

UML : TD2 - HORODATEUR

Sujet adapté d'une étude de cas du département informatique de l'IUT de Valence

Les fonctionnalisés suivantes doivent être assurées :

1. maintenir l'horodate (date et heure) courante ;
2. afficher l'horodate ;
3. l'Horodate est actualisé automatiquement une fois par jour en interrogeant un serveur distant SNTP (Simple Network Time Protocol).
4. saisir le paiement d'un utilisateur dès l'introduction d'une pièce ;
5. afficher la somme perçue et l'horodate limite de stationnement autorisé au fur à mesure de la saisie des pièces ;
6. encaisser les pièces engagées et délivrer un ticket de stationnement en cas de validation de la demande par l'utilisateur. Le ticket de stationnement doit comporter l'horodate de fin de stationnement autorisé ;
7. restituer les pièces engagées en cas d'annulation de la demande par l'utilisateur.

Les contraintes suivantes doivent être respectées :

1. Les pièces acceptées sont celles de 0,1 euro, 0,2 euro, 0,5 euro et 1 euro.
2. Tout objet non reconnu comme pièce valide doit être rejeté.
3. Les pièces acceptées sont stockées (avant l'encaissement ou la restitution).
4. 0,5 euro donne droit à 1/4 heure de stationnement.
5. La durée maximale de stationnement payant ne doit pas excéder 2 heures.
6. Toute pièce provoquant le dépassement de la durée maximale de stationnement payant doit être rendue.
7. Les heures de stationnement payantes sont de 9h à 12h et de 14h à 19h, tous les jours.
8. L'horodate de fin de stationnement autorisée peut être reportée à la journée suivante. Par exemple : un utilisateur qui paye 2 heures de stationnement à 18h le lundi 8 janvier 2007 est autorisé à stationner jusqu'au mardi 9 janvier 2007 à 10h.
9. Un capteur détecte l'absence de papier pour imprimer les tickets. En cas d'absence de papier, l'horodateur se met hors service ; la remise en service est effectuée par le technicien après avoir remis du papier.

Questions

1. Construisez le diagramme de contexte statique
2. Trouvez les divers groupes/package de cas d'utilisation
3. Elaborez un *diagramme de cas d'utilisation* par package
4. Etablissez la liste des scénarios possibles pour chaque cas d'utilisation
5. Complétez les descriptions précédentes par un diagramme de séquence par scénario