



UNIVERSITE IBN TOFAIL

FACULTE DES SCIENCES

Département Informatique
Filière Génie logiciel pour le cloud

Réalisé Par : Fatima BOUHRIGA

Fatiha AIT AADI

Rapport pour

Mini Project Web Service pour

gérer les candidatures d'un

concours.

Encadré par : Ismail Nadim

Année universitaire 2021-2023

Table de matières

Table de matières	3
Liste des figures	4
Introduction.....	5
Chapitre 1 :Web Services	6
1. Introduction	6
2. Architecture d'un service Web	6
3. Le fonctionnement d'un Web Service	7
4. Web Services SOAP WSDL UDDI Mise en œuvre avec JAXWS	8
A: Développer le web service	8
(1) Crée un projet java simple.....	8
(2) Crée l'objet metier bachelier.	9
(3) Crée le web service bachelierservice.	9
(4) Déployer le web service.....	10
(5) Vérifier si le web service est ça marche.	11
B: Test avec outil SoapUI.....	12
Chapitre 2 : Client SOAP	14
1. Introduction.....	14
2. Créer le web service client.....	14
Chapitre 3 : Réalisation de l'application web	17
1. Introduction.....	17
2. Diagramme de séquence	17
3. Démonstrations des interfaces	18
A: Page d'accueil.....	18
B: Page de resultats	20
Conclusion	22

Liste des figures

Figure 1:l'architecture de l'application	8
Figure 2:projet web service	8
Figure 3:Objet metier Bachelier	9
Figure 4:le web service bachelierservice	10
Figure 5:déploiement de web service	11
Figure 6:WSDL de web service.....	11
Figure 7:opérations de web service	11
Figure 8:creer le projet soap	12
Figure 9:structure de projet soap	12
Figure 10:test de methode findOne.....	13
Figure 11:BachelierClientt	14
Figure 12:select a wizard pour service client	15
Figure 13:select a service definition	15
Figure 14:src/main/java et les jars ajouter	16
Figure 15:Diagramme de séquence	17
Figure 16:la page d'accueil	18
Figure 17:formulaire pour candidater	19
Figure 18:alert d'admission	19
Figure 19:alert de refus	20
Figure 20:page de resultats	21

Introduction

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

La portée de notre projet de fin de semestre est la mise en place d'une application pour gérer les candidatures d'un concours. Programmé en java/jee pour le backend et pour le frontend j'ai choisi jsp/css/html/bootstrap/javascript et communiquant avec une base de données MySQL pour aider les candidats qui veut candidater à un concours réaliser par l'institut basée sur un micro service qui permet de gérer les candidats et les notes d'un 'institut.

Ce projet java de gestion permet principalement d'avoir une Page d'accueil et une page de résultats. En suite l'application permet Un bachelier qui veut candidater à un concours réaliser par un institut devrait saisir son numéro d'étudiant et son CIN. L'application fait appel au web service pour vérifier si l'étudiant concerné dispose d'une note supérieure à 12 dans toutes les matières précitées. Si oui, l'étudiant sera accepté pour passer le concours, sinon sa candidature sera rejetée. Consulter la liste des résultats existants pour les bacheliers acceptés pour passer le concours.

Chapitre 1 : Web Services

1. Introduction

Un Web Service est une application qui permet d'échanger des données avec d'autres applications web. Même si ces dernières sont construites dans des langages de programmation différents. Parmi les Web Services les plus connus on peut citer SOAP, REST ou HTTP.

2. Architecture d'un service Web

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML), et les appliquent à des interactions entre machines. Comme pour le World Wide Web, les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Les technologies utilisées par les services Web sont HTTP, WSDL, REST, XML-RPC, SOAP et UDDI.

Rest:

REST (Representational State Transfer) est une architecture de services Web. Élaborée en l'an 2000 par Roy Fielding, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd et d'autres travaux fondamentaux, REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

XML-RPC :

XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC. Les requêtes sont écrites en XML et envoyées via HTTP POST. Les requêtes sont intégrées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plate-forme, ce qui lui permet de communiquer avec diverses applications. Par exemple, un client Java peut parler de XMLRPC à un PerlServer.

SOAP :

SOAP (Simple object Access Protocol) est un protocole standard de communication. C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes. Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

WSDL :

WSDL (Web Services Description Language) est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

UDDI :

UDDI (Universal Description, Discovery and Integration) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients

3. Le fonctionnement d'un Web Service

Un Web Service fonctionne de la manière suivante :

- ✓ Le client (en général un utilisateur sur un ordinateur ou un support équivalent) effectue une requête dans un des langages suivants : XML, JSON ou HTTP.
- ✓ Cette requête est transmise à un serveur distant via les protocoles SOAP, REST ou HTTP.
- ✓ La réponse est ensuite délivrée sous le même format que sa demande : XML, JSON ou HTTP.

Et ce grâce à un modèle en couche. La 1re couche est l'invocation : elle décrit la structure des messages échangés par le client et le serveur. Elle s'effectue à l'aide des standards XML-RPC ou SOAP. La 2e est la découverte. C'est la phase de recherche et de localisation des données demandées par le client, le plus souvent via le protocole UDDI. La 3e est la description qui stipule les paramètres des fonctions et les types de données des services web utilisés. Son protocole standard est le WSDL qui repose sur la notation XML.

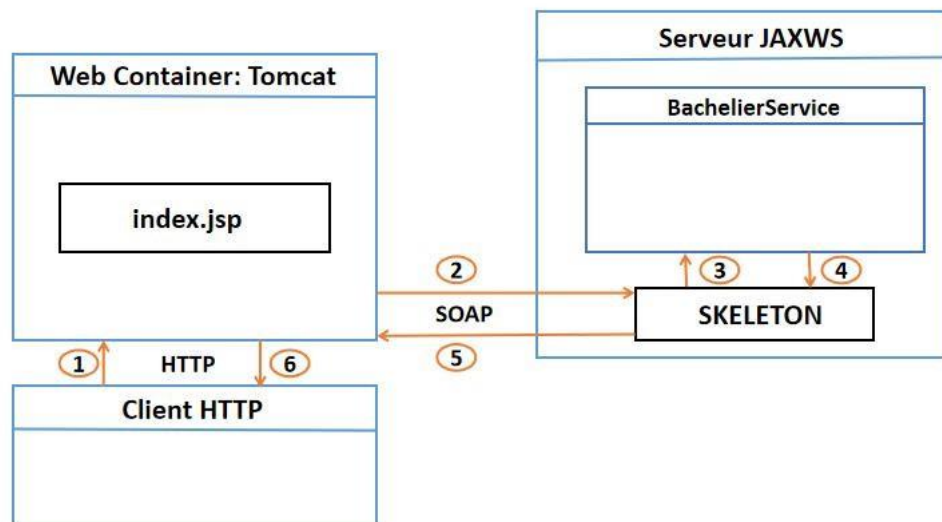


Figure 1: l'architecture de l'application

4. Web Services SOAP WSDL UDDI Mise en œuvre avec JAXWS

A: Développer le web service

(1) Crée un projet java simple.

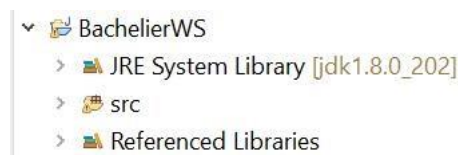


Figure 2: projet web service

(2) Crée l'objet métier bachelier.

```
package metier;

import java.io.Serializable;
@XmlRootElement(name="Bachelier")
public class Bachelier implements Serializable {

    private static final long serialVersionUID = 1L;
    private int id;
    private String cin;
    private String numetud;
    private double notemath;
    private double notephys;
    private double notearabe;
    private double noteanglais;
    private String nomprenom ;
    public Bachelier() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Bachelier(int id, String cin, String numetud, double notemath, double notephys, double notearabe,
        double noteanglais, String nomprenom) {
        super();
        this.id = id;
        this.cin = cin;
        this.numetud = numetud;
        this.notemath = notemath;
        this.notephys = notephys;
        this.notearabe = notearabe;
        this.noteanglais = noteanglais;
        this.nomprenom = nomprenom;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
```

Figure 3:Objet metier Bachelier

(3) Crée le web service bachelierservice.

Bachelierservice il nous permet d'assurer si le bachelier à des notes supérieures à 12 et ajouter les bacheliers accepter au tableau résultat de la base de données

```

package service;

import java.sql.Connection;
import javax.xml.ws.WebService;
import javax.xml.ws.WebServiceContext;
import javax.xml.ws.WebParam;
import javax.xml.ws.WebParamName;
import javax.xml.ws.WebMethod;

public class BachelierService {
    private WebServiceContext c;

    public boolean findOne(@WebParam(name="cin") String cin,@WebParam(name="numetud") String numetud) {
        boolean b = false;
        try {
            Connection conn = Singleton.getInstance().getConnection();
            PreparedStatement stmt = conn.prepareStatement("SELECT * FROM bacheliers WHERE cin = ? and numetud = ?");
            stmt.setString(1, cin);
            stmt.setString(2, numetud);
            ResultSet result = stmt.executeQuery();
            if (result.next()) {
                int id = result.getInt("id");
                double notemath = result.getDouble("notemath");
                double notephys = result.getDouble("notephys");
                double notearabe = result.getDouble("notearabe");
                double noteanglais = result.getDouble("noteanglais");
                String nomprenom = result.getString("nomprenom");

                if(notemath > 12 && notephys > 12 && notearabe > 12 && noteanglais > 12) {
                    b = true;
                    PreparedStatement stm = conn.prepareStatement("INSERT INTO resultat VALUES (NULL, ?, ?, ?, ?, ?)");
                    stm.setString(1, cin);
                    stm.setString(2, numetud);
                    stm.setString(3, nomprenom);
                    stm.execute();
                }
            }
        } catch (SQLException ex) {
            System.out.println("Erreur SQL : " + ex.getMessage());
        }
    }
}

```

Figure 4:le web service bachelierservice

(4) Déployer le web service.

Dans notre cas on a une application.java donc on aura besoin de créer notre propre serveur.

C'est une classe java main on va spécifier "le point d'accès" "le port" et pour publier le web service on utilise une classe qui s'appelle Endpoint qui fait partie de JAXWS avec la méthode publish qui contient le lien et l'instance de web service (ce serveur permet d'ouvrir un service http sur le port spécifier qui va être utilisé uniquement pour consulter ce web service).

```

package serveur;

import javax.xml.ws.Endpoint;

public class ServeurJWS {

    public static void main(String[] args) {
        String url="http://localhost:8553/";
        Endpoint.publish(url, new BachelierService());
        System.out.print(url);
    }
}

```

Figure 5:déploiement de web service

(5) Vérifier si le web service est ça marche.

Après l'exécution du projet en lance un navigateur et en tape l'adresse localhost:8553/BachelierWS?wsdl.

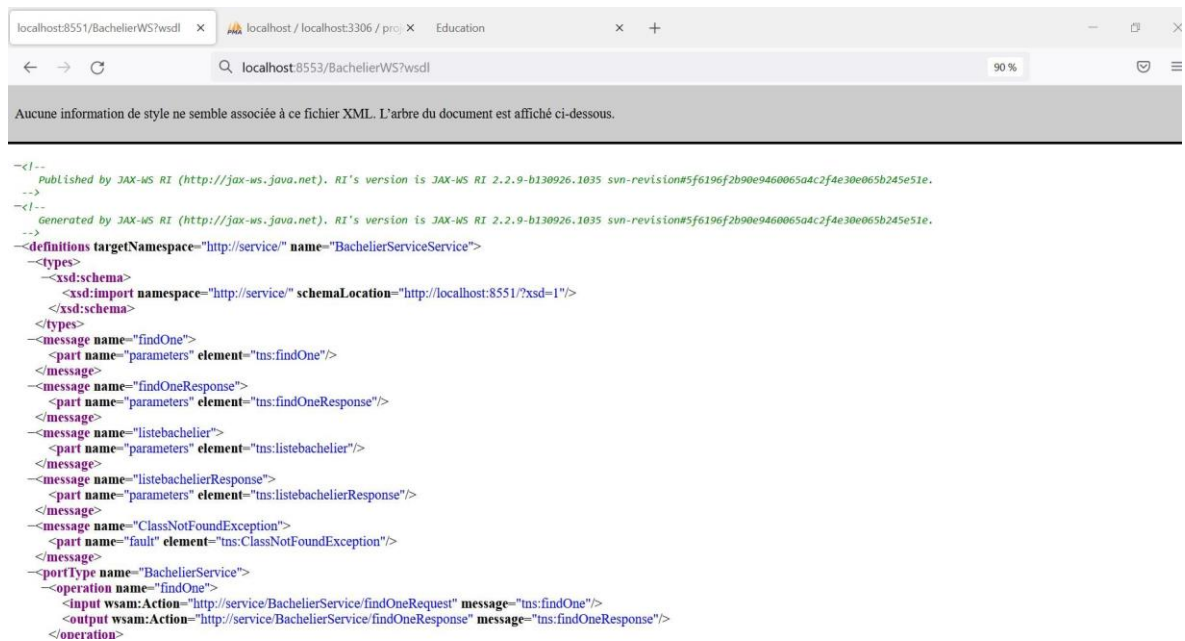


Figure 6:WSDL de web service

Ce wsdl c'est le contrat de web service il contient les composants de web service, par exemple les opérations de web service.

```

<portType name="BachelierService">
  <operation name="findOne">
    <input wsam:Action="http://service/BachelierService/findOneRequest" message="tns:findOne"/>
    <output wsam:Action="http://service/BachelierService/findOneResponse" message="tns:findOneResponse"/>
  </operation>
  <operation name="listebachelier">
    <input wsam:Action="http://service/BachelierService/listebachelierRequest" message="tns:listebachelier"/>
    <output wsam:Action="http://service/BachelierService/listebachelierResponse" message="tns:listebachelierResponse"/>
    <fault message="tns:ClassNotFoundException" name="ClassNotFoundException" wsam:Action="http://service/BachelierService/listebachelier/Fault/ClassNotFoundException"/>
  </operation>
</portType>

```

Figure 7:opérations de web service

B: Test avec outil SoapUI

SoapUI est un outil très utile dans la SOA. Il permet d'appeler des services et de contrôler les réponses. Il est aussi possible de lancer certains projets SoapUI en ligne de commande.

On va créer un projet soap en donnant le wsdl de notre web service.

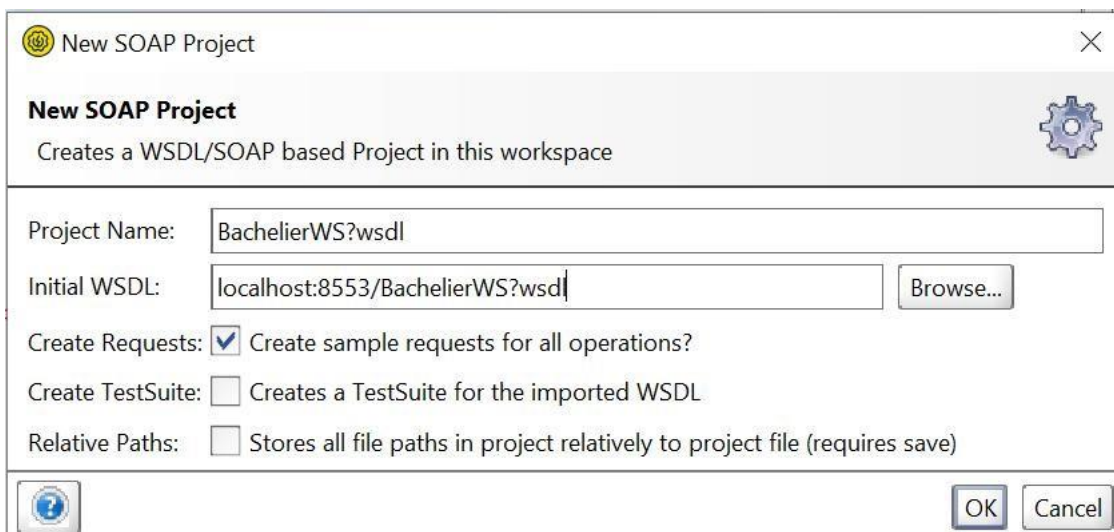


Figure 8:créer le projet soap



Figure 9:structure de projet soap

On va tester maintenant la méthode `findOne` qui return false si le bachelier à des notes inférieures à 12 ou true si le bachelier à des notes supérieures à 12.



Figure 10: test de methode findOne

Chapitre 2 : Client SOAP

1.Introduction

Le client en général un utilisateur sur un ordinateur ou un support équivalent effectue une requête dans un des langages suivants : XML, JSON ou HTTP. 2. Cette requête est transmise à un serveur distant via les protocoles SOAP, REST ou HTTP.

Dans cette partie du rapport on va créer le web service client de notre projet.

2.Créer le web service client

On souhaite créer Dynamique web project “BachelierClientt” en utilise le serveur tomcat .



Figure 11:BachelierClientt

On va créer ensuite un proxy, on va cliquer sur new :

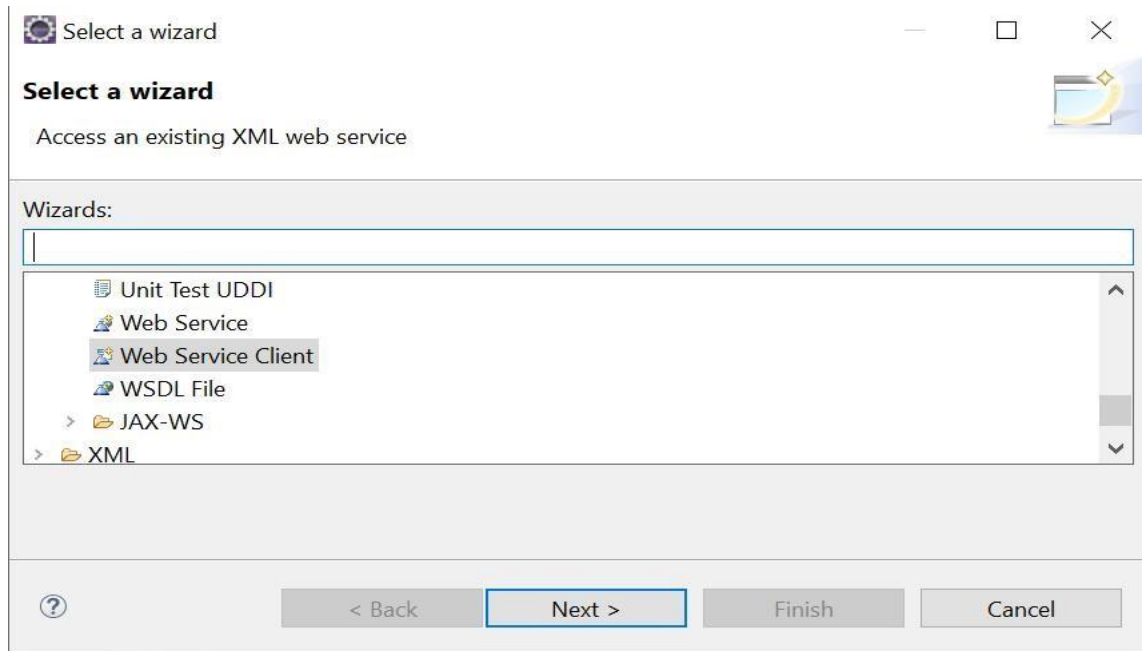


Figure 12:select a wizard pour service client

On va choisir le Web Service Client et on lui donne le lien de la WSDL :

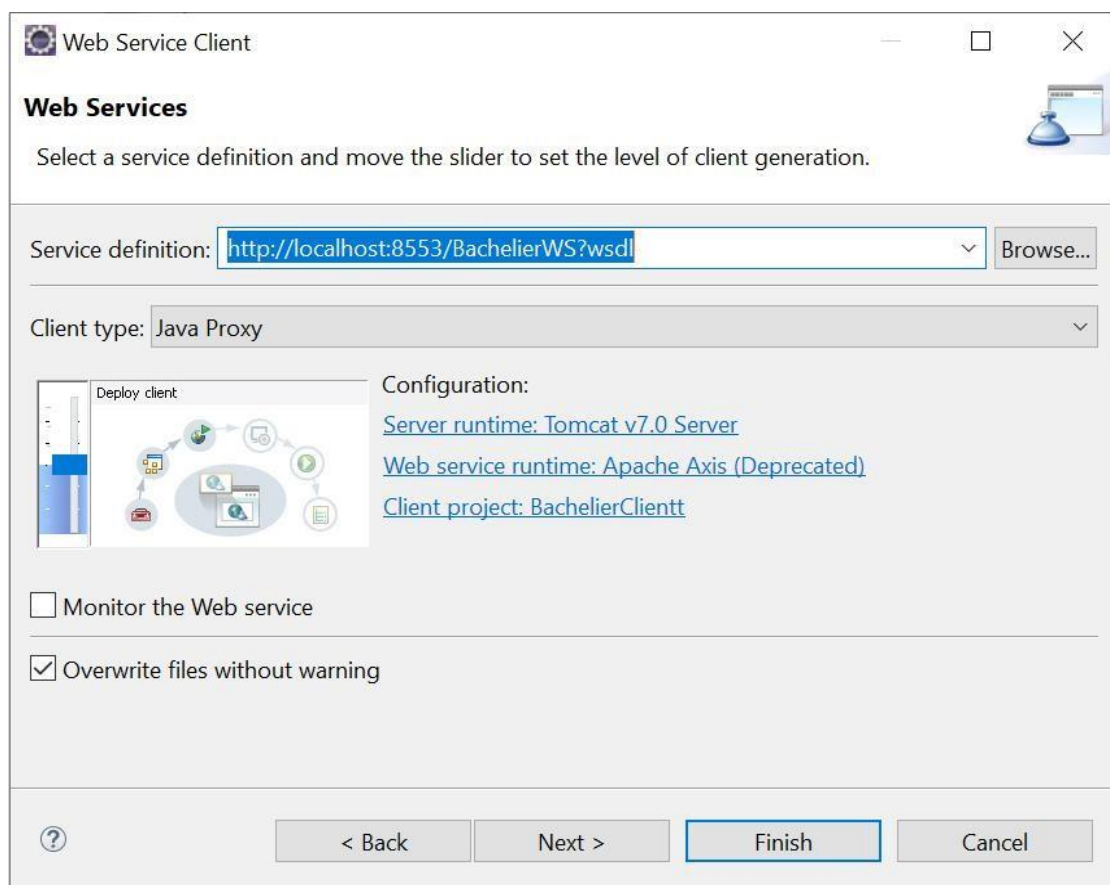


Figure 13:select a service definition

Donc le projet chargeait le WSDL et générait le proxy dans le dossier src/main/java aussi il va générer les jars à utiliser :

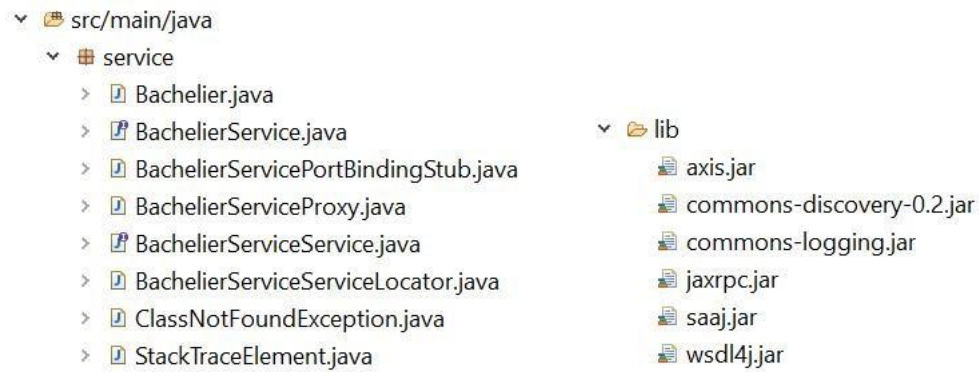


Figure 14:src/main/java et les jars ajouter

Maintenant on peut consommer notre méthode de web service dans le client

Chapitre 3 : 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.Diagramme de séquence

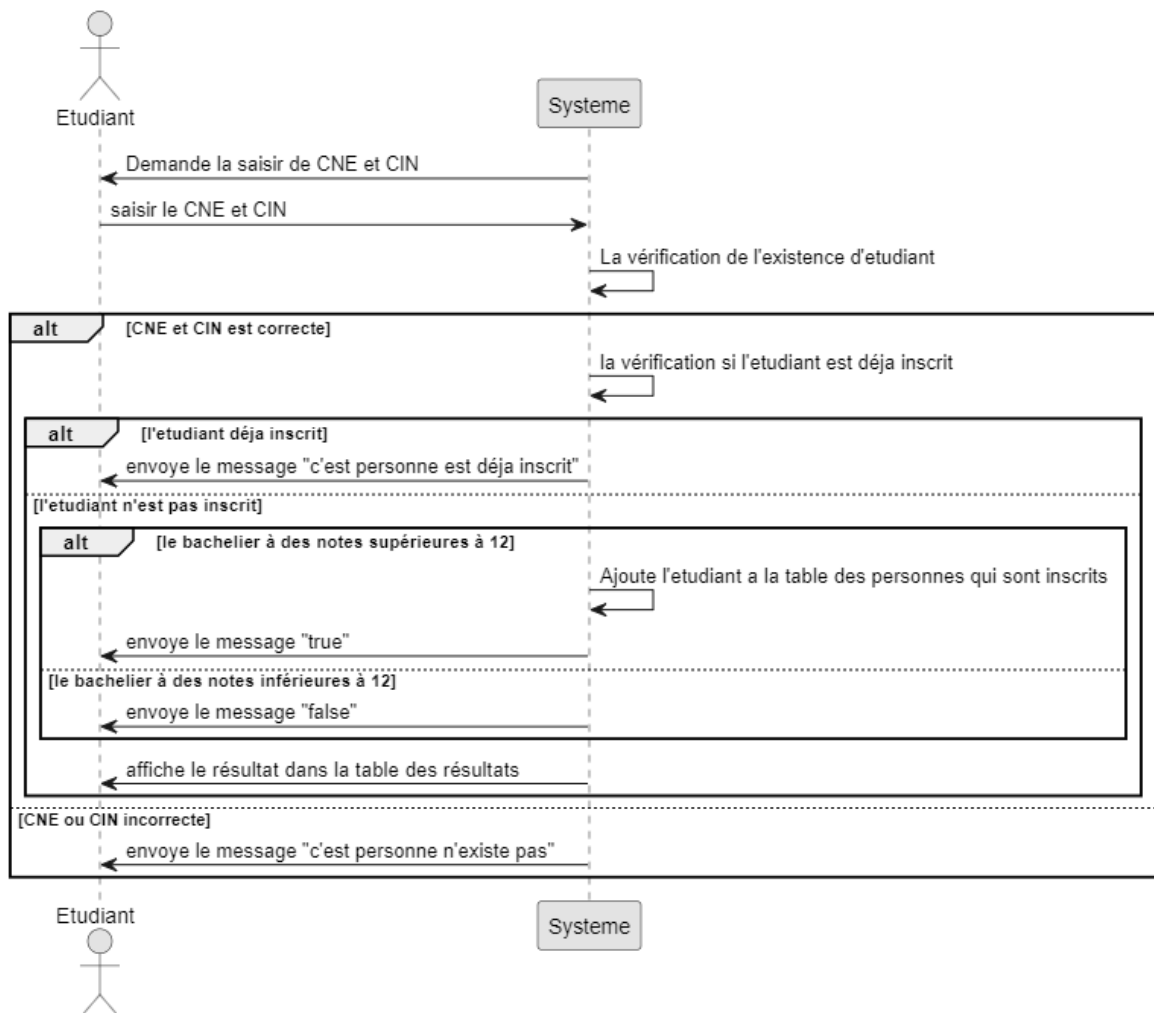


Figure 15:Diagramme de séquence

3. 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.

A: Page d'accueil

C'est la page d'accueil qui s'affiche dès l'accès au site web.

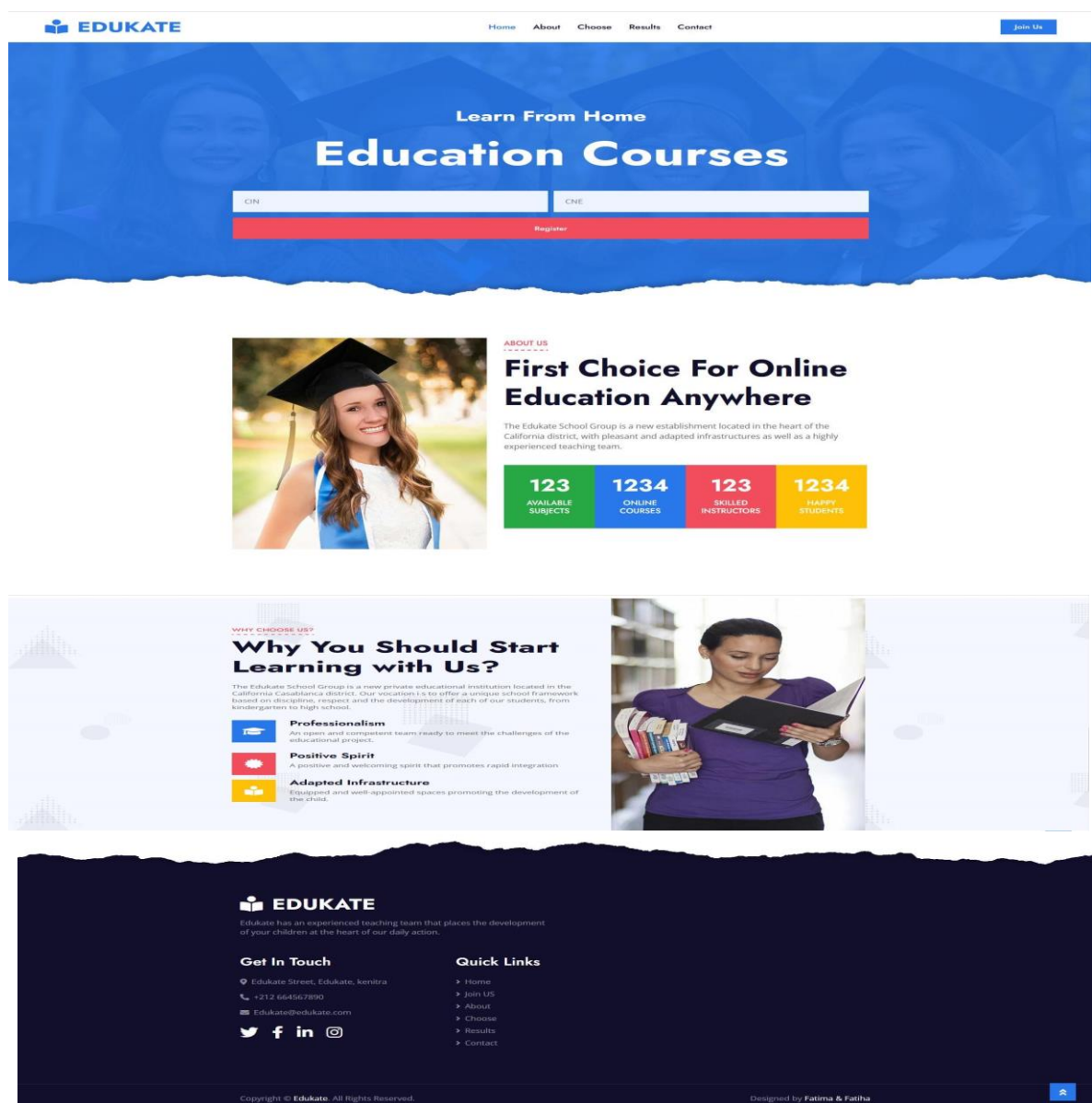


Figure 16: la page d'accueil

Dans cette page on a une partie pour le bachelier qui veut candidater au concours réaliser par notre institut EDUKATE. Le bachelier devrait saisir son numéro d'étudiant et son CIN.

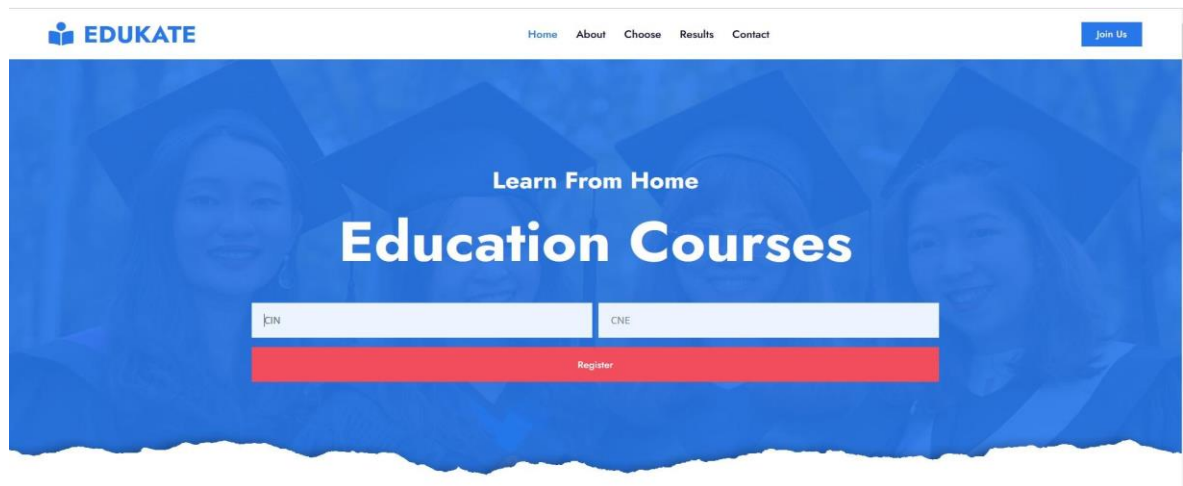
The image shows the top section of the EDUKATE website. At the top, there is a navigation bar with the EDUKATE logo on the left and links for Home, About, Choose, Results, and Contact on the right. A 'Join Us' button is also present. Below the navigation bar is a large blue banner with a background image of students. The banner contains the text 'Learn From Home' and 'Education Courses'. Below this text are two input fields labeled 'CIN' and 'CNE'. A red 'Register' button is positioned below these fields.

Figure 17:formulaire pour candidater

Et L'application fait appel au web service pour vérifier si l'étudiant concerné dispose d'une note supérieure à 12 dans toutes les matières précitées. Si oui, l'étudiant sera accepté pour passer le concours et l'application affiche une alert.

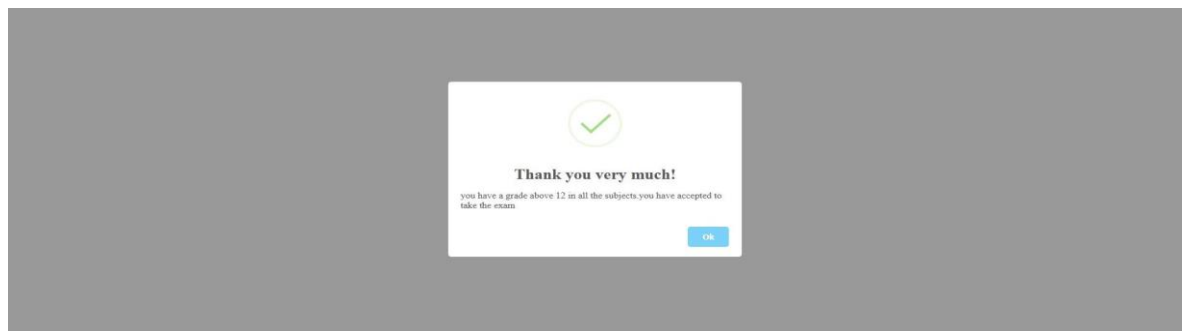


Figure 18:alert d'admission

Sinon sa candidature sera rejetée et l'application affiche une alert.



Figure 19:alert de refus

B: Page de resultats

C'est la page de résultats qui affiche la liste des bacheliers admis pour passer le concours.

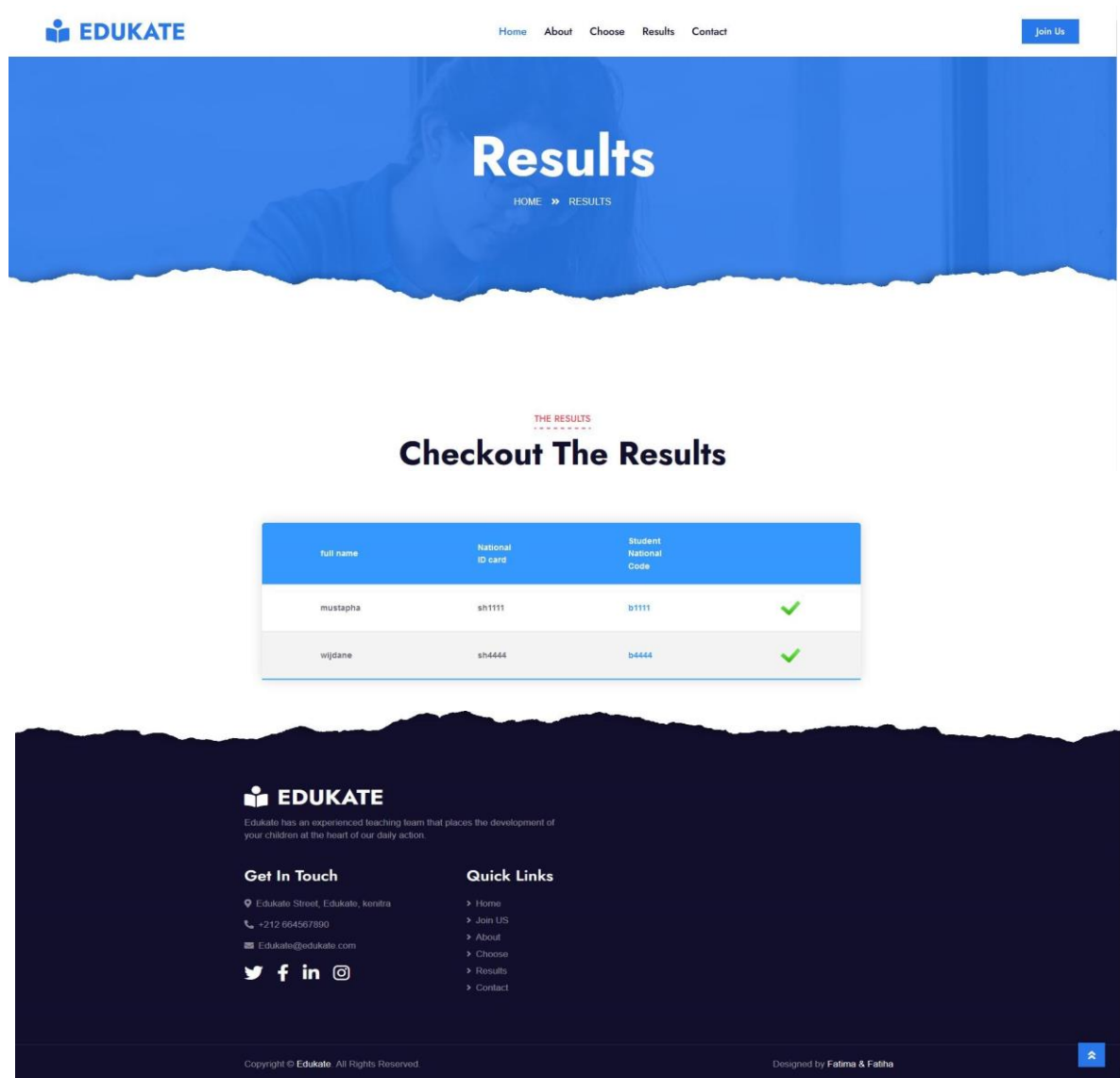


Figure 20:page de resultats

Conclusion

L'objectif visé à travers ce rapport est de présenter le mini projet réalisée pour le de module SOA. Dans un premier temps, Nous avons consacré nos réflexions à l'étude de l'existant pour améliorer notre application.

Cette étude nous permettons de passer à une phase plus importante qui est : citer les besoins et les répartir en besoins fonctionnels. En effet, nous posons 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. Pour les manipuler et vérifier les candidatures dans l'application Web. Pour procéder à cette application, j'ai utilisé MySQL, jsp, css, Bootstrap, javascript, soap, wsdl et uddi pour développer cette application.

Ce projet a été très enrichissant pour nous, nous avons eu l'opportunité d'enrichir nos compétences en web service par l'apprentissage de comment lier un web service avec une application. Par ailleurs durant ce projet, nous avons rencontré plusieurs difficultés au cours de la réalisation mais ça nous n'a pas empêché d'avance et d'améliorer nos techniques de travail.

Finalement, nous aimions remercier Monsieur pour cette occasion.