

# Informatiser le service Ressources Humaines

Rchid Anas

Introduction

Cahier des Charges

Conception

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. . .

- ▶ *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

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

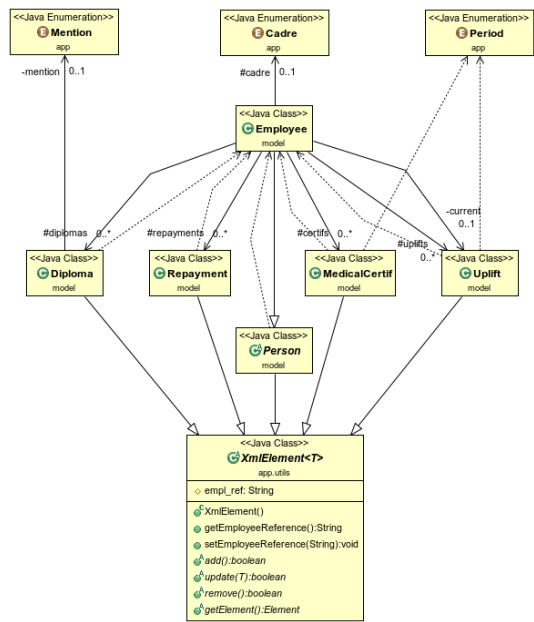
- ▶ Le root-tag est `<Employee>` et qui contient **0 ou plusieurs** tags de type `<employee>` qui représente des employés.
- ▶ Chaque tag `<employee>` contient **un seul** tag `<personal>` et **un seul** tag `<administrative>` qui peut contenir **0 ou plusieurs** tags `<uplift>`.
- ▶ Le tag `<employee>`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

- `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`
- `views` contient des pages générées pour l'impression.

# 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 }
```



## Hirite 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
4      @Override
5      public boolean update(Diploma updated) {
6          try {
7              /* process la mise à jour */
8              return true;
9          } catch (Exception e) {
10             System.err.println(e.getMessage());
11             return false;
12         }
13     }
14 }
```

# L'Interface Graphique

La relation entre les différents Paquets graphiques