



Université du Québec  
École de technologie supérieure

## Département de génie logiciel

### Rapport de laboratoire (Travail T3) : ToDo.Any

Cours	LOG350
Session	Automne 2012
Groupe	01
Numéro de votre équipe	
Chargé(e) de laboratoire	
Étudiant(s)	Grondin, Simon Desharnais, Martin Destrempes, Marc-André
Adresse(s) de courriel	
Code(s) permanent(s)	GROS15059104 DESM21099102 DESM14119009
Date	6 décembre 2012

## CONSIGNES POUR LA RÉDACTION

Le format de ce document doit être respecté.

Remplissez complètement l'en-tête précédent.

Ne touchez pas à la grille de correction qui suit.

Ne modifiez ni l'ordre ni les titres des sections, n'omettez aucune section. Vous pouvez reformater les titres si vous voulez utiliser une table des matières et créer des sous-titres pour les sous-sections de votre travail (fortement suggéré).

Citez vos sources, c'est à dire mentionnez explicitement l'origine de vos idées, algorithmes, exemples, figures, etc., et ce quelle que soit la langue d'origine. Voir le lien *plagiat*, dans la page *laboratoires et travaux pratiques*).

Pour la présentation des textes, figures, tables et références, utilisez les *normes de rédaction des mémoires de maîtrise* (disponibles sur le site de l'ÉTS).

Seule exception : ne mettez pas de table des matières, ni de listes d'abréviations, de figures ou de tables.

## CRITÈRES DE CORRECTION

Dans ce laboratoire, tous les critères seront appliqués strictement (orthographe, gabarit, etc.)

Voici quelques points pour vous aider à rédiger de bons rapports.

1. Gabarit. Dans les sections, effacer toutes le texte venant du gabarit (remplacez les par votre texte!).
2. Rédaction et orthographe. On retire des points pour fautes de frappe, d'orthographe, de grammaire, et de rédaction : phrases trop longues, tournures maladroites ou embrouillées, texte incompréhensible. Relisez vous avant de livrer!
3. Figures et tables : pas de figures ou de tables dans la section introduction ni dans la section interprétation et discussion. Légendes obligatoires. Attention : on doit pouvoir comprendre parfaitement la figure à l'aide de la légende sans revenir au texte.
4. Les figures viennent en complément du texte mais ne le remplacent pas. *Le texte doit être clair et compréhensible sans les figures.*
5. Sur le contenu : tout ce que vous écrivez doit être 1) objectif (neutre, non biaisé par des préférences personnelles, etc.). 2) critique (pas de louange excessive, et aller au fond des choses). 3) justifié (par des résultats, des faits) 4) informatif (clair, et pas redondant, c'est à dire pas de redite, ni de phrases qui n'apportent rien).
6. Style : évitez le "je" et les tournures informelles du langage parlé. Les paroles s'envolent, les écrits restent.

## GRILLE DE CORRECTION

1 Introduction, avec sommaire de T2 /0.5	
2 Planification du travail pour T3 /0.5	
3 Réalisation du prototype dynamique /4	
4 Démonstration du prototype dynamique en laboratoire /4	
5 Tests utilisateurs - évaluations /6	
6 Changements recommandés à l'interface /4	
7 Conclusion /1	
Total partiel /20	
<b>Points négatifs – rapport</b>	
Références (-10% max)	
Orthographe et grammaire (-10% max)	
Présentation (-10% max)	
Retard (-10% par jour)	
<b>Note du rapport / 20</b>	

## TABLE DES MATIÈRES

	Page
Glossaire .....	i
CHAPITRE 1 INTRODUCTION ET SOMMAIRE DU TRAVAIL EFFECTUÉ EN TP2	1
CHAPITRE 2 PLANIFICATION DU TRAVAIL.....	2
CHAPITRE 3 RÉALISATION DU PROTOTYPE DYNAMIQUE.....	3
3.1 Choix des outils .....	3
3.2 Fenêtres développées .....	4
3.2.1 Fenêtre principale .....	4
3.2.2 Événement.....	5
3.2.3 Tâche .....	6
3.2.4 Priorités .....	7
3.3 Justification des choix de conception .....	9
3.4 Améliorations possibles .....	11
CHAPITRE 4 DÉMONSTRATION DU PROTOTYPE DYNAMIQUE EN LABORA- TOIRE .....	12
CHAPITRE 5 TESTS AVEC UTILISATEURS .....	13
5.1 Méthodologie .....	13
5.2 Liste des tâches .....	14
5.3 Liste des utilisateurs .....	15
5.4 Résultats.....	16
5.5 Discussion et recommandations.....	18
CHAPITRE 6 CHANGEMENTS RECOMMANDÉS .....	20
6.1 Améliorer la barre d'ajout rapide .....	20
6.2 Simplifier le champ complétion .....	21
6.3 Simplifier la barre de tags .....	21
6.4 Ajouter des infobulles .....	22
6.5 Ajouter le support du glisser & déposer .....	23
CHAPITRE 7 CONCLUSION .....	24
ANNEXE I DOCUMENT DE PRÉSENTATION DU PROJET .....	26
ANNEXE II FORMULAIRES DE CONSENTEMENT SIGNÉS .....	36
ANNEXE III NOTES PRISES.....	37
ANNEXE IV TRAVAIL PRATIQUE 2.....	38

## CHAPITRE 0

### GLOSSAIRE

**C#** est un langage de programmation orienté objet à typage fort, créé par la société Microsoft.<sup>1</sup>

**Windows Presentation Foundation (WPF)** est la spécification graphique de Microsoft .NET 3.0. Il intègre le langage descriptif XAML qui permet de l'utiliser d'une manière proche d'une page HTML pour les développeurs.<sup>2</sup>

**WinForms** est le nom de l'interface graphique qui est incluse dans .NET Framework, fournissant l'accès via du Managed code à l'API Windows.<sup>3</sup>

**framework** est un kit de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).<sup>4</sup>

**.NET 4.5** est un framework pouvant être utilisé par un système d'exploitation Microsoft Windows et Microsoft Windows Mobile depuis la version 5 (.NET Compact Framework).<sup>5</sup>

**SQLite** est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL.<sup>6</sup>

**Git** est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, le créateur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.<sup>7</sup>

**DataGridView** est un élément graphique de type grille (table) qui est connecté directement sur la base de données.

**LaTeX** est un langage et un système de composition de documents créé par Leslie Lamport en 1983. Plus exactement, il s'agit d'une collection de macro-commandes destinées à faciliter l'utilisation du « processeur de texte » TeX de Donald Knuth.<sup>8</sup>

---

1. [https://fr.wikipedia.org/wiki/C\\_sharp](https://fr.wikipedia.org/wiki/C_sharp)

2. [https://fr.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](https://fr.wikipedia.org/wiki/Windows_Presentation_Foundation)

3. [https://fr.wikipedia.org/wiki/Windows\\_Forms](https://fr.wikipedia.org/wiki/Windows_Forms)

4. <https://fr.wikipedia.org/wiki/Framework>

5. [https://fr.wikipedia.org/wiki/Framework\\_.NET](https://fr.wikipedia.org/wiki/Framework_.NET)

6. <https://fr.wikipedia.org/wiki/SQLite>

7. <https://fr.wikipedia.org/wiki/Git>

8. <https://fr.wikipedia.org/wiki/LaTeX>

## **CHAPITRE 1**

### **INTRODUCTION ET SOMMAIRE DU TRAVAIL EFFECTUÉ EN TP2**

L'application permet la gestion simple de tâches et d'événements tout en offrant des fonctionnalités de gestion et de classification avancées. Parmi celles-ci, notons la possibilité de définir des rappels, de catégoriser les éléments, de définir un événement comme échéance d'une tâche ou encore de définir des sous-tâches à une tâche imposante.

Durant notre analyse de tâches, nous avons déterminé que le public cible se compose de personnes de 15 ans et plus désirant organiser leur temps et leur vie professionnelle. De plus, nous avons conclu que notre interface devrait paraître familière et garder une courbe d'apprentissage faible. Pour se donner un point de départ, nous avons ainsi dressé une liste des cas d'utilisation que l'application doit traiter. Par la suite, nous avons construit nos prototypes statiques.

Nous avons utilisé des fenêtres virtuelles où chaque point de chaque fenêtre a été détaillé pour s'assurer que le travail devant être effectué par chaque interface a bel et bien été compris. De plus, chaque fonctionnalité a aussi été détaillée dans le but qu'elles soient bien comprises.

Jusqu'à présent, presque aucune modification n'a été apportée entre les prototypes statiques et dynamiques sauf à quelques endroits pour des raisons de manque de temps et de personnel. Par exemple, dans la fenêtre Tâche, la section échéance a été changée. Pour le reste, tout va rester comme indiqué dans le document se trouvant à l'annexe III.

Dans le présent document, les tests qui seront effectués avec des utilisateurs sont listés et détaillés dans ce document. Par la suite, un consensus est fait pour chaque tâche et une amélioration possible est proposée pour améliorer le comportement du logiciel.

## CHAPITRE 2

### PLANIFICATION DU TRAVAIL

Semaine	Travail accomplis
4 novembre	Choix de la technologie et apprentissage de celle-ci.
11 novembre	Marc-André : Conception des interfaces. Martin : Conception de la base de données. Simon : Conception de la base de données.
18 novembre	Marc-André : Début de la rédaction du rapport. Martin : Programmation du prototype dynamique. Simon : Programmation du prototype dynamique.
25 novembre	Marc-André : Rédaction du rapport. Martin : Programmation du prototype dynamique. Simon : Programmation du prototype dynamique.
2 décembre	Marc-André : Rencontre avec les utilisateurs. Martin : Préparation de la présentation. Simon : Préparation de la présentation.
9 décembre	Marc-André : Rédaction du rapport. Martin : Rédaction du rapport. Simon : Rédaction du rapport.

**Tableau 2.1** Échéancier

## CHAPITRE 3

### RÉALISATION DU PROTOTYPE DYNAMIQUE

Dans la section suivante, nous vous présenterons notre prototype dynamique et donc, avant d'entreprendre la lecture de cette section, nous vous conseillons de lire le document se trouvant à l'annexe III. Ce document contient les informations concernant le prototype statique et toute la logique derrière le prototype dynamique.

#### 3.1 Choix des outils

Pour réaliser le prototype dynamique, nous avons décidé d'utiliser le C# et le Windows Presentation Foundation (WPF) comme langages de programmation. Nous avons choisi ces langages tout simplement parce que ceci nous permet de séparer le code des interfaces et que le WPF va nous permettre de créer des composantes et de les implémenter dans notre interface personnalisée plus facilement.

Comme environnement de développement, nous utilisons Visual Studio 2012 avec le framework .NET 4.5 pour faciliter et accélérer notre développement. Ce framework nous donne des composantes graphiques de base qui nous simplifie la vie en nous permettant d'économiser du temps pour ne pas à avoir à développer des composants de base comme un TextBox ou un Label, par exemple.

De plus, comme base de données, nous utilisons SQLite parce que cela nous permet d'avoir un endroit pour stocker nos données de façon structuré et le tout dans un seul et unique fichier. Ce qui nous permet d'avoir une base de données portable.

Pour assurer la gestion des versions, nous utilisons un logiciel connu sous le nom de Git.

Pour la production de la documentation, nous utilisons le processeur de texte  $\text{\LaTeX}$ .

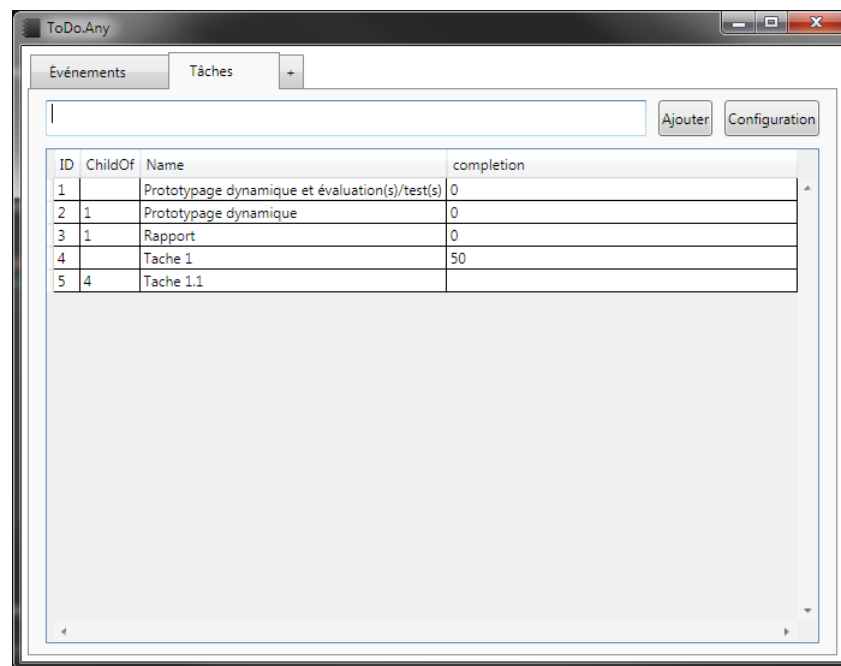


## 3.2 Fenêtres développées

Dans cette section, chaque fenêtre de l'application est décrite brièvement et, par la suite, une justification est apportée pour différents choix de conception.

### 3.2.1 Fenêtre principale

Cette fenêtre comporte deux onglets offrant, pour le premier, une vue sur l'ensemble des événements et, pour le second, une vue sur l'ensemble des tâches. Le principe fondamental de l'application étant de permettre la gestion des tâches et des événements, cette fenêtre occupe la place centrale de l'interface utilisateur.



**Figure 3.1** Fenêtre principale de l'application

Les onglets se présentent tous selon la même mise en page. Une barre comportant un champ texte et deux boutons se trouve dans la partie supérieure alors qu'une grille occupe l'espace restant. La grille permet d'afficher une liste d'éléments de façon hiérarchisée. Chaque ligne représente un élément et chaque colonne représente un champ d'information sur cet élément.

Lorsqu'un clic droit est fait sur une ligne de la grille, un menu contextuel apparaît et offre la possibilité de supprimer l'élément concerné. De plus, double cliquer sur une ligne ouvrira une fenêtre apportant plus d'informations sur l'élément sélectionné.

La barre permet, quant à elle, d'ajouter un élément à la liste. Pour ce faire, il suffit d'entrer le nom du nouvel élément dans le champ texte et d'appuyer sur le bouton « Ajouter ». Le bouton « Configuration » permet, quant à lui, d'accéder aux configurations de l'application <sup>1</sup>.

En plus des deux onglets permettant de visualiser l'ensemble des événements et l'ensemble des tâches, il est possible de créer de nouveaux onglets personnalisés à l'aide du bouton « + » prévu à cet effet. Un onglet personnalisé offre exactement les mêmes fonctionnalités qu'un onglet de base, mais offre en plus la possibilité de filtrer la liste d'éléments affichés selon leur respect de certains critères <sup>2</sup>. Ainsi, il est possible de créer une liste n'affichant que les tâches concernant le travail ou bien encore une liste n'affichant que les tâches dont l'échéance se trouve dans les 48 prochaines heures.

Les critères utilisés pour filtrer la liste d'élément sont demandés lors de la création de la liste.

De plus, si une liste personnalisée n'est plus nécessaire, il est possible de la supprimer en cliquant sur le bouton « X » situé sur l'onglet correspondant.

### 3.2.2 Événement

Cette fenêtre contient les informations détaillées sur un événement. Elle est ouverte lorsque l'utilisateur double-clic sur une ligne de l'onglet « Événements ».

Action	Quand	Attribut du quand

**Figure 3.2** Fenêtre concernant un événement

Le nom est un texte identifiant un événement de façon unique (e.g. « Sortie à la ronde »). L'échéance est une date (e.g. « 2 juin 2013 ») et peut être entrée à l'aide d'un calendrier

1. à l'heure actuelle, la seule configuration possible est la gestion de la liste de priorités.

2. En raison de contraintes de temps, nous n'avons malheureusement pas eu le temps d'implémenter cette fonctionnalité. Elle est présentée ici afin de bien saisir le fonctionnement de l'application, une fois celle-ci parfaitement fonctionnelle.

apparaissant lors d'un clic sur l'icône correspondante à droite. Les tags sont des informations permettant de catégoriser un événement (e.g. « [PERSONNEL][SORTIE] »). La description est un champ texte pouvant contenir plusieurs lignes et paragraphes.

La grille d'alertes permet d'entrer des actions (e.g « Envoyer un courriel ») qui devront être exécutées à un moment précis (e.g « 2 jours » avant la date d'échéance). Pour entrer une nouvelle alerte, il suffit d'entrer les informations dans la ligne vide. Un clic droit sur une ligne fera apparaître un menu contextuel permettant de supprimer l'alerte sélectionnée.

Lorsque l'une ou l'autre des informations est modifiée, le bouton « Enregistrer » est déverrouiller afin de permettre de sauvegarder les nouvelles informations. De plus, si la fenêtre est fermée alors qu'il y a des modifications en suspend, une boîte de dialogue demande s'il faut enregistrer avant de procéder à la fermeture de la fenêtre.

### 3.2.3 Tâche

Cette fenêtre contient les informations détaillées sur un événement. Elle est ouverte lorsque l'utilisateur double-clic sur une ligne de l'onglet « Événements ».

The screenshot shows a window titled 'ToDo.Any - Tâche'. It contains the following fields and sections:

- Nom:** Tâche #1
- Échéance:**
  - ☐ Aucune
  - ☒ Événement: Noël
  - ☐ Date précise: 25/12/2012
- Priorité:** haute
- Complétion:** 50
- Tags:** [ÉTS][LOG350]
- Description:** Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Alertes:** A table with columns ID, Action, and Quand. It is currently empty.
- Sous-tâches:** A table with columns ID, Tâche parent, Nom, Échéance, Priorité, Complétion, and Description. It is currently empty.
- Buttons:** 'Enregistrer' and 'Fermer' at the bottom right.

**Figure 3.3** Fenêtre concernant une tâche

Le nom est un texte identifiant une tâche de façon unique (e.g. « Faire le devoir de MAT265 »). L'échéance peut être un événement (e.g. doit être complété à l'événement « Examen final de MAT265 ») ou encore une date (e.g. doit être complété le « 14 décembre 2012 »). Dans le premier cas, l'événement peut être choisi dans une liste alors que, dans le deuxième cas, la date peut être entrée de la même manière que pour l'échéance d'un événement.

La priorité est un qualificatif permettant de spécifier qu'une tâche est plus importante qu'une autre et peut être choisie dans une liste. Si la priorité souhaitée ne se trouve pas dans la liste, il suffit de cliquer sur l'hyperlien « Priorité : » et la fenêtre permettant de configurer les priorités apparaîtra.

La complétion est un pourcentage indiquant la proportion de la tâche qui est complétée. Les tags sont des informations permettant de catégoriser une tâche (e.g. « [ÉTS][MAT265] »). La description est un champ texte pouvant contenir plusieurs lignes et paragraphes.

La grille d'alerte fonctionne exactement de la même façon que pour un événement.

La grille de sous-tâches permet de séparer une tâche imposante en plusieurs tâches plus petites. Par exemple, la tâche « TP3 de LOG350 » peut être sous-divisée en deux sous-tâches : « Réalisation du prototype dynamique » et « Rédaction du rapport ». Pour créer une nouvelle tâche, il suffit d'entrer les informations dans la ligne vide. Un clic droit sur une ligne fera apparaître un menu contextuel permettant de supprimer la sous-tâche sélectionnée.

De plus, comme les sous-tâches sont des tâches à part entière. Il est possible d'ouvrir la fenêtre de détails d'une sous-tâche en double-cliquant sur la ligne désirée.

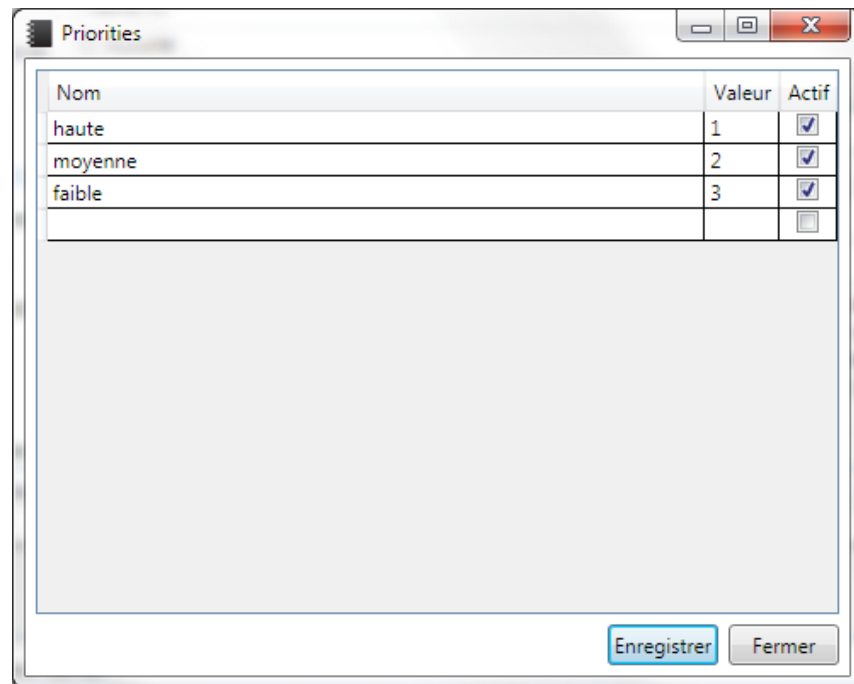
Lorsque l'une ou l'autre des informations est modifiée, le bouton « Enregistrer » est déverrouillé afin de permettre de sauvegarder les nouvelles informations. De plus, si la fenêtre est fermée alors qu'il y a des modifications en suspend, une boîte de dialogue demande s'il faut enregistrer avant de procéder à la fermeture de la fenêtre.

### 3.2.4 Priorités

Cette fenêtre permet de faire la gestion des priorités. Elle est ouverte à partir du bouton « Configuration » disponible dans chaque onglet de la fenêtre principale, ainsi qu'à partir de l'hyperlien « Priorité : » de la fenêtre de détails d'une tâche.

Le nom est un texte identifiant une priorité de façon unique. La valeur est un nombre donnant un poids aux différentes possibilités et permettant de les comparer entre elles. Ainsi, la priorité « Haute », qui a une la valeur un, est plus importante que la priorité « Faible », qui a la valeur trois.

L'attribut « Actif » indique s'il est possible<sup>3</sup> d'utiliser cette priorité lors de la création de nouvelles tâches. Si l'attribut est coché, la priorité apparaîtra dans la liste de priorité de la fenêtre de détails d'une tâche. Si l'attribut n'est pas coché, la priorité n'apparaîtra pas dans la liste, sauf si la tâche utilisait déjà la priorité lorsque celle-ci a été désactivée.



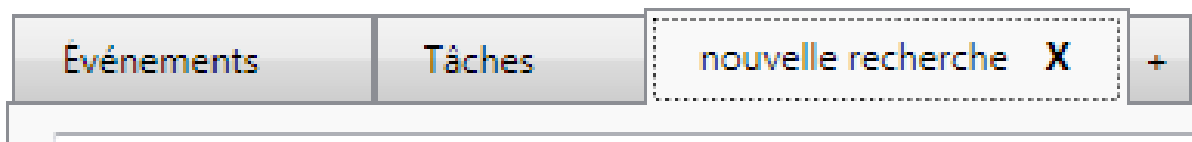
**Figure 3.4** Fenêtre concernant les priorités

Lorsque l'une ou l'autre des informations est modifiée, le bouton « Enregistrer » est déverrouillé afin de permettre de sauvegarder les nouvelles informations. De plus, si la fenêtre est fermée alors qu'il y a des modifications en suspend, une boîte de dialogue demande s'il faut enregistrer avant de procéder à la fermeture de la fenêtre.

### 3.3 Justification des choix de conception

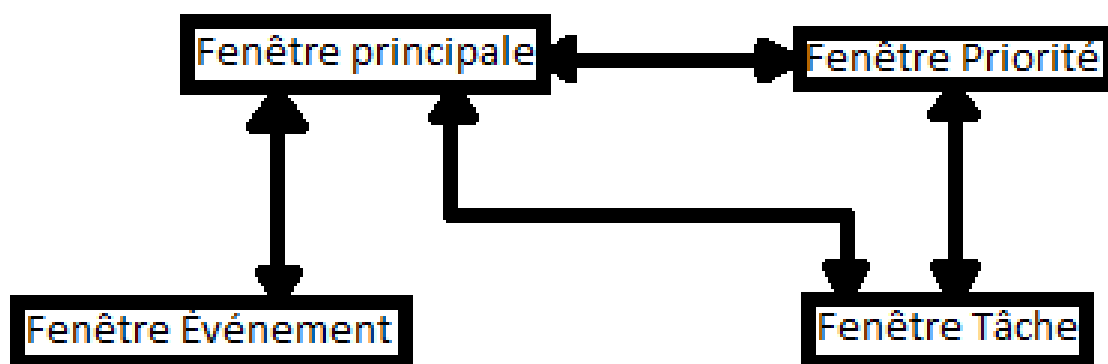
Divers patrons de conception ont été utilisés pour parvenir à la conception de l'application que nous avons présentement.

On compte entre autres le patron *Many Workspaces* qui se retrouve à la fenêtre principale de l'application où l'utilisateur a la possibilité d'avoir plusieurs listes ouvertes et de les organiser comme il veut via les onglets.



**Figure 3.5** Onglets représentant différent espace de travail

Pour la navigation entre les fenêtres de l'application nous utilisons le patron *Escape Hatch*. Ainsi, la navigation entre les fenêtres est limité et l'utilisateur peut toujours revenir a la fenêtre principale sans chercher pendant de longues minutes comment y revenir. Cela simplifie grandement l'interaction entre l'application et l'utilisateur en réduisant le processus de navigation au maximum.



**Figure 3.6** Navigation entre les fenêtres

Ensuite, pour ce qui est des listes, nous utilisons le patron *Tree Table* sur, par exemple, la fenêtre principale. Avec ce patron, nous listons donc les tâches et ses sous-tâches sous la tâche correspondante sous forme d'arborescence. Nous pouvons donc afficher un maximum d'information utile à l'utilisateur lorsque celui-ci le demande. Par manque de temps et de ressources humaines, ce patron a été implanté partiellement.

Nom	Complétion	Description
Prototypage dynamique et évaluation(s)/test(s)	50	Lorem ipsum dolor sit amet, consectetur adipiscing el
>Prototypage dynamique	75	Lorem ipsum.
>Rapport	15	Dolor sit amet.

**Figure 3.7** DataGridView avec le patron *Tree Table*

Nous utilisons aussi le patron *New-Item Row* dans la fenêtre Priorités, par exemple, pour permettre l'ajout rapide et infini de lignes dans le tableau sans que l'utilisateur n'ait besoin d'appuyer sur quoi que se soit.

Nom	Valeur	Actif
haute	1	<input checked="" type="checkbox"/>
moyenne2	2	<input checked="" type="checkbox"/>
faible	3	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

**Figure 3.8** DataGridView avec le patron *New-Item Row*

Pour ce qui est de toutes les grilles, le patron *Sortable Table* s'applique et permet de trier l'information affichée en tout temps pour permettre à l'utilisateur de trouver ce qu'il veut plus rapidement.

Nom	Description
Examen final de LOG240	Lorem ipsum
Examen final de LOG350	Lorem ipsum
Examen final de MAT265	Lorem ipsum
Examen final de PHY332	Lorem ipsum
Noël	Plein de cadeaux!!!
Présentation oral du TP3 de LOG350	Dernière semaine de la session

**Figure 3.9** Exemple d'une colonne trié

De plus, nous avons respecté le plus possible le contenu de la norme *ISO 9241-120* à *ISO 9241-129* qui traite sur les interactions sur les entrées, les sorties et les interactions.

Quelques changements ont été effectués entre les prototypes statique et dynamique. Parmi ces changements, on compte les options possibles pour une échéance dans la fenêtre sur les tâches. Ce changement a été fait dû aux limitations logicielles et par faute de temps. Il est à noter que ces changements n'influencent en aucun cas l'efficacité de l'application.

### **3.4 Améliorations possibles**

Quelques améliorations possibles seraient d'utiliser les lois psychomotrices de Fitts et Miller pour optimiser l'interface. Nous pourrions ainsi optimiser les déplacements que l'utilisateur doit faire avec sa souris pour cliquer sur les boutons et les champs de saisies dans les fenêtres. Par faute de temps et de ressources humaines, il nous est donc impossible d'effectuer ces tests.



## **CHAPITRE 4**

### **DÉMONSTRATION DU PROTOTYPE DYNAMIQUE EN LABORATOIRE**

Un prototype dynamique de l'application est disponible à l'adresse suivante :  
<https://github.com/xeph/LOG350.TP3>.

Évaluation du prototype en laboratoire : le 4 décembre 2012 au laboratoire (pour les 2 groupes).

Exposé oral : le 6 décembre 2012 dans un local à définir (pour les 2 groupes).

Remise du code final : le 11 décembre 2012 à 23h59 maximum par courriel à [log350a2012@gmail.com](mailto:log350a2012@gmail.com).

Remise du rapport : le 11 décembre 2012 à 23h59 maximum en version papier dans la chute du département.

Remise de la présentation de l'exposé oral : le 6 décembre 2012 à 23h59 maximum par courriel à [log350a2012@gmail.com](mailto:log350a2012@gmail.com).

## **CHAPITRE 5**

### **TESTS AVEC UTILISATEURS**

#### **5.1 Méthodologie**

Avant d'effectuer les tests, un document expliquant la nature du projet et le déroulement des tests a été fourni aux utilisateurs. Les membres de l'équipe assument que le document a été lu lors de la rencontre. De plus, un formulaire de consentement sera obligatoirement signé par l'utilisateur et un membre de l'équipe pour que les deux parties s'entendent sur ce qui sera fait lors de la rencontre.

Pour faire les tests avec les utilisateurs, le ou les membres de l'équipe s'asseoiront avec l'utilisateur qui testera l'application. Par la suite, un membre de l'équipe se chargera de dire chaque tâche que l'utilisateur devra effectuer et si l'utilisateur a des questions, il sera chargé de lui fournir les informations. Lorsque l'utilisateur aura terminé de faire une tâche, il y aura une brève pause pour que tout le monde ait le temps de prendre des notes. L'exécution de la tâche suivante se fera lorsque tout le monde aura terminé de prendre les notes pour la tâche courante. Quand toutes les tâches auront été exécutées, l'utilisateur pourra donner son opinion et ses impressions sur l'application, suggérer améliorations, les points positifs et des fonctionnalités qui pourraient être intéressantes à ajouter ou bien améliorer.

Ce seront principalement des notes concernant la performance et l'usabilité de l'application qui seront prises, car les principaux objectifs que nous voulons atteindre avec notre application sont 1- la facilité d'utilisation et 2- l'efficacité. Les notes seront tirées de l'interaction de l'utilisateur avec l'application et de la rapidité avec laquelle l'utilisateur effectue les tâches (l'efficacité).

Les tests se feront dans le calme et le respect. De plus, l'utilisateur peut, en tout temps, décider d'arrêter les tests sans devoir fournir une explication. Les données recueillies seront cependant conservées à moins que l'utilisateur ne demande à ce qu'elles soient détruites.

## 5.2 Liste des tâches

No tâche	Titre et description	Éléments à vérifier et hypothèses
1	Ajouter l'événement « événement 1 ».	Détection de raccourcis.
2	Définir l'échéance de « événement 1 » au 1 janvier 2013.	
3	Ajouter le tag « [TAG1] » à « événement 1 ».	
4	Ajouter une alerte « Envoyer un courriel », « 2 », « jours ».	
5	Enregistrer l'événement.	
6	Fermer la fenêtre.	
7	Supprimer l'événement « Examen final de PHY332 ».	Détection de menu contextuel.
8	Ajouter la tâche « tâche 1 ».	Détection de raccourcis.
9	Définir l'échéance de « tâche 1 » comme étant l'événement « événement 1 ».	
10	Définir la priorité de « tâche 1 » à « Très faible ». <NE PEUT PAS ÊTRE FAIT>	Détection de raccourcis.
11	Ajouter la priorité « très faible » avec une valeur de « 4 » et Actif à « Vrai » et enregistrer.	
12	Définir la priorité de « tâche 1 » à « Très faible ».	L'utilisateur s'attend-il que la liste de priorités ait été automatiquement actualisée ?
13	Définir la complétion de « tâche 1 » à « 42 ».	
14	Ajouter les tags « [TAG1] » et « [TAG2] » à « tâche 1 ».	
15	Ajouter la description « lorem ipsum » à « tâche 1 ».	
16	Ajouter une alerte « afficher un rappel », « 5 », « minutes » à « tâche 1 ».	
17	Créer une sous-tâche « tâche 1.1 » à « tâche 1 ».	
18	Créer une sous-tâche « tâche 1.2 » à « tâche 1 ».	
19	Enregistrer la tâche.	
20	Fermer la fenêtre.	
21	Supprimer la « tâche 1.2 »	
22	Modifier l'échéance de « tâche 1 » pour le « 3 février 2013 ».	

23	Fermer la fenêtre.	L'utilisateur a-t-il pensé qu'il devait enregistrer ?
24	Appuyer sur « Annuler ».	
25	Définir la complétion de la « tâche 1 » à « 50 ».	
26	Fermer la fenêtre.	
27	Appuyer sur « Oui ».	
28	Appuyer sur le bouton « Configuration ».	
29	Définir la priorité « très faible » à « non-actif »	
30	Fermer la fenêtre.	
31	Appuyer sur « Oui ».	
32	Modifier la priorité de la « tâche 1 » à « faible ».	L'utilisateur se remarque-t-il que la tâche est toujours valide même si sa priorité vient d'être désactivée ?
33	Enregistrer la tâche.	
34	Fermer la fenêtre.	

**Tableau 5.1** Liste des tâches

### 5.3 Liste des utilisateurs

Les caractéristiques principales des utilisateurs choisis sont la gestion de leurs tâches pour les travaux et devoirs au CÉGEP ou bien à l'université et la gestion des rendez-vous pour les professionnels. Nous avons choisis ce nombre d'utilisateurs, car il nous fallait un petit échantillon oeuvrant dans le même type de vie que le public visé, mais dans des sphères professionnelles différentes.

## 5.4 Résultats

no Tâche	Points importants de l'observation
1	a ajouté une ligne vide et double cliquer sur l'entrée ensuite
2	
3	
4	a pesé sur $\triangle$ en 1 <sup>er</sup> n'entre pas le 2 (date à la place)
5	
6	
7	ne trouve pas comment supprimer
8	n'utilise pas la barre d'ajout rapide
9	
10	
11	
12	
13	
14	; au lieu de les coller
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	

**Tableau 5.2** Résultats de l'utilisateur 1

no Tâche	Points importants de l'observation
1	n'utilise pas la barre d'ajout rapide
2	
3	
4	a pesé sur $\triangle$ en 1 <sup>er</sup>
5	
6	
7	ne trouve pas coment supprimer
8	ne trouve pas tâche
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	changer le nom au lieu du check box
30	
31	
32	
33	
34	

**Tableau 5.3** Résultats de l'utilisateur 2

## 5.5 Discussion et recommandations

no Tâche	Résumé	Recommandation
1	n'utilise pas la barre d'ajout rapide	mettre la barre plus visible
2		
3		
4	a pesé sur $\triangle$ en 1 <sup>er</sup>	donner plus d'informations sur les expander
5		
6		
7	ne trouve pas comment supprimer	fournir un bouton supprimer
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		

**Tableau 5.4** Recommandations par les observateurs

no Tâche	Résumé	Recommandation
1	n'utilise pas la barre d'ajout rapide	modifier la barre d'ajout
2		
3		
4	a pesé sur $\triangle$ en 1 <sup>er</sup>	donner plus d'information sur les expander
5		
6		
7	ne trouve pas coment supprimer	changer l'endroit ou se trouve supprimer
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		

**Tableau 5.5** Recommandations par les utilisateurs

La barre d'ajout rapide plante dans un nouvel onglet (SearchTasks.xaml.cs), les bugs n'ont pu être reproduits.



## CHAPITRE 6

### CHANGEMENTS RECOMMANDÉS

À la suite des tests avec les utilisateurs, nous avons établi une liste de modifications recommandées afin de rendre l'interface utilisateur plus simple et plus efficace.

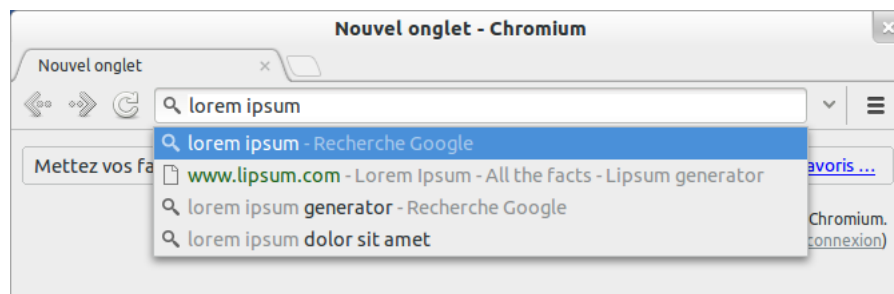
#### 6.1 Améliorer la barre d'ajout rapide

Nous avons observé deux faits étonnants concernant la fenêtre principale : la fonctionnalité d'ajout rapide n'a pas été utilisée et les utilisateurs croyaient que le champ texte servait à rechercher un élément dans la grille. Ceci nous amène à revoir le concept de la barre d'ajout rapide.

Après quelques réflexions et discussions avec les utilisateurs, nous avons décidé qu'il serait mieux de revoir la fonction du champ texte, situé en haut des listes de tâche et d'événements, et le transformer en barre de recherche et d'ajout. Ainsi, lorsque du texte sera tapé à l'intérieur, la grille au-dessous sera automatiquement actualisée pour ne contenir que les éléments dont le nom contenant le texte entré. Ceci permettra donc de retrouver plus rapidement un élément précis lorsque la grille sera pleine.

De plus, dès que l'utilisateur commencera à entrer du texte, une bulle contenant le texte « Créer un nouvel élément portant le nom ... » apparaîtra sous la barre. Si l'utilisateur appuie sur le bouton « entrer » ou clique sur le texte, un nouvel élément sera créé, ajouté à la liste et la barre sera vidée de son texte. Comme l'option d'ajouter une nouvelle tâche n'apparaît que lorsque nécessaire, nous allons enlever le bouton « Ajouter » qui serait redondant.

Ce comportement n'est pas sans rappeler la barre de navigation du logiciel Google Chrome qui, lorsque l'utilisateur commence à y entrer du texte, lance une recherche dans l'historique de navigation, en affiche les résultats et offre la possibilité de lancer une recherche Google.



**Figure 6.1** Barre de navigation de Google Chrome

## 6.2 Simplifier le champ complétion

Nous avons aussi remarqué que les utilisateurs n'étaient pas certains de ce qu'ils devaient entrer dans le champ « Complétion » de la fenêtre « Tâche ». Est-ce un qualificatif, un pourcentage ou bien un nombre quelconque ?

Pour remédier à cette incertitude, nous avons décidé de remplacer le champ texte actuellement présent par un curseur de défilement (mieux connu sous le nom « slider », en anglais). Celui-ci aura la double responsabilité de rendre plus explicite le fait que la complétion est un pourcentage, s'assurera qu'il est impossible d'entrer une valeur erronée et sera beaucoup plus facile d'utilisation.

Pour entrer un niveau de complétion, l'utilisateur pourra simplement glisser le curseur jusqu'à ce que le pourcentage affiché corresponde à la valeur recherchée. D'un autre côté, il aura aussi le choix de garder les mains sur le clavier en utilisant les flèches du clavier pour faire glisser le curseur vers la droite ou la gauche.

Une possibilité supplémentaire qui mérite d'être étudiée est de verrouiller le contrôle et calculer automatiquement le niveau de complétion lorsqu'une tâche à des sous-tâches qui ont elles-mêmes un niveau de complétion. Ainsi, une tâche ayant deux sous-tâches, une complétée à 100 pour cent et l'autre complétée à 50 pour cent, pourrait se faire assigner automatiquement un niveau de complétion de 75 pour cent.



**Figure 6.2** Slider de complétion

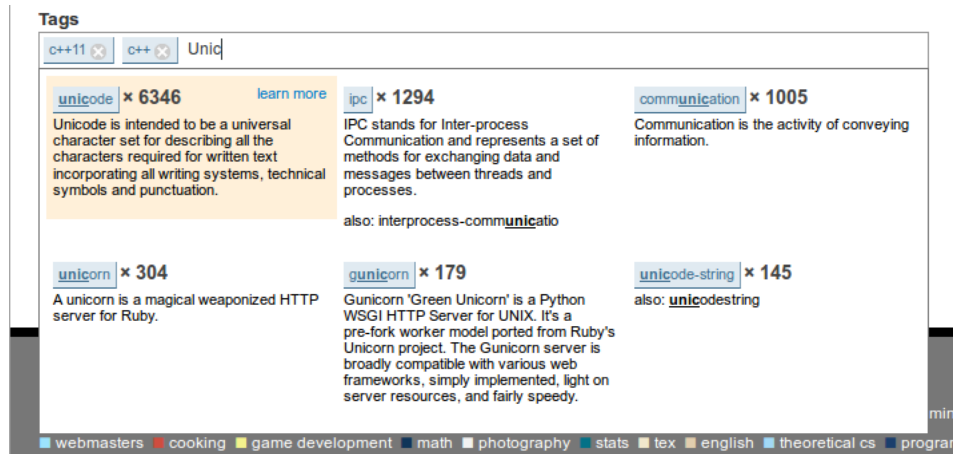
## 6.3 Simplifier la barre de tags

Les utilisateurs ont unanimement affirmé que la façon dont les tags sont actuellement entrés est trop compliquée et sujette à erreurs. Pour rappel, le format actuel est complètement textuel. Un tag est une suite de caractères entourée de crochets et il est possible d'entrer plusieurs tags les plaçant les uns à la suite des autres (e.g. « [ÉTS][LOG350][TP3] »).

Pour faciliter l'entrée de tags, nous avons décidé qu'il serait plus efficace de se baser sur le système d'entrée de tags du site StackOverflow. Le principe est que le champ contenant n'est pas un simple champ texte, mais une liste de tags qu'il est possible de manipuler avec la souris. Ainsi, chaque tag a un petit « x » qui permet de l'enlever de la liste et la liste peut être réordonnée en glissant & déposant les tags les uns à côté des autres.

Cependant, c'est lors de l'ajout d'un nouveau tag que la différence est la plus flagrante. Lorsque l'utilisateur commencera à entrer du texte, une fenêtre apparaîtra et offrira une liste de suggestions basées sur les tags déjà présents dans les autres tâches. L'utilisateur pourra alors cliquer sur le tag de son choix pour le sélectionner. De plus, le premier élément de la liste sera surligné par défaut et sera choisi automatiquement si l'utilisateur appuie sur la touche « entrer ».

Enfin, si l'utilisateur veut entrer un nouveau tag qui n'apparaît pas dans la liste, il suffira d'appuyer sur la barre d'espace et le nouveau tag sera ajouté à la liste.



**Figure 6.3** Barre de tags du site [www.stackoverflow.com](http://www.stackoverflow.com)

## 6.4 Ajouter des infobulles

Lorsque nous avons présenté notre application à différents utilisateurs, nous avons occasionnellement eu à répondre à des questions telles que :

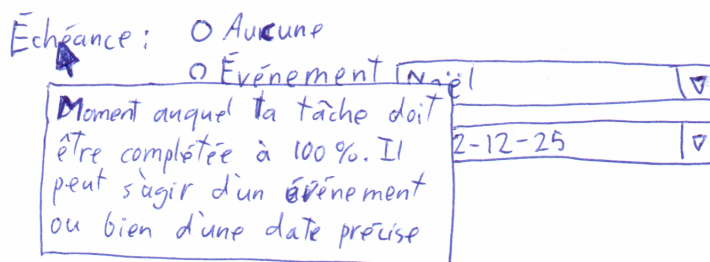
- Qu'est-ce qu'une échéance ?
- Qu'est-ce qu'un tag ?
- À quoi sert une alerte ?

Comme nous nous trouvions juste à côté des utilisateurs, il était facile pour nous de leur donner des explications. Seulement, s'ils avaient été seuls, il aurait pu être difficile pour eux d'obtenir des réponses à leurs questions.

Pour tenter de minimiser la courbe d'apprentissage et d'améliorer la compréhension des utilisateurs, nous avons décidé de placer des infobulles expliquant les différentes informations qu'ils peuvent entrer et les différentes actions qu'ils peuvent poser.

Par exemple, le concept de tag peut sembler étrange à quelqu'un qui n'a jamais été confronté à ce concept. Aussi, le fait qu'un événement puisse être considéré comme l'échéance d'une tâche peut ne pas sembler évident au premier abord.

Les infobulles ne serviront donc pas seulement pour les boutons ne comportant pas de texte, mais aussi pour offrir une explication pour la signification de certains composants.



**Figure 6.4** Infobulle expliquant le concept d'échéance

## 6.5 Ajouter le support du glisser & déposer

L'objectif principal de la fenêtre d'ajout rapide est de permettre de faire une gestion simple de ses tâches sans avoir à entrer dans la fenêtre de détails d'une tâche. Seulement, s'il est actuellement possible de créer autant de tâches que l'on veut à partir de la fenêtre principale, il est nécessaire d'ouvrir la fenêtre de détails afin de pouvoir créer des sous-tâches.

La méthode que nous avons choisie pour remédier à cette situation est d'ajouter le support du glisser & déposer dans la grille de la fenêtre principale. L'objectif est de pouvoir modifier la structure de tâches parent-enfant en glissant les tâches les unes sur les autres.

Ainsi, pour créer la tâche du laboratoire 3 de LOG350, il suffira de :

- 1 Entrer « Laboratoire 3 de LOG350 » dans la barre d'ajout.
- 2 Appuyer sur la touche « entrer ».
- 3 Entrer « Réaliser le prototype dynamique » dans la barre d'ajout.
- 4 Appuyer sur la touche « entrer ».
- 5 Sélectionner la tâche « Réaliser le prototype dynamique » et la glisser sur la tâche « Laboratoire 3 de LOG350 ».
- 6 Entrer « Rédaction du rapport » dans la barre d'ajout.
- 7 Appuyer sur la touche « entrer ».
- 8 Sélectionner la tâche « Rédaction du rapport » et la glisser sur la tâche « Laboratoire 3 de LOG350 ».

## CHAPITRE 7

### CONCLUSION

En conclusion, ce projet a été l'occasion de faire une incartade dans le monde du développement d'interfaces utilisateurs. Nous avons eu l'occasion de participer et d'en apprendre plus sur les différentes étapes de conception : analyse de tâches, prototype statique et prototype dynamique.

D'abord, l'analyse de tâches et le prototype statique nous ont amené à comprendre l'importance de bien connaître son public afin de produire des interfaces utilisateurs qui correspondent réellement à ses besoins. Pour ce faire, nous avons utilisé le vieux principe d'être notre propre public cible. Nous nous sommes demandé quels sont les besoins que nous avons pour gérer notre emploi du temps efficacement et avons tenté de concevoir un système qui réponde à ces exigences. Nous sommes partis du principe que, si nous arrivions à produire un logiciel qui réponde à nos propres besoins, le projet pourrait être considéré comme une réussite.

Par la suite, nous nous sommes demandé si une telle application pouvait être utile à un auditoire plus élargi. Nous avons présenté notre concept et nos idées à des personnes tierces et avons recueilli leurs commentaires et réactions afin de peaufiner quelques détails. Cette expérience nous a confortés dans notre idée que, si des concepteurs connaissent bien leurs utilisateurs et arrivent à penser comme eux, le système produit est plus adapté et plus cohérent.

Ensuite, la réalisation du prototype dynamique nous a appris l'importance, pour un développeur, de bien connaître ses outils avant de s'engager dans la conception d'un système. Au début du projet, nous avons décidé d'utiliser la bibliothèque graphique WPF, une technologie nouvelle pour nous, et nous sommes dit que nous l'apprendrions au fil de l'avancement du projet. Ceci s'est avéré plus difficile que prévu.

Il s'est d'abord avéré assez rapide et facile de faire fonctionner une interface simple utilisant les composants fournis de base. Malheureusement, la situation s'est complexifiée lorsque nous avons voulu implémenter des interfaces nécessitant des composants qui n'existent pas dans la bibliothèque. Toutes les briques nécessaires pour construire de tels composants sont disponibles, mais acquérir les compétences pour développer de nouveaux contrôles WPF prend beaucoup plus de temps que ce que nous avions à notre disposition.

Le meilleur exemple de ces difficultés rencontrées est le système multionglets. Dans la phase d'analyse et de réalisation du prototype statique, nous avons l'intention d'offrir la possibilité de définir des vues plus spécialisées de la liste de tâches dans des onglets supplémentaires. Il aurait alors été possible de créer une vue qui n'affiche que les tâches concernant l'université ou bien uniquement les tâches qui doivent être complétées dans les 48 prochaines heures. Malheureusement, il a fallu tellement de temps pour réussir à implémenter un contrôle permettant à l'utilisateur de créer et supprimer des onglets que nous n'avons pas eu le temps et les ressources nécessaires pour implémenter la fonctionnalité de requête pour configurer une vue.

Enfin, il est ressorti de nos différentes conversations avec des utilisateurs que plusieurs trouvaient l'application visuellement « ordinaire », mais qu'aucun n'avait la même opinion de ce qui permettrait de la rendre plus « belle ». Parmi les membres de notre équipe, aucun ne se considère comme un artiste et ne sait comment créer un arrangement de formes et de couleurs qui plairait à une majorité. Cette constatation met de l'avant le fait qu'il y a une différence importante entre développer une interface efficace et une interface visuellement attirante. Le cours que nous avons suivi tente de nous apprendre à développer la première, mais ne prétend nullement nous apprendre à développer la seconde.

Concernant le cours, nous suggérons que celui-ci donne une plus grande place aux applications dites de bureau. Celles-ci étant encore très présentes comme applications « grand public » et largement dominantes dans le monde professionnel, il serait intéressant qu'elles occupent une plus grande place. Certaines sont très simples d'utilisation alors que d'autres sont incroyablement complexes. Qu'est-ce qui les distingue ? Quelles sont les bonnes habitudes à imiter ? Quelles sont les erreurs à ne pas commettre ?

## **ANNEXE I**

### **DOCUMENT DE PRÉSENTATION DU PROJET**

17 novembre 2012

# Tests avec utilisateurs

Description du projet



## TABLE DES MATIÈRES

	Page
CHAPITRE 1    CONTENU DU DOCUMENT .....	4
CHAPITRE 2    GLOSSAIRE .....	5
CHAPITRE 3    PRÉSENTATION DE L'APPLICATION .....	6
3.1      Description de l'application .....	6
3.2      Contexte du domaine d'application et motivation.....	6
3.3      Description du public cible et des types d'intervenants.....	7
CHAPITRE 4    DESCRIPTION DE L'ÉVALUATION .....	8
4.1      Méthodologies utilisées .....	8
4.2      Déroulement de l'évaluation .....	8
4.3      Exemple .....	8
4.3.1    Fiche de tâches à effectuer.....	8
4.3.2    Fiche de discussion et recommandation .....	9

## LISTE DES TABLEAUX

	Page
Tableau 4.1    Tâches à effectuer .....	8
Tableau 4.2    Discussion et recommandation .....	9

## **CHAPITRE 1**

### **CONTENU DU DOCUMENT**

Le présent document contient une présentation du projet ainsi qu'une description des différentes évaluations : les méthodologies, le déroulement de l'évaluation ainsi qu'un exemple des fiches qui seront utilisées durant l'évaluation.

Ce document se veut à but descriptif afin bien comprendre le contexte de l'application et le but de l'évaluation.

## **CHAPITRE 2**

### **GLOSSAIRE**

**Tâche** Travail à effectuer

**Échéance** Date limite pour effectuer un travail

**Alerte** Rappel avant une date

**Événement** Activité, occurrence à une date donnée

## CHAPITRE 3

### PRÉSENTATION DE L'APPLICATION

#### 3.1 Description de l'application

L'application permet la gestion simple de tâches et d'événements tout en offrant des fonctionnalités de gestion et de classification avancées. Parmi celles-ci, notons la possibilité de définir des rappels, de catégoriser les éléments, de définir un événement comme échéance d'une tâche ou encore de définir des sous-tâches à une tâche imposante.

Le système de classification modulaire permet à l'utilisateur de définir des listes thématiques n'affichant que les informations pertinentes dans un contexte défini. Par exemple, l'on pourrait définir une liste « Université » n'affichant que les tâches non complétées catégorisées comme du travail universitaire ou bien une liste « Tâches urgentes » n'affichant que les tâches non complétées dont l'échéance se trouve dans prochaines 48 heures, indépendamment de leur catégorie.

Ces fonctionnalités avancées ne signifient nullement que l'application est complexe ou réservée aux professionnels. L'application n'oblige jamais d'en faire plus que nécessaire. En fait, si aucune gestion avancée n'est nécessaire, une tâche peut être réduite à un strict minimum : un nom. En effet, il n'est pas très utile de définir une échéance pour la tâche « Acheter du lait et une michette de pain ».

Lors d'une utilisation quotidienne, les tâches et événements sont généralement créés de façon itérative. Par exemple, un étudiant assiste à un cours et l'enseignant lui indique qu'il devra remettre son travail de session le jour de l'examen final. Bien sûr, la date de l'examen ne sera définie que bien plus tard. Qu'à cela ne tienne, notre étudiant crée un événement « Examen du cours de ... » et l'assigne à la catégorie « Université » sans en définir de date. Il crée ensuite une tâche « Travail de session » dans la même catégorie et lui assigne le nouvel événement comme échéance. L'enseignant indique alors que le travail sera constitué de trois parties : une analyse de situation, la réalisation du projet et un rapport. Notre étudiant crée alors une sous-tâche pour chaque partie.

Notre étudiant peut maintenant garder un œil sur son travail de session dans la liste « Université » où il regroupe tout ce qui a trait à ses études.

#### 3.2 Contexte du domaine d'application et motivation

Dans le monde moderne, tout va toujours extrêmement vite : des rendez-vous, des conférences, des fêtes, des travaux à remettre, des entretiens à faire, des pneus à changer, etc. L'homme moderne doit être à même de gérer un très grand nombre d'obligations et de choses à faire. Clairement, il doit y avoir une façon de s'y retrouver dans tout ça. La façon la plus simple est de simplement utiliser sa mémoire. Seulement, cette technique montre rapidement ses limites

lorsque le nombre d'éléments dont il faut se souvenir augmente. Il devient alors nécessaire d'utiliser un outil plus fiable.

Au début une simple liste sur un bout de papier peut faire l'affaire. Vient ensuite l'utilisation d'un calendrier ou encore d'un agenda. Dans des environnements professionnels, des logiciels d'envergures sont utilisés pour soutenir la gestion des choses à faire : on peut penser à Outlook, Trac ou encore Microsoft Project.

Pour beaucoup de gens, la plupart des solutions disponibles sont soit trop simples, soit trop complexes. Le besoin de développer une solution plus puissante tout en restant simple était évident. Pour répondre à ce besoin, nous avons développé un outil léger et efficace qui permet d'offrir une gestion des tâches et événements. Avec cet outil, il est donc possible d'effectuer un suivi des choses à faire et des événements auxquels assister. Lorsqu'une unique liste devient trop volumineuse, il est possible de modifier le classement comme bon nous semble. Par exemple, il peut sembler logique de classer ses tâches par sujet, mais il peut être tout aussi logique de les classer par échéance si ce qui doit être fait d'ici un mois n'est pas pertinent dans le contexte actuel.

### **3.3 Description du public cible et des types d'intervenants**

L'utilisateur cible est une personne cherchant à organiser son temps et sa vie professionnelle de façon simple et efficace. Cette personne doit avoir accès à un ordinateur et connaître les rudiments de base de son utilisation tels que la manipulation de fenêtre et la saisie de données.

## CHAPITRE 4

### DESCRIPTION DE L'ÉVALUATION

#### 4.1 Méthodologies utilisées

Les tests effectués seront les suivants :

##### **psychomoteurs**

Relatif aux fonctions psychiques et motrices <sup>1</sup>

##### **de performance**

Test dont l'objectif est de déterminer la performance d'un système informatique <sup>2</sup>

##### **d'usabilité (tests, enquêtes et focus groups)**

Consiste à observer directement l'utilisateur en train de se servir de l'application <sup>3</sup>

#### 4.2 Déroulement de l'évaluation

Chaque évaluation se déroulera en présence de l'évaluateur et celui-ci dictera les actions qui devront être effectuées. Durant la réalisation du test, l'évaluateur prendra des notes ayant trait aux résultats du test, aux difficultés rencontrées ainsi que toute autre observation jugée pertinente.

De plus, l'utilisateur pourra donner son point de vue sur des améliorations possibles ou bien critiquer les défauts que celui-ci voit dans l'application.

#### 4.3 Exemple

##### 4.3.1 Fiche de tâches à effectuer

Cette fiche contient les tâches que l'évaluateur demandera à l'utilisateur d'effectuer.

No de tâche	Titre et description	Éléments que vous voulez vérifier et hypothèses
1	Ajouter une tâche en utilisant la barre d'ajout rapide.	L'utilisation des raccourcis disponible pour l'utilisateur.
...	...	...

**Tableau 4.1** Tâches à effectuer

1. <http://dictionnaire.reverso.net/francais-definition/psychomoteur>

2. [https://fr.wikipedia.org/wiki/Test\\_de\\_performance](https://fr.wikipedia.org/wiki/Test_de_performance)

3. <http://www.usabilis.com/methode/test-utilisateur.htm>

### 4.3.2 Fiche de discussion et recommandation

C'est dans cette fiche que l'évaluateur noteras les observations effectuées sur l'utilisateur et les recommandations que l'utilisateur mentionne à l'évaluateur.

no Tâche	Points importants de l'observation de A	Recommandations
1	L'utilisateur ne trouve pas comment ajouter.	Rendre l'option plus visible et intuitive.
...	...	...

**Tableau 4.2** Discussion et recommandation



## **ANNEXE II**

### **FORMULAIRES DE CONSENTEMENT SIGNÉS**

## **ANNEXE III**

### **NOTES PRISES**

## **ANNEXE IV**

### **TRAVAIL PRATIQUE 2**