

Ecole Polytechnique de l'Université François Rabelais de Tours

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

[polytech.univ-tours.fr](http://polytech.univ-tours.fr)

**Projet Recherche & Développement**

**2018-2019**

# **Affectation des projets au sein du département informatique de polytech Tours**



**POLYTECH<sup>®</sup>**  
TOURS

**Tuteur académique**  
Ronan BOCQUILLON

**Étudiant**  
Mariam KONATÉ (DI5)

2 avril 2019



# Liste des intervenants

| Nom              | Email  | Qualité  |
|------------------|--|--|
| Mariam KONATÉ    | <a href="mailto:mariam.konate-20etu.univ-tours.fr">mariam.konate-20etu.univ-tours.fr</a> | Étudiant DI5                                   |
| Ronan BOCQUILLON | <a href="mailto:ronan.bocquillon@univ-tours.fr">ronan.bocquillon@univ-tours.fr</a>       | Tuteur académique,<br>Département Informatique |



# Avertissement

Ce document a été rédigé par Mariam Konaté surnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Ronan Bocquillon surnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



## Pour citer ce document

Mariam Konaté, *Affectation des projets au sein du département informatique de polytech Tours*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2018-2019.

```
@mastersthesis{
  author={Konaté, Mariam},
  title={Affectation des projets au sein du département informatique de polytech Tours},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2018-2019}
}
```

# Table des matières

|   |          |
|---|----------|
| Liste des intervenants                    | a        |
| Avertissement                             | b        |
| Pour citer ce document                    | c        |
| Table des matières                        | i        |
| Table des figures                         | iv       |
| <b>1 Introduction</b>                     | <b>1</b> |
| 1 Acteurs, enjeux et contexte .....       | 1        |
| 2 Objectifs .....                         | 1        |
| 3 Hypothèses .....                        | 1        |
| 4 Bases méthodologiques.....              | 2        |
| <b>2 Description générale</b>             | <b>3</b> |
| 1 Environnement du projet .....           | 3        |
| 2 Caractéristiques des utilisateurs ..... | 3        |
| 3 Fonctionnalités du système .....        | 4        |
| 4 Structure générale du système.....      | 7        |
| <b>3 Etat de l'art/veille</b>             | <b>8</b> |
| 1 Introduction .....                      | 8        |
| 2 Modélisation .....                      | 8        |
| 3 Exemple de problème.....                | 10       |
| 4 Méthodes de résolution .....            | 10       |
| 4.1 Exemple méthode hongroise.....        | 10       |

|          |  |           |
|----------|--|-----------|
| 5        | Veille technologique .....   | 12        |
| <b>4</b> | <b>Analyse et conception</b>   | <b>14</b> |
| 1        | Analyse .....  | 14        |
| 2        | Conception .....   | 15        |
| 2.1      | LDAP .....   | 16        |
| 2.2      | Fichier csv d'import .....   | 17        |
| <b>5</b> | <b>Mise en oeuvre</b>  | <b>18</b> |
| 1        | Base de données .....  | 18        |
| 2        | Framework Spring MVC .....   | 19        |
| 3        | Optimisation et notification .....   | 20        |
| 4        | Qualité .....  | 20        |
| <b>6</b> | <b>Bilan et conclusion</b>   | <b>21</b> |
| 1        | Fait .....   | 21        |
| 2        | Reste à faire .....  | 21        |
| 3        | Planning .....   | 22        |
| 4        | Bilan sur la qualité .....   | 22        |
| 5        | Bilan auto critique sur la gestion de projet .....                             | 22        |
| 6        | Améliorations .....  | 22        |
| 7        | Conclusion .....   | 22        |
|          | <b>Annexes</b>   | <b>23</b> |
| <b>A</b> | <b>Spécifications fonctionnelles</b>   | <b>24</b> |
| 1        | Partie optimisation .....  | 24        |
| 1.1      | Fonction 1 : Importation fichier csv contenant les résultats du formulaire ... | 24        |
| 1.2      | Fonction 2 : Calculer la solution du projet .....                              | 24        |
| 2        | Partie interface .....   | 25        |
| 2.1      | Fonction 1 : Affichage de l'historique des sujets .....                        | 25        |
| 2.2      | Fonction 2 : Authentification .....  | 25        |
| 2.3      | Fonction 3 : Création d'enseignement .....                                     | 25        |
| 2.4      | Fonction 4 : Ajouter un sujet .....  | 25        |
| 2.5      | Fonction 5 : Supprimer un sujet .....  | 26        |
| 2.6      | Fonction 6 : Valider l'affectation .....                                       | 26        |
| 2.7      | Fonction 7 : Créer un formulaire pour l'enseignement .....                     | 26        |
| 2.8      | Fonction 8 : Affichage solution projet .....                                   | 26        |
| 2.9      | Fonction 9 : Envoyer notification aux étudiants .....                          | 26        |
| 2.10     | Fonction 10 : Répondre au formulaire .....                                     | 27        |

|          |  |           |
|----------|--|-----------|
| <b>B</b> | <b>Spécifications non fonctionnelles et interfaces</b> | <b>28</b> |
| 1        | Contraintes de développement et conception .....       | 28        |
| 2        | Contraintes de fonctionnement et d'exploitation.....   | 28        |
| 3        | Performances .....                                     | 28        |
| 4        | Capacités.....   | 29        |
| 5        | Mode de fonctionnement .....                           | 29        |
| 6        | Contrôlabilité .....                                   | 29        |
| 7        | Sécurité .....   | 29        |
| 8        | Interfaces.....  | 29        |
| 8.1      | Interfaces matériel/logiciel .....                     | 30        |
| 8.2      | Interfaces homme/machine .....                         | 30        |
| 8.2.1    | Interface Etudiant .....                               | 31        |
| 8.2.2    | Interface Enseignants.....                             | 32        |
| <b>C</b> | <b>Analyse et cahier développeur</b>                   | <b>35</b> |
| 0.1      | Interface logiciel/logiciel .....                      | 35        |
| 1        | Diagramme de classes .....                             | 35        |
| 2        | Description détaillées des classes .....               | 36        |
| 3        | Structure des fichiers utilisés.....                   | 36        |
| <b>D</b> | <b>Gestion de projet</b>                               | <b>38</b> |
| 1        | Découpage des tâches.....                              | 38        |
| 1.1      | Sprint 1 .....   | 38        |
| 1.2      | Sprint 2 .....   | 38        |
| 1.3      | Sprint 3 .....   | 39        |
| 1.4      | Sprint 4 .....   | 39        |
| 1.5      | Sprint 5 .....   | 39        |
| 1.6      | Sprint 6 .....   | 39        |
| 1.7      | Sprint 7 .....   | 39        |
| <b>E</b> | <b>Documentation d'installation et de maintenance</b>  | <b>40</b> |
| 1        | Installation .....                                     | 40        |
| 2        | Maintenance .....                                      | 40        |
| <b>F</b> | <b>Documentation d'utilisation</b>                     | <b>42</b> |
| <b>G</b> | <b>Cahier de tests</b>                                 | <b>50</b> |

# Table des figures

## 2 Description générale

|   |   |   |
|---|---|---|
| 1 | Caractéristiques des utilisateurs .....       | 4 |
| 2 | Diagramme de cas d'utilisations .....         | 5 |
| 3 | Diagramme de séquences tout utilisateur ..... | 6 |
| 4 | Diagramme de séquences enseignant .....       | 6 |
| 5 | Diagramme de séquences étudiant .....         | 7 |
| 6 | Arborescence du site .....                    | 7 |

## 3 Etat de l'art/veille

|   |   |   |
|---|---|---|
| 1 | Représentation du problème sous forme de graphe ..... | 9 |
|---|---|---|

## 4 Analyse et conception

|   |                           |    |
|---|---------------------------|----|
| 1 | Diagramme de classe ..... | 16 |
|---|---------------------------|----|

## 5 Mise en oeuvre

|   |                            |    |
|---|----------------------------|----|
| 1 | UML Base de données .....  | 18 |
| 2 | FrameWork Spring MVC ..... | 19 |

## B Spécifications non fonctionnelles et interfaces

|   |                                      |    |
|---|--------------------------------------|----|
| 1 | Diagramme de cas d'utilisation ..... | 29 |
| 2 | Interface homme machine .....        | 30 |
| 3 | Authentification étudiant .....      | 31 |
| 4 | Voeux étudiants .....                | 31 |
| 5 | Authentification enseignant .....    | 32 |



|  |  |    |
|--|--|----|
| 6  | Ajout Sujet.....                                 | 32 |
| 7  | Liste des souhaits .....                         | 33 |
| 8  | Optimisation.....                                | 34 |
| <br><b>C Analyse et cahier développeur</b> |  |    |
| 1  | Diagramme de classe.....                         | 36 |
| <br><b>D Gestion de projet</b>             |  |    |
| 1  | diagramme de Gantt.....                          | 38 |
| <br><b>F Documentation d'utilisation</b>   |  |    |
| 1  | Recherche de sujets.....                         | 42 |
| 2  | Authentification étudiant .....                  | 43 |
| 3  | Consultation étudiant .....                      | 43 |
| 4  | Clé enseignement .....                           | 44 |
| 5  | Voeux étudiant .....                             | 45 |
| 6  | Authentification enseignant.....                 | 45 |
| 7  | Recherche enseignement.....                      | 46 |
| 8  | ouvrir enseignement .....                        | 46 |
| 9  | ouvrir enseignement si on est pas référent ..... | 47 |
| 10   | Liste Souhaits.....                              | 48 |
| 11   | Lancer l'optimisation .....                      | 48 |
| 12   | Tables de la base de données.....                | 49 |
| <br><b>G Cahier de tests</b>               |  |    |
| 1  | .....  | 51 |
| 2  | .....  | 52 |
| 3  | .....  | 53 |
| 4  | .....  | 54 |
| 5  | .....  | 55 |

# 1

## Introduction

Dans ce chapitre, on décrit le contexte du projet et la structure du rapport ainsi que les acteurs, les objectifs, les hypothèses, et les bases méthodologiques sur la gestion de projet.

### 1 Acteurs, enjeux et contexte

Ce document présente la conception d'un outil permettant de gérer les affectations aux différents projets que les étudiants en informatique à Polytech Tours effectuent au cours de leurs cursus.

A Polytech les affectations des projets ne se font pas de manière automatique, cela prend du temps aux encadrants, de plus il arrive que les étudiants soient mécontents du projet qu'ils vont devoir faire.

Afin de résoudre ce problème, ce projet a été mis en place par Ronan Bocquillon, enseignant chercheur à polytech Tours. En effet cet outil sera utilisé par les enseignants et les étudiants du département informatique de Polytech Tours puis sera éventuellement ouvert aux autres départements afin d'optimiser les affectations de projet.

### 2 Objectifs

L'objectif du projet est de créer une application web, interne à Polytech. Elle sera alimentée par la création de projet par des encadrants, de formulaires affectés à ce projet et d'optimisation des résultats. D'un autre côté les étudiants eux vont répondre aux questions du formulaire et vont recevoir une notification (mail) des résultats de l'affectation. La création et la gestion de projet nécessitera une authentification, ainsi que le remplissage des formulaires. Une partie historique de projets sera disponible pour les étudiants et les encadrants. Elle pourra donc :

- Automatiser le processus d'affectation des projets.
- Satisfaire au mieux les étudiants dans leurs choix.

### 3 Hypothèses

Afin de gagner de l'expérience en développement web Java Enterprise Edition, je souhaite utiliser le java comme langage pour ce projet. Si Coin-Or ou Or tools ne possèdent pas de librairie

pouvant au mieux s'adapter à mon projet, je le ferai en python car il existe déjà des bibliothèques qui sont adaptées dans ce langage là.

## 4 Bases méthodologiques

— Les outils

- Pour la gestion de projet, on utilise le diagramme de Gantt pour contrôler le déroulement du projet et définir le temps estimé pour chaque tâche. On utilise la plateforme « Trello » pour visualiser le travail à effectuer et ce qui nous reste à faire afin de gérer de manière plus précise le travail à faire.
- Pour la modélisation logicielle, on utilise les diagrammes d'UML notamment diagramme de cas d'utilisation et le diagramme de séquences. Pour la partie web nous utilisons une arborescence du site et un impact mapping.
- Pour la rédaction de rapport, nous utilisons « LaTeX » en version en ligne overleaf.

— La méthode de gestion de projet : On utilise la méthode agile scrum donc 7 listes sur l'outil trello je me suis inspirée de ce site [Agile-Scrum](#) pour mettre en place mon tableau :

- Ressources : dans cette liste, je place toutes les tâches récurrentes.
- Backlog : c'est mon répertoire des toutes les tâches.
- To do(Sprint n° X) : quand je planifie mon sprint, je déplace les tâches du backlog vers cette liste.
- En cours : quand je commence à travailler sur une tâche je la déplace dans cette liste.
- Contrôle de Qualité : lorsque les tâches sont complétées, elles sont déplacées dans la liste "CQ". A la fin de la semaine, je vérifie chaque élément pour m'assurer que tout est nickel.
- Fait : une fois le contrôle de qualité réussi, c'est je les déplace dans cette liste ! A partir de là, il n'y a plus de modifications.
- Bloqué : j'utilise cette liste lorsque la finalisation d'une tâche dépend d'un facteur externe (par exemple, si j'ai besoin de voir un professeur pour avancer et que j'ai pas de rdv), en spécifiant les raisons du blocage dans un commentaire

La planification détaillée dans le chapitre Gestion de projet à la page 38.

# 2

## Description générale

Dans ce chapitre, on décrit l'environnement du projet, les caractéristiques des utilisateurs, les fonctionnalités du système, et la structure générale du système.

### 1 Environnement du projet

L'environnement de développement consiste aux points suivants :

- Le système d'exploitation : Windows
- Le langage de programmation : Java 8
- L'IDE pour le développement : Eclipse

### 2 Caractéristiques des utilisateurs

Il y a trois types d'utilisateur pour ce logiciel : les étudiants, les enseignants et l'administrateur.

|                        | Droits                                  | Actions  |
|------------------------|---|--|
| <b>Étudiants</b>       | Restreints aux vœux                     | -consulter des sujets et émettre des vœux  |
| <b>Enseignants</b>     | Tous les droits sur leurs enseignements | -consulter des sujets, modifier leur enseignement, ouvrir des formulaires, faire l'optimisation, Notifier. |
| <b>Administrateurs</b> | Droits sur tous les enseignements       | Tout.  |

Figure 1 – Caractéristiques des utilisateurs

### 3 Fonctionnalités du système

Nous avons trois utilisateurs : les étudiants, les enseignants et l'administrateur. Nous allons détailler les fonctionnalités pour chaque utilisateur :

Pour les enseignants :

1. Consulter l'historique des sujets.
2. Insérer les sujets, valider et ouvrir le formulaire aux étudiants.
3. Interface pour fermer les choix et entamer la phase d'optimisation.
4. Dans l'interface d'optimisation pouvoir rajouter des options (contraintes sur les choix) : 1 étudiant doit avoir les vœux 1,2,3 ou 4).
5. Création de formulaire : Pouvoir rajouter des contraintes (nb de personnes pouvant choisir un même projet etc..).
6. Notification aux étudiants par mail.

Pour l'administrateur :

1. Consulter l'historique des sujets.
2. Créer de nouveaux enseignements.

3. Modifier des enseignements.
4. Ajout et suppression sur toutes les tables de la base de données.

Pour les étudiants :

1. Insérer les choix dans un formulaire.
2. Recevoir les résultats par mail.
3. Consulter l'historique des sujets.

Dans ce diagramme de cas d'utilisations nous montrons les différentes actions que peuvent faire les utilisateurs et les réponses renvoyées par le système :

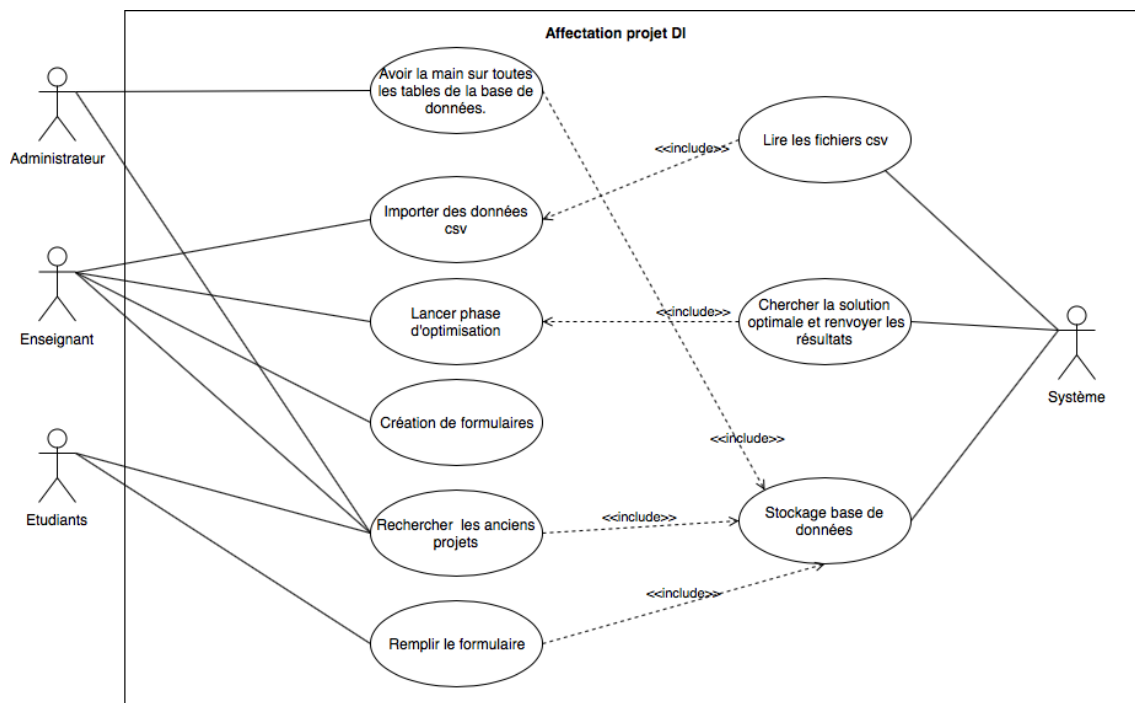
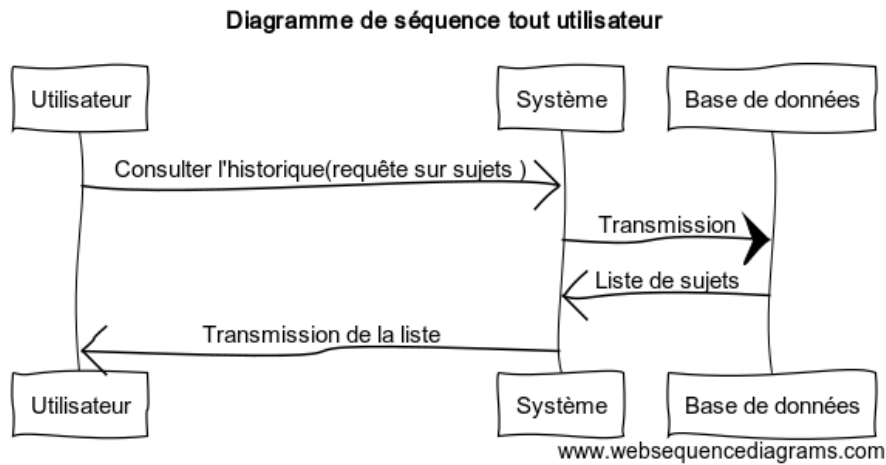


Figure 2 – Diagramme de cas d'utilisations

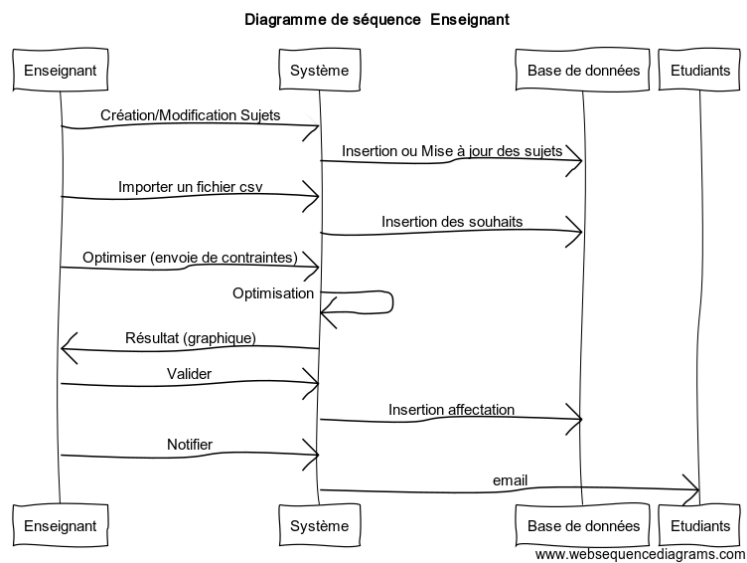
De plus nous avons représenté les interactions entre le système et les utilisateurs dans ces diagrammes de séquence :

— Tout utilisateur :



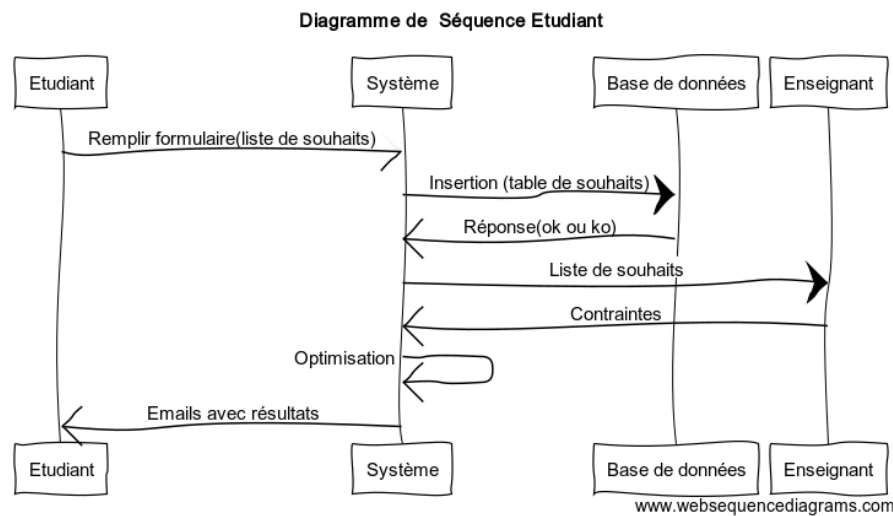
**Figure 3** – *Diagramme de séquences tout utilisateur*

— Enseignant :



**Figure 4** – *Diagramme de séquences enseignant*

— Etudiant :



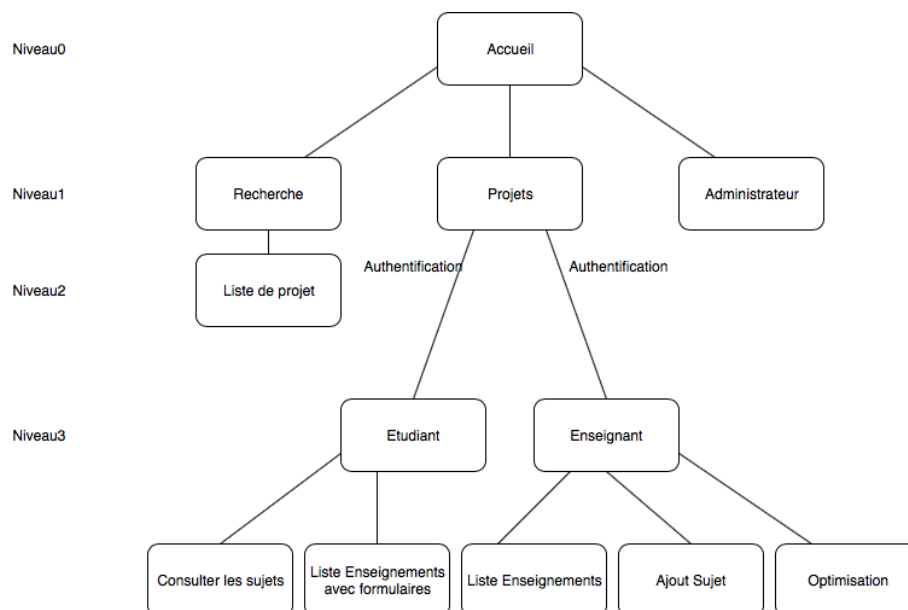
**Figure 5** – Diagramme de séquences étudiant

Pour plus de détails sur les fonctionnalités vous pouvez consulter les spécifications fonctionnelles à la page 24.

## 4 Structure générale du système

En analysant le projet nous avons identifié deux parties principales : une partie développement web et une partie d’optimisation.

— Partie web : Cette partie consiste à développer une application web qui peut gérer toutes les actions prévues dans le diagramme de cas d’utilisations à la page 4. Cette application est mieux expliquée à travers cette arborescence :



**Figure 6** – Arborescence du site

— Partie Optimisation : Cette partie permet au système à partir des données stockées dans la base de données, de calculer une solution optimale du problème d’affectation. Ce calcul va être affecté par les différentes contraintes posées par l’utilisateur(enseignant).



# 3

## Etat de l'art/veille

### 1 Introduction

Le problème d'affectation est un problème classique de recherche opérationnelle. L'objectif est de déterminer un couplage maximum dans un graphe biparti valué. Le problème d'affectation peut être résolu en temps polynomial par l'algorithme hongrois, il appartient par conséquent à la classe de complexité P.

### 2 Modélisation

Le "problème d'affectation" consiste à établir des liens entre les éléments de deux ensembles distincts, de façon à minimiser un coût et en respectant des contraintes d'unicité de lien pour chaque élément. On considère  $m$  tâches et  $n$  agents, avec  $n \geq m$ . Pour tout couple  $(i,j)$  ( $i$  allant de 1 à  $m$ ,  $j$  allant de 1 à  $n$ ), l'affectation de la tâche  $i$  à  $j$  entraîne un coût de réalisation noté  $C_{ij}$  ( $C_{ij} \geq 0$ ). Chaque tâche doit être réalisée exactement une fois et chaque agent peut réaliser au plus une tâche. Le problème consiste à affecter les tâches aux agents, de façon à minimiser le coût total de réalisation et en respectant les contraintes de réalisation des tâches et de disponibilité des agents [Nemhauser et Wolsey 88]. À tout couple tâche/agent  $(i,j)$ , on associe une variable d'affectation  $X_{ij}$  où :

$$X_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ est affectée à l'agent } j \\ 0 & \text{sinon} \end{cases}$$

Le problème d'affectation peut être représenté sous forme d'un graphe biparti de la manière suivante :

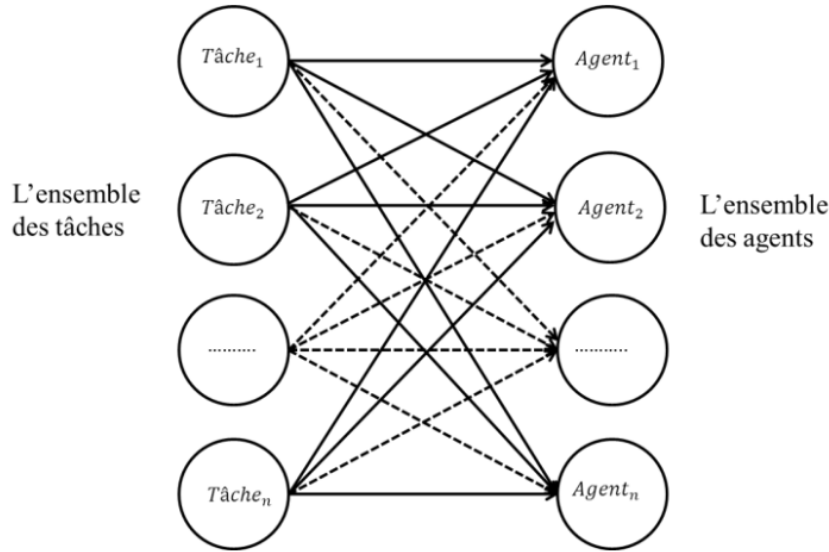


Figure 1 – Représentation du problème sous forme de graphe

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n X_{ij} C_{ij}$$

$$\text{Sc} = \begin{cases} \sum_{j=1}^n X_{ij} = 1 \quad i=1..n \\ \sum_{i=1}^n X_{ij} = 1 \quad j=1..n \\ X_{ij} \in \{0, 1\} \quad i=1..n \text{ et } j=1..n \end{cases}$$

Variables :

$C_{ij}$  : Matrice des coûts d'affectation.

Contraintes :

- La première contrainte donne le nombre d'agents réalisant la tâche  $i$  qui vaut 1 pour chaque  $i = 1..n$ .
- La deuxième contrainte donne le nombre de tâches réalisées par l'agent  $j$  qui vaut 1 pour chaque  $j = 1..n$ .

Les contraintes de ce problème se retrouvent dans de nombreuses applications mettant en jeu des problèmes d'allocation de ressources. Elles sont généralement appelées "contraintes d'affectation".

But :

Le problème consiste à affecter les tâches aux agents, de façon à minimiser le coût total de réalisation :  $\sum_{i=1}^n \sum_{j=1}^n X_{ij} C_{ij}$  et en respectant les contraintes de réalisation des tâches et de disponibilité des agents.

Remarques :

- i. Le problème d'affectation est dit non standard, si on a  $m$  agents et  $n$  tâches avec  $m \neq n$ .

Mais on peut transformer ce problème non standard à un problème standard de la manière suivante :

- Si  $m > n$  alors nous créons  $m-n$  agents fictifs.
- Si  $n > m$  alors nous créons  $n-m$  tâches fictives.

Le coût d'affectation de ces éléments fictifs est égale à  $M$  (ou  $M > 0$ ).

- ii. Le problème standard d'affectation peut être résolu comme un problème de transport ou les  $a_i = b_j = 1$  et les  $C_{ij}$  représentent les coûts unitaires de transport.
- iii. Le problème d'affectation peut être défini comme un problème de maximisation.

### 3 Exemple de problème

Supposons que, dans une entreprise,  $n$  agents puissent travailler indifféremment sur  $n$  tâches, mais avec plus ou moins d'efficacité; l'efficacité peut se mesurer par le revenu provenant de la vente des produits fabriqués par les divers ouvriers travaillant sur les différentes machines. Répartir les agents sur les machines de la façon la plus efficace est un problème d'affectation. Soit  $n$  unités à affecter à  $n$  tâches; une unité ne peut être affectée qu'à une tâche et une tâche ne peut employer qu'une unité. L'unité  $i$  exécute la tâche  $j$  avec un coût ou un profit  $c_{ij}$ . Comment doit-on affecter chaque unité à une seule tâche pour que la rentabilité soit optimale ?

### 4 Méthodes de résolution

- Adaptation de la méthode du simplexe :  
Les propriétés de la matrice des contraintes nous permettent d'utiliser le simplexe pour résoudre ce problème de programmation linéaire en nombres entiers. Inconvénients :
  - Cela conduit à des tableaux très grands (d'ordre  $n$ ).
  - La solution est fortement dégénérée, i.e. elle contient un nombre de variables positives inférieures à  $2n - 1$ .
  - La méthode du simplexe peut s'appliquer mais certaines itérations n'ont aucune incidence sur la solution réalisable courante. Seul la base change : une variable hors base est échangée avec une variable de base dégénérée.
- Enumération de toutes les solutions possibles :  
 $n=5 \rightarrow 120$  solutions réalisables.  
 $n=20 \rightarrow$  Une calculatrice analysant une affectation par microseconde travaillant 8 heures par jour et 365 jours par année, exigerait 2500 siècles pour ce travail.
- Méthode Hongroise : L'algorithme hongrois (Kuhn, 1955) permet de résoudre le problème d'affectation. Il repose sur les propriétés énoncées précédemment, et fonctionne de la façon suivante :
  1. Phase 1 : Obtention des zéros (propriété 1) : On retranche de chaque ligne l'élément minimum puis, dans la nouvelle matrice, de chaque colonne l'élément minimum.
  2. Phase 2 : On recherche un ensemble de cardinalité maximale de zéros indépendants (ce qui revient à chercher un couplage maximal dans un graphe biparti). Si l'affectation est complète, on a une solution optimale. Sinon on passe à la phase 3.
  3. Phase 3 : On cherche le nombre minimal de lignes et de colonnes contenant tous les zéros (d'après la propriété 3, il est égal à la cardinalité de l'ensemble calculé à la phase 2). Pour cela, il faut suivre ces étapes :
    - Marquer les lignes ne contenant pas de 0 encadrés (les 0 encadrés sont ceux obtenus à la phase 2).
    - Marquer les colonnes ayant un 0 sur une ou plusieurs lignes marquées
    - Marquer toute ligne ayant un 0 encadré dans une colonne marquée
    - Répéter b. et c. jusqu'à ce que plus aucune ligne ni colonne ne puisse être marquée
    - Rayer les lignes non marquées et les colonnes marquées.
  4. Phase 4 : Ajout ou déplacement de zéros (propriété 2) : On retranche le plus petit nombre non rayé (forcément strictement positif) aux éléments non rayés et on l'ajoute aux éléments rayés 2 fois, puis on retourne à la phase 2.

#### 4.1 Exemple méthode hongroise

Nous avons un tableau de départ contenant les coûts d'affectation des agents aux tâches :

|    |    |    |    |    |
|----|----|----|----|----|
| 17 | 15 | 9  | 5  | 12 |
| 16 | 16 | 10 | 5  | 10 |
| 12 | 15 | 14 | 11 | 5  |
| 4  | 8  | 14 | 17 | 13 |
| 13 | 9  | 8  | 12 | 17 |

Nous allons appliquer la méthode hongroise étape par étape :

#### 1. REDUCTION DU TABLEAU INITIAL :

On soustrait à chaque ligne du tableau initial, le plus petit élément de la ligne :

|    |    |    |    |   |
|----|----|----|----|---|
| 12 | 10 | 4  | 0  | 7 |
| 11 | 11 | 5  | 0  | 5 |
| 7  | 10 | 9  | 6  | 0 |
| 0  | 4  | 10 | 13 | 9 |
| 5  | 1  | 0  | 4  | 9 |

On fait de même avec les colonnes :

|    |    |    |    |   |
|----|----|----|----|---|
| 12 | 9  | 4  | 0  | 7 |
| 11 | 10 | 5  | 0  | 5 |
| 7  | 9  | 9  | 6  | 0 |
| 0  | 3  | 10 | 13 | 9 |
| 5  | 0  | 0  | 4  | 9 |

#### 2. ENCADRER ET BARRER DES ZEROS :

On cherche ligne comportant le moins de zéros non barrés (en cas d'égalité, choisir arbitrairement la plus haute). On encadre un des zéros de cette ligne (arbitrairement le plus à gauche). On barre tous les zéros se trouvant sur la même ligne ou sur la même colonne que le zéro encadré. On recommence l'opération jusqu'à ce qu'on ne puisse plus encadrer ni barrer des zéros :

|          |          |    |          |          |
|----------|----------|----|----------|----------|
| 12       | 9        | 4  | <b>0</b> | 7        |
| 11       | 10       | 5  | ∅        | 5        |
| 7        | 9        | 9  | 6        | <b>0</b> |
| <b>0</b> | 3        | 10 | 13       | 9        |
| 5        | <b>0</b> | ∅  | 4        | 9        |

#### 3. MARQUER ET BARRER DES LIGNES ET DES COLONNES :

- On marque d'une croix toutes les lignes ne contenant aucun zéro encadré
- On marque toute colonne ayant un zéro barré sur une ligne marquée
- On marque toute ligne ayant un zéro encadré dans une colonne marquée

On répète alternativement les opérations b) et c) jusqu'à ne plus pouvoir marquer de rangée.

On trace alors un trait sur toute ligne non marquée et sur toute colonne marquée.

|    |    |    |    |   |    |
|----|----|----|----|---|----|
|    |    |    | X2 |   |    |
| 12 | 9  | 4  | 0  | 7 | X3 |
| 11 | 10 | 5  | ∅  | 5 | X1 |
| 7  | 9  | 9  | 6  | 0 |    |
| 0  | 3  | 10 | 13 | 9 |    |
| 5  | 0  | 0  | 4  | 9 |    |

#### 4. MODIFICATION DU TABLEAU :

Les cases non traversées par un trait constituent un tableau partiel. On retranche à toutes les cases de ce tableau partiel le plus petit élément de celui-ci. On ajoute ce même élément à toutes les cases du tableau initial barrées deux fois. On obtient alors un nouveau tableau sur lequel on pourra répéter la succession des étapes 1 à 3.

|   |   |    |    |   |
|---|---|----|----|---|
| 8 | 5 | 0  | 0  | 3 |
| 7 | 6 | 1  | 0  | 1 |
| 7 | 9 | 9  | 10 | 0 |
| 0 | 3 | 10 | 17 | 9 |
| 5 | 0 | 0  | 8  | 9 |

Ainsi, dans l'exemple, la valeur de l'affectation minimale est  $9+5+5+4+9=32$

## 5 Veille technologique

Afin de résoudre le problème d'affectation de manière automatique nous allons utiliser un solveur. Afin de faire un choix nous avons décidé de faire une comparaison des solveurs gratuits et opensource.

- GLPK : (GNU Linear Programming Kit) logiciel écrit en ANSI C qui permet de résoudre des problèmes de programmation mixte en nombre entier.
- LP\_SOLVE : Egalement un solveur écrit en ANSI C et qui peut être utilisé pour résoudre des problèmes de programmation mixte en nombre entier.
- CLP : C'est un solveur créé par le projet Coin-OR écrit en C++ pour résoudre les problèmes d'optimisation linéaires.

Tous ces solveurs peuvent être utilisés sans restrictions dans tous logiciels et il existe des API pour échanger des données entre logiciels.

Hans Mittelmann ([source](#)) a produit une comparaison détaillée des différents solveurs de programmation mixte d'entiers. Il a utilisé des benchmark pour comparer leurs performance. Les problèmes étaient catégorisées de cette manière :

- toutes les variables sont binaires
- toutes les variables sont entières
- toutes les variables sont binaires ou continues
- toutes les variables sont entières ou continues

Ces données sont basées sur une recherche faite à partir du 9 janvier 2012. Les solveurs ont été testé sur un PC avec un processeur Intel Xeon X5680 avec 2\*6 cores, 32 gigabytes de RAM sur un linux 64bit. Les versions des solveurs utilisés :

- GLPK (4.47)
- LP\_SOLVE(5.5.2)
- SCIP-L(2.1.1 en utilisant CLP et LPSolver)

Le tableau ci-dessous montre les résultats principaux :

| solver   | running time | intances solved | solved(%) |
|----------|--------------|-----------------|-----------|
| GLPK     | 22.11        | 3               | 3.45      |
| LP_SOLVE | 19.40        | 5               | 5.75      |
| SCIP-L   | 6.40         | 52              | 59.77     |

Dans ce tableau la première colonne montre le solveur utilisé, la seconde montre le running time, le running time était limité à 60 secondes. La troisième colonne montre le nombre de problème qui ont été résolu avec succès pour chaque solveur et dans la dernière colonne nous avons le pourcentage correspondant.

Ainsi lorsque nous classons les solveurs en terme de performance nous avons SCIP-L en premier puis LP\_SOLVE et enfin GLPK. Scip-l utilisant CLP nous allons partir sur le solveur du projet coin-OR. Il y a des études qui ont montré que CLP fonctionnait bien avec le Java qui est le langage que l'on a choisit : [Clp-java](#).

# 4

## Analyse et conception

Dans notre cas nous avons plusieurs problèmes d'affectations à gérer car chaque projet possède ses propres contraintes. Il faudra donc une application capable de résoudre tous ces problèmes. Afin d'avoir une vue globale des projets nous avons interrogé les encadrants.

### 1 Analyse

Dans notre cas le problème d'affectation se déroulera ainsi : les agents seront les étudiants et les tâches à effectuer seront les projets. Pour chaque projet nous allons parler du déroulement et des contraintes.

- Projet parcours des écoles d'ingénieurs polytech :
  - Déroulement : Projets par binômes
  - Contraintes : 1 sujet par binôme, si un étudiant n'a pas de binôme il faut lui affecter un binôme. Chaque enseignant a un nombre max et min de sujets qu'il peut prendre. Un sujet peut avoir plusieurs copies.
  - Souhaits : classements de vœux billes.

Il s'agit du même problème d'affectation, il faut juste gérer l'affectation à un projet et à un binôme.
- Projet génie logiciel :
  - Déroulement : Projets par binômes
  - Contraintes : Plusieurs binômes peuvent avoir le même sujet.
  - Souhaits : Plusieurs types de classements de vœux (Billes ou ordre)

Il s'agit du même problème d'affectation, il faut juste gérer l'affectation à un projet et à un binôme.
- Projet de recherche et développement
  - Déroulement : Projet seul
  - Contraintes : 1 sujet par personne
  - Souhaits : Plusieurs types de classements de vœux (Billes ou ordre)

Ceci est un problème classique nous avons juste à faire remplir un formulaire par personne.
- Projet collectif
  - Déroulement : Projets en groupe
  - Contraintes : 1 sujet par groupe
  - Souhaits : Plusieurs types de classements de vœux (Billes ou ordre)

Ceci est un problème classique nous avons juste à faire remplir un formulaire par groupe, la création de groupe est gérée par l'encadrant.

— Projet tutoré 2

- Déroulement : Projets par binômes
- Contraintes : 3 sujets pour la promo, équilibrer les groupes
- Souhaits : Plusieurs types de classements de voeux(Billes ou ordre)

Il s'agit du même problème d'affectation, il faut juste gérer l'affectation à un projet et à un binôme.

— Projet systèmes d'information

- Déroulement : Projets par groupe
- Contraintes : 1 sujet par groupe
- Souhaits : Plusieurs types de classements de voeux(Billes ou ordre)

Ceci est un problème classique nous avons juste à faire remplir un formulaire par groupe.

— Projet architecture systèmes réseaux

- Déroulement : Projets par groupe
- Contraintes : 1 sujet par groupe
- Souhaits : Plusieurs types de classements de voeux(Billes ou ordre)

Ceci est un problème classique nous avons juste à faire remplir un formulaire par groupe.

— Options S10

- Déroulement : 3 options par personne parmi 9
- Contraintes : options équilibrées groupes d'option (3 options par groupe).
- Souhaits : Plusieurs types de classements de voeux(Billes ou ordre)

Il s'agit d'un problème d'affectation qui permet de créer des groupes d'options à partir des voeux (ACP) puis d'affecter chaque étudiant à un groupe d'option.

— Choix Spécialité

- Déroulement : Choix par personne
- Contraintes : 2 spécialité au choix, les spécialité doivent être équilibrées.
- Souhaits : Plusieurs types de classements de voeux(Billes ou ordre)

Il s'agit du même problème d'affectation, il faut juste gérer l'affectation d'un étudiant à une spécialité de son choix.

## 2 Conception

Nous allons donc faire une modélisation de projet pouvant permettre de résoudre les contraintes de ces différents projets. Ainsi nous avons le diagramme de classe suivant :



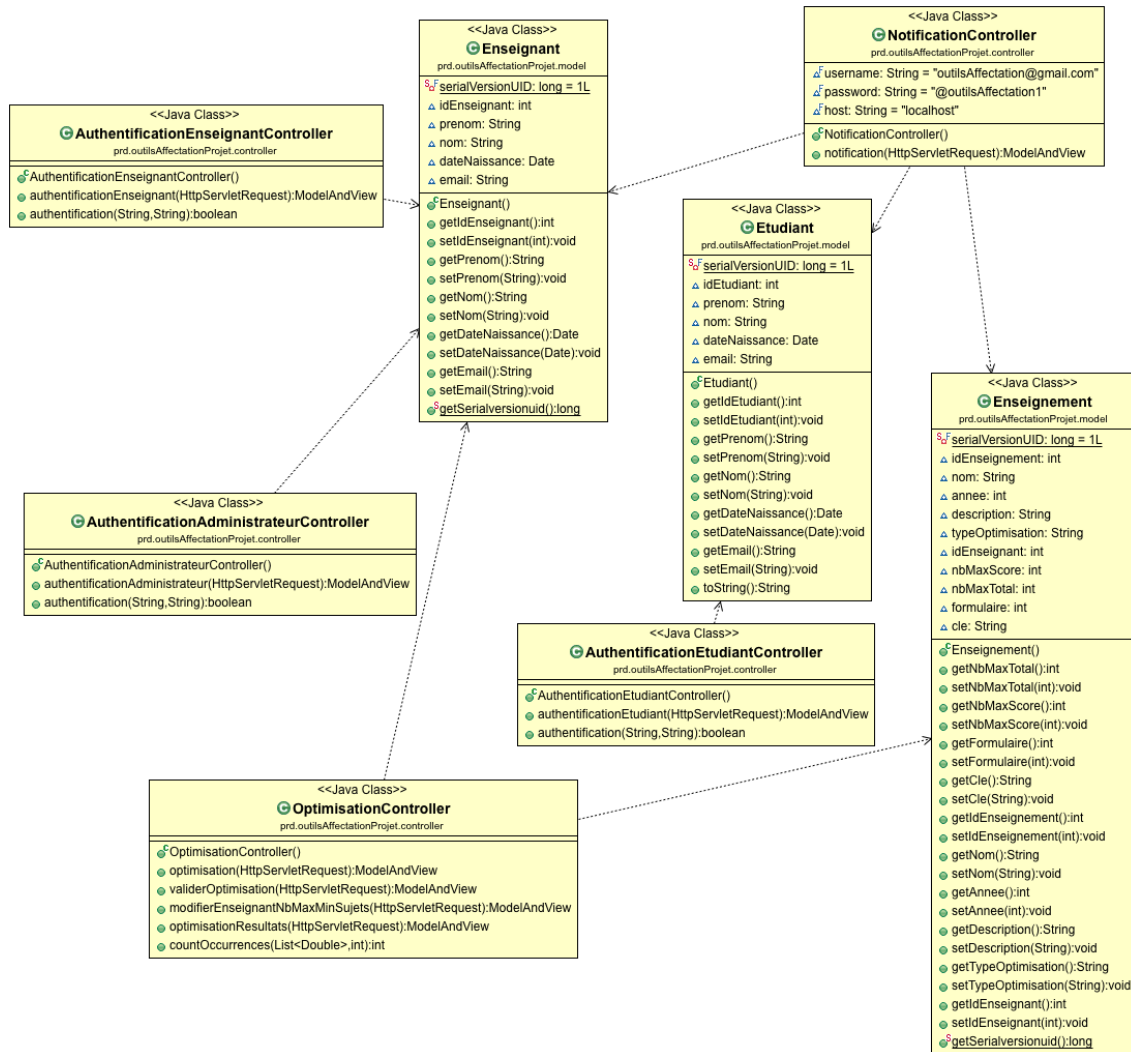


Figure 1 – Diagramme de classe

Comme présenté ci-dessus, les enseignants et les étudiants s'authentifient. Les enseignants afin de gérer les projets et les étudiants pour remplir les formulaires. Cette authentification se fera à travers l'active directory de l'école. Les enseignants peuvent également faire de l'optimisation en important un fichier csv ou en laissant les étudiants insérer leurs vœux.

## 2.1 LDAP

LDAP (Lightweight Directory Access Protocol) est une norme Internet qui permet d'accéder à des informations provenant de différents systèmes et applications informatiques. LDAP utilise un ensemble de protocoles pour accéder aux répertoires contenant des informations et extraire ces dernières. Un répertoire est similaire à une base de données, mais contient des informations plus descriptives et basées sur des attributs. Les informations d'un répertoire sont généralement lues plus souvent qu'elles ne sont écrites ou modifiées. LDAP permet à une application, fonctionnant sur la plate-forme informatique d'une école, d'obtenir des informations telles que des noms d'utilisateur et des mots de passe. La centralisation de ce type d'informations simplifie votre travail en vous dotant d'un point d'administration unique. Les informations sur les utilisateurs sont fournies à un seul endroit, ce qui réduit le stockage des informations en double. Cela conduit également à une réduction des besoins de maintenance. L'authentification LDAP permet également aux utilisateurs d'avoir un seul identifiant et mot de passe de connexion pour accéder

à différentes applications.

## 2.2 Fichier csv d'import

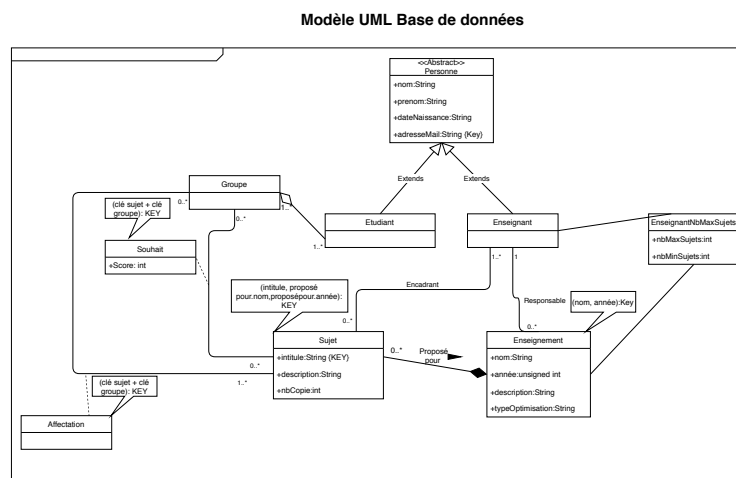
Le fichier csv qui sera importé aura cette forme : emails étudiants, sujet, score,email encadrant du sujet. S'il y a plusieurs étudiants leurs emails peuvent être séparés de ';'. Exemple :

- m@etu.univ-tours.fr;a@etu.univ-tours.fr,analyse de donnee,1, ens1@univ-tours.fr
- mi@etu.univ-tours.fr;i@etu.univ-tours.fr,analyse de donnee,2, ens1@univ-tours.fr.

## Mise en oeuvre

# 1 Base de données

Nous avons modélisé ainsi la base de données :



**Figure 1 – UML Base de données**

Les étudiants et les enseignants sont des personnes. Un étudiant peut appartenir à 1 ou plusieurs groupes. Ces groupes sont de tailles variables et peuvent émettre des souhaits sur des sujets.

Les enseignants peuvent encadrer des sujets et peuvent également être responsable d'un ou plusieurs enseignements. Pour un enseignement donné un enseignant peut encadrer un nombre

minimum et maximum de sujets.

## 2 Framework Spring MVC

Spring MVC est un framework qui permet d'implémenter des applications selon le design pattern MVC. Pour plus de détails Spring MVC fonctionne comme décrit dans le schéma ci dessous :

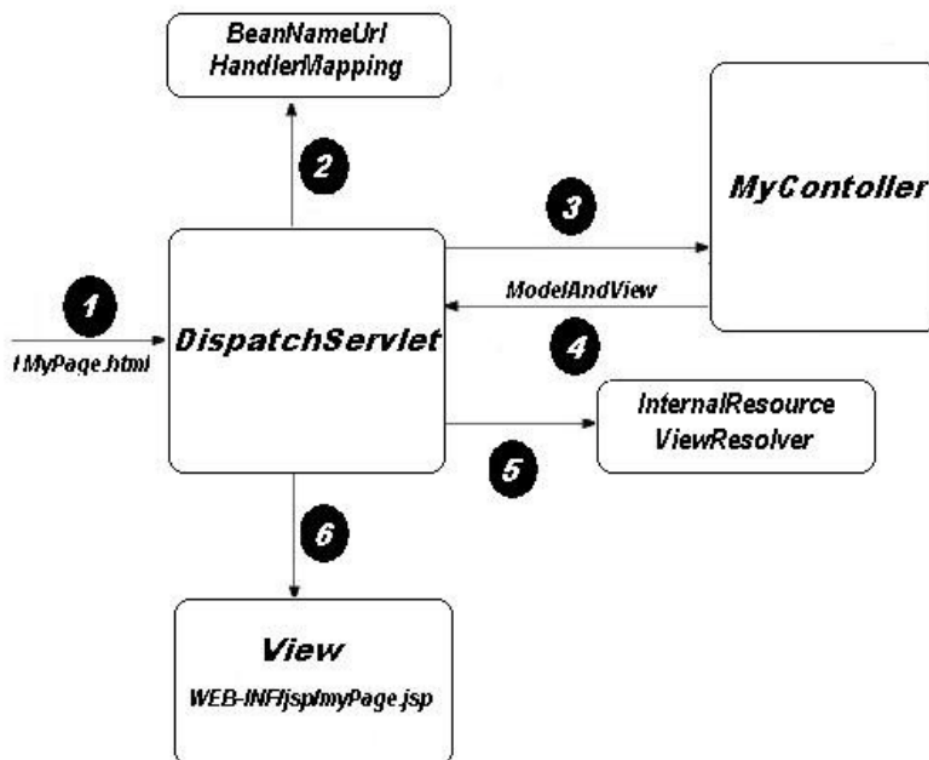


Figure 2 – Framework Spring MVC

Les étapes sont :

1. Le `DispatcherServlet` reçoit une requête dont l'URI-pattern est `/myPage.html`.
2. Le `DispatcherServlet` consulte son Handler Mapping (Exemple : `BeanNameUrlHandlerMapping`) pour connaître le contrôleur dont le nom de bean est `/myPage.html`.
3. Le `DispatcherServlet` dispatche la requête au contrôleur identifié (Exemple : `MyController`).
4. Le contrôleur retourne au `DispatcherServlet` un objet de type `ModelAndView` possédant comme paramètre au minimum le nom logique de la vue à renvoyer.
5. Le `DispatcherServlet` consulte son View Resolver lui permettant de trouver la vue dont le nom logique est `'myPage'`. Ici le type de View Resolver choisit est `InternalResourceViewResolver`.
6. Le `DispatcherServlet` forwarde la requête à la vue associée. Ici la page `/WEB-INF/jsp/myPage.jsp`.

La raison pour laquelle nous avons décidé d'utiliser Spring MVC : Spring un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'applications J2EE. Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ces objets. Le gros avantage par rapport aux serveurs d'application est qu'avec Spring, les classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le framework (au

contraire des serveur d'applications J2EE et des EJBs). C'est en ce sens que Spring est qualifié de conteneur « léger ».

### 3 Optimisation et notification

Pour cette partie nous avons utilisé la librairie clp-java. Grâce à cette librairie il est possible de créer des algorithmes avec des contraintes de programmation linéaire en nombres entiers. Ainsi nous l'avons utilisé pour les peip en énumérant les contraintes de cette manière : en tenant en compte que les groupes sont déjà créés.

- Chaque groupe est affecté à un seul sujet
- Chaque groupe a un entier Bmin comme score minimum sur ses souhaits (pire affectation)
- Chaque sujet peut être pris un nombre de fois égal à son nombre de copies.
- Chaque enseignant peut encadrer un nombre maximum de sujets (par défaut 10).
- Chaque enseignant peut encadrer un nombre minimal de sujets (par défaut 1).

Après avoir spécifié toutes ces contraintes à la librairie il suffit de lui dire de maximiser et nous obtenons un résultats.

### 4 Qualité

Afin de pouvoir attester de la qualité de notre travail nous avons décidé de mettre en place plusieurs types de tests :

- Test unitaires : les tests unitaires ont été mis en place sur les services (dao) car ce sont les services principalement utilisés par les contrôleurs pour envoyer des informations à la vue.
- Test des liens du site : les tests sur les liens permettent de vérifier que lorsqu'on clique sur un lien on aille bien au lien voulu, qu'il n'y ait pas d'erreur de frappe.
- Test sur les formulaires : Les formulaires ont été vérifiés afin qu'ils fassent bien les actions voulues conformément au cahier de spécifications.
- Test sur l'optimisation : Des jeux de données ont été créés dont on connaît déjà la solution afin de vérifier que la solution que l'on attendait soit la solution qui est affichée.
- Test d'interface selenium : Ces tests d'interfaces automatisent les tests déjà effectués ci-dessus.
- Test d'intégration gitlab ci : À l'aide de gitlab ci nous pouvons vérifier que les ajouts au code ne dégradent pas le code existant à l'aide de tests automatisés.

# 6

## Bilan et conclusion

### 1 Fait

Les tâches suivantes ont été faites :

- Analyse des problèmes d'affectation :  
Comme décrit plus haut au chapitre 3 nous avons étudié les différents problèmes d'affectation à mettre en oeuvre que nous résoudrons à l'aide du solveur clp.
- Modélisation des classes/Méthodes à réaliser :  
Comme décrit plus haut au chapitre 4 le diagramme de classe a été fait, il contient les différentes classes qui permettront de résoudre nos problèmes d'affectation.
- Modélisation du site à réaliser :  
Nous avons également réalisé une arborescence du site, figure 6 (Chapitre 2)(temporaire) afin de mieux appréhender la partie développement :
- La partie réalisation
  - Le développement :
    1. Création de la base de donnée.
    2. Création de formulaire automatique.
    3. Mise en place du projet avec Spring MVC.
    4. Création des vues.
    5. Notification.
  - La phase de test et documentation :
    1. Rédaction des tests.
    2. Correction des beugs.
    3. Documentation et préparation de la soutenance.

### 2 Reste à faire

Comme décrit ci-dessus il nous reste :

1. Authentification active directory
2. Rajouter les modèles des autres enseignements.

### 3 Planning

Le planning est décrit dans le chapitre [D](#).

### 4 Bilan sur la qualité

La qualité des travaux est assurée dans ces aspects :

- La qualité du code
- La qualité du modèle

Le code ayant été testé sur plusieurs plans et les bugs ayant été corrigés on peut en conclure que la qualité du code est attesté. Le modèle ayant été fait avec l'encadrant on peut en juger la qualité également.

### 5 Bilan auto critique sur la gestion de projet

Le projet a été bien géré étant donné que tous les livrables attendus ont été remis à l'exception de l'authentification par l'active directory.

### 6 Améliorations

Il serait peut-être bénéfique de rajouter une contrainte qui permet de favoriser un étudiant qui n'a pas de compétences dans un domaine particulier par rapport à d'autres étudiants qui en ont.

### 7 Conclusion

Ce projet nécessite des connaissances en recherche opérationnelle ainsi qu'en développement web. Le mélange de ces deux m'a permis d'acquérir beaucoup de connaissances scientifique et de mesurer mes capacités dans un travail en autonomie.

## Annexes



# A

## Spécifications fonctionnelles

On va introduire les fonctions principales qui peuvent être utilisées dans la partie optimisation et dans la partie interface.

### 1 Partie optimisation

#### 1.1 Fonction 1 : Importation fichier csv contenant les résultats du formulaire

1. Présentation :
  - Nom : Importation fichier csv.
  - Rôle : Cette fonction sert à entrer les informations tirées d'un formulaire.
  - Priorité : fonctionnalité secondaire.
2. Description : Lorsqu'un enseignant ne souhaite pas créer un projet et insérer les sujet directement sur l'application web, on lui permettra d'importer un fichier csv contenant les adresses mails et les poids de chaque projet.
  - Entrées : fichier csv
  - Sortie : création d'un projet dans la base de données.

#### 1.2 Fonction 2 : Calculer la solution du projet

1. Présentation :
  - Nom : Calcul de la solution du projet.
  - Rôle : Cette fonction sert à résoudre le problème d'affectation des étudiants aux projets.
  - Priorité : fonctionnalité principale.
2. Description : Cette fonction permet à l'enseignant de calculer une solution optimale du problème d'affectation des étudiants aux projets.
  - Entrées : Nom du projet, Type de classement, Liste de contraintes.
  - Sortie : Graphique affichant la solution (nombre d'étudiants ayant leur premier voeu, second voeu etc..)

## 2 Partie interface

### 2.1 Fonction 1 : Affichage de l'historique des sujets

1. Présentation :
  - Nom : Affichage de l'historique.
  - Rôle : Cette fonction sert à afficher la liste des sujets d'anciens projets.
  - Priorité : fonctionnalité secondaire.
2. Description : N'importe quel utilisateur (étudiant ou enseignant) peut consulter la liste d'anciens projets. Cette fonction sera utilisée par une page de recherche sur mot clé.
  - Entrées : l'année de l'enseignement, le thème du sujet, le nom de l'enseignement.
  - Sortie : liste de sujet qui correspond aux critères.

### 2.2 Fonction 2 : Authentification

1. Présentation :
  - Nom : Authentification.
  - Rôle : Cette fonction sert à authentifier les utilisateurs (étudiants et enseignants).
  - Priorité : fonctionnalité principale.
2. Description : Cette fonction fait appel à l'active directory afin d'authentifier les utilisateurs du site.
  - Entrées : Adresse mail et mot de passe (univ tours) de l'utilisateur.
  - Sortie : Accès à l'interface projet pour les enseignants ou aux formulaires pour les étudiants.

### 2.3 Fonction 3 : Création d'enseignement

1. Présentation :
  - Nom : Création d'un enseignement.
  - Rôle : Cette fonction sert à créer un enseignement qui aura des sujets, et à partir duquel on pourra créer un formulaire pour les étudiants.
  - Priorité : fonctionnalité principale.
2. Description : Cette fonction crée un enseignement.
  - Entrées : nom, annee, responsable, booléen formulaire ou pas
  - Sortie : Une entrée en base de données.

### 2.4 Fonction 4 : Ajouter un sujet

1. Présentation :
  - Nom : Ajout de sujet .
  - Rôle : Cette fonction sert à ajouter un sujet à un projet existant.
  - Priorité : fonctionnalité principale.
2. Description : Cette fonction prend un projet existant et lui rajoute un sujet.
  - Entrées : intitulé du sujet, identifiant de l'enseignement, nombre de copie, description.
  - Sortie : Une entrée en base de donnée.

**2.5 Fonction 5 : Supprimer un sujet**

1. Présentation :
  - Nom : Suppression de sujets .
  - Rôle : Cette fonction sert à supprimer un sujet d'un projet existant.
  - Priorité : fonctionnalité secondaire.
2. Description : Cette fonction prend un projet existant et lui retire un sujet.
  - Entrées : Nom du sujet.
  - Sortie : Un retrait de la base de donnée.

**2.6 Fonction 6 : Valider l'affectation**

1. Présentation :
  - Nom : Valider l'affectation.
  - Rôle : Cette fonction sert à confirmer qu'un projet contient toutes les information nécessaire à l'affectation.
  - Priorité : fonctionnalité principale.
2. Description : A l'aide d'un bouton on peut valider l'affectation et l'enregistrer dans la base de donnée
  - Entrées : l'affectation en question.
  - Sortie : Une insertion dans la base de données.

**2.7 Fonction 7 : Créer un formulaire pour l'enseignement**

1. Présentation :
  - Nom : Création de formulaire.
  - Rôle : Cette fonction sert à ajouter un formulaire à un projet existant.
  - Priorité : fonctionnalité secondaire.
2. Description : Cette fonction prend un projet existant et lui rajoute un formulaire qui contient une liste de sujet et prend en compte la méthode de rajout de coûts.
  - Entrées : booléen 1 ou 0.
  - Sortie : Affichage ou Fermeture du formulaire.

**2.8 Fonction 8 : Affichage solution projet**

1. Présentation :
  - Nom : Affichage solution projet.
  - Rôle : Cette fonction sert à calculer des statistiques sur le projet et à les afficher : taux de satisfaction des étudiants(quel pourcentage a eu son 1er voeu, second voeu ou troisième voeu).
  - Priorité : fonctionnalité secondaire.
2. Description : Cette fonction calcule des statistique sur un projet
  - Entrées : liste de souhaits, solution trouvée.
  - Sortie : Une page contenant les statistiques du projet.

**2.9 Fonction 9 : Envoyer notification aux étudiants**

1. Présentation :

- Nom : Envoyer notification.
  - Rôle : Cette fonction sert à notifier les étudiants des résultats de l'affectation.
  - Priorité : fonctionnalité secondaire.
2. Description : Cette fonction envoie un mail aux étudiants avec les résultats de l'affectation
- Entrées : solution de l'affectation.
  - Sortie : Un mail que les étudiants reçoivent.

### 2.10 Fonction 10 : Répondre au formulaire

1. Présentation :
- Nom : Répondre au formulaire.
  - Rôle : Cette fonction sert à stocker les réponses du formulaire par les étudiants
  - Priorité : fonctionnalité principale.
2. Description : Cette fonction permet aux étudiants de répondre au formulaire
- Entrées : le nom de l'étudiant, la liste des choix .
  - Sortie : Une entrée en base de données.

# B

## Spécifications non fonctionnelles et interfaces

### 1 Contraintes de développement et conception

Les contraintes de développement :

- Matériels : Environnement de développement intégré Eclipse
- Langage de programmation : Java 8
- Framework : Spring mvc
- Logiciels et bibliothèques à utiliser pour le développement : CLP-Java, JSTL, SPRING, HIBERNATE.
- Bibliothèques de programmes imposées : aucune

### 2 Contraintes de fonctionnement et d'exploitation

L'application consiste à résoudre le problème d'affectation selon des contraintes choisies par l'enseignant. Les contraintes de fonctionnement se basent sur ces différentes contraintes données par l'enseignant.

1. Authentification des étudiants.
2. Possibilité de rajouter des contraintes du style (chaque étudiant doit avoir au moins son troisième vœu).
3. Changer le nombre de sujet qu'un enseignant encadre etc..

### 3 Performances

Comme nous l'avons décrit dans la veille technologique le solveur clp est capable de résoudre en 6 secondes plus de 50 instances de problèmes bidimensionnels beaucoup plus complexes que le notre. En ce qui concerne la partie web un serveur java ee est capable de garantir une haute performance malgré une montée en charge.

## 4 Capacités

Le solveur est capable de résoudre un grand nombre d'instances en peu de temps. Nous pouvons fixer une limite d'itération au bout de laquelle le solveur s'arrête. Ce cas serait rare dans notre cas donc nous pouvons assumer qu'il n'y a pas de limite quand à la résolution d'une affectation de projet au sein du département informatique.

## 5 Mode de fonctionnement

La partie optimisation nécessite l'authentification de l'enseignant. Lorsque celui-ci accède à l'interface, il peut créer un projet ou importer un fichier csv contenant les informations nécessaires à l'optimisation. Lorsque cela est fait il peut ainsi indiquer les contraintes qui seront énumérées dans un combobox il pourra en sélectionner plusieurs, puis cliquer sur un bouton lancer.

## 6 Contrôlabilité

Afin de suivre l'exécution des différentes actions, des messages vont apparaître dans des fenêtres.

## 7 Sécurité

L'authentification des enseignants permet de sécuriser la manipulation des projets. L'authentification des étudiants permettra de rendre les données plus intègres.

## 8 Interfaces

Ce diagramme de cas d'utilisation résume l'interface homme machine.

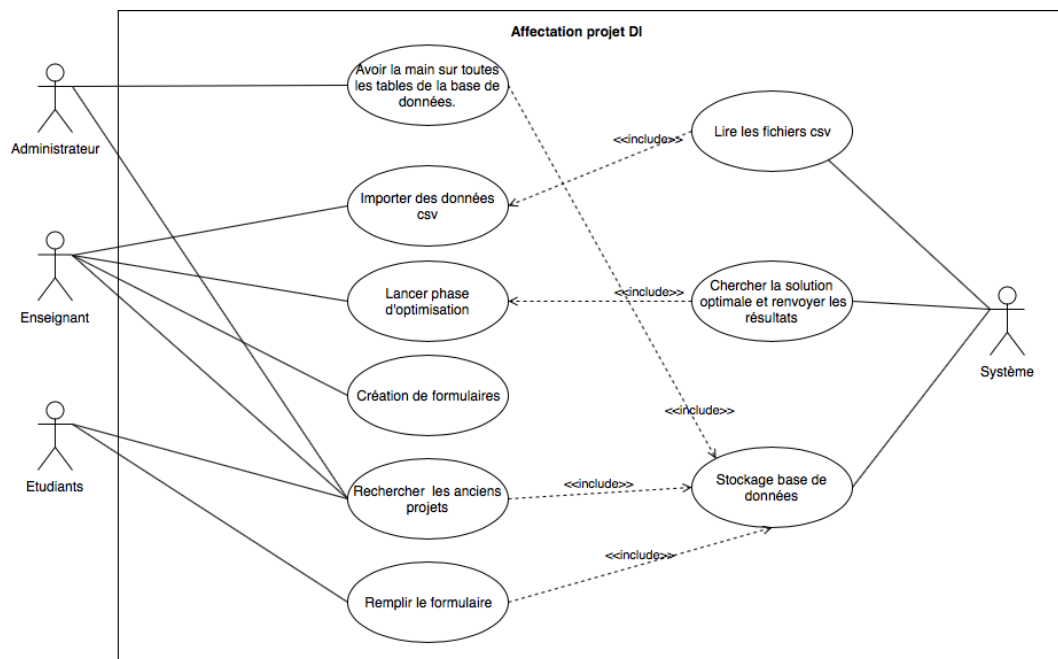


Figure 1 – Diagramme de cas d'utilisation

## 8.1 Interfaces matériel/logiciel

Dans ce projet, nous allons développer une interface web. Pour cela, l'utilisateur devra être muni d'un ordinateur et plus précisément d'un navigateur et d'un service d'accès Internet afin de pouvoir exécuter toutes les requêtes. De plus, le navigateur communiquera avec une base de données centralisée située sur un serveur virtualisé interne à l'Université. La plateforme sera donc accessible depuis l'intérieur du réseau universitaire.

## 8.2 Interfaces homme/machine

Les différents utilisateurs pourront avoir accès à la plateforme. Une gestion de l'authentification devra être mise en place pour différents types d'utilisateurs ayant des droits différents sur la plateforme.

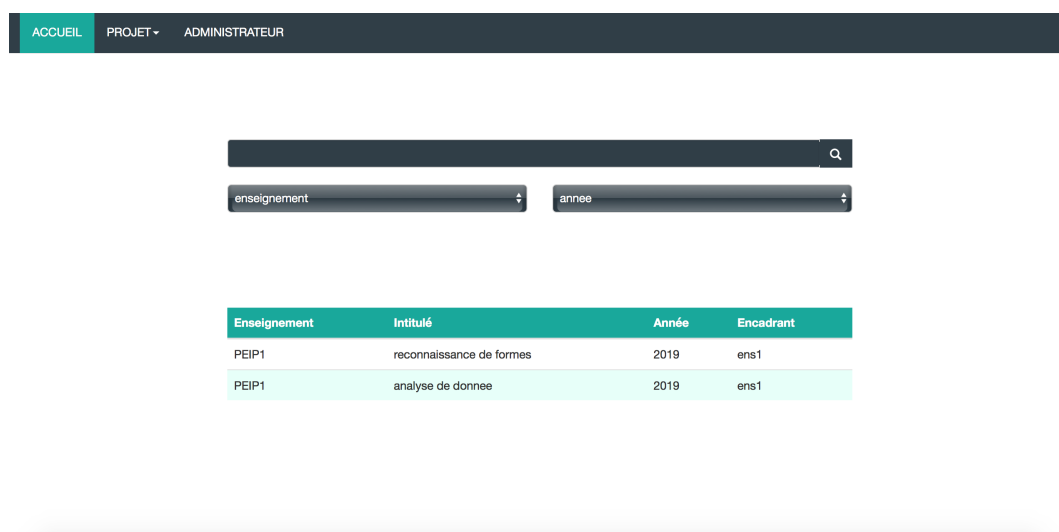


Figure 2 – Interface homme machine

## 8.2.1 Interface Etudiant

Les étudiants auront accès à toutes les parties de l'application sauf la partie projet qui elle est réservée aux enseignants. Ils devront s'authentifier pour accéder aux formulaires.

Veillez saisir un email\*

Veillez saisir un mot de passe\*

Opération réussie

Fermer

Figure 3 – Authentification étudiant

Ils peuvent émettre des vœux sur des sujets.

ACCUEIL PROJET ADMINISTRATEUR

Entrez les scores sans dépasser 12 !!!

Adresses mails des membres du groupe

enregistrer

| Sujet                    | Score |
|--------------------------|-------|
| analyse de donnée        | 1     |
| reconnaissance de formes | 3     |

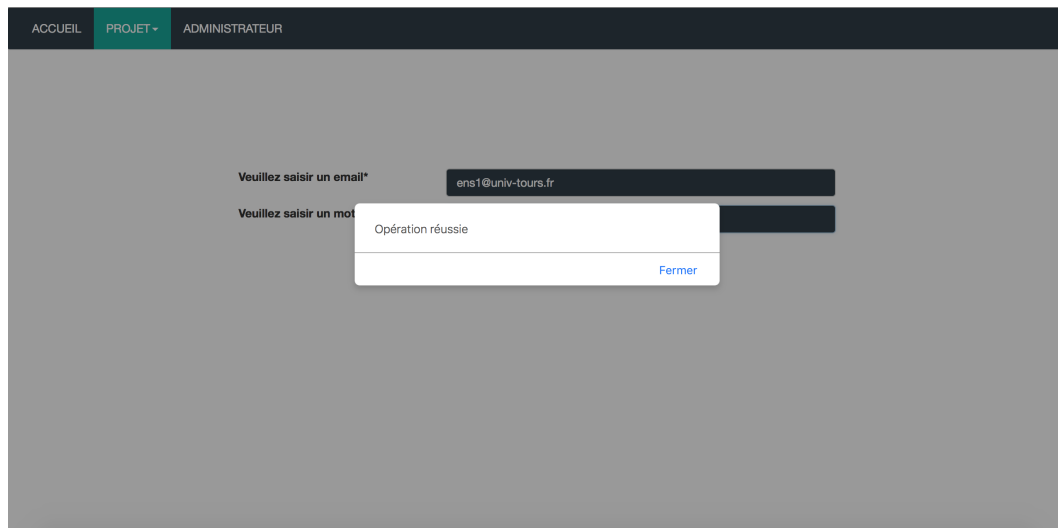
valider

Figure 4 – Vœux étudiants



## 8.2.2 Interface Enseignants

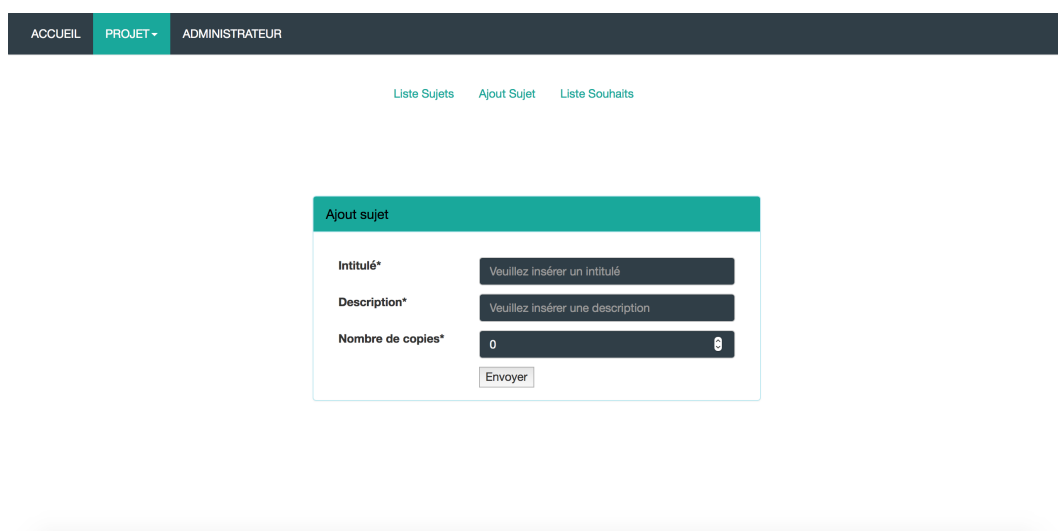
Les enseignants auront accès à toutes les parties de l'application sauf l'administration. Ils devront s'authentifier.



The screenshot shows a web interface with a dark header containing three menu items: 'ACCUEIL', 'PROJET' (highlighted in green), and 'ADMINISTRATEUR'. The main content area is light gray and contains two input fields. The first field is labeled 'Veillez saisir un email\*' and contains the text 'ens1@univ-tours.fr'. The second field is labeled 'Veillez saisir un mot de passe\*' and is empty. A white modal dialog box is centered over the form, displaying the message 'Opération réussie' and a blue 'Fermer' button.

Figure 5 – *Authentification enseignant*

Ils peuvent également ajouter des sujets :



The screenshot shows a web interface with a dark header containing three menu items: 'ACCUEIL', 'PROJET' (highlighted in green), and 'ADMINISTRATEUR'. Below the header, there are three links: 'Liste Sujets', 'Ajout Sujet' (highlighted in green), and 'Liste Souhaits'. The main content area is light gray and contains a form titled 'Ajout sujet' with a teal header. The form has three input fields: 'Intitulé\*' with placeholder text 'Veillez insérer un intitulé', 'Description\*' with placeholder text 'Veillez insérer une description', and 'Nombre de copies\*' with a value of '0' and a small icon. An 'Envoyer' button is at the bottom of the form.

Figure 6 – *Ajout Sujet*

De plus les enseignants peuvent importer un fichier csv ou passer directement à l'optimisation ainsi que de créer des groupes d'une certaine taille ou non :

ACCUEIL

PROJET

ADMINISTRATEUR

Liste SujetsAjust SujetListe Souhaits

PEIP1

fichier de souhaits  
en csv

Choisir le fichier aucun fichier sél.

optimiser

enregistrer

| Groupe | Sujet | Encadrant | Score |
|--------|-------|-----------|-------|
| 3      | 1     | ens1      | 1     |
| 4      | 1     | ens1      | 2     |
| 3      | 2     | ens2      | 2     |
| 4      | 2     | ens2      | 1     |

Type de création de groupe

Aucun

Taille min groupe 1

Taille max groupe 1

Figure 7 – Liste des souhaits

Enfin les enseignants peuvent émettre des contraintes et lancer l'optimisation :

ACCUEIL

PROJET ▾

ADMINISTRATEUR

### PEIP1

Problème à résoudre ▾

2 0

optimiser

| Nom  | Prénom | Nombre de sujets min | Nombre de sujets max | Action      |
|------|--------|----------------------|----------------------|-------------|
| ens1 | ens1   | 1 0                  | 10 0                 | enregistrer |
| ens2 | ens2   | 1 0                  | 10 0                 | enregistrer |

Figure 8 – *Optimisation*

# C

## Analyse et cahier développeur

### 0.1 Interface logiciel/logiciel

Concernant les interfaces logiciel/logiciel, il s'agit d'abord de la liaison entre la base de données et notre application (situées sur le même serveur). En effet, étant donné que nous manipulons une base de donnée il faudra régulièrement communiquer avec celle-ci. De plus, nous utiliserons des fichiers csv importés on peut donc également considérer ces fichiers comme interface entre Celene et notre base de données.

### 1 Diagramme de classes

Voici notre diagramme de classe de nos modèles.

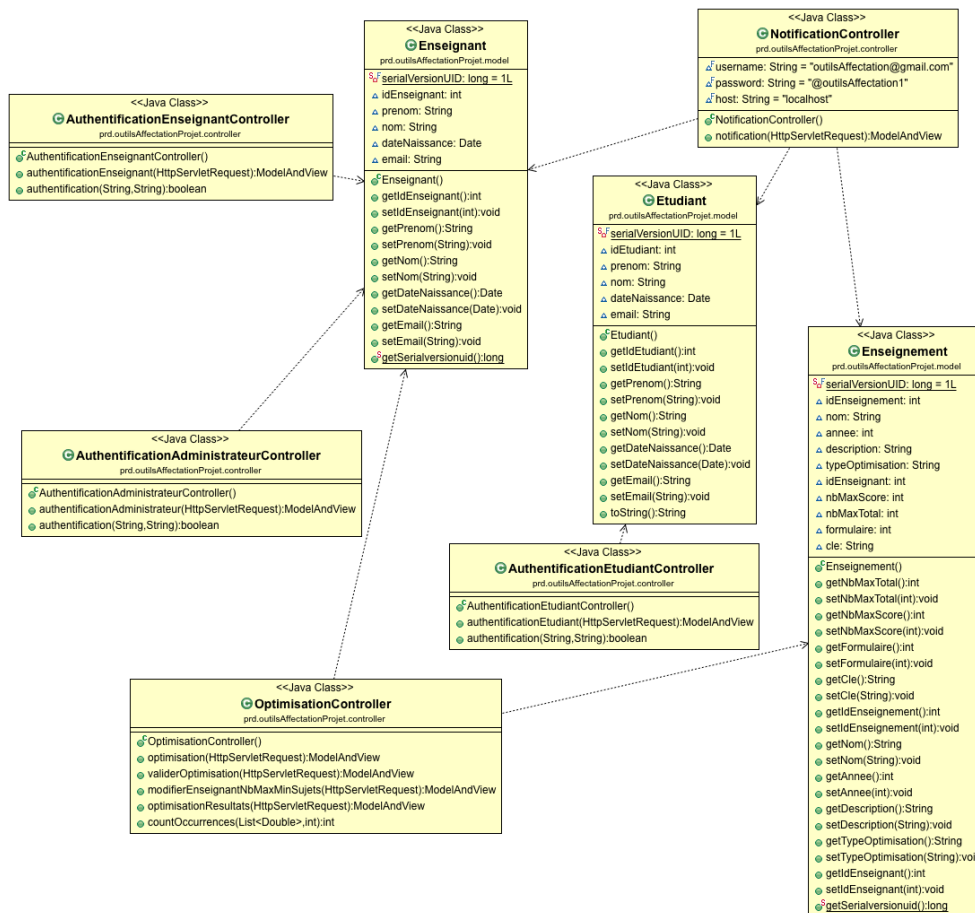


Figure 1 – Diagramme de classe

## 2 Description détaillées des classes

- Enseignant : La classe la plus importante du projet car c'est celle qui fait le plus d'action. Elle peut utiliser la classe Enseignement afin d'exécuter toutes les actions sur le projet (importation, ajout de sujet, suppression etc..) et également résoudre le problème d'affectation. L'enseignant peut également s'authentifier et notifier les étudiants du résultat.
- NotificationController : Cette classe permet d'envoyer à un étudiant le résultat de son affectation. Il y aura une boucle dans la méthode notifier qui fera que chaque étudiant recevra uniquement son affectation à lui et pas celui des autres.
- Etudiant : Cette classe peut remplir un formulaire afin que ses vœux soient pris en compte. Elle peut également s'authentifier.
- AuthentificationEtudiant et AuthentificationEnseignant : Ces classes permettent de faire appel à l'active directory de l'université afin d'authentifier un étudiant.
- Enseignement : Cette classe représente un enseignement avec sa liste de sujet et d'autres informations nécessaires.
- Optimisation : Cette classe gère l'optimisation et traduit les contraintes au solveur.

## 3 Structure des fichiers utilisés

Le fichier csv qui est composé comme ceci : emails étudiants, sujet, score, email encadrant du sujet. S'il y a plusieurs étudiants leurs emails peuvent être séparés de ',' ; Exemple :

- m@etu.univ-tours.fr ; a@etu.univ-tours.fr, analyse de donnée, 1, ens1@univ-tours.fr

— `mi@etu.univ-tours.fr;i@etu.univ-tours.fr,analyse de donnee,2,ens1@univ-tours.fr`.

# D

## Gestion de projet

La gestion de projet se fait avec la méthode Agile Scrum. Chaque semaine je fais un rapport sur ce que j'avais prévu, ce que j'ai pu réaliser (documents) et ce que je compte faire la semaine suivante. Après, l'encadrant va donner ses remarques sur mon travail et également répondre aux éventuelles question que j'ai et ce qui me bloque.

Les outils de gestion de projet sont :

- la diagramme de Gantt : pour le planning
- l'application Trello : pour suivre les tâches en respectant le délais
- git : pour le contrôle de version

### 1 Découpage des tâches

Notre diagramme de gantt nous a permis de planifier notre projet.

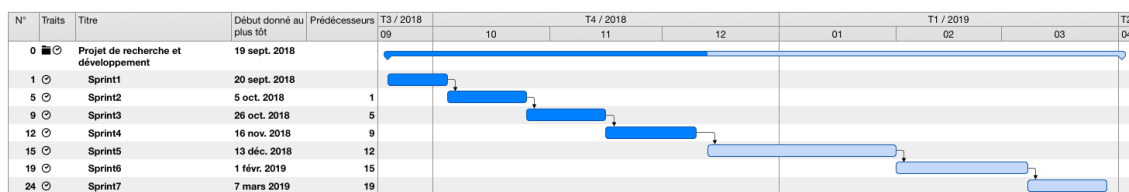


Figure 1 – diagramme de Gantt

#### 1.1 Sprint 1

- Description de la tâche : Compréhension du projet.
- Estimation de la charge : 4 jours homme.
- Livrables : Document résumant les différentes attentes des encadrants de projet par rapport à l'application.

#### 1.2 Sprint 2

- Description de la tâche : Cadrage du projet et planification du projet.

- Estimation de la charge : 6 jours homme
- Livrables : Diagramme de gantt.

### 1.3 Sprint 3

- Description de la tâche : Rédaction des spécifications fonctionnelles et non fonctionnelles.
- Estimation de la charge : 4 jours homme
- Livrables : Cahier de spécification.

### 1.4 Sprint 4

- Description de la tâche : Rédaction du rapport et préparation de la soutenance.
- Estimation de la charge : 8 jours homme
- Livrables : Rapport et Diapositives de la soutenance.

### 1.5 Sprint 5

- Description de la tâche : Création de la base de données, mise en place du projet spring mvc et création des vues.
- Estimation de la charge : 20 jours homme
- Livrables : Projet avec les pages

### 1.6 Sprint 6

- Description de la tâche : Création des actions sur les vues, de l'optimisation et de la notification
- Estimation de la charge : 35 jours homme
- Livrables : Projet fonctionnel

### 1.7 Sprint 7

- Description de la tâche : Recette
- Estimation de la charge : 15 jours homme
- Livrables : Projet de qualité, fiches de tests, Documentation



# E

# Documentation d'installation et de maintenance

## 1 Installation

1. Le matériel : Il faudrait un ordinateur
2. Les prérequis :
  - Tomcat version :8.5
  - Serveur base de donnée : wamp ou mamp
  - Ide java eclipse version : oxygen 2017
  - Maven version : 3
  - Jar driver Mysql
3. Configuration de la base de données :
  - Créer une base de donnée nommée PRD
  - username : root
  - mot de passe : root
4. Lancement de l'application : Il faut changer l'adresse mail dans la classe notification controller ligne 34 puis lancer l'application sur le serveur tomcat 8.5 et un navigateur s'ouvre. Pour utiliser l'interface enseignant et administrateur il faut utiliser l'adresse mail : ens1@univ-tours.fr et le mot de passe : Pop2212!

## 2 Maintenance

1. Structure du code : J'utilise le framework SpringMVC. Le code est structuré en plusieurs package :
  - config : les fichiers de configuration.
  - controller : les classes qui font les traitements sur le modèle et mettent à jour la vue.
  - dao : les classes qui permettent de faire des requêtes sur la base de données.
  - model : les classes entité qui sont mappées dans la base de données.
  - view : les pages jsp.
2. Fichiers de configuration :
  - HibernateConfiguration : fichier contenant les configurations de la base de données.
  - MvcConfiguration : fichier contenant les configurations pour les imports, les vues et les ressources (css, bootstrap, jquery etc..).
3. Accès à l'active directory

- Adresse IP : 10.236.5.1
- Compte de service : CN=service.polyprojects,OU=Utilisateurs,OU=Projet PolyProjects,DC=mydomain,DC=local
- Pass : Pop2212!
- Comptes utilisateurs :
  - CN=etu1,OU=Utilisateurs,OU=Projet PolyProjects,DC=mydomain,DC=local
  - CN=etu2,OU=Utilisateurs,OU=Projet PolyProjects,DC=mydomain,DC=local
  - CN=ens1,OU=Utilisateurs,OU=Projet PolyProjects,DC=mydomain,DC=local
  - CN=ens2,OU=Utilisateurs,OU=Projet PolyProjects,DC=mydomain,DC=local
- Pass : Pop2212! pour tous
- Groupes :
  - CN=grp-etu,OU=Groupes,OU=Projet PolyProjects,DC=mydomain,DC=local
  - CN=grp-ens,OU=Groupes,OU=Projet PolyProjects,DC=mydomain,DC=local

- Utilisateur quelconque : Tous les utilisateurs peuvent faire des recherches sur les sujets en marquant l'intitulé, en choisissant l'enseignement et l'année.

ACCUEIL

PROJET ▾

ADMINISTRATEUR

big data

Q

enseignement ▾

annee ▾

| Enseignement | Intitulé                 | Année | Encadrant |
|--------------|--------------------------|-------|-----------|
| Option       | Securite                 | 2019  | ens1      |
| Option       | big data                 | 2019  | ens1      |
| PEIP1        | reconnaissance de formes | 2018  | ens2      |
| PEIP1        | analyse de donnee        | 2018  | ens2      |
| PEIP2        | reconnaissance de formes | 2019  | ens1      |
| PEIP2        | analyse de donnee        | 2019  | ens1      |

Figure 1 – Recherche de sujets

- Utilisateur étudiant :
  1. S'authentifier L'étudiant peut se connecter à l'aide de son email et son mot de passe stockés dans l'active directory.

ACCUEIL PROJET ADMINISTRATEUR

Veuillez saisir un email\* mariam.konate-2@etu.univ-tours.fr

Veuillez saisir un mot de passe\*

Opération réussie

Fermer

**Figure 2** – *Authentification étudiant*

2. Consulter : L'étudiant peut voir la liste des enseignements ayant un formulaire ouvert. Il peut uniquement les ouvrir s'il connaît une clé permettant de l'ouvrir. L'étudiant

ACCUEIL PROJET ADMINISTRATEUR

Se deconnecter

| Enseignement | Année | Intitulé                            | Ouvrir |
|--------------|-------|-------------------------------------|--------|
| PEIP2        | 2019  | peip2 Big data science informatique | ouvrir |

**Figure 3** – *Consultation étudiant*

peut ouvrir un enseignement :

The screenshot shows a web interface with a dark blue header bar containing three navigation links: 'ACCUEIL', 'PROJET -', and 'ADMINISTRATEUR'. The 'PROJET -' link is highlighted in green. Below the header, the main content area is white. It features a text prompt 'Veuillez saisir une clé pour cet enseignement\*' followed by a dark blue rectangular input field. Below the input field is a small, light gray button labeled 'valider'.

**Figure 4** – *Clé enseignement*

ACCUEIL PROJET ADMINISTRATEUR

## Entrez les scores sans dépasser 12 !!!

Adresses mails des membres du groupe

enregistrer

| Sujet                    | Score |
|--------------------------|-------|
| analyse de donnée        | 1     |
| reconnaissance de formes | 3     |

valider

Figure 5 – *Voeux étudiant*

— Utilisateur enseignant :

1. S'authentifier : Les enseignants peuvent s'authentifier en indiquant leur adresse mail et leur mot de passe qui sont stockés dans l'active directory.

ACCUEIL PROJET ADMINISTRATEUR

Veillez saisir un email\*

Veillez saisir un mot de passe\*

Opération réussie

Fermer

Figure 6 – *Authentification enseignant*

2. Faire des recherche sur les enseignements : L'enseignant peut faire des recherches sur les intitulés et les descriptions des enseignements.

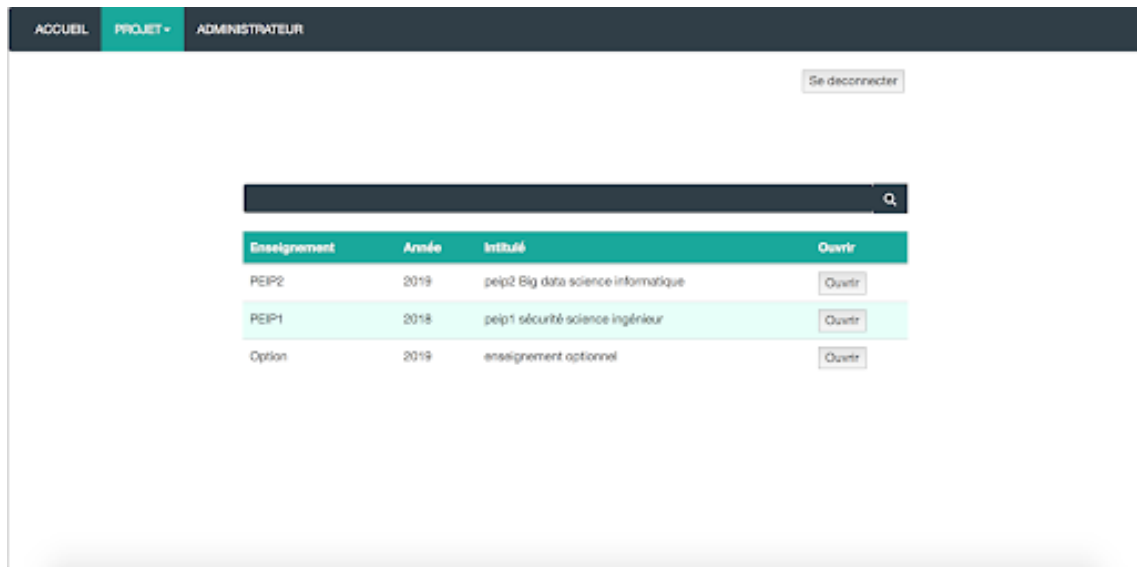


Figure 7 – Recherche enseignement

3. Ouvrir un enseignement :

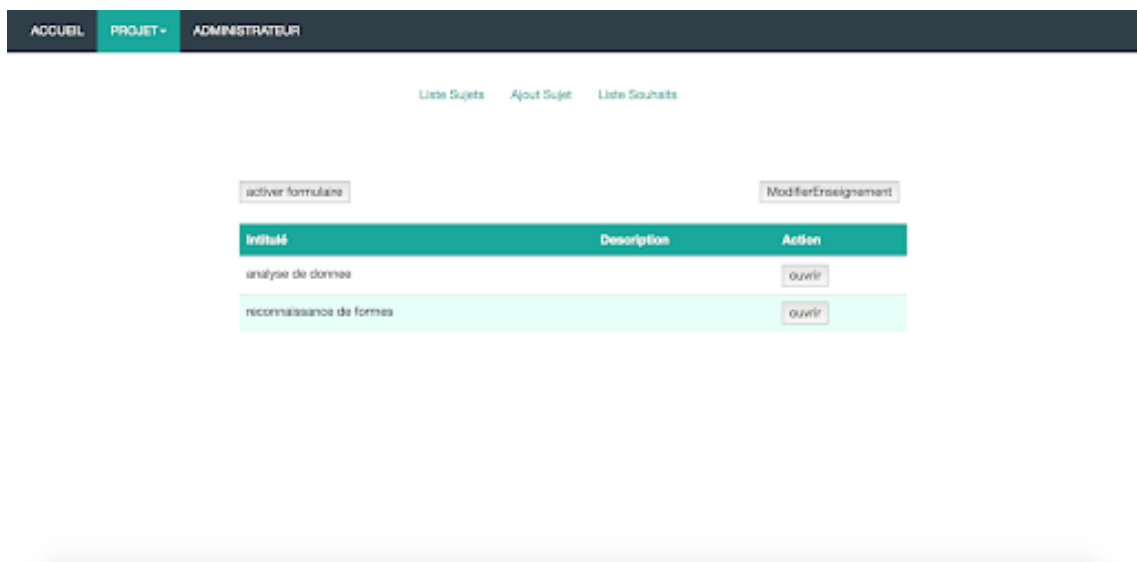
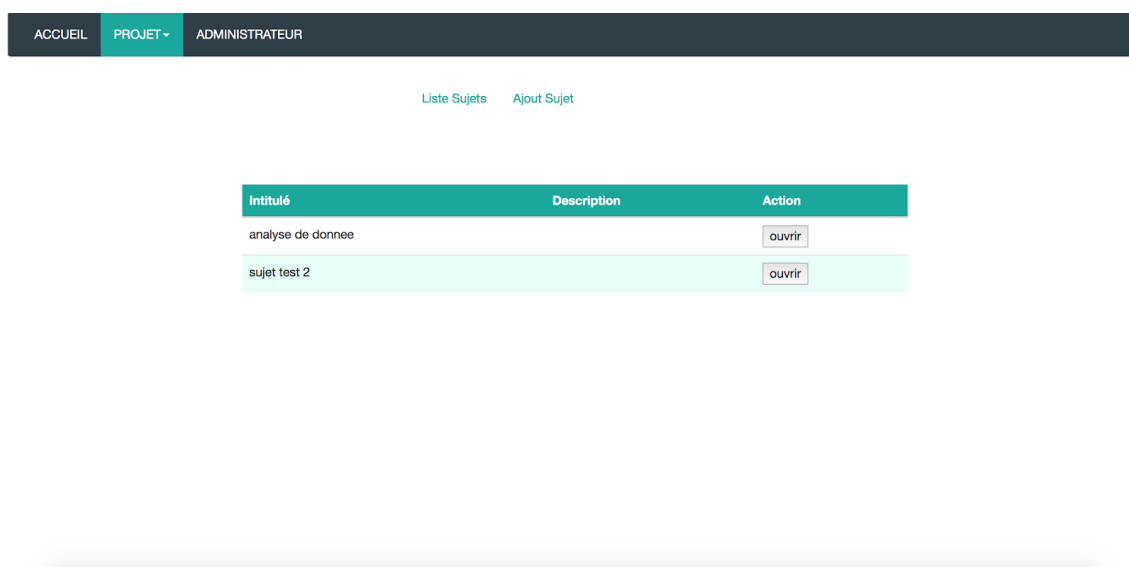


Figure 8 – ouvrir enseignement

En ouvrant un enseignement il peut consulter la liste des sujets de cet enseignement. Ajouter un sujet dans l'onglet Ajout Sujet et modifier un sujet s'il est l'enseignant référent de l'enseignement. Il peut activer ou désactiver le formulaire de l'enseignement afin que les étudiants y aient accès ou non. De plus il peut consulter les souhaits émis. Pour finir il peut modifier l'enseignement s'il est le référent.

Lorsqu'un enseignant ouvre un enseignement dont il n'est pas le référent. Il ne peut qu'ajouter des sujets et consulter/modifier les sujets qu'il a lui même créé.



**Figure 9** – ouvrir enseignement si on est pas référent



4. Lancer l'optimisation : L'enseignant peut importer un fichier contenant les voeux de ses étudiants. Le fichier en question doit être un fichier csv avec comme colonnes : emails étudiants, sujet, score, email encadrant du sujet. S'il y a plusieurs étudiants leurs emails peuvent être séparés de ';'.

ACCUEIL PROJET ADMINISTRATEUR

Liste Sujets Ajout Sujet Liste Souhaits

### PEIP1

fichier de souhaits en csv Choisir le fichier aucun fichier sél. optimiser

enregistrer

| Groupe | Sujet | Encadrant | Score |
|--------|-------|-----------|-------|
| 3      | 1     | ens1      | 1     |
| 4      | 1     | ens1      | 2     |
| 3      | 2     | ens2      | 2     |
| 4      | 2     | ens2      | 1     |

Type de création de groupe

Aucun ✓

Taille min groupe 1 Taille max groupe 1

Figure 10 – Liste Souhaits

L'enseignant peut créer des groupes en indiquant une taille minimum et une taille maximum. Il peut également décider de lancer la phase d'optimisation :

ACCUEIL PROJET ADMINISTRATEUR

### PEIP1

Problème à résoudre 2 optimiser

| Nom  | Prénom | Nombre de sujets min | Nombre de sujets max | Action      |
|------|--------|----------------------|----------------------|-------------|
| ens1 | ens1   | 1                    | 10                   | enregistrer |
| ens2 | ens2   | 1                    | 10                   | enregistrer |

Figure 11 – Lancer l'optimisation

- Utilisateur administrateur : L'administrateur a la main sur toutes les tables de la base de données.

|         |          |                |
|---------|----------|----------------|
| ACCUEIL | PROJET ▾ | ADMINISTRATEUR |
|---------|----------|----------------|

[Liste Enseignements](#)
[Ajout Enseignement](#)
[Gestion Tables](#)

| Table          | Ajouter                                | Vider                                |
|----------------|--|--------------------------------------|
| Affectation    | <a href="#">Ajouter Affectation</a>    | <a href="#">Vider Affectation</a>    |
| Encadrant      | <a href="#">Ajouter Encadrant</a>      | <a href="#">Vider Encadrant</a>      |
| Enseignant     | <a href="#">Ajouter Enseignant</a>     | <a href="#">Vider Enseignant</a>     |
| Enseignement   | <a href="#">Ajouter Enseignement</a>   | <a href="#">Vider Enseignement</a>   |
| Etudiant       | <a href="#">Ajouter Etudiant</a>       | <a href="#">Vider Etudiant</a>       |
| Groupe         | <a href="#">Ajouter Groupe</a>         | <a href="#">Vider Groupe</a>         |
| MembreDuGroupe | <a href="#">Ajouter MembreDuGroupe</a> | <a href="#">Vider MembreDuGroupe</a> |
| Souhait        | <a href="#">Ajouter Souhait</a>        | <a href="#">Vider Souhait</a>        |
| Sujet          | <a href="#">Ajouter Sujet</a>          | <a href="#">Vider Sujet</a>          |

Figure 12 – Tables de la base de données

G

Cahier de tests

Cahier de test:

## Test sur les liens du site

|                |                              |                                      |  |                        |
|----------------|------------------------------|--------------------------------------|--|------------------------|
|                | Auteurs:<br>Konaté<br>Mariam |                                      |  | Tests du<br>18/03/2019 |
| Spécifications | Spécifications               | Test effectué                        | Résultats attendus   | Commentaire            |
|                | Gestion du menu principal    |                                      |  |                        |
|                |                              | Je clique sur le lien "Accueil"      | La page d'accueil apparaît   | OK                     |
|                |                              | Je clique sur le lien "Projet"       | Le menu déroulant apparaît contenant "Enseignant" et "Etudiant"  | OK                     |
|                |                              | Je clique sur le lien Enseignant     | Si l'enseignant ne s'est pas connecté on atterrit sur sa page d'authentification.                          | OK                     |
|                |                              | Je clique sur le lien Enseignant     | Si l'enseignant est connecté on atterrit sur la page affichant les enseignements.                          | OK                     |
|                |                              | Je clique sur le lien Etudiant       | Si l'étudiant ne s'est pas connecté on atterrit sur sa page d'authentification.                            | OK                     |
|                |                              | Je clique sur le lien Etudiant       | Si l'étudiant est connecté on atterrit sur la page affichant les enseignements ayant un formulaire ouvert. | OK                     |
|                |                              | Je clique sur le lien Administrateur | S'il ne s'est pas connecté on atterrit sur sa page   | OK                     |

Figure 1 –

|  |                            |  |   |    |
|--|----------------------------|--|---|----|
|  |                            |  | d'authentification.   |    |
|  |                            | Je clique sur le lien Administrateur   | Si l'administrateur est connecté on atterrit sur la page affichant les enseignements.                         | OK |
|  | Gestion du menu accueil    |  |   |    |
|  |                            | Je clique sur le bouton "recherche" sans rien sélectionner                     | Cela m'affiche une liste des sujets limitée à 20  | OK |
|  |                            | Je clique sur le bouton "recherche" en sélectionnant l'enseignement            | Cela m'affiche une liste de tous les sujets de cet enseignement limitée à 20                                  | OK |
|  |                            | Je clique sur le bouton "recherche" en sélectionnant l'année                   | Cela m'affiche une liste de tous les sujets de cette année limitée à 20                                       | OK |
|  |                            | Je clique sur le bouton "recherche" en sélectionnant l'enseignement et l'année | Cela m'affiche tous les sujets de cet enseignement en l'année sélectionné                                     | OK |
|  |                            | Je clique sur le bouton "recherche" en ayant écrit un texte.                   | Cela m'affiche la liste des sujets ayant une description contenant ce texte et un intitulé contenant ce texte | OK |
|  | Gestion du menu Enseignant |  |   |    |
|  |                            | Je clique sur ouvrir un enseignement.  | Cela m'affiche un tableau avec la liste des sujets avec leurs descriptions et un bouton ouvrir.               | OK |
|  |                            | Je clique sur ouvrir un sujet  | Cela m'affiche un formulaire pour   | OK |

Figure 2 –

|  |                                      |  |  |    |
|--|--------------------------------------|--|--|----|
|  |                                      |  | modifier le sujet  |    |
|  |                                      | Je clique sur le lien Ajout<br>Sujet         | Cela m'affiche un<br>formulaire pour<br>ajouter un sujet                               | OK |
|  |                                      | Je clique sur le lien liste<br>souhais       | Cela m'affiche la<br>liste souhaits pour<br>cet enseignement                           | OK |
|  | Gestion du<br>menu<br>Etudiant       |  |  |    |
|  |                                      | Je clique sur le lien ouvrir<br>enseignement | Cela m'affiche un<br>formulaire pour<br>insérer la clé<br>d'accès                      | OK |
|  |                                      | J'insère la clé                              | Cela m'affiche un<br>formulaire avec les<br>sujets et les scores<br>à mettre.          | OK |
|  | Gestion du<br>menu<br>Administrateur |  |  |    |
|  |                                      | Je clique sur ouvrir un<br>Enseignement      | Cela m'affiche un<br>formulaire pour<br>modifier<br>l'enseignement                     | OK |
|  |                                      | Je clique sur Ajout<br>Enseignement          | Cela m'affiche un<br>formulaire pour<br>ajouter un<br>Enseignement                     | OK |
|  |                                      | Je clique sur Gestion<br>Tables              | Cela m'affiche la<br>liste des tables avec<br>les options d'ajout<br>et de suppression | OK |

#### Test sur les formulaires de connexion

|                |                          |               |                    |             |
|----------------|--------------------------|---------------|--------------------|-------------|
|                | Auteur: Konaté<br>Mariam |               |                    |             |
| Spécifications | Spécifications           | Test effectué | Résultats attendus | Commentaire |
|                | Connexion                |               |                    |             |

Figure 3 –

|  |  |   |   |    |
|--|--|---|---|----|
|  |  | Enseignant<br>:Identifiant et mot de<br>passe correctes je<br>clique sur le bouton<br>valider       | Ma session s'ouvre<br>avec un message<br>"Opération réussie".                                 | OK |
|  |  | Enseignant<br>:Identifiant et mot de<br>passe incorrectes je<br>clique sur le bouton<br>valider     | Ma session ne<br>s'ouvre pas et j'ai<br>un message<br>"Désolé(e)<br>l'opération a<br>échoué". | OK |
|  |  | Etudiant<br>:Identifiant et mot de<br>passe correctes je<br>clique sur le bouton<br>valider         | Ma session s'ouvre<br>avec un message<br>"Opération réussie".                                 | OK |
|  |  | Etudiant<br>:Identifiant et mot de<br>passe incorrectes je<br>clique sur le bouton<br>valider       | Ma session ne<br>s'ouvre pas et j'ai<br>un message<br>"Désolé(e)<br>l'opération a<br>échoué". | OK |
|  |  | Administrateur<br>:Identifiant et mot de<br>passe correctes je<br>clique sur le bouton<br>valider   | Ma session s'ouvre<br>avec un message<br>"Opération réussie".                                 | OK |
|  |  | Administrateur<br>:Identifiant et mot de<br>passe incorrectes je<br>clique sur le bouton<br>valider | Ma session ne<br>s'ouvre pas et j'ai<br>un message<br>"Désolé(e)<br>l'opération a<br>échoué". | OK |

#### Test sur les formulaires de création et de modification

|                |                               |   |             |
|----------------|-------------------------------|---|-------------|
|                | Auteure: Konaté<br>Mariam     |   |             |
| Spécifications | Test effectué                 | Résultats attendus                                    | Commentaire |
| Enseignant     |                               |   |             |
|                | Je clique sur ajout<br>Sujet. | Le formulaire d'ajout<br>de sujet s'ouvre<br>avec les | OK          |

Figure 4 –

|  |                                     |  |    |
|--|-------------------------------------|--|----|
|  |                                     | Formulaire,cle   |    |
|  | Lorsque je modifie un enseignement. | la modification apparaît dans la base de donnée.   | OK |
|  | Je clique ajouter un enseignement.  | Un formulaire de création s'ouvre avec les informations suivantes;<br>Nom,<br>Description,Type d'optimisation,<br>NbMaxScore,<br>NbMaxTotal,<br>Formulaire,cle,<br>Encadrant | OK |

#### Test sur l'optimisation et la notification

|                | Auteure: Konaté Mariam |  |  |             |
|----------------|------------------------|--|--|-------------|
| Spécifications | Spécifications         | Test effectué  | Résultats attendus   | Commentaire |
|                |                        | Importer un fichier de souhaits dans un enseignement.<br>(Enseignement vide) | Un message apparaît avec opération réussie et Les souhaits apparaissent dans la page.      | OK          |
|                |                        | Création de groupe   | Les étudiants fusionnent dans des groupes de tailles définies                              | OK          |
|                |                        | Optimiser  | On nous affiche une page de solutions nous spécifiant si la solution est réalisable ou pas | OK          |
|                |                        |  |  |             |
|                |                        |  |  |             |

Figure 5 –



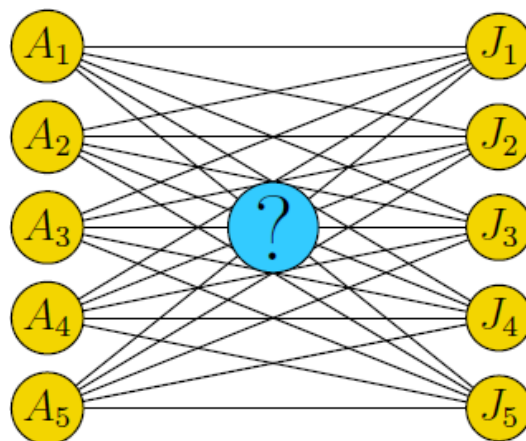
# Affectation des projets au sein du département informatique de polytech Tours

**Mariam Konaté**

**Encadrement : Ronan Bocquillon**

## Objectifs

Polytech Tours propose différents projets aux étudiants pour différents enseignements. Les enseignants proposent des sujets et les étudiants émettent des vœux. Pour améliorer ce processus ce projet vise à automatiser le recueil des sujets et des vœux ainsi que l'automatisation des affectations.



## Mise en œuvre

La recherche consiste à proposer des modèles pour résoudre les problèmes d'affectation liés aux différents enseignements par exemple (une fille par groupe, un étudiant étranger par groupe etc..). Le développement consiste en créer une application web qui résout le problème d'affectation.

PEIP1

Peip1

| Nom  | Prénom | Nombre de sujets min           | Nombre de sujets max            | Action                                     |
|------|--------|--------------------------------|---------------------------------|--|
| ens1 | ens1   | <input type="text" value="1"/> | <input type="text" value="10"/> | <input type="button" value="enregistrer"/> |
| ens2 | ens2   | <input type="text" value="1"/> | <input type="text" value="10"/> | <input type="button" value="enregistrer"/> |

## Résultats attendus

Le résultat attendu est une application complète permettant de faire une affectation qui respecte les objectifs. De ce fait nous avons le résultat de l'affectation avec les contraintes émises par l'encadrant.



# Affectation des projets au sein du département informatique de polytech Tours

Mariam Konaté

Encadrement : Ronan Bocquillon

## Objectifs

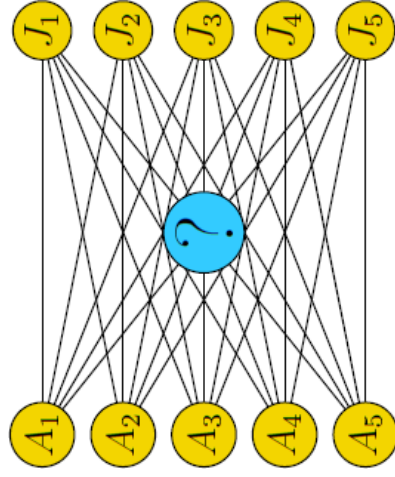
Polytech Tours propose différents projets aux étudiants pour différents enseignements. Les enseignants proposent des suggestions et les étudiants émettent des vœux. Pour améliorer ce processus ce projet vise à automatiser le recueil des sujets et des vœux ainsi que l'automatisation des affectations.

## Mise en œuvre

La recherche consiste à proposer des modèles pour résoudre les problèmes d'affectation liés aux différents enseignements par exemple (une fille par groupe, un étudiant étranger par groupe etc...). Le développement consiste en créer une application web qui résout le problème d'affectation.

## Résultats attendus

Le résultat attendu est une application complète permettant de faire une affectation qui respecte les objectifs. De ce fait nous avons le résultat de l'affectation avec les contraintes émises par l'encadreur.



PEIP1

| Nom  |      | Prénom |      | Nombre de sujets min |   | Nombre de sujets max |    | Action      |             |
|------|------|--------|------|----------------------|---|----------------------|----|-------------|-------------|
| ens1 | ens2 | ens1   | ens2 | 1                    | 8 | 8                    | 10 | enregistrer | enregistrer |
| ens2 | ens2 | ens2   | ens2 | 1                    | 8 | 8                    | 10 | enregistrer | enregistrer |



# Affectation des projets au sein du département informatique de polytech Tours

## Résumé

Projet de recherche et développement visant à résoudre le problème d'affectation concernant les projets au sein du département informatique de polytech Tours.

## Mots-clés

Problème d'affectation

## Abstract

Research and development project aiming at solving the assignment problem concerning projects withing the computer science department at Polytech Tours.

## Keywords

Assignment problem