

Précisions

Pré-conditions à une tâche :

- pour autoriser l’U à cocher cette tâche il faut avoir coché les tâcheA, tâcheB,...
- ou, on se limite à :
 - ajout de liste ordonnées de tâches
 - ajout d’ensemble de tâches(liste non ordonnée). L’U se débrouille pour spécifier dans ce canevas même si cela engendre des regroupements sans signification précise.

exemple :

1. réserver le bateau
2. préparation
 - sécurité
 - sac perso

Ce que nous appellerons un *template* de liste hiérarchique de tâches correspond à une liste hiérarchique spécifique (exemple : préparation de cours) qui va être instantiée par le nom du module et des dates adaptées.

Qt designer est autorisé Les classes spécifiques à certain widget Qt sont autorisées Les dates sont indicatives (pas de vérification d’antériorité) (on pourra prévenir l’U qu’une tâche dont la date est dépassée n’est pas cochée) Lors d’une tentative de validation d’une tâche, on vérifie que les tâches antérieures sont effectuées, si ce n’est pas le cas, on avise l’U. Vous n’êtes pas obligé de proposer des corrections intelligentes du genre : ”voulez-vous valider les tâches précédentes ?”.

Pour l’édition des dates relatives : on pourra se limiter à des choix multiples (1,2..6 jours, 1,2,3 semaines, 1..3,6 mois)

L’application est prévue pour être mono-utilisateur. Au lancement de l’appli, on retrouve l’état de l’appli quand on l’a quittée (mais pas obligatoirement pour les menus, et fenêtres) Donc dans le fichier XML, on trouvera

- les *template* déjà enregistrés ;
- les listes hiérarchiques de tâches courantes.

Rendu

Les démonstration du logiciel et le rendu du rapport devront impérativement se faire le mardi 19 février (derniers TP d’ IHM1).

Dans le rapport, on se concentrera sur l’aspect IHM du logiciel développé. On y trouvera (à minima) :

- un storyboard ;
- plusieurs scénari d’usage ;
- les fonctionnalités du noyau qui ont gouverné l’interface ;
- les *paper prototypes* scannés avec les arguments qui ont guidés vos choix ;
- un test de ce prototype avec un des scénarios (l’idéal serait de fournir une vidéo de ce test) ;
- la structure de widgets de votre IHM , avec les *signals/slots* correspondants.

Vous fournirez, en outre, une archive contenant les sources et autre makefiles autorisant une reconstruction complète.

Pas besoin de grosse précisions sur l’aspect génie logiciel. Toutefois le code devra être ré-utilisable, donc commenté. La structure de données interne (liste de tâches) expliquée.