

RAPPORT DE PROJET

Projet : Gestionnaire de listes de tâches

COUTABLE Guillaume, RULLIER Noémie
6 février 2013

Table des matières

1	Introduction	2
2	Les fonctionnalités	3
2.1	Fonctionnalités principales	3
2.2	Fonctionnalités secondaires	3
3	Storyboard, Scénarios, PaperPrototype	4
3.1	Storyboard	4
3.2	Scénarios	4
3.2.1	Scénario 1 - Création/Utilisation/Suppression des listes	4
3.2.2	Scénario 2 - Utilisation des templates	4
4		5
5	Conclusion générale	6

1 Introduction

L'objectif de ce projet fut de développer un gestionnaire avancé de tâches. Celui-ci devait permettre de créer des listes de tâches et de suivre l'avancement de celles-ci.

Afin de créer cette application que nous avons appelé *Taskinator*, nous avons dû établir plusieurs étapes d'avancement du projet. Ce rapport présentera ces étapes les unes après les autres (même si lors de ce projet certaines étapes se sont croisées).

2 Les fonctionnalités

La première étape fut d'analyser l'ensemble des fonctionnalités que notre application devait proposer.

2.1 Fonctionnalités principales

Voici dans un premier temps les fonctionnalités principales :

Créer une liste : cette fonctionnalité permet à l'utilisateur de créer une liste vide.

Créer une liste ordonnée : cette fonctionnalité permet à l'utilisateur de créer une liste ordonnée vide. L'ensemble des éléments de cette liste doivent être effectués dans un ordre précis.

Créer une tâche : cette fonctionnalité permet à l'utilisateur de créer une tâche.

Supprimer un élément : cette fonctionnalité permet de supprimer une tâche ou une liste (ordonnée ou non). Cette fonctionnalité est à manipuler avec précaution, en effet dans le cas d'une liste, la suppression de celle-ci implique aussi la suppression de tous ses éléments (listes ou tâches).

Enregistrer : cette fonctionnalité permet à l'utilisateur d'enregistrer sa liste dans un document sur son disque dur.

Enregistrer un template : cette fonctionnalité permet à l'utilisateur d'enregistrer la liste qu'il vient de créer comme un template afin que la structure de celle-ci soit réutilisable.

Ouvrir un template : cette fonctionnalité permet à l'utilisateur de créer une liste à partir d'un template enregistré. Il devra cependant renseigner le nom de cette liste ainsi que toutes les dates de tous les éléments. Il peut ensuite continuer à modifier cette liste.

2.2 Fonctionnalités secondaires

Voici les fonctionnalités secondaires :

Paramètre : cette fonctionnalité permet à l'utilisateur de modifier le type de l'élément sélectionné. Il pourra par exemple choisir de modifier une liste en liste ordonnée ou en une tâche. Cette fonctionnalité est à manipuler avec précaution, en effet si l'utilisateur décide de transformer une liste en tâche l'ensemble des éléments de la liste seront supprimés.

Monter / Descendre : cette fonctionnalité permet de monter ou descendre un élément dans l'arborescence de la liste. Dans le cas d'une liste, tous ces éléments sont aussi montés/descendus d'un rang. Si le changement se fait au sein d'une liste ordonnée l'ordre des éléments est aussi changé. A chaque déplacement, une vérification de la cohérence des dates est effectuée et l'utilisateur en est informé.

Historique : cette fonctionnalité permet d'annuler ou rétablir des actions faites par l'utilisateur.

Gérer ses templates : cette fonctionnalité permet à l'utilisateur de supprimer les templates qu'il a enregistrés.

3 StoryBoard, Scénarios, PaperPrototype

3.1 StoryBoard

3.2 PaperPrototype

3.3 Scénarios

Afin de créer et tester notre PaperPrototype, nous avons créer plusieurs scénarios.

3.3.1 Scénario 1 - Création/Utilisation/Suppression des listes

Ce premier scénario a été utilisé pour tester le paperPrototype. Il permet de créer une liste non ordonnée et d'y ajouter une liste ordonnée avec ses propres tâches et des tâches.

1. L'utilisateur choisit quel type de liste il souhaite créer, il choisit ici une liste non ordonnée. Il lui donne un nom et une date de fin. (Pour notre scénario cette liste sera appelée la liste mère)
2. Il crée ensuite une liste ordonnée (qui est le premier élément de la liste mère). Il lui donne un nom et une date. (Pour notre scénario cette liste sera appelée la liste 1)
3. Il crée ensuite une tâche qui sera le premier élément de la liste 1. Il lui donne un nom et une date. (Pour notre scénario cette tâche sera appelée la tâche 1.1)
4. Il crée ensuite une tâche qui sera le deuxième élément de la liste 1. Il lui donne un nom et une date. (Pour notre scénario cette tâche sera appelée la tâche 1.2)
5. Il souhaite maintenant échanger les tâches 1.1 et 1.2, il sélectionne donc la tâche 1.1 et la place au dessous de la tâche 1.2
6. Une popup peut apparaître et informe l'utilisateur que l'action qu'il vient d'effectuer provoque un conflit de date. Le champs date concerné par le conflit prend une couleur orange d'avertissement.
7. L'utilisateur change le type de la liste 1 en une liste non ordonnée.
8. L'utilisateur supprime la tâche 1.2
9. L'utilisateur supprime la tâche 1. Une popup apparaît pour l'avertir que cette suppression supprimera aussi tous les éléments de la liste.

3.3.2 Scénario 2 - Utilisation des templates

Ce scénario permet de créer une liste à partir d'un template. De modifier cette liste et de la réenregistrer comme un nouveau template.

1. L'utilisateur choisit de créer une liste à partir d'un template. Il choisit ici le template qui correspond à la préparation d'un cours, puis donne un nom et une date à sa liste.
2. Il va ensuite pour tous les éléments de ce template donner une date.
3. L'utilisateur va ensuite modifier cette liste en y ajoutant une tâche à la liste mère (il lui donne un nom et une date).
4. Il va ensuite vouloir enregistrer cette nouvelle liste comme un nouveau template.

4

5 Conclusion générale

Ce projet nous a permis de contribuer au Web Sémantique en apportant notre propre lot de données ouvertes prêtes à être utilisées. Ce travail nous a montré que chacun pouvait contribuer au Cloud of Linked Data et participer de façon à obtenir une base de données géante et présente sur le web.