

# L3 Miage, projet de programmation

## Description

L'objectif de ce projet est de développer une application dédiée à la gestion des compétences d'une entreprise de services et à la gestion des missions de ses personnels. Dans cette entreprise, les compétences de chaque personne sont identifiées suivant un référentiel donné.

Chaque mission doit pouvoir être programmée, avec une date de début et une durée définie. L'entreprise doit alors pouvoir programmer des missions en utilisant au mieux les compétences disponibles parmi son personnel. Le processus de sélection des personnes participant à une mission donnée doit être semi-automatique. C'est-à-dire que le logiciel devra recommander des personnes pour une mission parmi lesquelles ils sera possible de choisir. Le logiciel devra également permettre d'entrer de nouveaux personnels et de mettre à jour les compétences d'une personne lorsque cela est nécessaire.

## 1 Ressources

Les données utilisables par le logiciel sont présentées sous la forme de fichiers csv. Vous disposez :

- D'un fichier `liste_personnel.csv` listant les personnels et leurs caractéristiques.
- D'un fichier `liste_compétences.csv` listant les compétences recensées par l'entreprise
- D'un fichier `compétences_personnel.csv` listant les compétences de chaque personnel de l'entreprise.

Il sera nécessaire d'ajouter des fichiers de données supplémentaires pour la gestion des missions par exemple.

## 2 Principes

Une mission est préparée en amont de son démarrage. Chaque mission est caractérisée par des besoins : pour chaque compétence utile à la mission, un certain nombre de personnes possédant la compétence est requis ; il y a également un besoin en nombre total de personnes. Une même personne peut satisfaire plusieurs besoins d'une même mission si elle dispose de plusieurs compétences adéquates.

La cycle de vie d'une mission est le suivant :

- Lorsqu'elle est créée, une mission est une *mission en préparation*.
- Lorsqu'une mission en préparation est correctement préparée, elle devient alors une *mission planifiée*. Cela intervient lorsque toutes les informations de la mission sont correctement remplies et que des personnels ont été affectés à cette mission de manière à satisfaire tous les besoins. Une *mission planifiée* peut encore être modifiée (par exemple en remplaçant une personne par une autre).
- Lorsque la date de début est passée, une mission planifiée devient une *mission en cours*. À ce stade, la mission n'est plus modifiable.
- Il sera utile de prévoir une notion de mission terminée afin d'anticiper les évolutions du logiciel (par exemple pour une analyse statistique).

Dans certains cas, des missions peuvent provoquer des anomalies. Il est important d'avoir une solution logicielle structurée pour la gestion de ces anomalies.

## 3 Travail à réaliser

Il vous est demandé de réaliser les fonctionnalités suivantes (par ordre d'importance) :

- Un système permettant de charger/sauvegarder les fichiers de personnel et compétences dans l'application.
- Une interface permettant de visualiser/modifier les personnels et les compétences de chacun (il n'est pas demandé de pouvoir ajouter/éditer/modifier les compétences)
- Une interface permettant de saisir de nouvelles missions et de suivre leur cycle de vie.
- Un système de chargement/sauvegarde de l'ensemble des missions.
- Un système de recommandation de personnels pour les missions en préparation où l'opérateur peut valider les personnels choisis pour la mission.

Outre le logiciel, il vous est également demandé de prévoir les documentations suivantes :

- Une diagramme de classes mis a jour.
- La javadoc des classes de votre projet.

## 4 Recommandations

- Le projet que vous réalisez fera l'objet d'évolutions. Il est donc nécessaire de prévoir l'architecture de votre logiciel pour qu'elle soit évolutive.
- L'interface graphique devra être pensée avant le démarrage du code pour permettre une utilisation efficace des fonctionnalités relatives aux missions.
- Le système de recommandation de personnels repose sur un problème d'optimisation non trivial. Il est conseillé de commencer par une version simple qui pourra évoluer par la suite.
- Afin de tester le logiciel, il sera nécessaire de pouvoir avancer le temps dans le logiciel.