

Informatiser le service Ressources Humaines

Rchid Anas

- 1 Introduction
- 2 Cahier des Charges
- 3 Conception
- 4 L'Interface Graphique

Introduction

Ce projet est le résultat d'un stage que j'avais passé chez la Faculté des Lettres et Science Humaines, El Jadida sous le thème Informatiser le service Ressources Humaines. Sous l'encadrement de Mr. A. Madani, et la supervision du chef de service; Mr. Driss Dibaji.

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*
- *Gérer les diplômes et les grades*

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*
- *Gérer les diplômes et les grades*
- *Suivi des grades*

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*
- *Gérer les diplômes et les grades*
- *Suivi des grades*
- *Suivi d'absence*

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*
- *Gérer les diplômes et les grades*
- *Suivi des grades*
- *Suivi d'absence*
- *Suivi de rémunération du travail les fériés*

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*
- *Gérer les diplômes et les grades*
- *Suivi des grades*
- *Suivi d'absence*
- *Suivi de rémunération du travail les fériés*
- *Générer des attestations de travail*

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*
- *Gérer les diplômes et les grades*
- *Suivi des grades*
- *Suivi d'absence*
- *Suivi de rémunération du travail les fériés*
- *Générer des attestations de travail*
- *Générer des autorisations de congé*

Cahier des Charges

Un service qui est responsable de la gestion des employés et fonctionnaires, leurs diplômes et grades. . .

- *Implémenter un système de gestion des employés/fonctionnaires*
- *Gérer les diplômes et les grades*
- *Suivi des grades*
- *Suivi d'absence*
- *Suivi de rémunération du travail les fériés*
- *Générer des attestations de travail*
- *Générer des autorisations de congé*
- *Générer des fiches de notation annuelle*

Conception

Stockage des données en XML

Conception

Stockage des données en XML

Les données sont stockées dans un fichier XML;

Conception

Stockage des données en XML

Les données sont stockées dans un fichier XML; **puisque'il est lisible à la fois par la machine et l'humain.**

Conception

Stockage des données en XML

Les données sont stockées dans un fichier XML; **puisque'il est lisible à la fois par la machine et l'humain.**

- *Le root-tag est `<Employee>` et qui contient **0 ou plusieurs** tags de type `<employee>` qui représente des employées.*

Conception

Stockage des données en XML

Les données sont stockées dans un fichier XML; **puisque'il est lisible à la fois par la machine et l'humain.**

- Le root-tag est `<Employee>` et qui contient **0 ou plusieurs** tags de type `<employee>` qui représente des employées.
- Chaque tag `<employee>` contient **un seul** tag `<personal>` et **un seul** tag `<administrative>` qui peut contient **0 ou plusieurs** tags `<uplift>`.

Conception

Stockage des données en XML

Les données sont stockées dans un fichier XML; **puisque'il est lisible à la fois par la machine et l'humain.**

- Le root-tag est `<Employee>` et qui contient **0 ou plusieurs** tags de type `<employee>` qui représente des employées.
- Chaque tag `<employee>` contient **un seul** tag `<personal>` et **un seul** tag `<administrative>` qui peut contient **0 ou plusieurs** tags `<uplift>`.
- Le tag `<employee>` peut aussi avoir **0 ou plusieurs** tags de type `<diploma>`, `<medicalcertif>` et `<repayment>`.

Les Paquets et leurs Classes

Le code source de l'application est divisé en 4 paquets principales:

Les Paquets et leurs Classes

Le code source de l'application est divisé en 4 paquets principales:

`model` contient les différentes classe pour mobilisé les donnée en objet

Les Paquets et leurs Classes

Le code source de l'application est divisé en 4 paquets principales:

- `model` contient les différentes classe pour mobilisé les donnée en objet
- `app` contient les différentes énumération utilisées dans l'application.

Les Paquets et leurs Classes

Le code source de l'application est divisé en 4 paquets principales:

`model` contient les différentes classe pour mobilisé les donnée en objet

`app` contient les différentes énumération utilisées dans l'application. Ce paquet contient aussi `app.utils`, qui contient des utilitaires utiles pour le développement, notamment la gestion du **fichier XML**

Les Paquets et leurs Classes

Le code source de l'application est divisé en 4 paquets principales:

- `model` contient les différentes classe pour mobilisé les donnée en objet
- `app` contient les différentes énumération utilisées dans l'application. Ce paquet contient aussi `app.utils`, qui contient des utilitaires utiles pour le développement, notamment la gestion du **fichier XML**
- `wins` contient des interfaces graphiques, y compris celles qui sont responsables des opérations CRUD normales qui existent dans `wins.crud`

Les Paquets et leurs Classes

Le code source de l'application est divisé en 4 paquets principales:

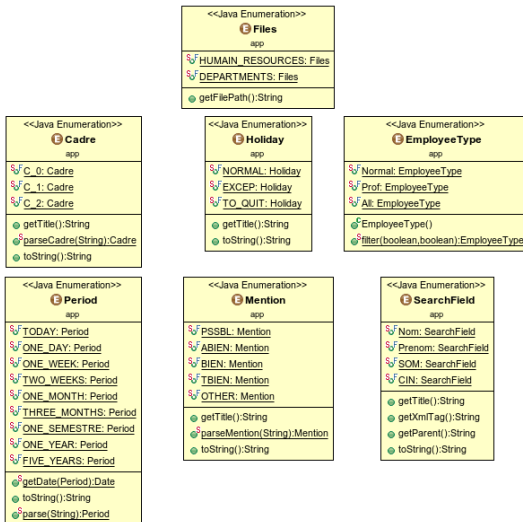
- `model` contient les différentes classe pour mobilisé les donnée en objet
- `app` contient les différentes énumération utilisées dans l'application. Ce paquet contient aussi `app.utilis`, qui contient des utilitaires utiles pour le développement, notamment la gestion du **fichier XML**
- `wins` contient des interfaces graphiques, y compris celles qui sont responsables des opérations CRUD normales qui existent dans `wins.crud`
- `views` contient des pages générées pour l'impression.

Paquet app

Ce paquet contient que les énumérations

Paquet app

Ce paquet contient que les énumérations

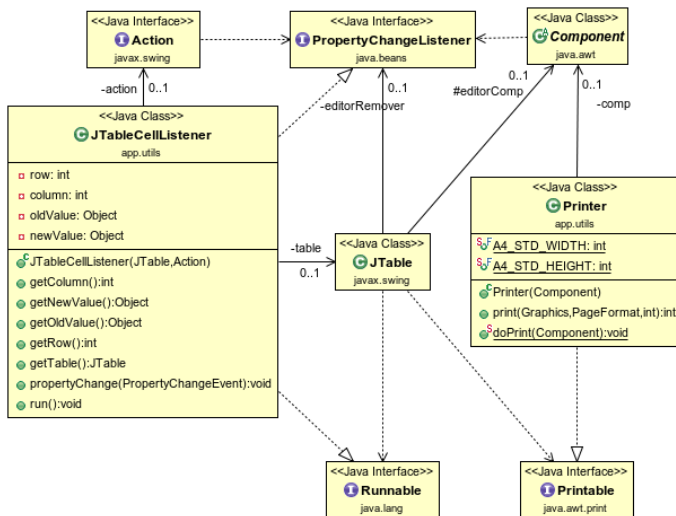


Paquet `app.utils`

Ce paquet contient des classes important pour l'application.

Paquet app.utils

Ce paquet contient des classes important pour l'application.



La classe Printer

La classe `Printer` est responsable de l'impression d'un `Component` (la classe des composants graphiques) avec l'aide de la méthode `static doPrint()` qui prend un `Component` comme paramètre.

La classe Printer

La classe `Printer` est responsable de l'impression d'un `Component` (la classe des composants graphiques) avec l'aide de la méthode statique `doPrint()` qui prend un `Component` comme paramètre.

Exemple de l'appel

```
import app.utils.Printer;
```

```
Printer.doPrint(/* le composant graphique */);
```

La classe Printer

La classe `Printer` est responsable de l'impression d'un `Component` (la classe des composants graphiques) avec l'aide de la méthode statique `doPrint()` qui prend un `Component` comme paramètre.

Exemple de l'appel

```
import app.utils.Printer;
```

```
Printer.doPrint(/* le composant graphique */);
```

Implementation de l'interface `java.awt.Printable`

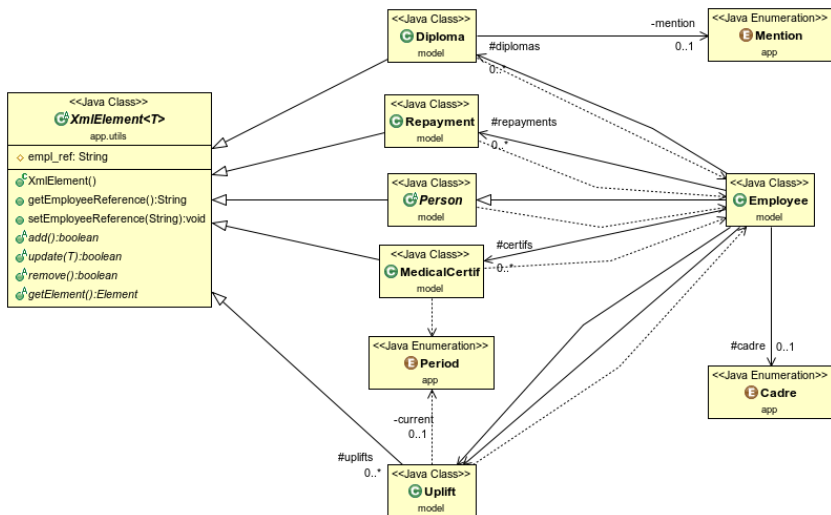
Aussi, `Printer` a une implémentation de la méthode abstraite `print()` de l'interface `Printable`

Exsrais de la classe

L'implementation du print()

```
1 public class Printer implements Printable {
2     private final Component comp;
3     public Printer(Component comp) {
4         this.comp = comp;
5     }
6     @Override
7     public int print(Graphics g, PageFormat format,
8                     int page_index) {
9         if (page_index > 0)
10             return Printable.NO_SUCH_PAGE;
11         /* ... */
12         return Printable.PAGE_EXISTS;
13     }
14 }
```


Paquet model



La classe XmlElement

```
1  public abstract class XmlElement<T> {
2      public abstract boolean add();
3      public abstract boolean update(T updated);
4      public abstract boolean remove();
5      public abstract Element getElement();
6
7      /* référence du employé */
8      protected String empl_ref;
9      public String getEmployeeReference( ) {
10         return empl_ref;
11     }
12
13     public void setEmployeeReference(String ref) {
14         this.empl_ref = ref;
15     }
16 }
```

Héritage de la classe XmlElement

La méthode `update()` prend un variable de type T ,

Héritage de la classe XmlElement

La méthode `update()` prend un variable de type `T`, ce type est décrit avec un héritage de la classe `XmlElement`

Héritage de la classe XmlElement

La méthode `update()` prend un variable de type `T`, ce type est décrit avec un héritage de la classe `XmlElement`

```
1  public class Diploma extends XmlElement<Diploma> {
2      /* les attributs du classe */
3      @Override
4      public boolean update(Diploma updated) {
5          try {
6              /* process la mise à jour */
7              return true;
8          } catch (Exception e) {
9              System.err.println(e.getMessage());
10             return false;
11         }
12     }
13 }
```

L'Interface Graphique

La relation entre les différents Paquets graphiques