



UNIVERSITE IBN TOFAIL

FACULTE DES SCIENCES

Département Informatique
Filière Génie logiciel pour le cloud

Réalisé Par : Fatima BOUHRIGA

Rapport pour
une application basée sur des
micros services pour gérer
une bibliothèque.

Encadré par : Soufiane HAMIDA

Année universitaire 2022-2024

Table de matières :

| | |
|--|-----------|
| Table de matières..... | 3 |
| Liste des figures:..... | 5 |
| Introduction..... | 6 |
| Chapitre 1 : L'Architecture..... | 7 |
| Technique du projet..... | 7 |
| 1. Introduction..... | 8 |
| 2. Le choix de l'architecture..... | 8 |
| a. L'architecture monolithique..... | 8 |
| ✓ Qu'est-ce qu'une architecture monolithique..... | 8 |
| ✓ Citons quelques inconvénients d'une architecture monolithique..... | 9 |
| b. L'architecture de microservices..... | 10 |
| ✓ Qu'est-ce qu'une architecture microservices..... | 10 |
| ✓ Citons quelques Avantages d'une architecture micro-services..... | 10 |
| 3. Les Outils Utilisés..... | 12 |
| a. Backend..... | 12 |
| b. frontend..... | 13 |
| c. Base de données..... | 13 |
| Chapitre 2: Conception UML..... | 14 |
| 1. Introduction..... | 15 |
| 2. Analyse des existants..... | 15 |
| 3. Le diagramme de classe..... | 16 |
| Chapitre 3..... | 17 |
| Architecture de projet sur Eclipse..... | 17 |
| 1. Introduction..... | 18 |
| 2. L'architecture sur eclipse(backend):..... | 18 |
| ✓ Architecture de service Adherent:..... | 18 |
| ✓ Architecture de service Emprunt:..... | 19 |
| ✓ Architecture de service document..... | 19 |
| 3. Description des packages du projet:..... | 19 |
| ✓ Dossier src/main/java:..... | 20 |
| 4. L'architecture sur visual studio code(frontend):..... | 21 |
| ✓ L'Architecture de notre frontend..... | 21 |
| Chapitre 4 : Réalisation de l'application web..... | 24 |
| 1. Introduction..... | 25 |
| 2. Démonstrations des interfaces..... | 25 |
| a. Les Transactions sur un adherent..... | 25 |
| ✓ Ajouter un adherent..... | 25 |

| | | |
|------------------------|------------------------------------|-----------|
| ✓ | Modifier un adherent..... | 27 |
| ✓ | Supprimer un adherent..... | 28 |
| ✓ | Rechercher adherent (par CIN)..... | 29 |
| ✓ | voire le profil d'un adhérent..... | 30 |
| b. | Les Transactions sur document..... | 30 |
| ✓ | Consulter tous les documents..... | 31 |
| ✓ | Liste des livres..... | 31 |
| ✓ | Liste des dictionnaires..... | 31 |
| ✓ | Liste des revues..... | 31 |
| c. | Les Transactions sur emprunt..... | 30 |
| Conclusion..... | | 32 |

Liste des figures :

| | |
|---|----|
| Chapitre 1 : L'Architecture Technique du projet..... | 7 |
| figure 1: l'architecture monolithique..... | 8 |
| figure 2: L'architecture monolithique vs l'architecture micro-services..... | 9 |
| Chapitre 2: Conception UML..... | 14 |
| Figure 4: Diagramme de classe..... | 16 |
| Chapitre 3: Architecture de projet sur Eclipse..... | 17 |
| figure 5: Présentation de l'architecture avec un schéma..... | 18 |
| figure 6: service adherent..... | 18 |
| figure 7: service emprunt..... | 19 |
| figure 8: service document..... | 19 |
| figure 9: service adherent..... | 20 |
| figure 10: dossier src/main/java..... | 20 |
| figure 11: l'architecture frontend..... | 22 |
| figure 12: composants adherent..... | 22 |
| figure 13: composants dictionnaire..... | 22 |
| figure 14: composants document..... | 23 |
| figure 15: composants emprunt..... | 23 |
| figure 16: composants livre..... | 23 |
| figure 17: composants revue..... | 23 |
| Chapitre 4 : Réalisation de l'application web..... | 24 |
| figure 18: la page d'accueil..... | 25 |
| figure 19: la page d'adherent..... | 26 |
| figure 20: formulaire pour ajouter adherent..... | 26 |
| figure 21: formulaire avec les donnees a ajouter..... | 26 |
| figure 22: adherent ajouter avec succes..... | 27 |
| figure 23: bouton modifier..... | 27 |
| figure 24: les donnees a changer..... | 28 |
| figure 25: les donnees modifier..... | 28 |
| figure 26: les donnees modifier avec succes..... | 28 |
| figure 27: bouton pour la suppression..... | 29 |
| figure 28: alerte de confirmation..... | 29 |
| figure 29: input pour chercher..... | 29 |
| figure 30: l'adherent avec cih 1999..... | 29 |
| figure 31: icon de profil..... | 30 |
| figure 32: le profil d'un adherent..... | 30 |
| figure 33: gerer les documents..... | 30 |
| figure 34: liste des documents..... | 31 |

Introduction

Dans le but de perfectionner les connaissances acquises en Module EJB aux étudiants de la filière Génie logiciel pour le cloud, Notre professeur organise à la fin de semestre un projet à faire.

La portée de notre projet de fin de semestre est la mise en place d'une application pour gérer une bibliothèque programmée en Spring Boot pour le backend et pour le frontend j'ai choisi l'Angular et communiquant avec une base de données MySQL pour aider les bibliothécaires à gérer les Adhérents des usagers de la bibliothèque ainsi que les emprunts des documents. basée sur des microservices qui permet de gérer les adhérents et les documents d'une bibliothèque.

Ce projet java de gestion permet principalement d'avoir une Page d'accueil permettant l'accès aux trois volets de gestion d'adhérent, document, et des emprunts. Pour les adhérents on a Ajouter, modifier, supprimer, rechercher (par CIN). la même chose pour les documents mais les documents sont par Id. Et on a Enregistrer, modifier, supprimer pour les emprunts. Ensuite l'application permet de Consulter le profil d'un adhérent, Consulter la liste des documents existants pour chaque catégorie des documents, Consulter tous les documents, Consulter la liste des emprunts.

Chapitre 1 : L'Architecture Technique du projet

1. Introduction :

Une architecture d'application d'écrit les modèles et les techniques utilisés pour concevoir et créer une application. L'architecture fournit une feuille de route ainsi que les meilleures pratiques à suivre pour créer une application bien structurée donc on est besoin de faire attention au **choix de l'architecture**.

2. Le choix de l'architecture :

a. L'architecture monolithique:

d'habitude on utilise **l'architecture monolithique** c'est pour une application qui est développée en un seul bloc, avec une même technologie et déployée dans un serveur d'application.

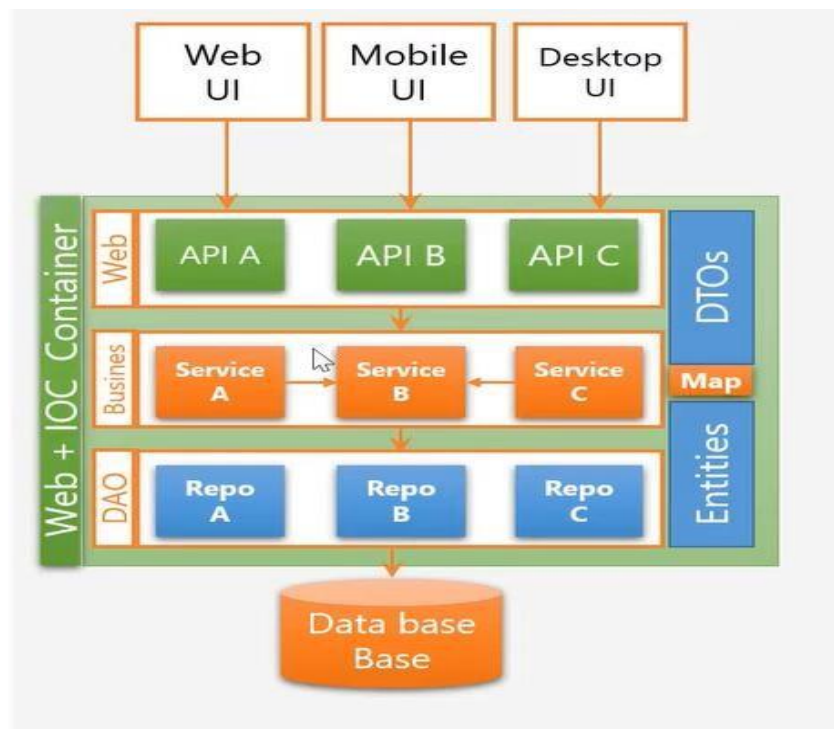


figure 1: l'architecture monolithique

✓ Qu'est-ce qu'une architecture monolithique:

Une architecture monolithique est un modèle traditionnel de programme de développement, conçu comme une unité unifiée autonome et indépendante d'autres apps. Cette architecture est rapide à développer. L'app étant basée sur une seule base de code, elle est simple à prendre en main.

les apps monolithiques peuvent être très efficaces jusqu'à ce qu'elles soient trop volumineuses et que la mise à l'échelle se complique. Apporter un petit changement à une seule fonction nécessite de compiler et de tester l'ensemble de la plateforme, ce qui va à l'encontre de l'approche Agile privilégiée par les développeurs d'aujourd'hui.

✓ Citons quelques inconvénients d'une architecture monolithique :

- **Vitesse de développement plus lente** : une app monolithique volumineuse complexifie et ralentit le développement.
- **Évolutivité** : vous ne pouvez pas mettre à l'échelle des composants individuels.
- **Fiabilité** : si une erreur survient dans un module, elle peut affecter la disponibilité de l'ensemble de l'app.
- **Obstacle à l'adoption de la technologie** : les changements apportés au framework ou au langage affectent l'ensemble de l'app, ce qui les rend souvent coûteux et chronophages.
- **Manque de flexibilité** : un monolithe est limité par les technologies déjà utilisées en son sein.
- **Déploiement** : un changement mineur apporté à une app monolithique nécessite le redéploiement de l'ensemble du monolithe.

➤ C'est pour ça notre professeur nous propose de découvrir l'architecture micro-services

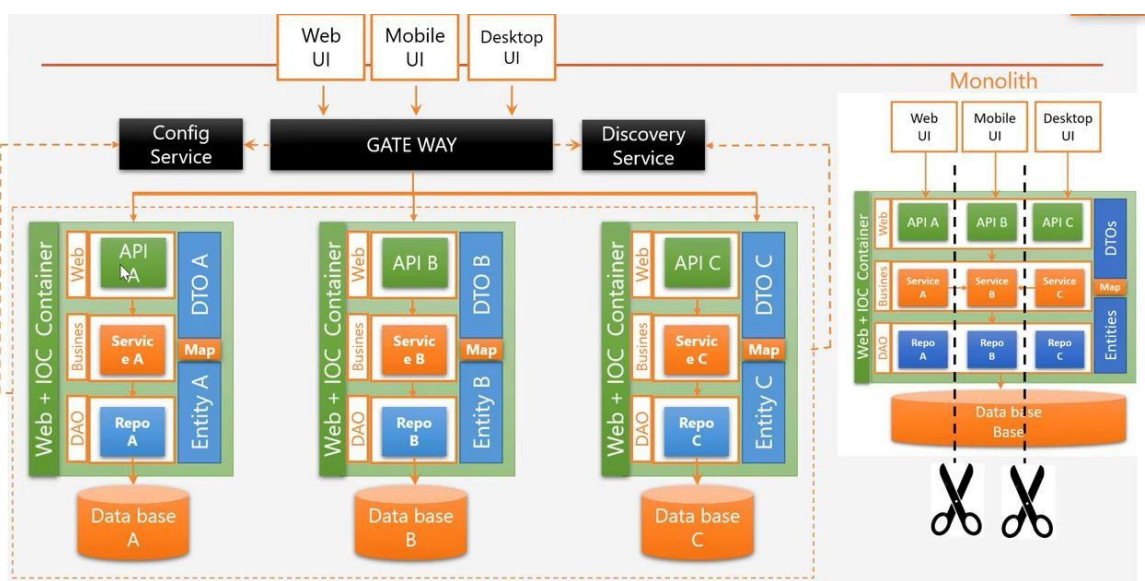


figure 2: L'architecture monolithique vs l'architecture micro-services

b. L'architecture de microservices:

✓ Qu'est-ce qu'une architecture microservices:

Une architecture de microservices, également appelée simplement microservices, est une méthode architecturale qui repose sur une série de services déployables indépendamment. Ces services ont leur propre logique métier et leur propre base de données avec un objectif précis. La mise à jour, les tests, le déploiement et la mise à l'échelle ont lieu dans chaque service. Les microservices dissocient les principales préoccupations spécifiques au domaine d'activité en bases de code indépendantes et distinctes. Ils ne réduisent pas la complexité, mais la rendent visible et plus facile à gérer en séparant les tâches en processus plus petits qui fonctionnent indépendamment les uns des autres et contribuent à l'ensemble.

L'architecture des microservices offre flexibilité, évolutivité et possibilité de modifier, d'ajouter ou de supprimer des composants logiciels sans affecter les autres parties de l'application.

Les microservices ne sont en aucun cas une solution miracle, mais ils résolvent un certain nombre de problèmes pour les logiciels et les entreprises en pleine croissance. Étant donné qu'une architecture de microservices se compose d'unités qui s'exécutent indépendamment, chaque service peut être développé, mis à jour, déployé et mis à l'échelle sans affecter les autres services. Les mises à jour logicielles peuvent être effectuées plus fréquemment, avec une fiabilité, une disponibilité et des performances améliorées. Nous sommes passés de mises à jour une fois par semaine à deux ou trois fois par jour.

✓ Citons quelques Avantages d'une architecture micro-services :

- **Agilité** : promouvoir des méthodes de travail Agile avec de petites équipes qui déploient fréquemment.
- **Évolutivité flexible** : si un microservice atteint sa capacité de charge, de nouvelles instances de ce service peuvent être rapidement déployées sur le cluster attendant afin de réduire la pression. Nous adoptons désormais une solution multilocataire et sans état avec des clients répartis sur plusieurs instances. Nous pouvons prendre en charge des instances bien plus vastes.
- **Déploiement continu** : nos cycles de livraison sont désormais fréquents et plus rapides. Auparavant, nous réalisions des mises à jour une fois par semaine.

Maintenant, nous sommes passés à un rythme de deux ou trois fois par jour.

- **Facilité d'administration et de test** : les équipes peuvent tester de nouvelles fonctionnalités et revenir en arrière en cas de problème. Cela facilite la mise à jour du code et accélère la commercialisation des nouvelles fonctionnalités. De plus, il est facile d'isoler et de corriger les pannes ainsi que les bugs des services individuels.
- **Déploiement indépendant** : les microservices étant des unités individuelles, ils permettent de déployer indépendamment des fonctionnalités individuelles de manière rapide et facile.
- **Flexibilité technologique** : les architectures de microservices permettent aux équipes de sélectionner les outils souhaités.
- **Fiabilité élevée** : vous pouvez déployer des changements pour un service spécifique, sans risquer de paralyser l'ensemble de l'app.
- **Équipes satisfaites** : les équipes Atlassian qui travaillent avec des microservices sont beaucoup plus satisfaites, car elles sont plus autonomes et peuvent créer et déployer elles-mêmes sans attendre l'approbation d'une pull request pendant des semaines.

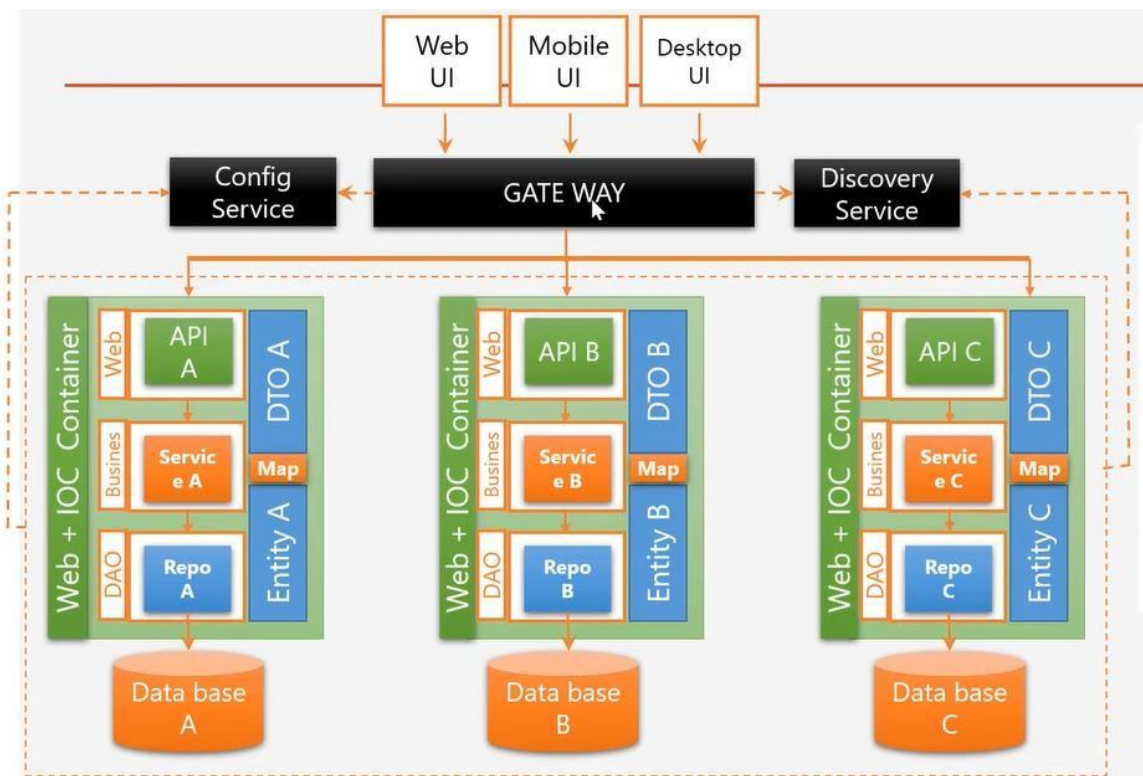


figure 3: L'architecture monolithique vs l'architecture micro-service

3. Les Outils Utilisés:

a. Backend:

Spring Boot: est une extension du Spring Framework pour mettre en place rapidement des applications Java. Grâce à un système modulaire de dépendances et un principe de configuration automatique, il permet de disposer d'une structure de projet complète et immédiatement opérationnelle. La perte de souplesse dans les choix d'architecture est contre-balançée par la rapidité de mise en place et un large choix de paramètres de configuration. Spring Boot n'est pas une nouvelle version du Spring Framework. Il s'agit d'une utilisation particulière du framework, évitant aux développeurs de gérer une complexité technique souvent inutile au moment de l'initialisation d'un projet.

Grâce à une configuration par défaut, une application Spring Boot permet de développer plus rapidement en se focalisant essentiellement sur le code métier. On dira que Spring Boot a une « opinion ». Bien entendu, ces fonctionnalités par défaut peuvent être modifiées pour correspondre aux besoins. Cela implique de respecter les conventions et règles imposées par Spring afin d'éviter de perdre trop de temps sur de la configuration applicative. Cela nous permet notamment d'éviter d'écrire de multiples fichiers de configuration en XML.

Spring Boot propose différentes annotations afin d'appliquer rapidement des rôles et des comportements à des classes de votre application. par exemple `@Autowired`, `@SpringBootApplication`, `@Component`..etc.

Toujours dans un objectif de simplicité, un large choix de dépendances Maven préconçus (starters) sont proposées par Spring et permettent d'ajouter des fonctionnalités à votre application. Un starter est en réalité un pom.xml ou un gradle build file qui définit et gère un ensemble de dépendances avec leurs versions compatibles. j'ai utilisé dans mon projet:

- ✓ **Starter Web:** Pour faire une application Spring MVC Web
- ✓ **Starter Data JPA:** Pour faire du Spring Data avec JPA
- ✓ **Starter Data REST:** Pour exposer des services REST
- ✓ **Spring-boot-devtools:** pour déclencher le rafraîchissement du navigateur lorsqu'une ressource est modifiée.

b. frontend:

Angular: est une plateforme de développement, construite sur TypeScript. En tant que plateforme, Angular comprend :

- ✧ Un cadre basé sur des composants pour la création d'applications web évolutives.
- ✧ Une collection de bibliothèques bien intégrées qui couvrent une grande variété de fonctionnalités, notamment le routage, la gestion des formulaires, la communication client-serveur, etc.
- ✧ Une suite d'outils de développement pour vous aider à développer, construire, tester et mettre à jour votre code.

c. Base de données:

MySQL: est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde³, autant par le grand public (applications web principalement) que par des professionnels.

Chapitre 2:

Conception UML

1. Introduction:

Au stade de la conception détaillée, nous disposons d'un ensemble de graphiques UML. Ces schémas sont utilisés pour structurer la couche métier de l'application, en réunissant les classes d'analyse par fonctionnalité.

L'UML (Unified Modeling Language ou Langage de modélisation unifiée en français) est un langage graphique de modélisation informatique. Ce langage est désormais la référence en modélisation objet, ou programmation orientée objet. Cette dernière consiste à modéliser des éléments du monde réel (immeuble, ingrédients, personne, logos, organes du corps...) ou virtuel (temps, prix, compétence...) en un ensemble d'entités informatiques appelées « objet ».

2. Analyse des existants:

On souhaite créer une application qui permet de gérer les Adhérents, les emprunts et les documents d'une bibliothèque.

- ✓ Adherents(id,nom,prenom,adresse,cin,dateNaissances)
- ✓ Document: Il existe trois types de documents, livre, revue, et dictionnaire.
 - Livre(id,auteur,nombredepages,prixdevente,nom)
 - Revue(reference,anneedepublication,prixvente,nom)
 - Dictionnaire(id,langue,prixvente,nom)
- ✓ Emprunt(id,datedemprunt,datederetoure)

Notre application basée sur plusieurs tâches qui sont :

- Page d'accueil permettre l'accès au trois volets de gestion d'adhérent, document, et des emprunts,
- Ajouter, modifier, supprimer, rechercher (par CIN) un adhérent,
- Ajouter, modifier, supprimer, rechercher (par Id) un document,
- Enregistrer, modifier, supprimer un emprunt,
- Consulter le profil d'un adhérent,
- Consulter la liste des documents existants pour chaque catégorie des documents,
- Consulter tous les documents,
- Consulter la liste des emprunts.

3. Le diagramme de classe:

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

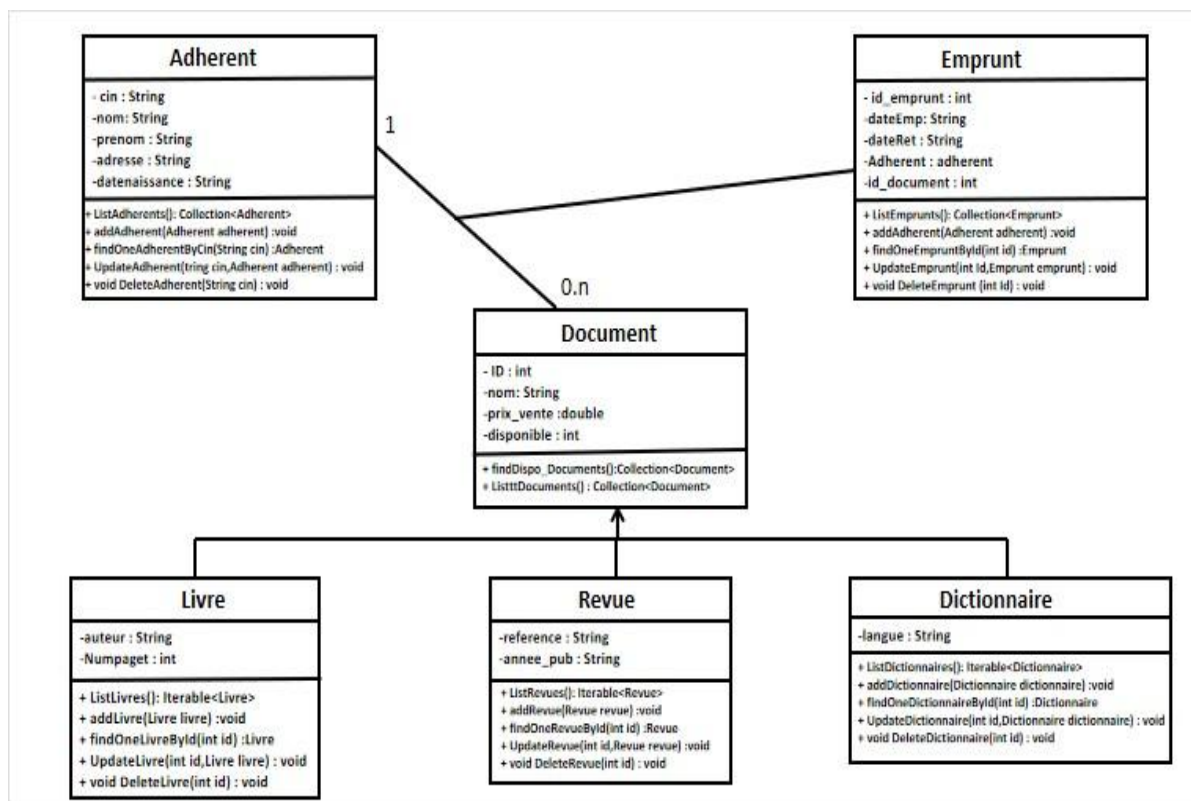


Figure 4: Diagramme de classe

Chapitre 3:

Architecture de projet sur Eclipse

1. Introduction :

Le style architectural des microservices est une approche permettant de développer une application unique sous la forme d'une suite logicielle intégrant plusieurs services. Ces services sont construits autour des capacités de l'application et peuvent être déployés de façon indépendante.

2. L'architecture sur eclipse(backend):

Concrètement, les microservices sont une méthode développement logiciel utilisée pour concevoir une application comme un ensemble de services modulaires. Chaque module répond à un objectif métier spécifique et communique avec les autres modules.

Pour notre application on a trois services principale.service d'emprunt ,document et adherent. c'est pour ça j'ai crée trois microservices différents comme suite.

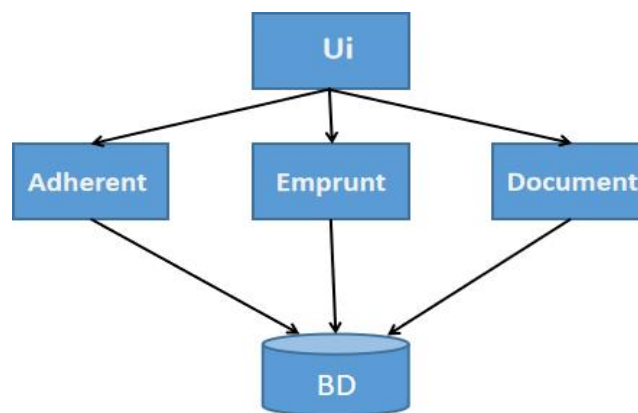


figure 5: Présentation de l'architecture avec un schéma

✓ Architecture de service Adherent:

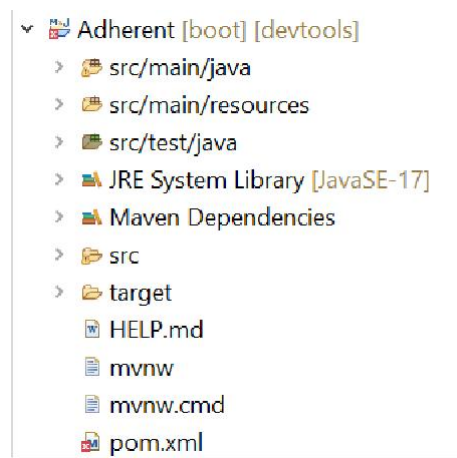


figure 6: service adherent

✓ Architecture de service Emprunt:



figure 7: service emprunt

✓ Architecture de service document :

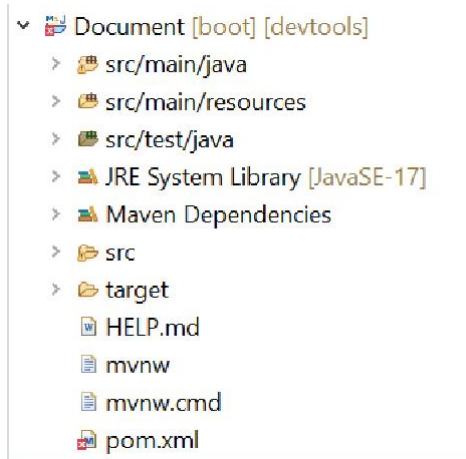


figure 8: service document

3. Description des packages du projet:

Pour décrire les packages je ne présente qu'un seul service (service adherent) sinon l'autre c'est la même chose.

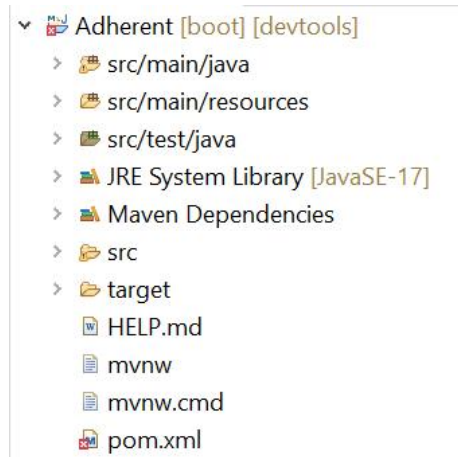


figure 9: service adherent

✓ Dossier src/main/java:

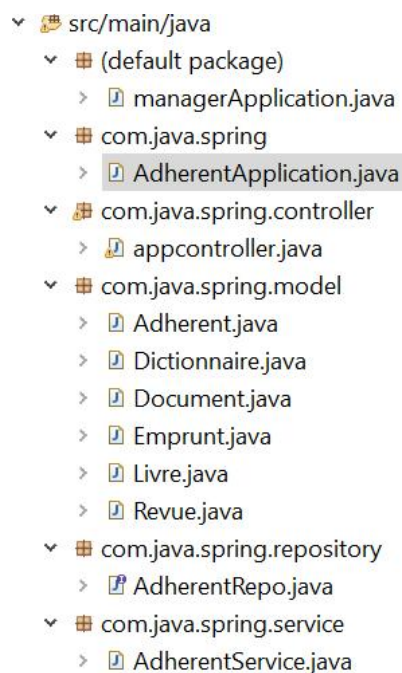


figure 10: dossier src/main/java

- ✧ Le premier package contient une classe de configuration du frontend
- ✧ Le deuxième package contient une classe automatiquement créée au moment de création du projet spring pour lancer tout le framework Spring
- ✧ Le troisième package c'est la classe controller portant l'annotation `@Controller`. Pour que il soit appelé lors du traitement d'une requête, il suffit d'ajouter l'annotation `@RequestMapping` sur une méthode publique de la classe en précisant la méthode HTTP concernée (par défaut GET) et le chemin d'URI (à partir du contexte de déploiement de l'application) pris en charge par la méthode.

- ✧ Le quatrième package contient les modèles de notre projet le modèle correspond à l'ensemble des données qui sont nécessaires à la construction de la vue.
- ✧ le cinquième package contient une classe repository fournit une interface marqueur générique Repository<T, ID>. pour générer automatiquement le code sous-jacent et exécuter la requête. La requête est déduite de la signature de la méthode (on parle de query methods).
- ✧ La sixième package contient la classe adherentservice pour les services disponibles par l'adhérent

4. L'architecture sur visual studio code(frontend):

Angular est écrit en TypeScript. Il implémente des fonctionnalités de base et optionnelles sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans vos applications.

Chaque application Angular fonctionne grâce à au moins un component. C'est le component racine qui lie la hiérarchie des components au DOM. Chaque component définit une classe qui contient les données et la logique de l'application, et est associé à un modèle HTML qui définit une vue à afficher dans un environnement cible.

Les autres éléments de l'architecture sont les modules et les services. Les modules, les services et les components sont des classes qui utilisent des decorators. Ces decorators fournissent des métadonnées qui indiquent à Angular comment les utiliser.

✓ L'Architecture de notre frontend :

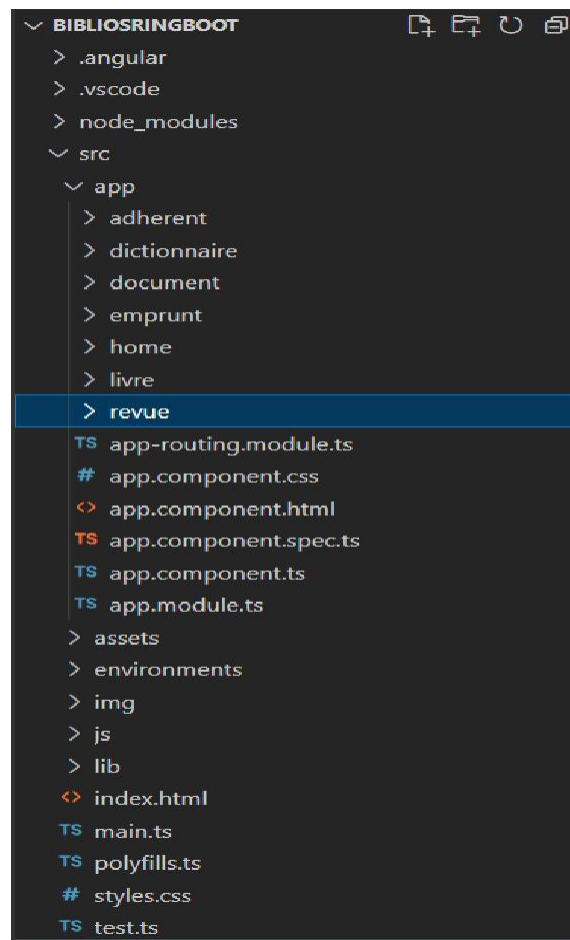


figure 11: l'architecture frontend

Pour chaque service a un composants

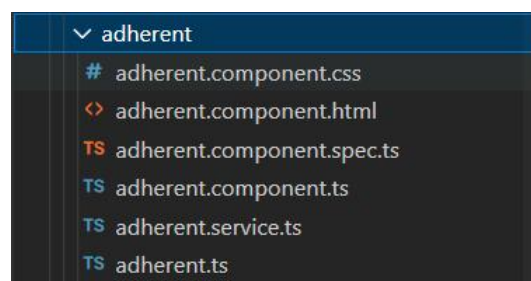


figure 12: composants adherent

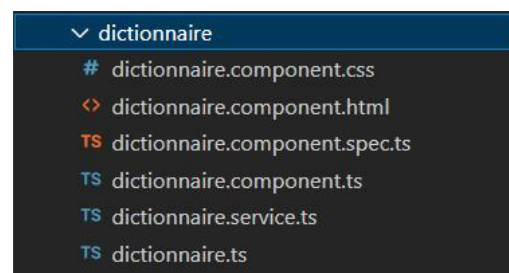


figure 13: composants dictionnaire

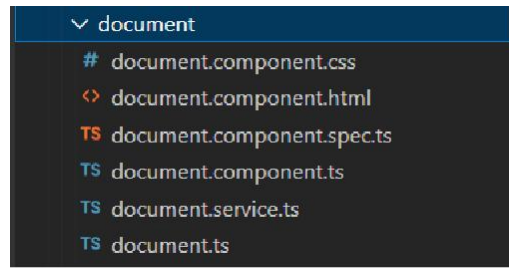


figure 14: components document

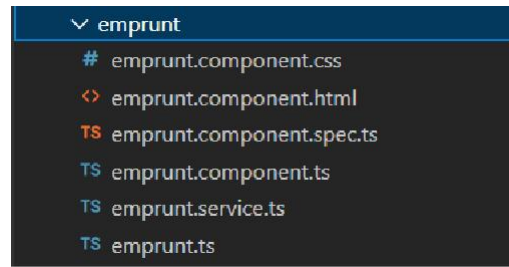


figure 15: components emprunt

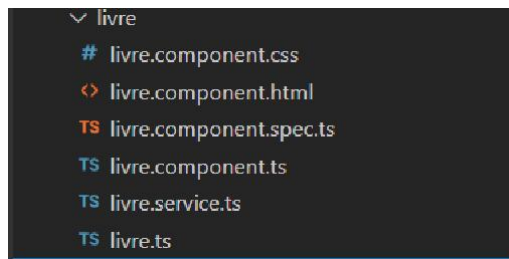


figure 16: components livre

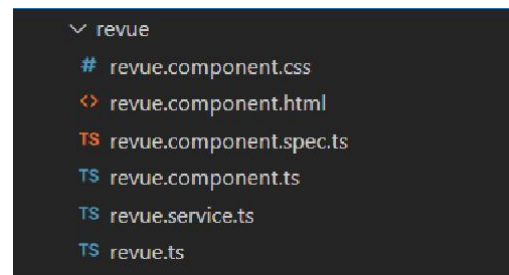


figure 17: components revue

Chapitre 4 : Réalisation de l'application web

1. Introduction :

Cette partie nous permet de montrer les résultats de notre application dans sa phase de test et d'essai, ce qui nous permettra d'envisager les améliorations possibles. Les difficultés rencontrées nous ont permis de dégager un ensemble de pistes susceptibles de nous aider.

2. Démonstrations des interfaces

Cette partie dénombre la présentation des Scénarios applicatifs de l'application. Je vais présenter dans ce qui suit, les imprimés-écran des principales interfaces réalisées dans le site web.

C'est la page d'accueil qui s'affiche dès l'accès au site web, elle permet l'accès aux trois volets de gestion d'adhérent, document, et des emprunts



figure 18: la page d'accueil

a. Les Transactions sur un adherent

Ajouter, modifier, supprimer, rechercher (par CIN) et voir le profil d'un adherent.

✓ Ajouter un adherent

Pour ajouter un nouveau adherent .il faut d'abord remplir le formulaire puis en cliquer sur le Button ajouter donc l'adherent s'enregistre sur la base de données et il s'affiche sur le tableau.

Bibliothèque bouhriga

Ajouter adherent

Search adherent avec CIN...

Liste des adherents
















| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|---------|--------|---------|-------------------|---|
| Bouhriga | ilyas | sh1111 | kenitra | 2000-03-10 |    |
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |
| idichou | anas | sh2901 | agadir | 2001-02-10 |    |
| fati | bouh | sh6789 | rabat | 2003-02-12 |    |
| bouhriga | wijdane | sh7845 | agadir | 1999-02-01 |    |

figure 19: la page d'adherent

Bibliothèque bouhriga

Ajouter adherent

Search adherent avec CIN...

| Nom | Prenom |
|----------|---------|
| Bouhriga | ilyas |
| Bouhriga | fatima |
| Idichou | anas |
| fati | bouh |
| bouhriga | wijdane |

Ajouter adherent

Nom

Entre le nom

Prenom

Entre le prenom

CIN

Entre le CIN

Address

Entre l'address

Date de Naissance

jj / mm / aaaa

Close

Ajouter

figure 20: formulaire pour ajouter adherent

Bibliothèque bouhriga

Ajouter adherent

Search adherent avec CIN...

| Nom | Prenom |
|----------|---------|
| Bouhriga | ilyas |
| bouhriga | fatima |
| Idichou | anas |
| fati | bouh |
| bouhriga | wijdane |

Ajouter adherent

Nom

Bouhriga

Prenom

mouhamed

CIN

sh7923

Address

agadir hay salame

Date de Naissance

01 / 01 / 1996

Close

Ajouter

figure 21: formulaire avec les donnees a ajouter

Bibliothèque bouhriga Ajouter adherent Search adherent avec CIN...

Liste des adherents


| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|----------|--------|-------------------|-------------------|---|
| Bouhriga | ilyas | sh1111 | kenitra | 2000-03-10 |    |
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |
| idichou | anas | sh2901 | agadir | 2001-02-10 |    |
| fati | bouh | sh6789 | rabat | 2003-02-12 |    |
| bouhriga | wijdane | sh7845 | agadir | 1999-02-01 |    |
| Bouhriga | mouhamed | sh7928 | agadir hay salame | 1996-01-01 |    |

figure 22: adherent ajouter avec succes

✓ Modifier un adherent

Pour modifier les données d'un adherent il faut d'abord cliquer sur le bouton Modifier il s'affiche un formulaire qui contient les données d'adherent de cette ligne. Après la modification des données on prend l'exemple de nom et la date en le changeant bouhriga avec ben youssef et 2000 avec 1999 puis on clique sur le bouton modifier.

Bibliothèque bouhriga Ajouter adherent Search adherent avec CIN...

Liste des adherents



















| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|----------|--------|-------------------|-------------------|---|
| Bouhriga | ilyas | sh1111 | kenitra | 2000-03-10 |    |
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |
| idichou | anas | sh2901 | agadir | 2001-02-10 |    |
| fati | bouh | sh6789 | rabat | 2003-02-12 |    |
| bouhriga | wijdane | sh7845 | agadir | 1999-02-01 |    |
| Bouhriga | mouhamed | sh7928 | agadir hay salame | 1996-01-01 |    |

figure 23: bouton modifier

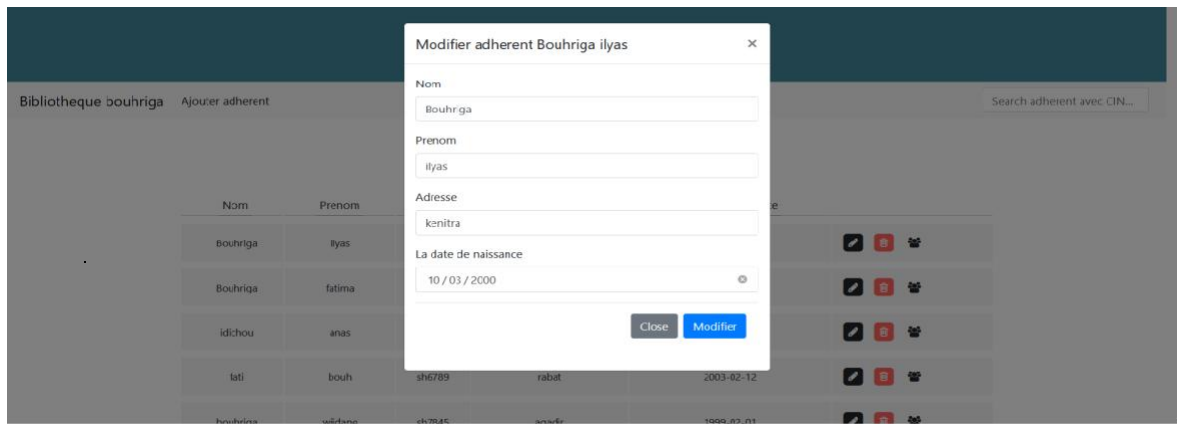


figure 24:les donnees a changer

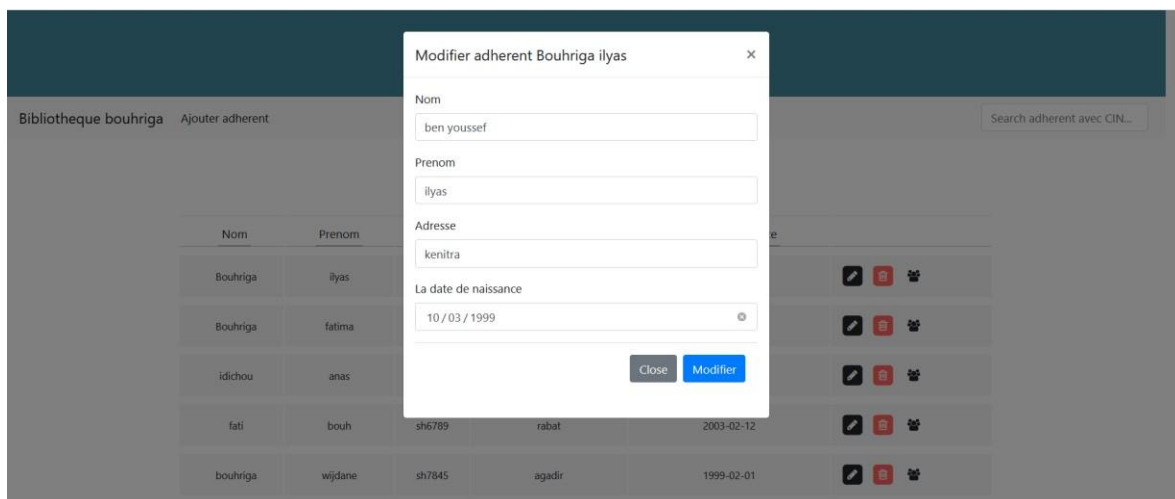


figure 25:les donnees modifier

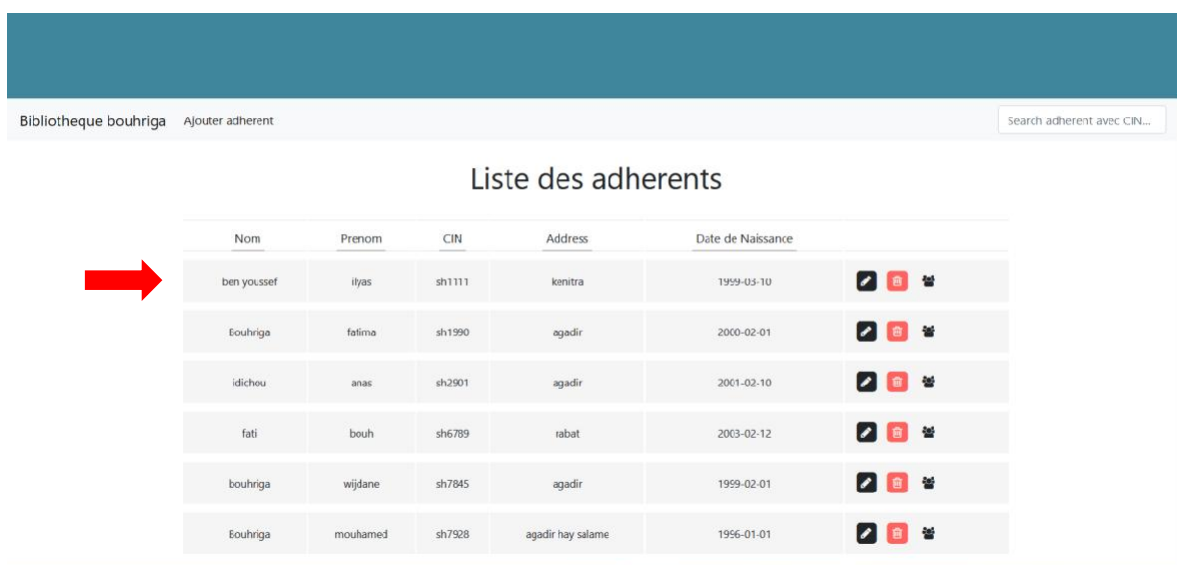


figure 26:les donnees modifier avec succer

✓ Supprimer un adherent

Le clique sur ce bouton “Supprimer”.Il s’affiche une alertde confirmation en suite

l'adherent supprimer sur la base de données et aussi sur le tableau.

Bibliothèque bouhriga Ajouter adherent Search adherent avec CIN...

Liste des adherents

| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|----------|--------|-------------------|-------------------|---|
| Bouhriga | ilyas | sh1111 | kenitra | 2000-03-10 |    |
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |
| Idichou | anas | sh2901 | agadir | 2001-02-10 |    |
| fati | bouh | sh6789 | rabat | 2003-02-12 |    |
| bouhriga | wijdane | sh7845 | agadir | 1999-02-01 |    |
| Bouhriga | mouhamed | sh7928 | agadir hay salame | 1996-01-01 |    |

figure 27:button pour la suppression

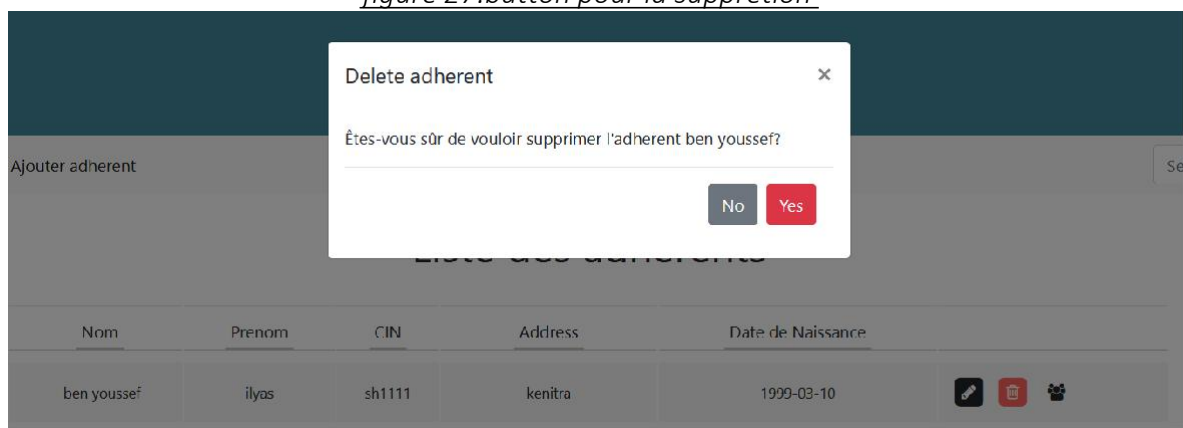


figure 28:alerte de confirmation

✓ Rechercher adherent (par CIN)

Pour chercher un adherent en ecrit dans ce input le cin ensuit l'adherent s'affiche

Bibliothèque bouhriga Ajouter adherent Search adherent avec CIN...

Liste des adherents




| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|--------|--------|---------|-------------------|---|
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |
| Idichou | anas | sh2901 | agadir | 2001-02-10 |    |

figure 29:input pour chercher

Bibliothèque bouhriga Ajouter adherent sh1990

Liste des adherents

| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|--------|--------|---------|-------------------|---|
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |

figure 30:l'adherent avec cin 1999

✓ voir le profil d'un adhérent.

Pour voir le profil d'un adhérent en click sur l'icône de profil

Bibliothèque bouhriga Ajouter adhérent Search adhérent avec CIN...


Liste des adhérents

| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|--------|--------|---------|-------------------|---|
| Bouhriga | ilyas | sh1111 | kenitra | 2000-03-10 |    |
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |
| idichou | anas | sh2901 | agadir | 2001-02-10 |    |

figure 31: icône de profil

Bibliothèque bouhriga Ajouter adhérent Search adhérent avec CIN...

Liste des adhérents

**Information**
Bouhriga fatima

Nom : Bouhriga
Prenom : fatima
CIN : sh1990
Adresse : agadir
Date de naissance : 2000-02-01







| Nom | Prenom | CIN | Address | Date de Naissance | |
|----------|--------|--------|---------|-------------------|---|
| Bouhriga | fatima | sh1990 | agadir | 2000-02-01 |    |
| idichou | anas | sh2901 | agadir | 2001-02-10 |    |

figure 32: le profil d'un adhérent

b. Les Transactions sur document

A propos du document on a quatre voles voire tous les documents ou voire chaque un indépendamment

bouhriga.biblio Gerer les adhérents Gerer les documents Gerer les emprunts

bouhriga.biblio

Les meilleures bibliothèques qui Chaque amateur de livres doit Visite!

les adhérents **les documents** **les emprunts**

Tous les documents
Revue
Dictionnaire
Livre

figure 33: gerer les documents

✓ Consulter tous les documents

| | | | |
|-----------------------|-------|--------------|-------|
| Bibliothèque bouhriga | Revue | Dictionnaire | Livre |
|-----------------------|-------|--------------|-------|

| Liste des dictionnaires | | | |
|-------------------------|-----------------------|---------------|----------------|
| id | nom | prix de vente | disponibiliter |
| 15 | lexicographe | 30 | 1 |
| 16 | rosace | 100 | 0 |
| 8 | harry potter | 0 | 1 |
| 9 | voyage de gulliver | 190 | 1 |
| 10 | le journal La Presse | 200 | 1 |
| 11 | The agony | 99 | 0 |
| 12 | scientific publishing | 70 | 0 |
| 13 | the ecstasy | 80 | 0 |
| 14 | CiteSeer | 81 | 0 |

figure 34: liste des documents

✓ Liste des livres

On peut aussi ajouter, modifier, supprimer, rechercher (par Id) un livre

| | | |
|-----------------------|---------------|----------------------------|
| Bibliothèque bouhriga | Ajouter livre | Chercher avec nom livre... |
|-----------------------|---------------|----------------------------|

| Liste des livres | | | | | |
|--------------------|---------------|----------------|----------------|----------------|---|
| nom | prix de vente | disponibiliter | l'auteur | numero de page | |
| harry potter | 0 | 1 | J. K. Rowling | 309 |   |
| voyage de gulliver | 190 | 1 | jonathan swift | 110 |   |

figure 35: liste des livres

✓ Liste des dictionnaires

On peut aussi ajouter, modifier, supprimer, rechercher (par Id) un dictionnaire

| | | |
|-----------------------|----------------------|-------------------------------|
| Bibliothèque bouhriga | Ajouter dictionnaire | Chercher avec nom dictionn... |
|-----------------------|----------------------|-------------------------------|

| Liste des dictionnaires | | | | |
|-------------------------|---------------|----------------|----------|---|
| nom | prix de vente | disponibiliter | langue | |
| lexicographe | 30 | 1 | français |   |
| rosace | 100 | 0 | français |   |

figure 36: liste des dictionnaires

✓ Liste des revues

On peut aussi ajouter, modifier, supprimer, rechercher (par Id) un revue

Bibliothèque bouhriga

Ajouter revue

Chercher avec nom revue...

Liste des revues

| nom | prix de vente | disponibiliter | la reference | année de publiciter | |
|-----------------------|---------------|----------------|---------------|---------------------|---|
| le journal La Presse | 200 | 1 | Falagasa | 1978 |   |
| The agony | 99 | 0 | Communication | 2021 |   |
| scientific publishing | 70 | 0 | cite | 2000 |   |
| the ecstasy | 80 | 0 | The role | 2007 |   |
| CiteSeer | 81 | 0 | biomedical | 2022 |   |

figure 37: liste des revues

c. Les Transactions sur emprunt

A propos d'emprunt on a quatre voles voire tous les emprunts ou supprimer,ajouter,modifier une emprunt

Bibliothèque bouhriga

Ajouter emprunt

Liste des emprunts







| Nom adherent | Prenom adherent | CIN | Id document | date d'emprunt | date de retour | |
|--------------|-----------------|--------|-------------|-------------------------------|-------------------------------|---|
| Bouhriga | fatima | sh1990 | 15 | Fri Sep 03 13:23:24 WEST 2022 | Wed Sep 08 13:23:24 WEST 2022 |   |
| Bouhriga | fatima | sh1990 | 10 | Fri Sep 02 19:09:23 WEST 2022 | Wed Sep 07 19:09:23 WEST 2022 |   |
| idichou | anas | sh2901 | 8 | Fri Sep 02 19:10:15 WEST 2022 | Wed Sep 07 19:10:15 WEST 2022 |   |

figure 38: liste des emprunts

Conclusion

L'objectif visé à travers ce rapport est de présenter l'application réalisée pour l'examen de module EJB. Dans un premier temps, j'ai consacré mes réflexions à l'étude de l'existant pour améliorer mon application.

Cette étude m'a permis de passer à une phase plus importante qui est : citer les besoins et les répartir en besoins fonctionnels. Une phase de conception détaillée basée sur le Processus Unifié, UML, a précédé l'implémentation de ce projet. En effet, j'ai pu, dans ce qui a resté du temps, développer une interface qui répond éventuellement aux exigences soulignées pendant l'analyse et la conception.

Au titre de notre projet, il fallait apprendre comment fonctionne l'application .et créer, supprimer, modifier, rechercher les modèles etc.. pour les manipuler et vérifier les emprunts dans l'application Web. Pour procéder à cette application, j'ai utilisé mysql, spring boot, rest et angular pour développer cette application.

Ce projet a été très enrichissant pour moi, j'ai eu l'opportunité d'enrichir mes compétences en technologies web par l'apprentissage d'une Framework. Par ailleurs, durant ce projet, j'ai rencontré plusieurs difficultés au cours de la réalisation mais ça me n'a pas empêché d'avancer et d'améliorer mes techniques de travail.

Finalement, j'aimerais remercier M.Soufiane HAMIDA de cette occasion.

