

Étude de cas : Informatisation d'une pompe à essence

L'existant :

Une station service possède plusieurs pompes à essence. C'est le pompiste qui sert les clients qui se présentent avec leur voiture.

Le pompiste doit décrocher le pistolet afin de remettre le compteur à zéro et remettre la pompe en fonction. Un contact situé sur la pompe détecte quand le pistolet est décroché puis raccroché.

Chaque pompe fournit plusieurs types de carburant aussi le pompiste doit sélectionner le carburant (essence ordinaire, essence sans plomb ou diesel) avant de commencer à remplir le réservoir. La sélection du carburant positionne le prix au litre qui est différent pour chaque carburant.

Le service se fait en appuyant sur la gâchette du pistolet. Quand le pompiste a terminé, il raccroche le pistolet et peut lire le montant à payer sur l'afficheur de la pompe. La pompe est désactivée jusqu'à ce que le pistolet soit de nouveau décroché.

Expression des besoins :

On souhaite permettre aux clients de se servir eux-mêmes de façon à ce que le gérant puisse contrôler simultanément le fonctionnement de toutes les pompes depuis l'ordinateur de la station.

C'est l'ordinateur du gérant qui pilote chaque pompe à l'aide des informations qu'elle lui communique : pistolet décroché ou raccroché, carburant sélectionné, prix du litre, quantité servie ...

Sur l'ordinateur et pour chaque pompe est associé l'affichage de son état qui peut indiquer le fait que la pompe est non autorisée ou autorisée, en train de servir du carburant, la nature du carburant sélectionné, le montant en cours, le paiement effectué ou non ...

Pour qu'une pompe puisse être utilisée, il faut que le gérant la remette en service (l'autorise) depuis son ordinateur. Cette remise en service peut avoir lieu dès que le client précédent a payé sa fourniture en carburant.

La pompe pourra ainsi redémarrer dès que le prochain client décrochera le pistolet.

Quand le client décroche le pistolet, si la pompe est autorisée l'afficheur se remet à zéro. Si la pompe n'est pas autorisée, le client doit attendre que le client précédent ait payé pour que gérant puisse l'autoriser.

Le client doit en premier sélectionner son carburant, puis peut se servir de la quantité souhaitée. Le montant est constamment mis à jour pendant que le client se sert.

Quand le client raccroche le pistolet, le montant est affiché sur l'ordinateur du gérant. Le client doit ensuite aller à la station pour payer, ce qui déclenchera l'autorisation de la pompe pour un client suivant.

Le soir, quand le gérant ferme la station, il désactive l'ensemble des pompes en leur enlevant l'autorisation. Il peut ensuite obtenir les chiffres pour la journée : pour chaque type de carburant le total vendu et le montant. Ainsi, il peut prévoir une demande de réapprovisionnement des cuves d'essence de la station quand le volume restant dans les cuves atteint le minimum autorisé.

Le matin quand il ouvre la station, la première des choses à faire est d'autoriser à nouveau les pompes.

Quand la station est réapprovisionnée, il faut mettre à jour la quantité de carburant disponible dans les

cuves pour pouvoir prévoir correctement le prochain approvisionnement.

Quand le carburant dans une cuve donnée atteint un minimum fixé, l'ordinateur reçoit un signal, envoyé par un capteur, qui prévient le gérant afin qu'il puisse désactiver la sélection de ce carburant pour toutes les pompes.

Travail à effectuer :

Établir la vue fonctionnelle.

Définir les acteurs du système (humains et système connexes). Faire la distinction entre systèmes connexes et simples éléments d'interface pilotés par l'ordinateur du gérant. Donner leur description textuelle et le diagramme de contexte statique.

Établir pour chaque cas d'utilisation la spécification textuelle complète, le diagramme de séquence système et le diagramme d'activité.

Établir les vues dynamiques et statiques.

Pour chaque scénario, donner le diagramme de séquence ou le diagramme de collaboration.

Parallèlement, ajouter les classes correspondantes dans le vue statique.

En déduire les diagrammes de classes partiels.

Établir pour les classes où cela a un intérêt le diagramme d'état.

Établir le diagramme complet des classes d'application, ainsi que le diagramme des classes d'interface homme-machine.

Pour les plus avancés, écrire le code Java correspondant permettant de simuler le fonctionnement des pompes. L'interface homme-machine pourra être simplement textuel (menus textuels dans la fenêtre de commande et saisie des choix de l'utilisateur au clavier) ou pourra être graphique (avec choix avec sélections grâce à des boutons par exemple).