



GESTION DES LIVRES (CASSANDRA- SPRING BOOT ET ANGULAR)



BINOME:

Mohammed KHARMICHI

Aya FELLAH

ENCADRANT:

Youness OUBENAALLA

Table des matières

I.	Analyse des besoins.....	3
II.	Présentation du système.....	4
1.	Spring boot.....	5
2.	Angular.....	10
III.	Présentation des interfaces.....	12
1.	Ajouter un livre	13
2.	Modifier un livre	14
3.	Supprimer un livre.....	15
4.	Rechercher un livre	15

Table des figures

Figure 1: Schéma de l'architecture du système	4
Figure 2: Architecture de l'application Spring Boot.....	5
Figure 3: Model du projet Spring Boot.....	6
Figure 4: CassandraRepository du projet Spring Boot	6
Figure 5: Service du projet Spring Boot	7
Figure 6: Controller du projet Spring Boot.....	8
Figure 7: Figure 7 : Arborescence du projet Spring Boot.....	9
Figure 8: Architecture de l'application Angular	10
Figure 9: Arborescence du projet Angular	11
Figure 10: Affichage de la liste des livres.....	12
Figure 11: Affichage de la liste des livres dans la base de données Cassandra	12
Figure 12: Formulaire d'ajout d'un livre	13
Figure 13: L'ajout du livre (Sauve-moi) dans la base de données Cassandra	13
Figure 14: Formulaire de modification d'un livre	14
Figure 15: La modification du livre (Sauve-moi) dans la base de données Cassandra	14
Figure 16: La suppression du livre.....	15
Figure 17: La suppression du livre (Sauve-moi) de la base de données Cassandra	15
Figure 18: La recherche du livre (la vie secrète des écrivains).....	15

I. Analyse des besoins

Le besoin de notre projet est de développer une application utilisant les technologies Spring Boot, Angular et Cassandra pour gérer une table "livre" avec les attributs suivant:

- ID: une clé primaire unique de type UUID, permettant d'identifier de manière unique chaque livre dans la base de données Cassandra.
- Titre: le titre du livre.
- Auteur: l'auteur du livre.
- Date d'apparition: la date à laquelle le livre a été publié.
- Édition: l'édition du livre.
- Genre: le genre auquel le livre appartient.

L'objectif du projet est de créer une application web qui offre les fonctionnalités CRUD (Create, Read, Update, Delete) pour gérer les livres. Pour cela, nous utiliserons Spring Boot en tant que framework côté serveur pour la gestion des requêtes et des opérations sur la base de données Cassandra. De plus, nous utiliserons Angular comme framework côté client pour créer une interface utilisateur interactive et conviviale.

Les fonctionnalités principales de l'application incluront l'ajout de nouveaux livres, la suppression des livres existants, la modification des attributs des livres, la recherche de livres en fonction de divers critères tels que le titre, l'auteur, la date d'apparition, l'édition ou le genre, ainsi que l'affichage de tous les livres disponibles.

II. Présentation du système

En utilisant la combinaison de Spring Boot et Angular, il est possible de créer une application web complète avec une architecture de type "client-serveur". Cette méthode permet de développer le backend avec Spring Boot et le frontend avec Angular. Cette architecture offre une séparation nette des responsabilités entre le frontend et le backend, ce qui facilite la maintenance et l'évolutivité de l'application. En effet, cette méthode permet de travailler sur les deux parties de manière indépendante, tout en garantissant une communication fluide entre les deux.

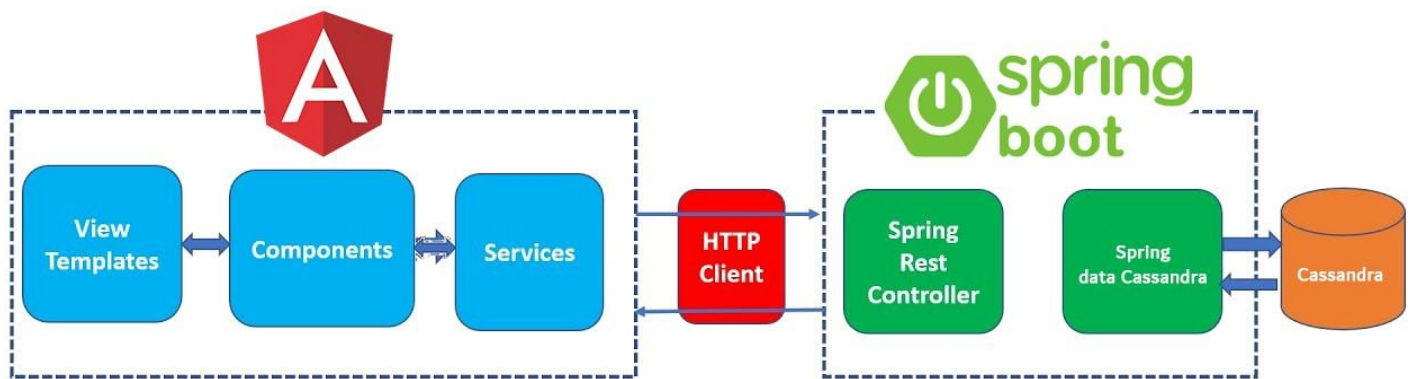


Figure 1: Schéma de l'architecture du système

1. Spring boot

Spring Boot est un framework Java utilisé pour créer le backend d'une application web. Il fournit une infrastructure robuste pour gérer la logique métier, la base de données et les services RESTful.

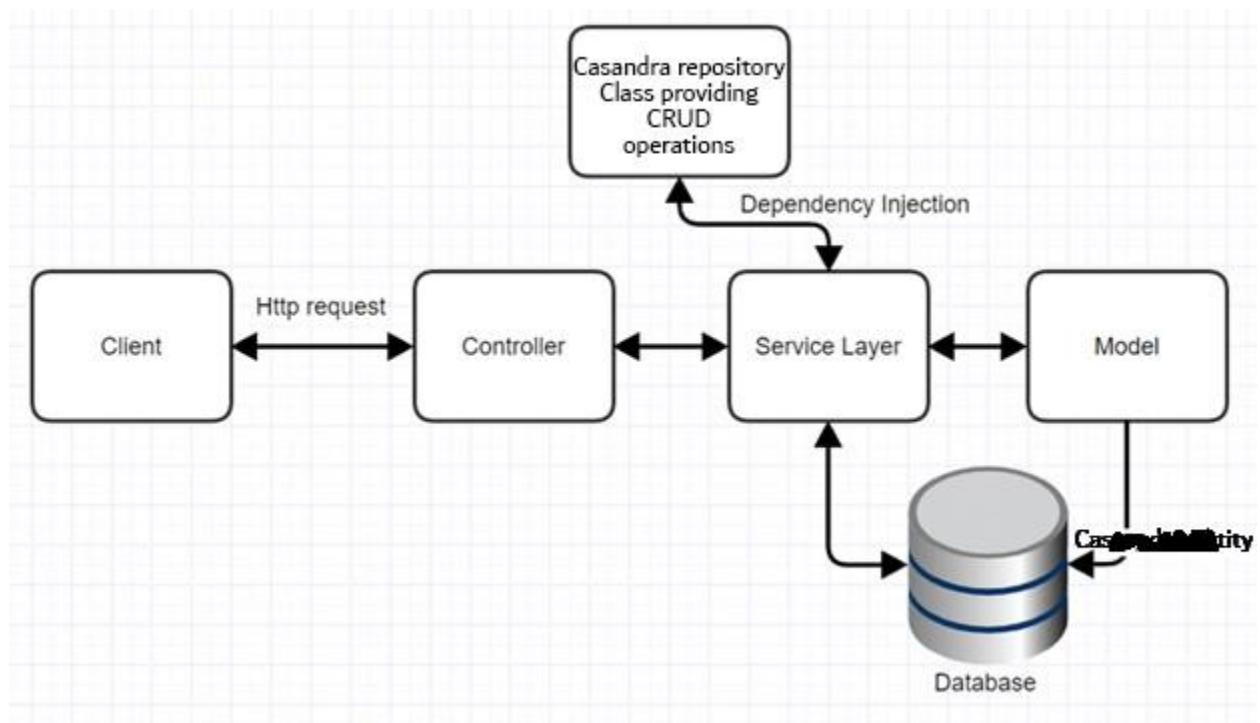


Figure 2: Architecture de l'application Spring Boot

Dans une RESTful API, plusieurs composants clés (Model, Service, Controller, CassandraRepository) sont utilisés pour structurer et organiser l'application.

- **Model :**

Le modèle (model) est la représentation des données de l'application. Il contient les classes POJO (Plain Old Java Object) qui représentent les entités métier de l'application, telles que les livres. Les classes POJO sont généralement annotées avec des annotations telles que `@Table`, `@PrimaryKey`, etc. pour faciliter leur utilisation avec une base de données NoSQL.

```
@Table
public class Livre {
    @PrimaryKey
    private UUID id = UUID.randomUUID();

    private String titre;

    private String auteur;

    private LocalDate date;

    private String edition;

    private String genre;

    public Livre() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```

Figure 3: Model du projet Spring Boot

- **Cassandra Repository :**

Le repository Cassandra est une interface qui facilite les opérations de persistance et d'interaction avec la base de données Cassandra, en fournissant des méthodes prédéfinies pour effectuer des opérations CRUD sur les entités. Il simplifie l'accès et la manipulation des données dans Cassandra.

```
public interface LivreRepository extends CassandraRepository<Livre, UUID> {

}
```

Figure 4: CassandraRepository du projet Spring Boot

- **Service :**

Le service (service) est la couche intermédiaire entre le contrôleur et le modèle. Il contient la logique métier de l'application et utilise le modèle pour effectuer des opérations sur les données. Pour interagir avec la base de données, les services utilisent des objets de type `CassandraRepository`, qui sont des interfaces fournies par Spring Data Cassandra et qui fournissent des méthodes pour effectuer des opérations de base de données telles que la création, la lecture, la mise à jour et la suppression des entités.

```
@Service
public class LivreServiceImpl implements LivreService {

    @Autowired // permet d'injecter LivreRepository
    private LivreRepository livreRepository;

    public LivreServiceImpl(LivreRepository livreRepository) {
        super();
        this.livreRepository = livreRepository;
    }

    public Livre saveLivre(Livre livre) {
        // TODO Auto-generated method stub
        return livreRepository.save(livre);
    }

    public List<Livre> getAllLivre() {
        // TODO Auto-generated method stub
        return livreRepository.findAll();
    }

    @Override
    public Livre updateLivre(Livre livre) {
        // TODO Auto-generated method stub
        return livreRepository.save(livre); // contient 1 id
    }

    public void deleteLivre(UUID id) {
        // TODO Auto-generated method stub
        livreRepository.deleteById(id);
    }

    @Override
    public List<Livre> RechercherLivre(String query) {
        // TODO Auto-generated method stub
        return null;
    }
}
```

Figure 5: Service du projet Spring Boot

- **Controller :**

Le contrôleur (controller) est la couche la plus haute du backend et elle est responsable de la gestion des requêtes HTTP entrantes et des réponses sortantes. Les contrôleurs exposent des endpoints sont des adresses URL spécifiques qui permettent aux clients de communiquer avec l'application en utilisant le protocole HTTP. Les contrôleurs sont généralement implémentés en utilisant des classes annotées avec des annotations telles que `@RestController`, `@GetMapping`, `@PostMapping`, etc. pour définir les endpoints et les méthodes qui traitent les requêtes et les réponses.

```
@RestController
@RequestMapping("api/livres")
public class LivreController {
    • @Autowired
      private LivreService livreService;

    • public LivreController(LivreService livreService) {
        super();
        this.livreService = livreService;
    }

    • @PostMapping
      public ResponseEntity<Livre> saveLivre(@RequestBody Livre livre) {
        return new ResponseEntity<Livre>(livreService.saveLivre(livre), HttpStatus.CREATED);
      }

    • @GetMapping
      public List<Livre> getAllLivres() {
        return livreService.getAllLivre();
      }

    • @PutMapping("{id}")
      public Livre updateLivre(@PathVariable("id") UUID id, @RequestBody Livre livre) {
        livre.setId(id);
        return livreService.updateLivre(livre);
      }

    • @DeleteMapping("{id}")
      public List<Livre> deleteLivre(@PathVariable("id") UUID id) {
        livreService.deleteLivre(id);
        return livreService.getAllLivre();
      }
}
```

Figure 6: Controller du projet Spring Boot

✓ *Arborescence du projet*

La structuration du projet Spring Boot en packages permet de séparer les différentes couches de l'application. Parmi ces packages, on retrouve un dédié aux contrôleurs pour la gestion des requêtes HTTP, ainsi qu'un autre réservé aux services métiers, aux Cassandra repositories et aux modèles. Cette approche de structuration facilite la compréhension du code, sa maintenance et son évolutivité.

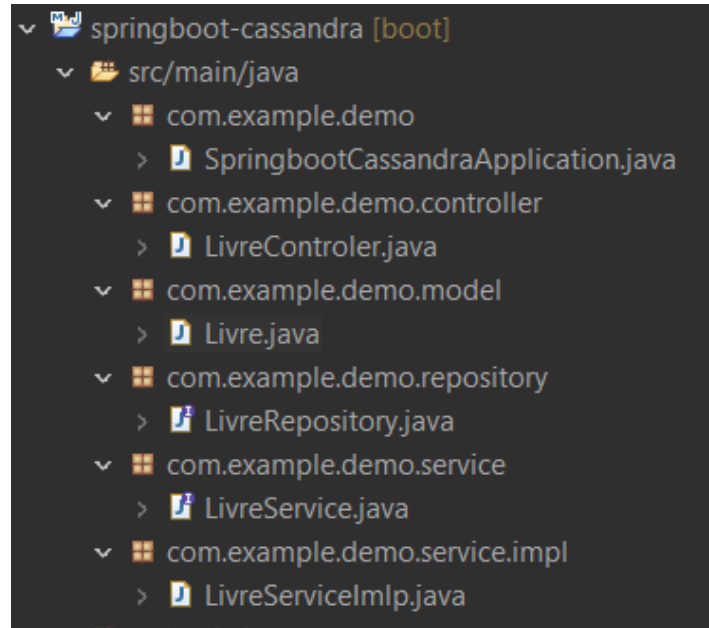


Figure 7: Figure 7 : Arborescence du projet Spring Boot

2. Angular

Angular est un framework JavaScript utilisé pour créer le frontend de l'application. Il permet de créer des interfaces utilisateur dynamiques et réactives qui communiquent avec le backend via des services RESTful.

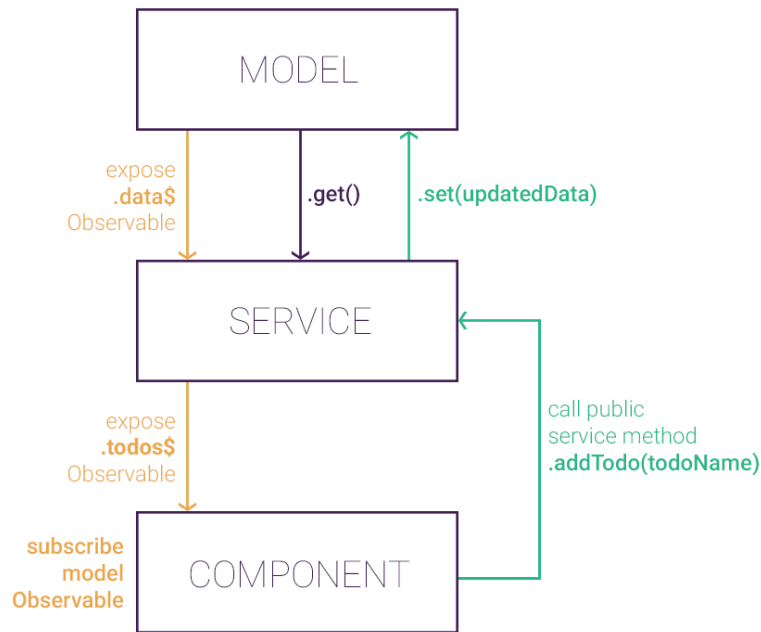


Figure 8: Architecture de l'application Angular

- **Model :**

Les modèles sont généralement définis à l'aide de classes TypeScript et peuvent inclure des propriétés, des méthodes et des fonctions d'assistance pour manipuler les données.

- **Service :**

Les services sont utilisés pour encapsuler la logique métier de l'application et peuvent être utilisés pour effectuer des opérations complexes sur les données, telles que l'appel d'une API, la gestion des données de l'application, etc. Les services peuvent être injectés dans les composants Angular pour fournir des fonctionnalités réutilisables et pour éviter la duplication de code.

- **Component :**

Les composants sont utilisés pour créer l'interface utilisateur de l'application et fournir une vue de l'application pour l'utilisateur. Les composants sont généralement définis à l'aide de classes TypeScript et de fichiers HTML, et peuvent inclure des styles CSS pour personnaliser l'apparence de l'interface utilisateur.

- ✓ *Arborescence du projet*

L'architecture d'un projet Angular suit une organisation en modules, avec des composants pour la partie présentation, des services pour la logique métier de l'application et des modèles pour la représentation des données. Les composants, modèles et services sont regroupés dans des dossiers séparés pour une organisation claire et modulaire du code.

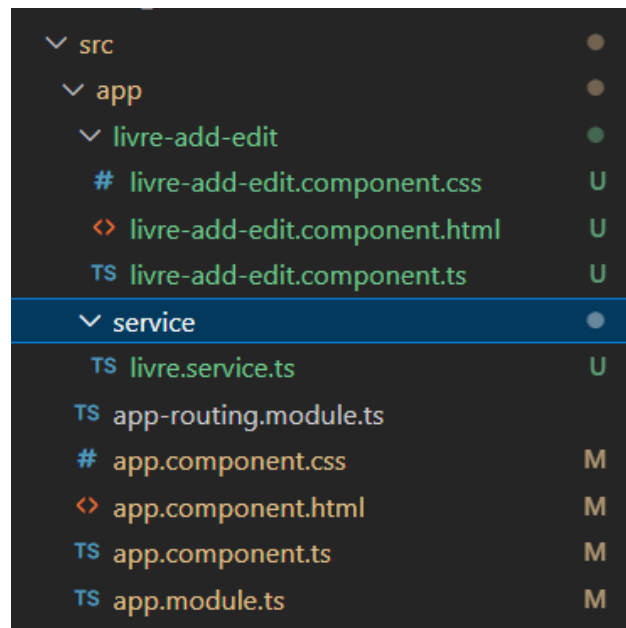


Figure 9: Arborescence du projet Angular

III. Présentation des interfaces

Gestion des livres						Ajouter un nouveau livre
Recherche						
Auteur	Date	Edition	Genre	Titre	Action	
Mohammed Choukri	Mar 14, 1972	Pocket	Autobiographique	Le pain nu	MODIFIER	SUPPRIMER
Victor Hugo	Jan 13, 1890	Albert Lacroix	Tragédie	les misérables	MODIFIER	SUPPRIMER
Yasmina Khadra	Feb 3, 2002	Julliard	Fiction	Les hirondelles de KABOUL	MODIFIER	SUPPRIMER
Guillaume Musso	Apr 1, 2019	Calmann-Lévy	Mystère	La vie secrète des écrivains	MODIFIER	SUPPRIMER
Collen Hover	Feb 25, 2016	Pocket	Fiction	It's ends with us	MODIFIER	SUPPRIMER
Items per page: 5 1 - 5 of 5 < >						

Figure 10: Affichage de la liste des livres

L'interface ci-dessus, représente la liste de tous les livres qui existe dans notre base de données et qui offre à l'utilisateur la possibilité de rechercher un livre avec son titre. L'utilisateur a la possibilité d'effectuer des recherches des livres, d'ajouter de nouveaux livres, de modifier leurs informations existantes et de supprimer des livres si nécessaires.

Les livres existent bien dans notre base de données :

```
cqlsh:projetnosql> select * from livre;
```

id	auteur	date	edition	genre	titre
404a4baa-d11d-402e-ae72-83b44f8b8ac8	Mohammed Choukri	1972-03-14	Pocket	Autobiographique	Le pain nu
ef271b32-194c-43a0-9a41-d0bf23bbe9be	Victor Hugo	1890-01-13	Albert Lacroix	Tragédie	les misérables
e66db173-e281-4f17-b409-a9b8c5d7951b	Yasmina Khadra	2002-02-03	Julliard	Fiction	Les hirondelles de KABOUL
3effa6a5-8819-4687-912a-50f2f891961d	Guillaume Musso	2019-04-01	Calmann-Lévy	Mystère	La vie secrète des écrivains
a7e6f153-b61c-4eae-b8dd-b05f94d2533b	Collen Hover	2016-02-25	Pocket	Fiction	It's ends with us

(5 rows)

Figure 11: Affichage de la liste des livres dans la base de données Cassandra

1. Ajouter un livre

L'interface présentée ci-dessous offre la possibilité d'ajouter les informations d'un livre en affichant un formulaire de saisie.

Figure 12: Formulaire d'ajout d'un livre

Après l'ajout du livre 'Sauve-moi' de l'écrivain 'Guillaume Musso', on le trouve stocker dans la base de données.

```
cqlsh:projetnosql> select * from livre;
```

id	auteur	date	edition	genre	titre
404a4baa-d11d-402e-ae72-83b44f8b8ac8	Mohammed Choukri	1972-03-14	Pocket	Autobiographique	Le pain nu
ef271b32-194c-43a0-9a41-d0bf23bbe9be	Victor Hugo	1890-01-13	Albert Lacroix	Tragédie	les misérables
e690c9c6-392b-4a40-9487-05f4e31c01b3	Guillaume Musso	2005-06-21	Pocket	Fiction	Sauve-moi
e66db173-e281-4f17-b409-a9b8c5d7951b	Yasmina Khadra	2002-02-03	Julliard	Fiction	Les hirondelles de KABOUL
3efffa6a5-8819-4687-912a-50f2f891961d	Guillaume Musso	2019-04-01	Calmann-Lévy	Mystère	La vie secrète des écrivains
a7e6f153-b61c-4eae-b8dd-b05f94d2533b	Collen Hover	2016-02-25	Pocket	Fiction	It's ends with us

(6 rows)

Figure 13: L'ajout du livre (Sauve-moi) dans la base de données Cassandra

2. Modifier un livre

L'interface présentée ci-dessous offre également la possibilité de modifier les informations d'un livre existant en affichant un formulaire pré-rempli avec les champs correspondants aux données du livre sélectionné.

Modifier un livre

Auteur: Guillaume Musso

Titre: Sauve-moi

Genre: Fiction

Date d'apparition: 6/21/2005

Edition: Pocket

Annuler Modifier

Figure 14: Formulaire de modification d'un livre

Par exemple si on modifie le genre du livre 'Sauv-moi', 'Fiction' en 'Policier' : On remarque alors que cette modification a été bien enregistrée dans la base de données.

```
cqlsh:projetnosql> select * from livre;
```

id	auteur	date	edition	genre	titre
404a4baa-d11d-402e-ae72-83b44f8b8ac8	Mohammed Choukri	1972-03-14	Pocket	Autobiographique	Le pain nu
ef271b32-194c-43a0-9a41-d0bf23bbe9be	Victor Hugo	1890-01-13	Albert Lacroix	Tragédie	les misérables
e690c9c6-392b-4a40-9487-05f4e31c01b3	Guillaume Musso	2005-06-21	Pocket	Policier	Sauve-moi
e66db173-e281-4f17-b409-a9b8c5d7951b	Yasmina Khadra	2002-02-03	Julliard	Fiction	Les hirondelles de KABOUL
3effa6a5-8819-4687-912a-50f2f891961d	Guillaume Musso	2019-04-01	Calmann-Lévy	Mystère	La vie secrète des écrivains
a7e6f153-b61c-4eae-b8dd-b05f94d2533b	Collen Hover	2016-02-25	Pocket	Fiction	It's ends with us

Figure 15: La modification du livre (Sauve-moi) dans la base de données Cassandra

3. Supprimer un livre

Lorsqu'on clique sur le bouton 'Supprimer', une pop up qui va apparaître :

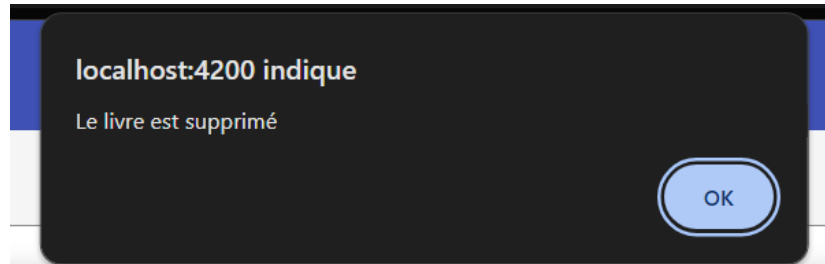


Figure 16: La suppression du livre

On remarque que le livre 'Sauve-moi' est supprimé de la base de données

```
cqlsh:projetnosql> SELECT * FROM livre WHERE titre = 'Sauve-moi' ALLOW FILTERING;  
  
id | auteur | date | edition | genre | titre  
---+-----+-----+-----+-----+-----  
(0 rows)
```

Figure 17: La suppression du livre (Sauve-moi) de la base de données Cassandra

4. Rechercher un livre

On peut effectuer la recherche d'un livre par son titre.

Recherche					
la vie					
Auteur	Date	Edition	Genre	Titre	Action
Guillaume Musso	Apr 1, 2019	Calmann-Lévy	Mystère	La vie secrète des écrivains	<button>MODIFIER</button> <button>SUPPRIMER</button>
					Items per page: 5 1 - 1 of 1

Figure 18: La recherche du livre (la vie secrète des écrivains)