**Projet de synthèse**

MODÈLE DE CONCEPTION DU SYTÈME DE GESTION DES RESSOUCES HUMAINES DE « MAJESTÉ IMPORT-EXPORT »

18 mars 2024

**Sébastien Luc KAMGA KAMDEM**

**Serge Paulindrix TCHIEDJO TAGUIADOM**

**Wesly LOUIS**

**Omar MEKNI**

Table des matières

[**1.** **Introduction** 1](#_Toc161663728)

[1.1 Rappel de l'énoncé du projet 1](#_Toc161663729)

[1.1.1 Présentation de l’entreprise « Majesté Import-Export » 1](#_Toc161663730)

[1.1.2 L'intérêt du projet 1](#_Toc161663731)

[1.1.3 Gains recherchés 1](#_Toc161663732)

[1.2 Définitions et acronymes 1](#_Toc161663733)

[1.3 Documents de référence 2](#_Toc161663734)

[**2.** **Décisions stratégiques de conception de haut-niveau** 2](#_Toc161663735)

[2.1 Plate-forme d'exécution 2](#_Toc161663736)

[2.2 Outils de développement 3](#_Toc161663737)

[2.3 Standards à suivre 3](#_Toc161663738)

[2.4 Librairies de composantes à utiliser 3](#_Toc161663739)

[**3.** **Description des divisions** 4](#_Toc161663740)

[3.1 Description de la division en modules (paquetages) (architecture globale) 4](#_Toc161663741)

[3.2 Standard de communication utilisé (pour les projet ayant une composante réseau seulement) 6](#_Toc161663742)

[3.3 Description de la division en fichiers de données persistantes 6](#_Toc161663743)

[**4.** **Les interfaces utilisateurs** 8](#_Toc161663744)

[**5.** **Conception détaillée** 11](#_Toc161663745)

[5.1 Paquetage 1 : model 12](#_Toc161663746)

[5.1.1 Diagramme de paquetage réduit 12](#_Toc161663747)

[5.1.2 Description de l'interface du paquetage 12](#_Toc161663748)

[5.1.3 Diagramme de classes du paquetage 12](#_Toc161663749)

[5.1.4 Description des classes 12](#_Toc161663750)

[5.1.5 Description des fonctions avec algorithmes pour les fonctions les plus complexes (ou risquées) 12](#_Toc161663751)

[5.2 Paquetage 2 : service 13](#_Toc161663752)

[5.2.1 Description de l'interface du paquetage 13](#_Toc161663753)

[5.2.2 Diagramme de classes du paquetage 13](#_Toc161663754)

[5.2.3 Description des classes 13](#_Toc161663755)

[5.2.4 Description des patrons de conception utilisés dans le paquetage 13](#_Toc161663756)

[5.3 Paquetage 3 : repository 14](#_Toc161663757)

[5.3.1 Description de l'interface du paquetage 14](#_Toc161663758)

[5.3.2 Diagramme de classes du paquetage 14](#_Toc161663759)

[5.3.3 Description des classes 14](#_Toc161663760)

[5.3.4 Description des patrons de conception utilisés dans le paquetage 14](#_Toc161663761)

[5.4 Paquetage 4 : exception 14](#_Toc161663762)

[5.4.1 Description de l'interface du paquetage 15](#_Toc161663763)

[5.4.2 Diagramme de classes du paquetage 15](#_Toc161663764)

[5.4.3 Description des classes 15](#_Toc161663765)

[5.4.4 Description des patrons de conception utilisés dans le paquetage 15](#_Toc161663766)

[5.5 Paquetage 5 : contoller 15](#_Toc161663767)

[5.5.1 Description de l'interface du paquetage 16](#_Toc161663768)

[5.5.2 Diagramme de classes du paquetage 16](#_Toc161663769)

[5.5.3 Description des classes 16](#_Toc161663770)

[5.5.4 Description des patrons de conception utilisés dans le paquetage 16](#_Toc161663771)

[**6.** **Conclusion** 16](#_Toc161663772)

[**ANNEXES** 17](#_Toc161663773)

# **Introduction**

Ce document présente en détail le travail effectué pour la conception de l'application de gestion des ressources humaines pour l'entreprise "Majesté Import-Export". Nous débutons par un rappel de l'énoncé du projet, en présentant l'entreprise ainsi que les raisons qui ont motivé le besoin d'une telle application. Ensuite, nous discutons des intérêts attendus de cette application en termes de gestion du personnel. Nous fournissons également des définitions et des références importantes pour mieux contextualiser le projet.

## Rappel de l'énoncé du projet

## Présentation de l’entreprise « Majesté Import-Export »

« ***Majesté Import-Export*** » est une jeune entreprise créée et fonctionnelle depuis 2017. Elle fait dans le commerce des produits qu’elle importe de certains pays pour les exporter ensuite dans d’autres. Jusqu’à présent, elle a toujours fonctionné avec un seul salarié s’occupant de toutes les tâches. L’entreprise étant en pleine croissance, projette d’augmenter son personnel. A cet effet, elle a besoin d’une application de gestion des ressources humaines.

## L'intérêt du projet

Jusqu’à ce jour, l’entreprise n’avait qu’un seul employé qui menait toutes les tâches manuellement sans avoir besoin d’un logiciel. Elle a eu le choix entre acheter un logiciel existant ou bien faire créer un logiciel qui lui est propre. Au vue des coûts et des fonctions spécifiques désirées, elle a choisi la deuxième option à savoir celle de développer une nouvelle application de gestion de ressources humaines.

L’application désirée a pour objectif d’automatiser la gestion des ressources humaines de l’entreprise « ***Majesté Import-Export*** ».

## Gains recherchés

Cette application va faciliter la gestion efficace du personnel notamment avec :

* la centralisation des données,
* l’automatisation des processus (gestion des congés et des absences, de la paie, des horaires, des avantages sociaux),
* la gestion des talents (identification des compétences, planification de la relève),
* le suivi des performances,
* l’amélioration de la communication interne,
* la facilitation du recrutement (publication des offres d’emploi, candidature, entrevue, sélection, contrat de service),
* l’analyse des données des ressources humaines, et
* la gestion des carrières (promotion, formation, sanctions).

## Définitions et acronymes

**RH** : Ressource humaine

## Documents de référence

Soraya Ferdenache (2024) Planification des activités du cours Projet de synthèse.

Modèle de document de conception (Livrable 2 : Document de conception)

Modèle de cahier de charges

IC-Projet-plan-Template-8538-V1

<https://www.geeksforgeeks.org/introduction-to-spring-boot/> : Introduction to Spring Boot

[Spring Boot Tutorial | Full Course [2023] [NEW] (youtube.com)](https://www.youtube.com/watch?v=9SGDpanrc8U) : Spring Boot Tutorial | Full Course [2023]

[Spring Boot MVC Project Architecture Diagram (youtube.com)](https://www.youtube.com/watch?app=desktop&v=gtBS7HKfy-k) : Spring Boot MVC Project Architecture Diagram

<https://www.youtube.com/watch?v=Ho08ahyc12s&list=PL5t0n9Tgavtm5xik2fzSFcJKW09Jdonk6&ab_channel=Vision6D> MS Project 2019 en français

<https://www.youtube.com/watch?v=fjdni96UuVo&list=PLNbb2hOYRUgz3dgfOTyUgVjYOUBmm_OAX&ab_channel=SciadoPartenaires> Tutoriel MSProject 2019

[Quelle part de marché pour Linux dans le monde ? (Février 2024) (goodtech.info)](https://goodtech.info/part-de-marche-linux/#:~:text=Pour%20les%20tablettes%2C%20Android%20occupe,et%20Linux%200%2C13%20%25.) : Quelle part de marché pour Linux dans le monde ? Derniers chiffres

<http://www.thymeleaf.org> : Thymeleaf

<https://app.uizard.io/prototypes>

<https://en.wikipedia.org/wiki/Convention_over_configuration> : Convention over configuration

<https://www.baeldung.com/mockito-series> : Mockito Tutorial

Philippe Nasr, Charlène Poirier (2022) La gestion de projet, 4e édition.

Cours 420-A03-BB Analyse des besoins à la première session 2023

Cours 420-A08-BB Analyse et modélisation à la deuxième session 2023

# **Décisions stratégiques de conception de haut-niveau**

## Plate-forme d'exécution

Nous avons travaillé dans l’environnement Windows pour les raisons suivantes :

* elle est la plateforme la plus utilisée de nos jours,
* la facilité d’utilisation pour les débutants,
* le support matériel et professionnel.

Nous avons choisi de développer une application Web pour plusieurs raisons :

* l’accessibilité universelle (desktop, portable, téléphone, tablette),
* la facilité de maintenance,
* la compatibilité multiplateforme,
* la collaboration simplifiée,
* l’évolutivité,
* la sécurité renforcée,
* le téléchargement local non requis, ce qui évite la surcharge de la machine client lors de l'utilisation.

## Outils de développement

**Tableau 1 :** Outils de développement

|  |  |
| --- | --- |
| **Outils de développement** | |
| IDE | Intellij IDEA |
| Langage | Java |
| Base des données | mySQL |
| Logiciels | - Visual paradigm (UML) pour le développement des diagrammes des cas d’utilisations, de classe et de séquence  - Uizard pour le développement des maquettes. |
| Besoins en licence | - Intellij IDEA  - Ms Project |

## Standards à suivre

~~Les différents standards suivis pour la conception (ex. : UML) et pour l'implantation (ex. : TCP/IP,....). Ce pourrait être également des standards de programmation (règle de nomenclature de variables par exemple).~~

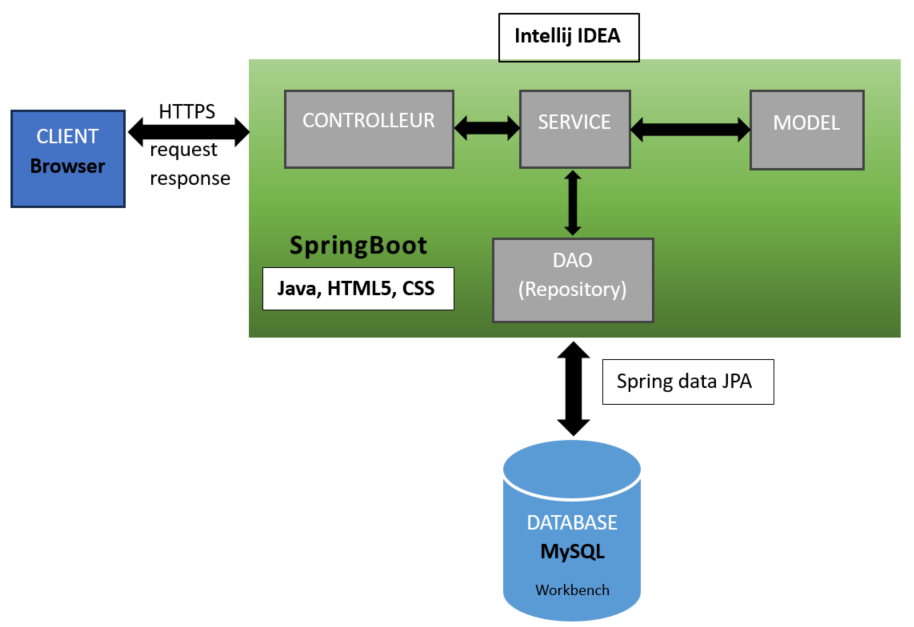
## Librairies de composantes à utiliser

**Tableau 2 :** Librairies de composantes utilisées

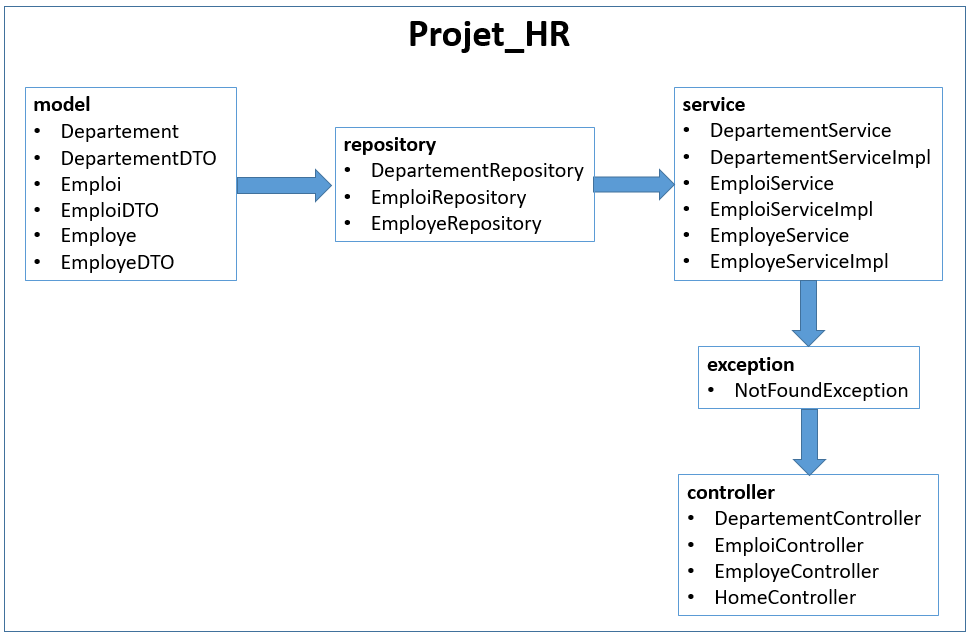
|  |  |  |
| --- | --- | --- |
| **Désignation** | **Source** | **Utilité** |
| SpringBoot | Cf. 1.3 | Pour simplifier le processus de création et de déploiement d’application Spring basé sur Java. |
| Thymeleaf | Cf. 1.3 | Pour afficher la vue |
| Mockito | Cf. 1.3 | Pour faire les tests unitaires |

# **Description des divisions**

## Description de la division en modules (paquetages) (architecture globale)



**Figure 1** : Architecture de l’application



**Figure 2 :** Diagramme de paquetages global

**Tableau 3 :** Justification de l'existence des paquetages, des classes et de leurs liaisons

|  |  |  |
| --- | --- | --- |
| **Désignation** | | **Justification** |
| ***package*** | ***Model*** | *Ce package contient des classes qui représentent les entités de données de l'application, ainsi que des objets de transfert de données (DTO) associés.* |
| **Classe** | Departement | Les classes Departement, Emploi, et Employe représentent les entités métier principales de l'application. Elles interagissent avec la couche d'accès aux données et la couche de présentation de l'application. |
| Emploi |
| Employe |
| DepartementDTO | Les classes DTO (DepartementDTO, EmploiDTO, EmployeDTO) sont utilisées pour transférer des données entre les différentes couches de l'application, souvent entre le contrôleur et les services. |
| EmploiDTO |
| EmployeDTO |
| ***package*** | ***service*** | *Ce package contient des classes qui fournissent des services métier pour manipuler les données et effectuer des opérations métier.* |
| **Classe** | DepartementService | DepartementService, EmploiService, et EmployeService définissent les interfaces pour les services associés à chaque entité. |
| EmploiService |
| EmployeService |
| DepartementServiceImpl | DepartementServiceImpl, EmploiServiceImpl, et EmployeServiceImpl fournissent les implémentations concrètes de ces services, où la logique métier est réellement mise en œuvre. |
| EmploiServiceImpl |
| EmployeServiceImpl |
| ***package*** | ***repository*** | *Le package repository sert à fournir une abstraction de la couche d'accès aux données, à définir les opérations de persistance des entités métier.* |
| **Classe** | DepartementRepository | DepartementRepository, EmploiRepository et EmployeRepository sont des interfaces qui définissent les opérations de persistance pour chaque entité. Ces interfaces sont implémentées par Spring Data JPA pour effectuer les opérations CRUD sur les entités. |
| EmploiRepository |
| EmployeRepository |
| ***package*** | ***controller*** | *Ce package contient des classes qui agissent comme des contrôleurs dans le modèle MVC, gérant les requêtes HTTP et les interactions entre l'interface utilisateur et le reste de l'application.* |
| **Classe** | DepartementController | DepartementController, EmploiController, et EmployeController sont responsables de la gestion des requêtes liées à chaque entité respective. Elles assurent la liaison entre Back-end et Front-end. |
| EmploiController |
| EmployeController |
| HomeController | HomeController est responsable de la gestion des requêtes liées à la page d'accueil de l'application. Elle assure la liaison entre Back-end et Front-end. |
| **package** | ***exception*** | *Ce package contient la classe qui définit des exceptions spécifiques et personnalisées à l'application.* |
| **Classe** | NotFoundException | NotFoundException est utilisée pour signaler qu'une ressource demandée n'a pas été trouvée, par exemple, lorsqu’on effectue une recherche et que l'élément recherché n'est pas présent dans la base de données. |

## Standard de communication utilisé (pour les projet ayant une composante réseau seulement)

~~Vous devez présenter le standard de communication que vous utiliserez au niveau de l'application. Comment allez-vous échanger les informations ?, quel sera le format des messages échangés ? (format des requêtes, échanges de fichiers, etc.) Votre protocole de communication doit être clairement défini.~~

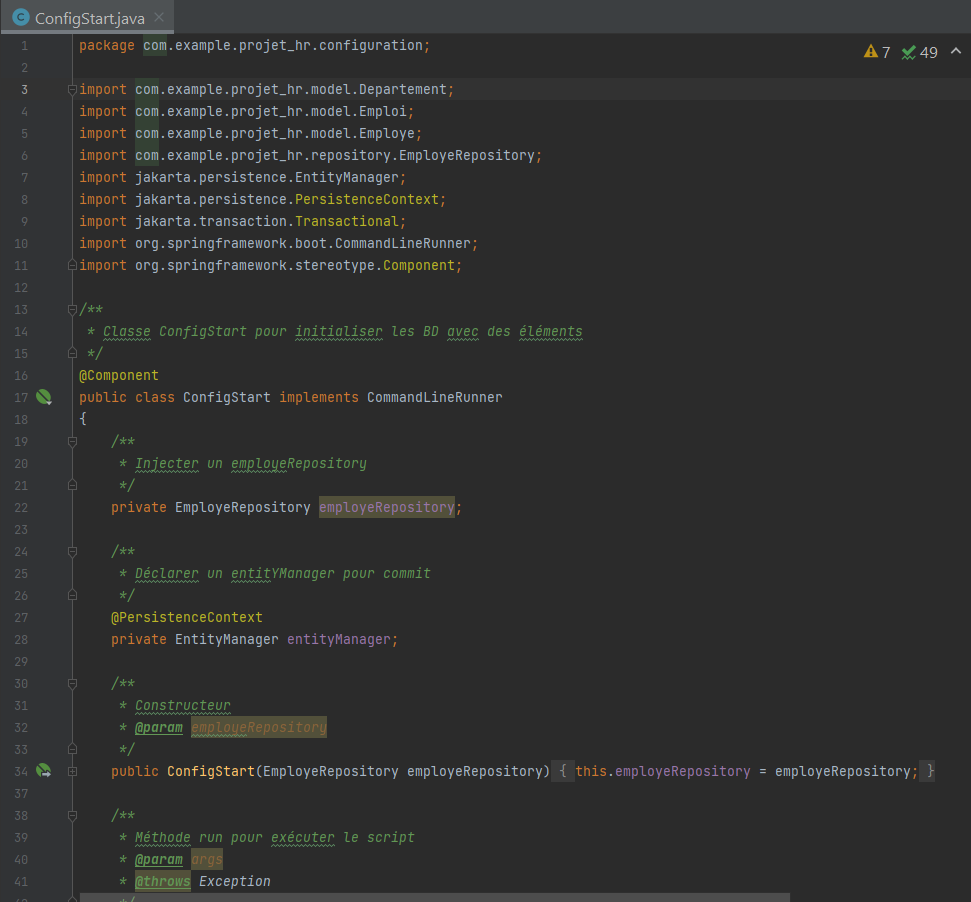
~~Note: On ne parle pas des standards réseaux tels TCP/IP ici. On parle plutôt de la couche 7 du modèle OSI (la couche Application). Bref, définissez le format des message échangés par votre application.~~

## Description de la division en fichiers de données persistantes

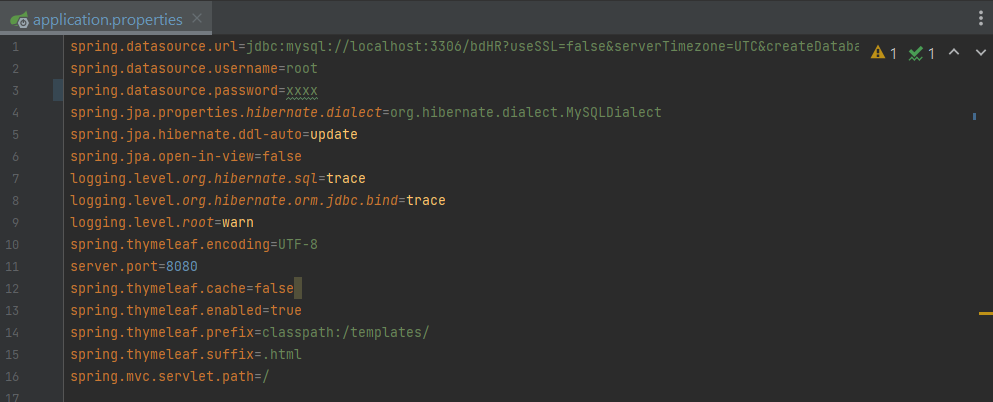
**Pom.xml**

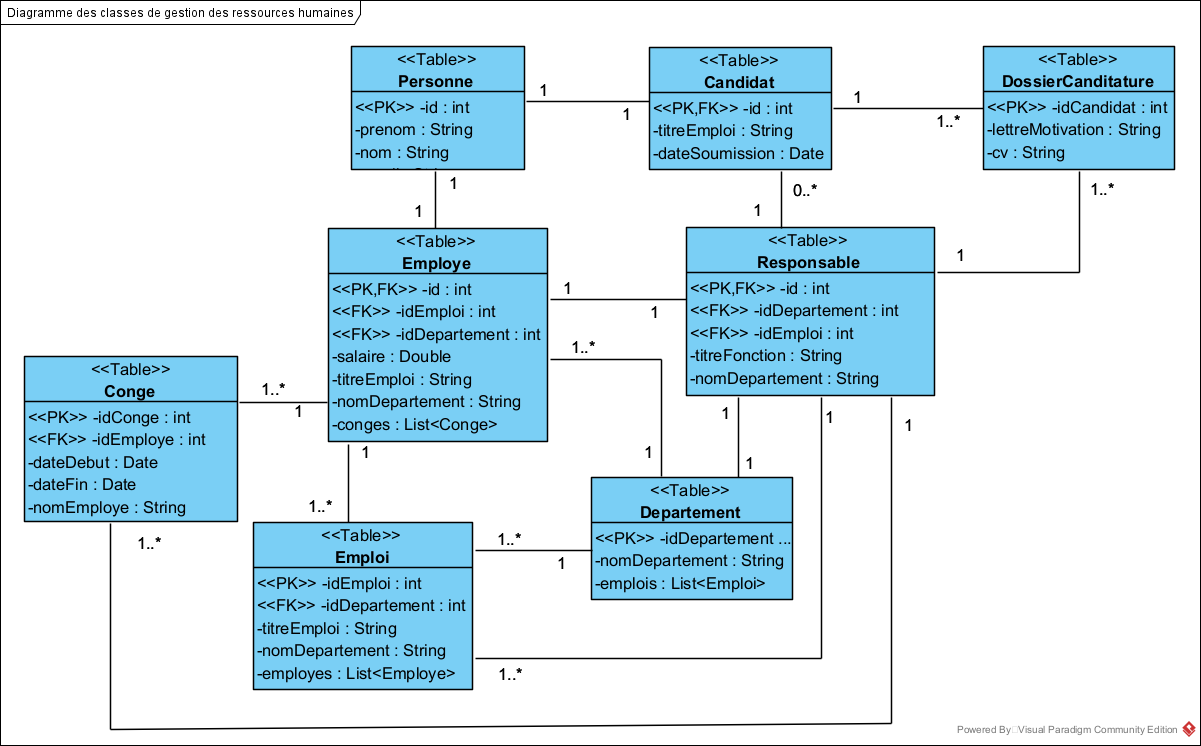
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.2</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.example</groupId>  
 <artifactId>projet\_HR</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>projet\_HR</name>  
 <description>projet\_HR</description>  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-thymeleaf</artifactId>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <configuration>  
 <excludes>  
 <exclude>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 </exclude>  
 </excludes>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**Fichiers de configuration (ConfigStart)**



**Fichier Properties de Spring Boot (log)**



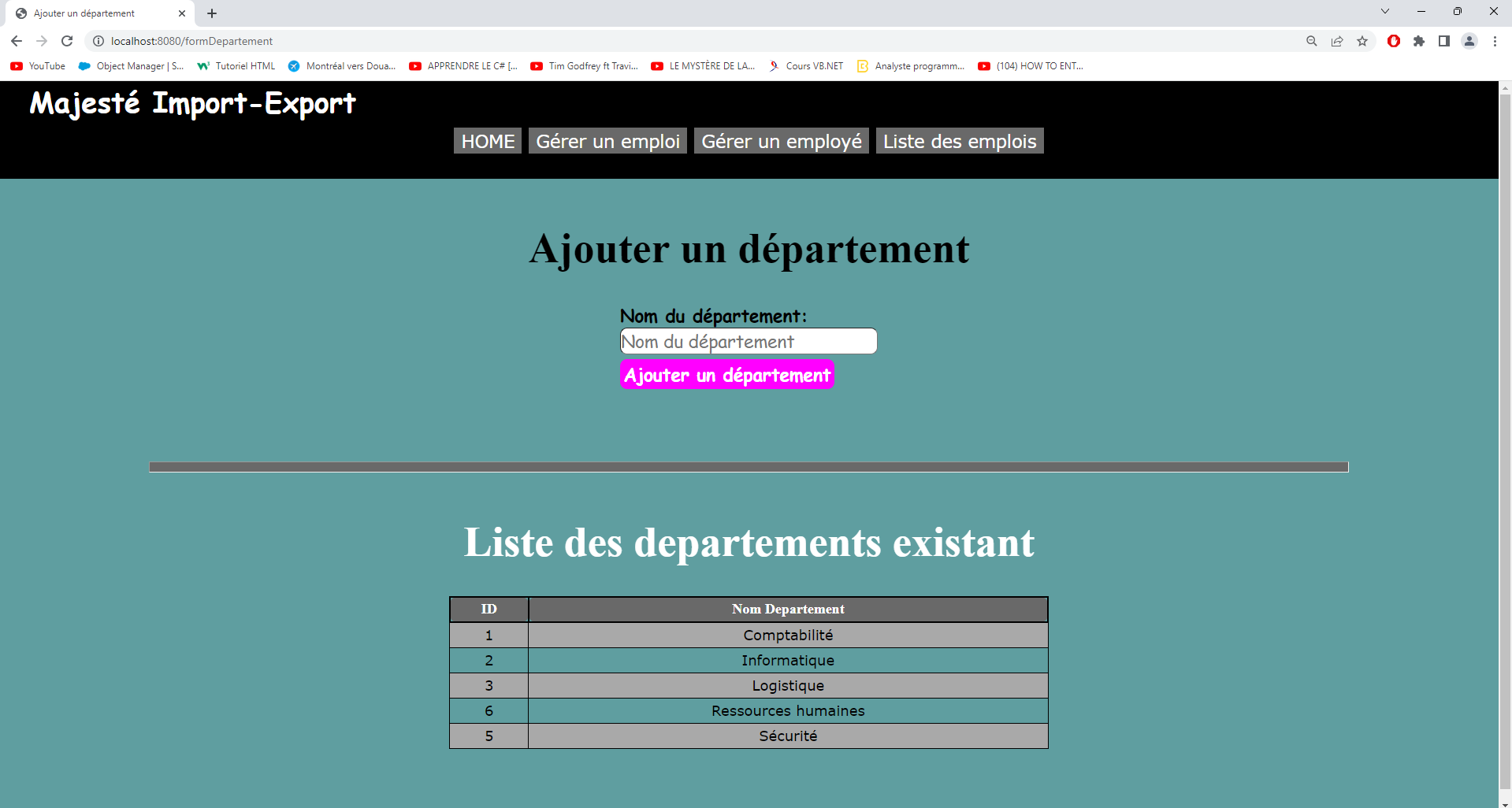


**Figure 3** : Modèle logique des données

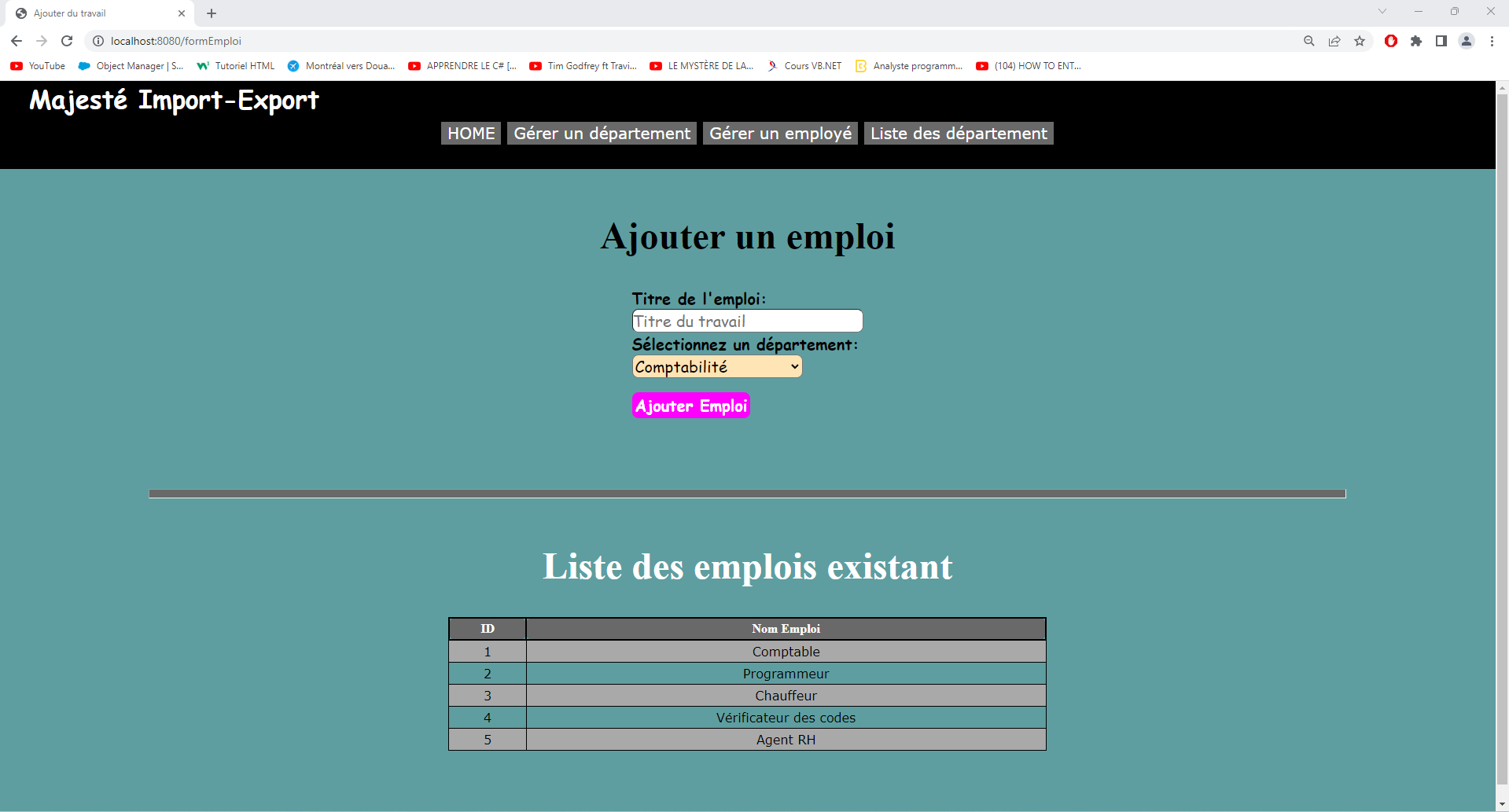
# **Les interfaces utilisateurs**



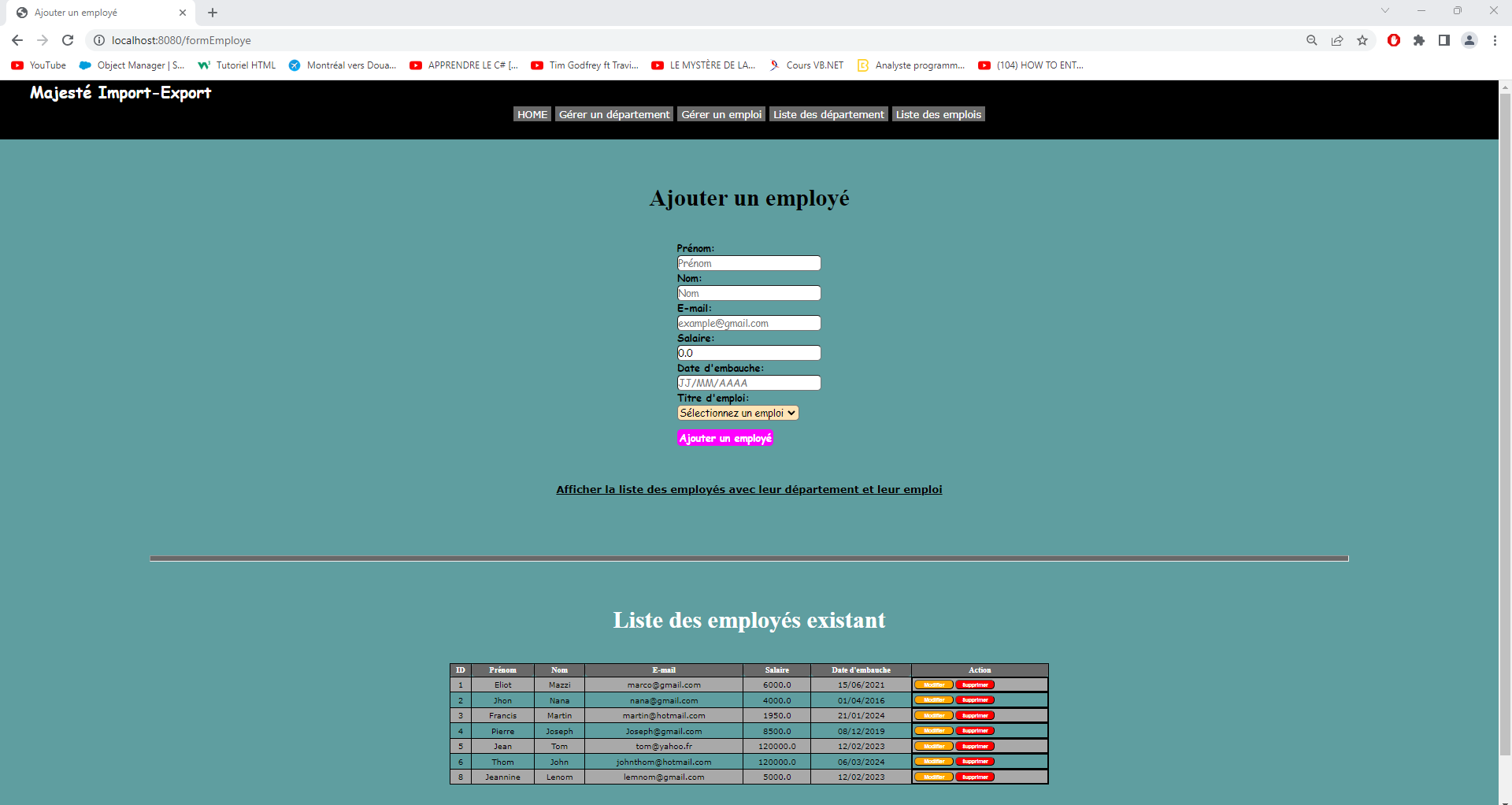
**Figure 4 :** Page d’accueil



**Figure 5** : Page d’ajout de département



**Figure 6 :** Page d’ajout d’emploi



**Figure 7 :** Page d’ajout d’employé



**Figure 8 :** Page d’affichage de la liste des départements

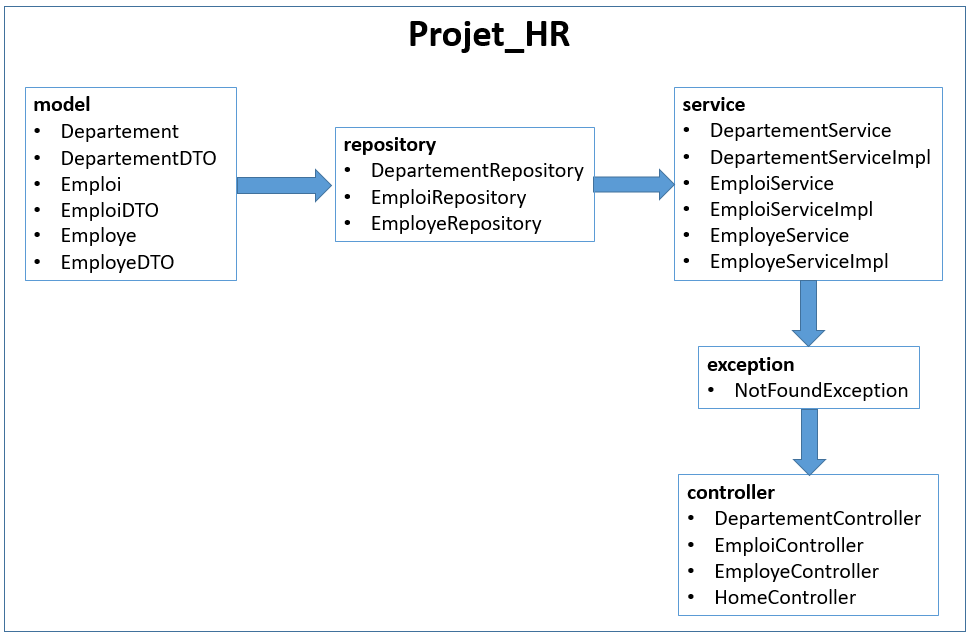


**Figure 9 :** Page d’affichage de la liste des emplois



**Figure 10 :** Page d’affichage de la liste des employés

# **Conception détaillée**



**Figure 9 :** Diagramme de paquetages global

## Paquetage 1 : model

## Diagramme de paquetage réduit

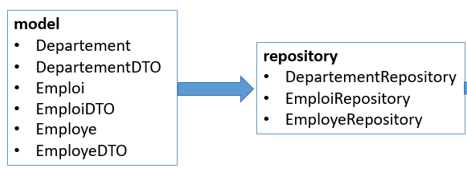
**Tableau 4 :** Description des classes du model

|  |  |  |
| --- | --- | --- |
| **Désignation** | | **Description** |
| ***Model*** | |  |
| **Classe** | Departement | **Attributs :**  Id  Nom du département  Liste des employés  Liste des emplois  **Méthodes :**  Ajouter employé()  Ajouter emploi()  Get departement() |

## Description de l'interface du paquetage

~~Comment les autres paquetages (ou classes) du système interagissent-ils avec le paquetage étudié ? Quel(s) est(sont) le(s) point(s) d'entrée du paquetage ? Est-ce qu'il y a une façade ?~~

## Diagramme de classes du paquetage



## Description des classes

La classe **Departement** du **model** représente un département dans notre système RH, avec des attributs tels que le nom du département et des relations avec les employés et les emplois associés à ce département. Elle est conçue pour être utilisée dans un environnement de persistance de données avec une base de données relationnelle.

## Description des fonctions avec algorithmes pour les fonctions les plus complexes (ou risquées)

~~Présenter une courte description de chacune des fonctions de chacune des classes (Entrées/Sorties, Arguments, Valeur de retour, une courte description (une ligne)). Pour les fonctions les plus complexes ou plus risquées, présenter des algorithmes. Les algorithmes peuvent être faits pseudo-code ou encore sous la forme d'un diagramme (de séquence, de collaboration ou d'état) selon ce qui vous semble le mieux approprié pour la fonction en question. On suggère fortement de placer cette section en annexe.~~

## Paquetage 2 : service

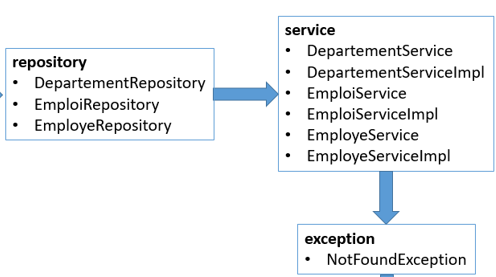
**Tableau 5 :** Description des classes du service

|  |  |  |
| --- | --- | --- |
| **Désignation** | | **Description** |
| ***service*** | |  |
| **Classe** | EmploiServiceImpl | **Attributs :**  emploiRepository  **Méthodes :**  saveEmploi()  findEmploiById()  deleteEmploiById()  findAll()  updateTitreEmploiById()  findEmploiTitreEmploi()  getAllEmplois() |

## Description de l'interface du paquetage

L’interface **EmploiService** fournit un ensemble de méthodes permettant d'effectuer des opérations CRUD (Create, Read, Update, Delete) sur les emplois dans le système RH. Elle définit les opérations nécessaires pour gérer les emplois au niveau de la couche de service de l'application.

## Diagramme de classes du paquetage



## Description des classes

La classe **EmploiServiceImpl** fournit une implémentation des méthodes définies dans l'interface **EmploiService**, permettant ainsi de gérer les emplois dans le système RH en interagissant avec la base de données.

## Description des patrons de conception utilisés dans le paquetage

~~Expliquer les patrons utilisés dans ce paquetage et justifier leur utilisation. Ont-ils été modifiés (ou adaptés) pour le système ? Si oui, de quelle manière et surtout, pourquoi ?~~

~~Il est possible que vous n'ayez pas utilisé de patrons de conception dans certains paquetages. Dans ce cas, laissez simplement tomber cette sous-section.~~

## Paquetage 3 : repository

**Tableau 6 :** Description des classes du repository

|  |  |  |
| --- | --- | --- |
| **Désignation** | | **Description** |
| ***repository*** | |  |
| **Classe** | EmployeRepository | **Attributs :**  **Méthodes :**  findByEmail() |

## Description de l'interface du paquetage

L’interface **EmploiRepository** permet d'interagir avec la table des emplois dans la base de données en fournissant des méthodes pour effectuer des opérations de recherche et de manipulation sur les emplois.

## Diagramme de classes du paquetage



## Description des classes

La classe **EmploiRepository** permet d'interagir avec la table des emplois dans la base de données en fournissant des méthodes pour effectuer des opérations de recherche et de manipulation sur les emplois.

## Description des patrons de conception utilisés dans le paquetage

~~Expliquer les patrons utilisés dans ce paquetage et justifier leur utilisation. Ont-ils été modifiés (ou adaptés) pour le système ? Si oui, de quelle manière et surtout, pourquoi ?~~

~~Il est possible que vous n'ayez pas utilisé de patrons de conception dans certains paquetages. Dans ce cas, laissez simplement tomber cette sous-section.~~

## Paquetage 4 : exception

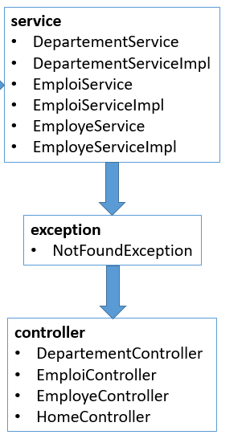
**Tableau 7 :** Description des classes du package exception

|  |  |  |
| --- | --- | --- |
| **Désignation** | | **Description** |
| ***exception*** | |  |
| **Classe** | NotFoundException | **Attributs :**  **Méthodes :**  NotFoundException() |

## Description de l'interface du paquetage

~~Comment les autres paquetages (ou classes) du système interagissent-ils avec le paquetage étudié ? Quel(s) est(sont) le(s) point(s) d'entrée du paquetage ? Est-ce qu'il y a une façade ?~~

## Diagramme de classes du paquetage



## Description des classes

La classe **NotFoundException** fournit une structure standardisée pour gérer les exceptions de type "Not Found" dans l'application, en permettant de transmettre des messages d'erreur explicites aux parties concernées lorsqu'une entité spécifique n'est pas trouvée.

## Description des patrons de conception utilisés dans le paquetage

~~Expliquer les patrons utilisés dans ce paquetage et justifier leur utilisation. Ont-ils été modifiés (ou adaptés) pour le système ? Si oui, de quelle manière et surtout, pourquoi ?~~

~~Il est possible que vous n'ayez pas utilisé de patrons de conception dans certains paquetages. Dans ce cas, laissez simplement tomber cette sous-section.~~

## Paquetage 5 : contoller

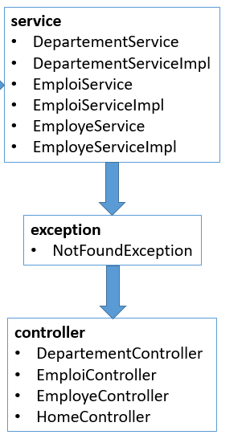
**Tableau 8 :** Description des classes du controller

|  |  |  |
| --- | --- | --- |
| **Désignation** | | **Description** |
| ***controller*** | |  |
| **Classe** | HomeController | **Attributs :**  employeService  **Méthodes :**  getAllEmployes()  convertToDTO() |

## Description de l'interface du paquetage

~~Comment les autres paquetages (ou classes) du système interagissent-ils avec le paquetage étudié ? Quel(s) est(sont) le(s) point(s) d'entrée du paquetage ? Est-ce qu'il y a une façade ?~~

## Diagramme de classes du paquetage



## Description des classes

La classe **HomeController** est responsable de gérer les requêtes GET vers l'URL "/home" et de fournir les données nécessaires à la vue associée en récupérant les employés à partir du service, en les convertissant en objets DTO et en les ajoutant au modèle.

## Description des patrons de conception utilisés dans le paquetage

~~Expliquer les patrons utilisés dans ce paquetage et justifier leur utilisation. Ont-ils été modifiés (ou adaptés) pour le système ? Si oui, de quelle manière et surtout, pourquoi ?~~

~~Il est possible que vous n'ayez pas utilisé de patrons de conception dans certains paquetages. Dans ce cas, laissez simplement tomber cette sous-section.~~

# **Conclusion**

Ce rapport détaille minutieusement les différentes phases de conception de l'application de gestion des ressources humaines pour "Majesté Import-Export". À travers l'analyse stratégique, les choix technologiques, et la description détaillée des divisions du système, nous avons travaillé à garantir une conception solide et efficace répondant aux besoins spécifiques de l'entreprise.

Compte tenu du temps et des difficultés rencontrées, nous n’avons pas pu respecter nos plans initiaux. Nous nous sommes concentrés sur les fonctionnalités de base demandées par le client. Avec une bonne organisation au départ, on aurait pu avancer plus que ça.

# **ANNEXES**

**Annexe1** : Tableau de répartition des tâches

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Package** | **Serge et Wesly** | | **Omar et Sébastien** | | **Tout le groupe** |
| **Model** | Departement  DepartementDTO | Serge  Wesly | Employe  EmployeDTO | Sébastien  Omar | Emploi  EmploiDTO |
| **Repository** | DepartementRepository | Serge | EmployeRepository | Sébastien | EmploiRepository |
| **Service** | DepartementService  DepartementServiceImpl | Serge  Wesly | EmployeService  EmployeServiceImpl | Sébastien  Omar | EmploiService  EmploiServiceImpl |
| **Controller** | DepartementController | Wesly | EmployeController | Omar | EmploiController  HomeController |
| **Configuration** |  |  |  |  | ConfigStart |
| **Exception** |  |  |  |  | NotFoundException |
| **Template/HTML** | formDepartement.html  employes.html | Serge  Wesly | formEmploye.html  editEmploye.html | Sébastien  Omar | formEmploi.html  home.html |
| **Static/SCRIPT** |  |  |  |  | script.js |
| **Static/CSS** |  |  |  |  | style.css |
| **Test** | DepartementServiceImplTest | Wesly | EmployeServiceImplTest | Omar | EmploiServiceImplTest |