

# Projet IdD - INI 1 - 2012/2013

## Gestion électronique de documents

Stéphane Lopes ([stephane.lopes@prism.uvsq.fr](mailto:stephane.lopes@prism.uvsq.fr))  
Laurent Yeh ([laurent.yeh@prism.uvsq.fr](mailto:laurent.yeh@prism.uvsq.fr))

27 mai 2013

L'objectif de ce projet est de réaliser une application de gestion de documents. Ce sujet décrit les fonctionnalités attendues sans toutefois imposer tous les détails. Quand ce n'est pas explicitement imposé par l'énoncé, vous avez toute latitude de décider du choix le plus pertinent pour votre implémentation.

## 1 Gestion électronique de documents

La **gestion électronique de documents** (GED) a pour objectif d'organiser un ensemble de documents informatiques de façon à pouvoir les retrouver et les utiliser de façon efficace.

Les principales fonctionnalités d'un tel système sont l'*acquisition*, le *stockage*, le *classement*, la *recherche* et la *diffusion* des documents. La phase d'acquisition permet d'intégrer un document dans la GED. Le document est alors enrichi avec des *métadonnées* (titre, auteur, date de création, ...). Un fois acquis, le document ainsi que ses métadonnées sont conservés au sein de la GED. Des informations de classement (*tags* par exemple) font parties des métadonnées des documents et permettent de les organiser. L'accès aux documents fait appel à des fonctions de recherche de documents qui sont une partie importante d'une GED. Enfin, la diffusion permet de restituer les documents aux utilisateurs.

De nombreux systèmes de gestion de contenus existent en *open source* comme par exemple [nuxeo](#) ou [Alfresco](#). On peut également citer quelques systèmes plus spécialisés comme [Calibre](#) pour les *e-books* ou [Shotwell](#) pour les photos.

## 2 L'application

L'application à réaliser devra respecter les contraintes ci-dessous.

### 2.1 Acquisition de documents

L'acquisition permettra à l'utilisateur d'ajouter dans le système les fichiers qu'il souhaite gérer. L'utilisateur aura la possibilité de choisir un ou plusieurs fichiers existants sur la machine locale qui seront intégrés à la GED.

### 2.2 Gestion des métadonnées

À chaque document de la GED seront associées des métadonnées que l'utilisateur pourra modifier. Par exemple, on pourra considérer le titre, l'auteur, ... Parmi ces métadonnées, on trouvera une représentation graphique du document (vignette pour une photo, première page pour un pdf, affiche pour un film, ...). Votre application devra supporter un type de document (photo, vidéo, pdf, ...).

### 2.3 Stockage

La GED devra conserver à la fois les documents eux-mêmes et leurs métadonnées. Quand cela est possible, les métadonnées devront être inscrites dans le document lui-même à la manière des données [Exif](#) pour les photos. D'autre part, ces métadonnées seront également stockées dans la GED pour fournir les fonctionnalités ci-dessous.

Le format de stockage des documents et des métadonnées n'est pas imposé. Par exemple, on pourra utiliser le système de fichiers, des fichiers textes, des fichiers [YAML](#) ou [JSON](#), une base de données relationnelle ou XML, ...

---

## 2.4 Organisation

Les documents pourront être organisés selon deux concepts : le *tag* et la *série*. Un tag est un mot-clé caractérisant un document. Un document peut être associé à plusieurs tags et un tag peut être attribué à plusieurs documents. Une série est une collection ordonnée de documents. Un document peut se trouver dans plusieurs séries.

D'autre part, les documents pourront être notés par l'utilisateur.

## 2.5 Recherche

Les recherches devront porter sur les métadonnées des documents. Le [tag-browser](#) du logiciel Calibre peut être une source d'inspiration.

## 2.6 Interface utilisateur

À tout moment, la fenêtre principale devra afficher des documents (tous les documents, résultats d'une recherche, séries, ...).

Deux possibilités de présentation seront toujours proposées à l'utilisateur pour les documents. La première est une simple liste sous forme de tableau, la seconde affichera les représentations graphiques des documents.

Enfin, l'utilisateur devra avoir la possibilité d'ouvrir un document à partir de la GED. Par exemple, une vidéo sera lancée avec [VLC media player](#). Pour chaque type de document, la commande d'ouverture devra être paramétrable.

## 2.7 Extensions facultatives

Une ou plusieurs des extensions facultatives ci-dessous pourront être intégrées au logiciel :

**Sauver les recherches** Une requête de recherche pourra être nommée et sauvee afin de pouvoir être réexécutée.

**Recherche plein texte** La recherche est étendue aux contenus textuels (métadonnées ou document) par exemple à l'aide d'une bibliothèque comme [Apache Lucene](#).

**Téléchargement de métadonnées** Pour éviter à l'utilisateur la saisie fastidieuse de toutes les métadonnées, le système proposera de compléter les métadonnées à partir du web. Par exemple, si l'utilisateur ajoute une vidéo représentant un film, il peut préciser son titre et le système complètera en interrogeant le web ([allociné](#) ou [imdb](#) par exemple).

**Autres types de documents** Le système supportera d'autres types de documents possédant des métadonnées différentes.

## 3 Contraintes techniques

Le projet sera réalisé en langage Java. Les fonctionnalités disponibles dans le langage (collections, exceptions, ...) devront être utilisées au mieux. L'interface graphique utilisera la bibliothèque Swing. L'utilisation de bibliothèques tierce est autorisée ([Google Guava](#), ...).

Le code source devra respecter les [conventions de codage de SUN](#). La documentation du code sera conforme à l'outil standard [javadoc](#).

Le projet final devra fonctionner et se compiler en dehors de tout IDE avec l'outil [maven](#). Idéalement, le projet sera fourni sous la forme d'une archive au format jar et « exécutable ». Le code source sera disponible sur un espace [GitHub](#).

## 4 Organisation et évaluation

Ce projet est à réaliser par groupe de 4 élèves. Le suivi du projet se fera sur l'espace GitHub.

Plusieurs livrables seront à rendre **sous forme électronique sur l'espace GitHub** :

**Code source** Le code source de l'application.

**Distribution binaire** Un exécutable de l'application.

## 5 Dates importantes

**29 mai 2012** lancement du projet

**11 juin 2012** démonstration du projet