Série n° 2 en systèmes d'information, méthodes avancées (CBF)

Description du cas

On veut développer une application de gestion des listes de tâches à faire (*to-do list manager).* Après authentification, l'utilisateur peut créer une tâche en mentionnant son titre, ses dates de début et de fin, sa priorité ainsi qu'un rappel. Le rappel se fait à travers le service de messagerie (email) par la réception d'un message de rappel en début de journées de(s) tâches. Aussi, l'utilisateur peut ajouter une lieu en le localisant sur une carte grâce à un service de cartes (ex : Google Maps), comme il peut ajouter des collaborateurs en mentionnant leur adresses emails. Les collaborateurs reçoivent la notification par email et ils peuvent accepter ou refuser la collaboration. Dans les deux cas, l'utilisateur reçoit un mail dans son service de messagerie.

Lorsque l'utilisateur réalise une tâche, il met à jour son statut. Il peut également supprimer une tâche non accomplie ou reporter/avancer une tâche en changeant sa date de début et/ou de fin. Lors de la mise à jour de la réalisation d'une tâche (statut *réalisée*), l'utilisateur ajoute éventuellement des commentaires (ex : difficultés rencontrées), et calcule le décalage entre dates prévues et réelles.

Pour mieux organiser les tâches, l'utilisateur peut éventuellement les regrouper en projets. Pour cela, il crée d'abord le projet, lui associe une description et les dates de début et de fin, puis il lui affecte les tâches. Pour plus de flexibilité, il est possible de créer une tâche au cours de l'opération d'affectation si la tâche n'existe pas dans la liste. Aussi, de la même manière que pour une tâche, l'utilisateur peut associer une localisation au projet par le service de cartes et ajouter des collaborateurs par leur adresses emails.

La mise à jour de réalisation d'un projet se fait à travers celles des tâches mais l'utilisateur peut toujours changer les dates de début et de fin du projet tout en garantissant l'intégrité du système (dates des tâches incluses dans celles du projet). Toutefois, l'utilisateur peut toujours (dés)associer des tâches. Lorsqu'une tâche est désassociée, le système propose de la supprimer complètement ou de la garder dans la liste, et elle devient non associée. A la fin d'un projet, l'utilisateur le clôture, ajoute des commentaires et calcule le décalage prévu/réel comme pour les tâches.

Le système permet à chaque utilisateur de connaître son évolution (novice, débutant, avancé) en effectuant des statistiques automatiques chaque mois, que l'utilisateur reçoit dans son email. Aussi, l'administrateur peut lancer des analyses d'utilisation du système. En particulier, le système affiche les utilisateurs n'ayant pas utilisé le système depuis un mois. L'administrateur trie ces utilisateurs dans l'ordre décroissant de la durée d'inactivité et sélectionne certains pour leur envoyer un message de relance ou d'encouragement à travers le système, qu'ils reçoivent dans leurs emails.

Travail demandé

1. Elaborer le diagramme de cas d'utilisations fonctionnels et identifier les éventuels liens entre les cas.
2. Identifier une exception dans le déroulement d'un cas d'utilisation de la question 2 et donner le traitement adéquat de l'exception.
3. Pour un cas de votre choix, identifier les classes candidates et les représenter par un diagramme de classes en indiquant la responsabilité de chaque classe.