

Université de Strasbourg
UFR Maths-Informatique

Licence 3 - Semestre 6

IHM : Manageur de tâches

Jonathan HAEHNEL

Strasbourg, le 4 mai 2012

Table des matières

1	Analyse	3
1.1	Structure de données	3
1.1.1	Nomenclature du fichier XML	3
1.1.2	Classe « Task »	4
1.1.3	Classe « ToDoList »	4
1.2	Fonctionnalités intégrées	5
2	Modélisation	6
2.1	Fenêtre principale	6
2.2	Boîtes de dialogue	7
2.2.1	Ajout d'une tâche	7
2.2.2	Modification et suppression d'une tâche	8
2.2.3	Tri et filtrage des tâches	8
2.2.4	Gestions des étiquettes (tags)	8

Chapitre 1

Analyse

1.1 Structure de données

1.1.1 Nomenclature du fichier XML

Afin de conserver la persistance des données, nous utilisons un **fichier XML**. Qt permet de parser facilement ce type de fichier avec la *bibliothèque QXml*.

A chaque ouverture de l'application, on importe les tâches sauvegardées dans ce fichier. De même à chaque fermeture de l'application l'on enregistre dans ce fichier la liste de tâche courante.

Ci-dessous, la structure du fichier XML :

```
1 <html><body>
2 blabla
3 </body></html>
4
5 <todoList>
6   <tags>
7     <tag>NOM DE L ETIQUETTE </tag>
8     {...}
9   </tags>
10  <task>
11    <name>NOM DE LA TACHE</name>
12    <priority>PRIORITE DE LA TACHE</priority>
13    <status>ETAT DE LA TACHE</status>
14    <description>DESCRIPTION DE LA TACHE</description>
15    <deadline>ECHEANCE</deadline>
16    <create>DATE DE CREATION</create>
17    <tags>
18      <tag>NOM DE L ETIQUETTE </tag>
19      {...}
20    </tags>
21  </task>
22  {...}
23 </todoList>
```

On remarque que un liste contient deux types d'éléments :

- **task** : qui représente une tâche, les champs d'une tâche ne sont pas obligatoire. (sauf le nom)
- **tags** : qui représente une liste d'étiquettes utilisée par l'application pour faciliter l'ajout d'une tâche

1.1.2 Classe « Task »

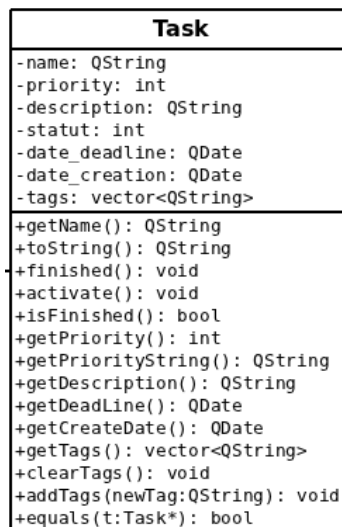


FIGURE 1.1 – UML de la classe « Task »

Un instance de la classe « Tache » contient obligatoirement un nom qui doit être uniquement, cela permet de rechercher une tâche parmi une liste de tâches. Par ailleurs, une tâche contient de nombreux autres attributs non-obligatoire comme une description, un statut (active ou terminée), une priorité et une échéance. On peut également lui affecter une ou plusieurs étiquettes (tags), matérialisé par un vecteur de chaînes de caractères.

Par défaut, lors de la création d'une tâche, celle ci possèdent une date de création et elle est toujours active.

1.1.3 Classe « ToDoList »

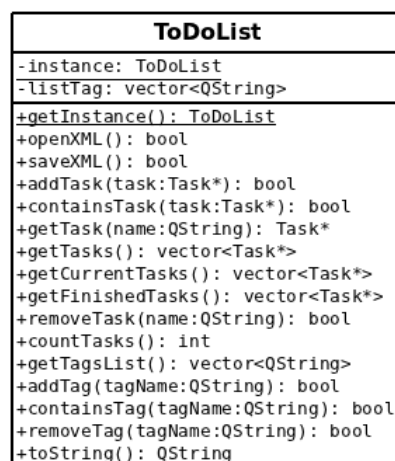


FIGURE 1.2 – UML de la classe « ToDoList »

Cette classe singleton (à instance unique) représente *une liste de tâche*. Celle-ci contient un *vecteur de tâches*, *mais aussi un vecteur d'étiquettes* (chaînes de caractères). L'application peut donc proposer un ensemble d'étiquettes sans qu'aucune des tâches n'y soit affectée. Je suis conscient que cela créer un peu de redondance dans certains cas, mais bon, cela facilite grandement le développement et ne ralentit pas cette petite application.

De plus, cette classe permet de **manipuler le fichier XML d'importation/-exportation** des tâches grâce aux méthodes : `openXML()` et `saveXML()`.

1.2 Fonctionnalités intégrées

Toutes les **fonctionnalités obligatoires ont été implémentées**. Je n'ai pas malheureusement pas intégrés certaines fonctions facultatives comme le tag cloud¹ ou l'import/export de tâches.

Voici, un petit rappel, des fonctionnalités intégrées dans l'application :

Description de la fonctionnalité	Type
Création de la structure représentant une liste de tâches	Core
Ouverture et sauvegarde des tâches dans un fichier XML	Core
Création d'une tâche	GUI
Édition et suppression d'une tâche	GUI
Affectation d'un ensemble d'étiquettes à une tâche	GUI
Ajout et suppression d'une étiquette	GUI
Trier les tâches (par défaut, triage par échéance)	GUI
Filtrage des tâches par un ensemble d'étiquettes	GUI
Séparation des tâches en cours et celles terminées	GUI

TABLE 1.1 – Liste des fonctionnalités

1. Le nuage de mots-clefs est une représentation visuelle des mots-clefs les plus utilisés. Généralement, les mots s'affichent dans des polices de caractères d'autant plus grandes qu'ils sont utilisés ou populaires

Chapitre 2

Modélisation

2.1 Fenêtre principale

J'ai décidé de réaliser une interface graphique simple, efficace et surtout ergonomique pour l'utilisateur. La *fenêtre principale permet de visualiser la liste des tâches*. Les tâches actives et terminées sont contenues dans une table (QTableWidget) et séparées par un composant à onglets (QTabWidget). On remarque aussi une barre de menu et une barre d'outils qui permettent d'effectuer toutes les actions sur ces tâches.

En fermant l'application et, si nous avons oublié d'enregistrer de nos modifications, une boîte de dialogue de sauvegarde apparaît. Tout au long de l'application, l'utilisateur peut être guidé par des infos-bulles en passant sur des composants. Je vais maintenant passer à la présentation des différentes boîtes de dialogues.

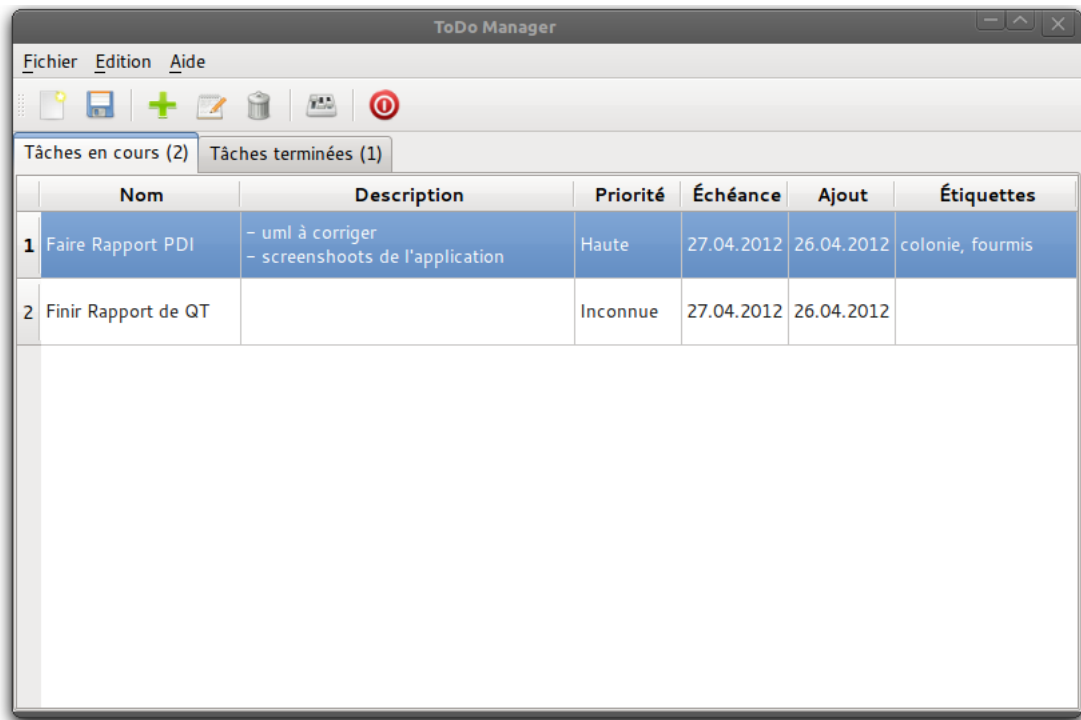


FIGURE 2.1 – Aperçu de la fenêtre principale de l'application

2.2 Boîtes de dialogue

2.2.1 Ajout d'une tâche

La boîte de dialogue permettant de créer une nouvelle tâche est séparée en deux parties à l'aide d'un composant à onglets : **le choix des informations de la tâche et l'affectation des tags**. Seul le nom est obligatoire. Le choix de l'échéance est réalisé grâce à un calendrier et l'affectation des tags se fait dans une liste à choix multiples.

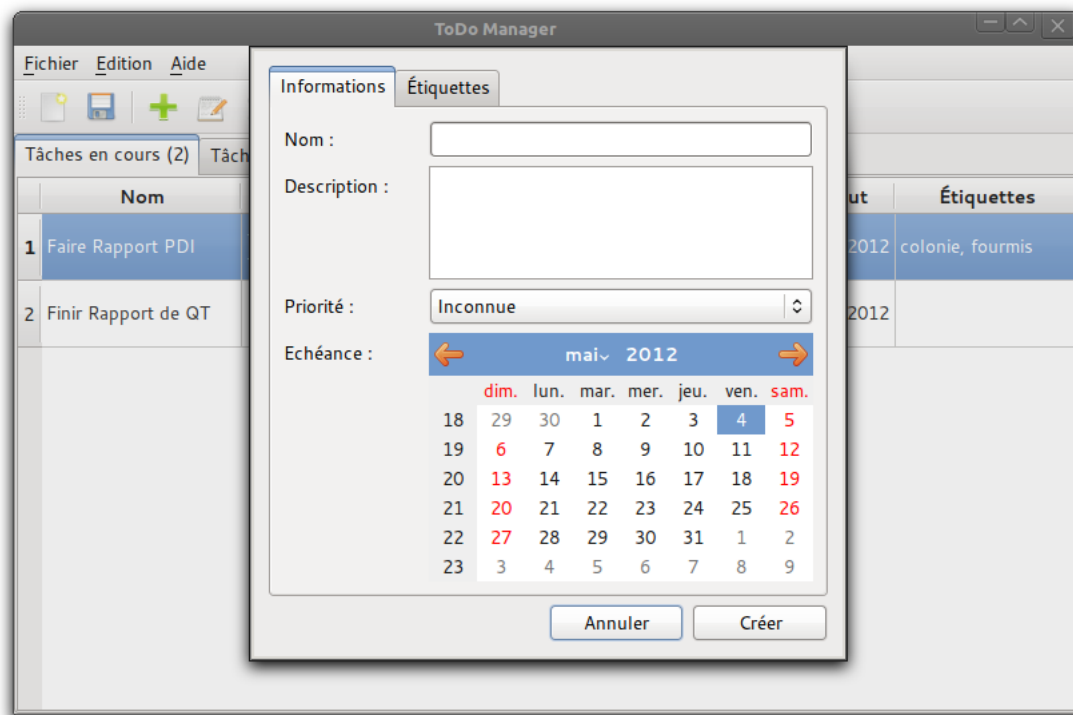


FIGURE 2.2 – Boîte de dialogue de création d'une tâche

2.2.2 Modification et suppression d'une tâche

La boîte de dialogue de modification s'ouvre en cliquant soit sur l'icône de la barre d'outils, soit en double-cliquant sur la tâche à modifier. Cette dernière ressemble fortement à une **boîte de dialogue de création pré-remplie**. Pour la suppression d'une tâche, il suffit de cliquer sur l'icône de la barre d'outils.

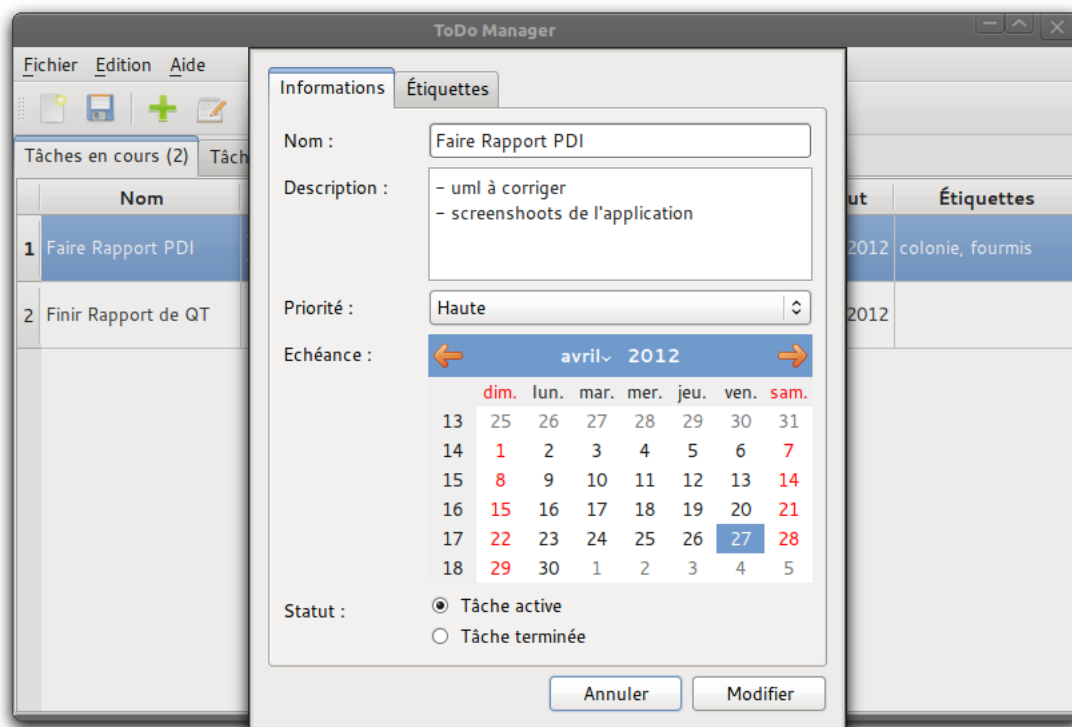


FIGURE 2.3 – Boîte de dialogue d'édition d'une tâche

2.2.3 Tri et filtrage des tâches

Le tri se fait directement en cliquant sur les noms des colonnes du `QTableWidget`. La boîte de dialogue de filtrage se lance en cliquant sur le menu correspondant ou sur l'icône dans la barre de d'outils. Les modifications sont directement prises en compte dans la fenêtre principale.

2.2.4 Gestions des étiquettes (tags)

Cette boîte de dialogue permet d'ajouter et de supprimer des mots clés, les modifications ne sont que prises en compte dès que l'on clique sur le bouton « Appliquer ».

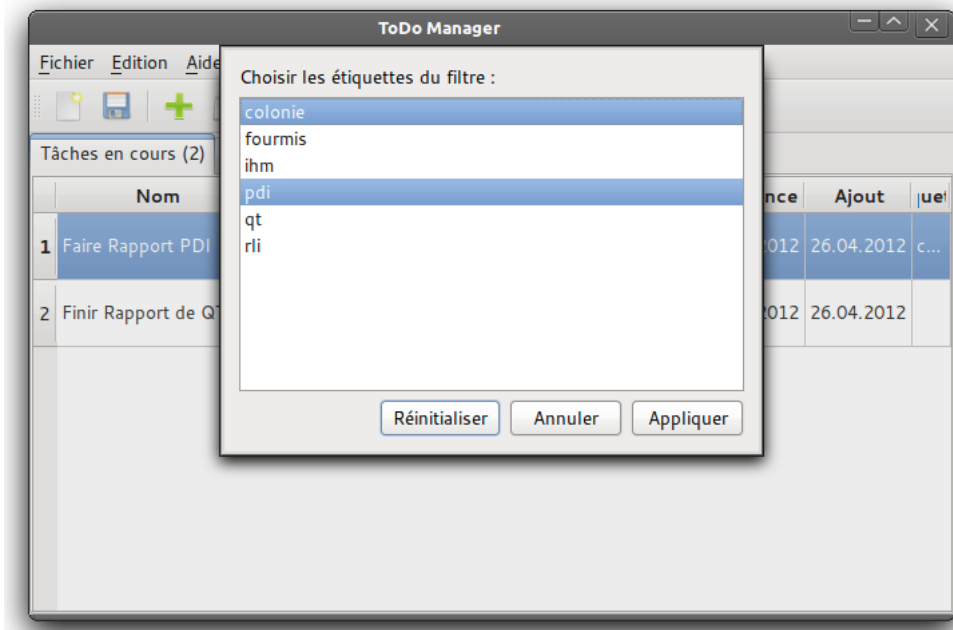


FIGURE 2.4 – Boite de dialogue de filtrage

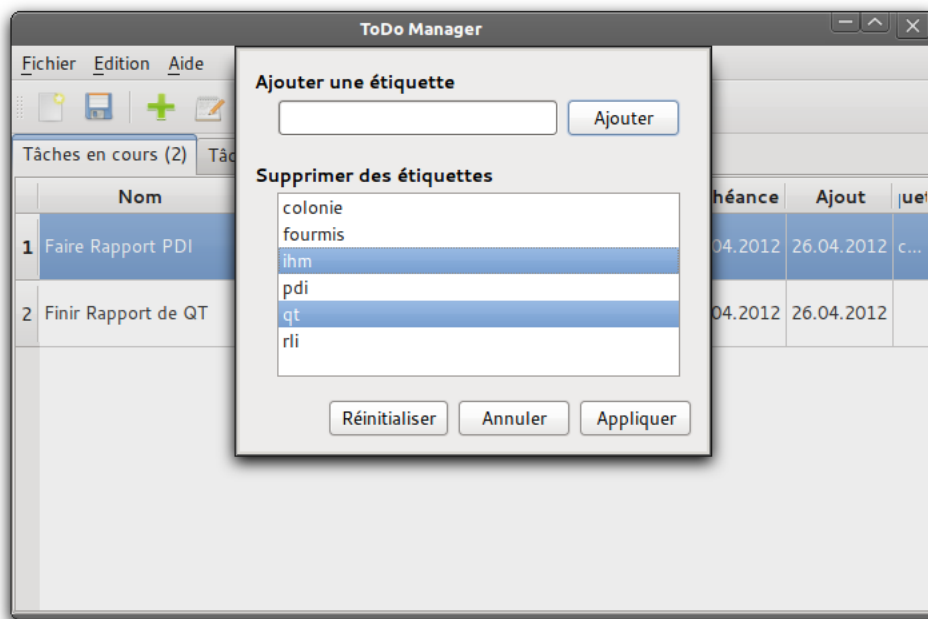


FIGURE 2.5 – Boite de dialogue de gestion des tags