution

## Diagramme des use cases

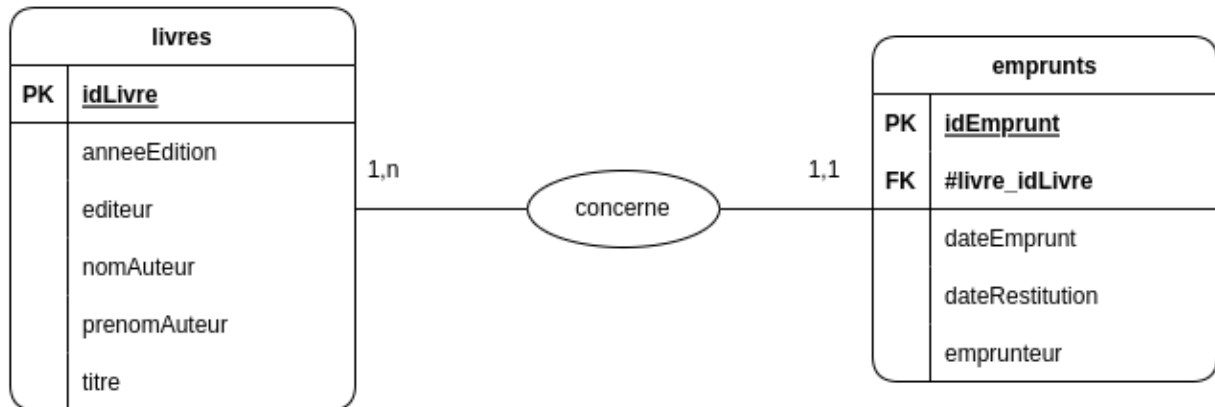


# Diagramme de classes



## Diagramme de base et schéma de table

### Diagramme de base



### Schéma de table

- Livre(IdLivre, anneeEdition, editeur, nomAuteur, prenomAuteur, titre)
- Emprunt(IdEmprunt, #livre\_idLivre, dateEmprunt, dateRestitution, emprunteur)

# Projet d'interface graphique

## Menu principal

**Gestionnaire de bibliothèque**

- Créer un livre
- Créer un emprunt, voir les emprunts en cours & modifier un emprunt

## 1ere page

**Créer un livre**

Titre

Année

Editeur

Nom auteur

Prenom auteur

## 2eme page

**Créer un emprunt** **Liste des emprunts en cours** **Modifier un emprunt**

Titre du livre

Emprunteur

Date d'emprunt

**(Liste des livres disponibles (2))**

| Id  | Titre | Année | Editeur | Nom auteur | Prenom auteur |   |
|-----|-------|-------|---------|------------|---------------|---|
| xxx | xxx   | xxx   | xxx     | xxx        | xxx           | <input type="button" value="Selectionner"/> |
| xxx | xxx   | xxx   | xxx     | xxx        | xxx           | <input type="button" value="Selectionner"/> |

## 2eme page

**Créer un emprunt** **Liste des emprunts en cours** **Modifier un emprunt**

**Liste des livres empruntés**

| Id  | Titre | Année | Editeur | Nom auteur | Prenom auteur | Date limite          | Perdu                               |   |
|-----|-------|-------|---------|------------|---------------|----------------------|-------------------------------------|---|
| xxx | xxx   | xxx   | xxx     | xxx        | xxx           | <input type="text"/> | <input checked="" type="checkbox"/> | <input type="button" value="Modifier"/> |



**Créer un emprunt** **Liste des emprunts en cours** **Modifier un emprunt**

**Liste des livres empruntés (ordre chronologique)**

| Id  | Titre | Année | Editeur | Nom auteur | Prenom auteur | Date de d'emprunt | Date de restitution  |                                       |
|-----|-------|-------|---------|------------|---------------|-------------------|----------------------|---------------------------------------|
| xxx | xxx   | xxx   | xxx     | xxx        | xxx           | xxx               | <input type="text"/> | <input type="button" value="Rendre"/> |

## Planning de réalisation

### Légende :

- Tâches attendues (soulignées)
- Tâches effectuées 
- Tâches non effectuées 

| Gestion des livres    |                   |                               |                    |
|-----------------------|-------------------|-------------------------------|--------------------|
| <u>Créer un livre</u> | Lister les livres | Modifier les infos d'un livre | Supprimer un livre |

| Gestion des emprunts    |                            |  |                      |
|-------------------------|----------------------------|--|----------------------|
| <u>Créer un emprunt</u> | <u>Lister les emprunts</u> | <u>Modifier les infos d'un emprunt</u> | Supprimer un emprunt |

État d'avancement : 62.5 %

### Détail des tâches :

| Tâche                                | Temps estimé (h) | Temps passé (h) |
|--------------------------------------|------------------|-----------------|
| Diagramme Use Case                   | 6                | 6               |
| Schéma d'interface graphique         | 6                | 6               |
| Diagramme de classes                 | 8                | 8               |
| Diagramme de base + schéma de tables | 3                | 3               |
| Configuration du projet dans Eclipse | 3                | 6               |
| Code Model/View/Controller           | 45               | 45              |
| CSS                                  | 10               | 10              |
| Tests                                | 3                | 4               |
| Présentation projet/montage dossier  | 21               | 17              |
| Total                                | 105              | 105             |
| Total / personne                     | 35               | 35              |

## Tableau de tests

| No. Test                   | Action                                      | Attendu  | Résultat                            |
|----------------------------|---|--|-------------------------------------|
| <b>Ajouter livre</b>       |   |  |                                     |
| 1                          | Mettre un chiffre dans le titre             | Accepter uniquement des lettres                                  | Accepte des chiffres et des lettres |
| 2                          | Mettre un chiffre dans le prénom d’auteur   | Accepter uniquement des lettres                                  | Accepte des chiffres et des lettres |
| 3                          | Mettre un chiffre dans le nom d’auteur      | Accepter uniquement des lettres                                  | Accepte des chiffres et des lettres |
| 4                          | Ne pas remplir tous les champs              | Ne peut pas envoyer le formulaire + afficher un message d’erreur | OK attendu                          |
| 5                          | Remplir avec des caractères spéciaux (%\$#) | Accepter les caractères spéciaux                                 | OK attendu                          |
| 6                          | Création de doublons                        | Accepte les doublons   | OK attendu                          |
| <b>Liste des livres</b>    |   |  |                                     |
| 7                          | Affichage de longs champs                   | La liste garde une bonne présentation                            | OK attendu                          |
| <b>Ajouter un emprunt</b>  |   |  |                                     |
| 8                          | Ajouter un emprunt                          | Ajout à la liste des emprunts                                    | OK attendu                          |
| 9                          | Emprunter un livre déjà emprunté            | Ne pas pouvoir emprunter le livre                                | Emprunt possible                    |
| 10                         | La même personne emprunte le même livre     | Ne pas pouvoir emprunter le livre                                | Emprunt possible                    |
| 11                         | Envoi du formulaire incomplet               | Pas possible   | OK attendu                          |
| 12                         | Mettre un chiffre dans le nom emprunteur    | Pas possible   | Possible                            |
| 13                         | Mettre l’ID d’un livre qui n’existe pas     | Pas possible   | OK attendu                          |
| 14                         | Mettre un id livre négatif                  | Pas possible   | OK attendu                          |
| <b>Modifier un emprunt</b> |   |  |                                     |



|    |  |              |            |
|----|--|--------------|------------|
| 15 | Saisie d'une date de restitution antérieur à celle d'emprunt | Possible     | OK attendu |
| 16 | Saisir ID d'emprunt qui n'existe pas                         | Pas possible | Possible   |

## Tests Junit

```
public class LivreTest {  
    //Test pour tester les méthodes de la class Livre  
    @Test  
    public void testGetId() {  
        //1er test pour obtenir l'identifiant d'un livre  
        //Au moment de la création d'un objet Livre son identifiant est 0  
        //Et il ne s'autoincrémente que quand il est persisté en base de donnée  
        Livre l1=new Livre("L'Homme chauve-souris","02-06-1997","Gaïa","Jo","Nesbo");  
        Livre l2=new Livre("L'Homme chauve-souris","02-06-1997","Gaïa","Jo","Nesbo");  
        int obtenu = l1.getId();  
        assertEquals(0, obtenu);  
        int obtenu2= l2.getId();  
        assertEquals(0,obtenu2);  
    }  
  
    @Test  
    public void testToString() {  
        //2d test pour afficher des attributs d'un objet Livre  
        //pour un Livre crée nous comparons l'affichage attendu avec celui que le test  
        //génère.  
        Livre l=new Livre("L'Homme chauve-souris","02-06-1997","Gaïa","Jo","Nesbo");  
        String obtenu = l.toString();  
        assertEquals("Livre [idLivre=0, titre=L'Homme chauve-souris, anneeEdition=1997,"  
            + " editeur=Gaïa, prenom=Jo, nom=Nesbo]",obtenu);  
    }  
}
```

```
}
```

```
@Test
```

```
public void testGetEditeur() {
```

```
    //3e test pour faire le test que le champ Editeur reste vide si laissé vide
```

```
    Livre l=new Livre("L'Homme chauve-souris","02-06-1997",null,"Jo","Nesbo");
```

```
    String obtenu=l.getEditeur();
```

```
    assertEquals(null,obtenu);
```

```
}
```

```
@Test
```

```
public void testSetNomAuteur() {
```

```
    //4e test pour vérifier que nous pouvons changer le nom du champ Nom dans
```

```
    //la class Livre
```

```
    Livre l=new Livre("L'Homme chauve-souris","02-06-1997",null,"Jo","Nesbo");
```

```
    l.setNomAuteur("Arnald");
```

```
    String obtenu =l.getNomAuteur();
```

```
    assertEquals("Arnald",obtenu);
```

```
}
```

```
}
```