

Cahier des charges

Voici le cahier des charges originel auquel on a ajouté nos choix (*en italique souligné*).

Le projet consiste à réaliser une petite application pour gérer les tâches d'une personne, à la manière d'une "todo list". Une tâche doit être réalisée pour une date donnée (l'échéance), la granularité sera la journée (quand une tâche doit être achevée pour un jour j, c'est sous-entendu qu'elle est due pour le jour j au soir (*on a initialisé nos deadline à 23h59m59s*), elle ne devient donc en retard qu'au jour j+1. L'échéance est modifiable à tout moment tant que la tâche n'est pas réalisée. L'échéance est impérativement postérieure à la date de création de la tâche. Une tâche a un titre et éventuellement une catégorie (voir plus loin), le titre et la catégorie sont modifiables à tout moment tant que la tâche n'est pas réalisée. Si elle n'a pas déjà été réalisée. *Il est possible de la supprimer dans ce cas elle ne sera pas gardée pour le bilan et sera définitivement effacée.*

Nous avons ajouté une importance à chacune des tâches (faible, moyenne et importante) elle est aussi modifiable dans les mêmes conditions que ses autres paramètres.

On distingue deux types de tâches : celles dites ponctuelles, et celles "au long cours".

Les tâches ponctuelles sont atomiques, elles ne peuvent pas être partiellement réalisées. Au contraire, les tâches au long cours sont à réaliser sur une grande période de temps, et on peut régulièrement venir renseigner le pourcentage d'achèvement courant. Pour cela, la granularité sera le pourcent. L'avancement ne peut que croître et n'excédera pas 100. Pour une tâche au long cours, on peut spécifier une date de début (par défaut, la date de début est la date de création de la tâche). La date de début est impérativement antérieure à l'échéance.

Catégories de tâches

Les catégories de tâches sont par défaut : Travail, ou Personnel.

Nous avons ajouté la catégorie "Aucune" c'est une catégorie permanente non modifiable par l'utilisateur contrairement à Travail et Personnel permettant de rassembler toute la tâche sans catégorie.

Une fonctionnalité attendue de l'application est de pouvoir modifier les catégories proposées : on peut en rajouter de nouvelles, renommer ou supprimer les catégories existantes. Dans le cas d'une suppression de catégorie, les tâches qui avaient cette catégorie deviennent sans catégorie (*Aucune*).

Retard de tâche

Une tâche ponctuelle est en retard a partir du jour suivant l'échéance prévue.

Une tâche au long cours est en retard si elle ne progresse pas a peu près linéairement.

Si on nomme d la durée impartie pour une tâche (différence entre son échéance et sa date de début), on vérifie l'avancement a chaque pas de $d/4$: l'avancement doit être au moins de 25% a $d/4$, de 50% a $d/2$, de 75% a $3d/4$ et de 100% a d .

L'heure de la date d'échéance ou de la date d'échéance intermédiaire (tâche au long cours) est à 23h59m59sec passé cette horaire elle sera considéré comme en retard.

Tâches réalisées

Les tâches ponctuelles sont réalisées quand l'utilisateur le spécifie. Les tâches au long cours sont réalisées quand leur avancement passe a 100. Les tâches réalisées disparaissent de la liste des choses a faire, mais sont gardées en mémoire, notamment pour l'édition de bilans.

En effet chaque tâche en cours, réalisée dans les temps ou non son stockées dans une liste permettant au bilan de générer différente information.

Organisation de la liste des choses a faire

On prévoit pour l'instant deux sortes de listes de choses a faire. La première, dite simple, maintient et présente simplement la liste des tâches non réalisées, par ordre de proximité d'échéance croissante pour les tâches dont l'échéance n'est pas échue, et décroissante pour les tâches dont l'échéance est échue (plus une tâche est urgente plus elle est en tête de liste). Cette liste est donc triée par date d'échéance croissante. La deuxième sorte intercale mieux les tâches ponctuelles et les tâches au long cours. On trie ainsi la liste de manière a prendre en compte les échéances intermédiaires (en termes de quart de durée a affectée) des tâches au long cours. Les tâches sont ainsi triées par ordre d'échéance intermédiaire croissante, avec : la date d'échéance intermédiaire d'une tâche ponctuelle est sa date d'échéance, et la date d'échéance intermédiaire d'une tâche au long cours est celle du prochain quart de tâche non achevé. Par exemple, soit une tâche au long cours prévue de $j+1$ a $j+4$, et réalisée a 30%. Au jour j , l'échéance intermédiaire de cette tâche est $j+2$ car c'est le deuxième quart de la tâche qui est a réaliser, le premier est déjà réalise. Une troisième sorte sera développée ultérieurement, gérant l'ensemble des tâches mais n'en présentant que 9 : une importante, 3 moyennes, et 5 plus petites. Dans tous les cas, les tâches en retard seront stigmatisées (par le moyen de votre choix, par exemple grâce a un code couleur). Quand une tâche apparaît dans une liste de tâches, on fait systématiquement figurer la catégorie si elle est présente, le délai associe et le nombre de jours restant avant l'échéance.

On a choisi de mettre un panel qui comporte les informations associées à la tâche sélectionnée (cf manuel d'utilisation pour ce qui est du code couleur des tâches).

Édition de bilans et mémoire des tâches réalisées

Pour l'instant, l'édition de bilan est minimal. Un bilan est réalisé pour une période donnée, et permet de connaître la liste des tâches à réaliser sur cette période (échéance incluse dans cette période), le pourcentage de tâches réalisées dans les temps sur cette période, le pourcentage de tâches réalisées en retard sur cette période, et le pourcentage de tâches non encore réalisées sur cette période alors que dues sur cette période.

Le bilan génère aussi une liste rassemblant toute les tâche (créées et non supprimées) dont l'échéance est incluse dans la période spécifié par l'utilisateur. différente couleur détermine si la tâche est terminée ou non (Voir manuel d'utilisation).

Interfaçage

L'application aura une interface graphique réalisée en Java Swing. Vous êtes laissés libres sur la réalisation de cette interface et sur les interactions entre l'application et son utilisateur. Veuillez a rester simples ! Veuillez également a réaliser une conception propre, et a bien dissocier la partie métier et la partie graphique de votre application, grâce a un modelé MVC.

On a utilisé une JList pour présenter la liste de tâche, et un panel d'information pour afficher les informations relative a chaque tâche.

Nous nous somme permis d'ajouter une librairie (JCalendar v1.4) permettant de manipuler les dates de java dans l'interface graphique plutôt facilement.

Persistance

Les données seront rendues persistante par écriture et lecture dans un fichier. Le format de sérialisations est laisse libre, la sérialisations proposée par Java convient très bien.

Nous avons effectivement utilisée la sérialisations proposée par Java qui sauvegardera les données nécessaires (l'instance courante du manager) dans un fichier nommée "sauvegarde.todo"