ASR2 Utilisation des systèmes Informatiques

TM

Département Informatique – IUT Bordeaux 1 – 2005-2006

Objectifs : Une entreprise souhaite gérer son ensemble de badges électromagnétiques servant à contrôler l'accès aux diverses portes d'un bâtiment. Elle fait donc appel à vous pour fournir une application *shell* permettant diverses opérations. Voici donc le cahier des charges de ce travail :

1 Gestion des badges

Votre base de données sera composée d'un ensemble de fichiers ; le premier fichier est fixé étant donné qu'il est utilisé dans une autre application :

Ce fichier fixé stocke les badges avec leur possesseur déclaré, il sera donc composé de quatre champs : nom et prénom de l'utilisateur suivi de son adresse e-mail et d'un numéro de badge (numéro unique sur 8 caractères), les champs étant séparés par le caractère « : ».

Par exemple :

Proprietaires.bd

```
dupont:franck:fdupont@labo.net:16444321
durand:luc:ldurand@dept-info.net:00433214
```

Les contraintes concernant les badges sont :

- Un numéro de badge est unique.
- Un utilisateur est unique (on peut quand même imaginer plusieurs personnes avec le même nom) et ne peut posséder qu'un seul badge.
- Un badge est relié à l'ensemble des portes qu'il permet d'ouvrir : bureaux de 001 à 399, la porte 000 étant la porte du bâtiment.
- Un badge peut-être *actif* ou pas, cela permet de bloquer un badge car il a été perdu par exemple.
- Par défaut un badge ouvre la porte 000.
- Un badge a une date de validité à partir de laquelle le badge ne pourra plus être utilisé.

TRAVAIL EN BINÔME À RENDRE AVANT LE 24/01/2006 17H00

2 Fonctionnalités

Notre client souhaite pouvoir

1. Récupérer le numéro de badge à partir du nom d'un utilisateur (et de son prénom si nécessaire) ou inversement trouver le propriétaire d'un badge à partir du numéro.
2. Connaître toutes les informations sur un badge : son propriétaire, et (si il est actif) sa date de validité et les portes qu'il permet d'ouvrir.
3. Ajouter un badge dans la base de données (avec son propriétaire) en donnant toutes les informations nécessaires. On fera une deuxième fonction d'ajout qui utilisera des données par défaut : pour la date de validité, un an après la création du badge, et l'accès, seulement la porte 000.
4. Remplacer le badge d'une personne, il s'agira donc de seulement changer le numéro de badge d'un utilisateur tout en conservant les propriétés de son ancien badge.
5. Supprimer un badge de la base de données.
6. Ajouter ou supprimer des accès à un utilisateur, à partir de son nom et éventuellement de son prénom ou de son numéro de badge (on peut supprimer l'accès à la porte 000, il s'agira alors d'un badge utilisé en interne ou pour un stagiaire pour lequel on vérifiera l'identité avant de le laisser rentrer).
7. Faire des requêtes pour savoir si un utilisateur donné (ou un badge donné) peut ouvrir une porte particulière.
8. Vérifier les dates de validité de l'ensemble des badges et envoyer un mèl à chacun des utilisateurs dont la date de validité est dépassée.
9. Un script général permettant à travers un menu d'accéder aux opérations ci-dessus.

Votre travail consiste

- à **implémenter** l'ensemble de ces fonctionnalités
- et à fournir un **dossier** expliquant la structure de données utilisée, les 3 ou 4 algorithmes les plus complexes que vous ayez eu à réaliser, et les sources imprimés de chacun des scripts.