

Examen d'Éléments de Génie Logiciel

Durée : 2 heures.

Seuls documents autorisés : mémento UML et mémento OCL.



PREMIERE PARTIE : REPONDEZ SUR LA COPIE CLASSIQUE

Remarque : Toute ambiguïté que vous pourriez rencontrer dans ce sujet devra être résolue en décrivant brièvement le choix que vous avez fait.

1. Conception en UML : étude de cas (12 points)

On souhaite développer un site web pour une centrale de réservation hôtelière. Ce site devra permettre de réserver des hôtels en France par l'intermédiaire d'un service audiotel.

On peut considérer que le site web peut être divisé en trois modules :

- un module de réservation des hôtels permettant aux clients d'accéder aux informations sur les hôtels ;
- un module de gestion des hôtels permettant aux hôteliers de gérer la fiche décrivant leur hôtel ;
- un module client permettant aux clients de consulter les réservations qu'ils ont faites auprès des établissements affiliés à la centrale et ayant été prises en compte.

La recherche d'un établissement (module de réservation) peut s'effectuer de deux manières : par un formulaire de recherche ou par choix dans une liste. Elle s'effectue selon les critères suivants : ville et/ou code postal (un champ au moins obligatoire) et nombre d'étoiles (facultatif) ou choix du département puis de la ville. Dans le résultat de sa recherche, le client peut voir le nom de l'établissement, le nombre d'étoile, la situation géographique (ville et code postal) et un lien pour voir la fiche de l'hôtel. La fiche descriptive comprend un numéro audiotel qu'il peut appeler et grâce à un code d'hôtel et un code d'accès indiqués sur la fiche de l'hôtel, il est mis en relation directement avec l'établissement concerné ce qui lui permet d'effectuer sa réservation. A partir du moment où la réservation est saisie par l'hôtelier, elle apparaît dans le module client permettant aux clients de consulter leurs réservations. Bien sûr, une réservation peut-être annulée et apparaîtra alors comme telle dans le module client. Une réservation peut porter sur plusieurs chambres de plusieurs types (simple, double, etc) qui sont toutes réservées pour la même période (date d'arrivée et date de départ).

Une interface devra permettre à l'hôtelier de modifier les informations concernant son hôte et de saisir les réservations (et leurs annulations éventuelles). Cet espace hôtelier (module de gestion des hôtels) sera accessible par identification (identifiant et mot de passe).

Le module client sera accessible par identifiant et mot de passe. Ces informations sont envoyées par e-mail au client lors de sa première réservation via la centrale de réservation. Le système attribuera automatiquement à chaque nouveau client (lors de la saisie d'une première réservation par un hôtelier) un numéro de client, un identifiant et un mot de passe. A chacune de ses réservations, le client pourra rappeler son numéro de client afin que toutes ses réservations soient rattachées à un même numéro de client et donc consultables simultanément (module client). S'il ne fournit pas son numéro de client lors d'une de ses réservations, il sera considéré comme un nouveau client, c'est-à-dire qu'un nouveau numéro de client, un nouvel identifiant et un nouveau mot de passe lui seront attribués pour la réservation courante. Dans cet exercice on considérera un module client minimal (par de gestion de profil client et pas de possibilité de modification des identifiants et des mots de passe).

Une interface d'administration du site doit également être prévue. Elle doit permettre de voir le détail des informations sur un hôtel, de modifier les informations sur l'hôtel, de le désactiver ou de le supprimer. La suppression d'un hôtel supprime tous les éléments associés à cet hôtel. En revanche, la désactivation ne supprime aucune information, elle permet uniquement de ne pas afficher l'hôtel dans les résultats de recherche. Un hôtel désactivé peut-être modifié, réactivé ou supprimé. Cette interface doit également permettre l'ajout d'hôtels pour lesquels il faudra alors renseigner les éléments suivants : coordonnées de l'hôtel, adresse e-mail, nombre d'étoiles, identifiant et mot de passe pour le gérant de l'hôtel (avec possibilité de les générer automatiquement).

Travail demandé :

- a) Donner un (ou plusieurs) diagrammes de cas d'utilisation de plus haut niveau afin de décrire le fonctionnement souhaité du site (2 points).
- b) Préciser comment spécifier, à l'aide de la notation UML, les différents modes de recherche demandés dans le module de réservation des hôtels. Le(s) diagramme(s) ne sont pas demandés (1 point).

La société souhaite dans un deuxième temps intégrer à ce système une fonction de réservation en temps réel par carte bancaire. Cette fonctionnalité sera externalisée car elle requiert de la part des hôteliers la mise en place d'un système de réservation en temps réel afin de pouvoir vérifier la disponibilité des chambres et il n'est pas envisageable pour le moment de demander aux hôteliers non équipés d'avoir un tel système. La solution retenue est donc de s'affilier à un partenaire de réservation afin d'intégrer sa base d'hôtels à celle de la centrale et de profiter ainsi du paiement en ligne déjà en place chez ce partenaire. La recherche de disponibilités n'ayant pas pour le moment été intégrée au site, un lien vers un formulaire de recherche de disponibilités est présent sur les fiches des hôtels provenant de la base d'hôtel du partenaire. Ces formulaires sont ceux proposés par le partenaire. Ils permettent la réservation en ligne.

- c) Indiquer ce qu'il est nécessaire de changer ou de rajouter dans le(s) diagramme(s) de cas d'utilisation données en (1) pour prendre en compte cette nouvelle fonctionnalité (1 points).
- d) Donnez le diagramme de classe complet correspondant à cette application (3 points).
- e) Préciser où sont conservés le numéro audiotel, les identifiants et mot de passe des hôteliers, les identifiants et mot de passe des client et les identifiants et mot de passe des administrateurs du site (1 point).
- f) Donner le diagramme état-transition de la classe modélisant un hôtel (2 points).

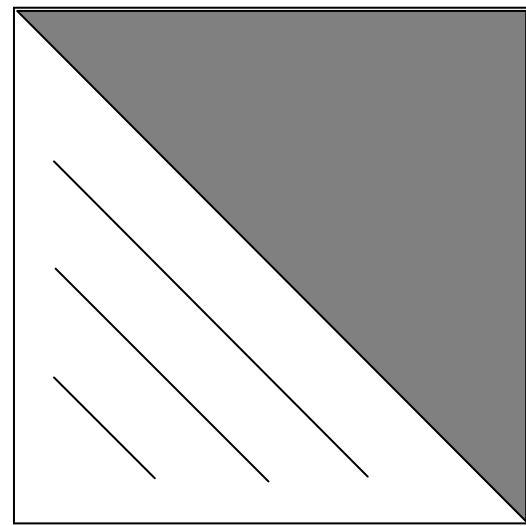
Examen d'Éléments de Génie Logiciel

Durée : 2 heures.

Seuls documents autorisés : mémento UML et mémento OCL.



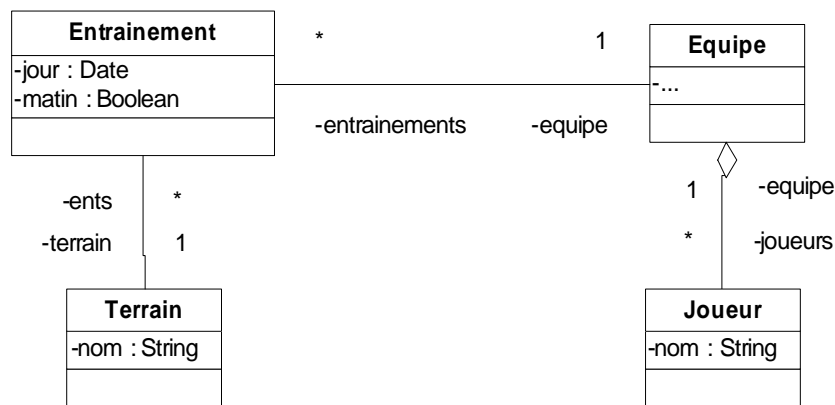
DEUXIEME PARTIE : REPONDEZ SUR CE SUJET



Remarque : Toute ambiguïté que vous pourriez rencontrer dans ce sujet devra être résolue en décrivant brièvement le choix que vous avez fait.

1. Spécification en OCL : étude de cas (8 points)

On considère le diagramme de classes suivant, version très simplifiée d'un modèle qui prend en compte la gestion des entraînements d'équipes sportives sur des terrains :



Des équipes, composées de joueurs, peuvent réserver des terrains. La réservation du terrain pour un entraînement se fait pour un jour donné, le matin (booléen *matin* à vrai) ou l'après-midi (booléen à faux). Dans les questions suivantes, on considérera que des objets du type Date peuvent se comparer avec des opérations comme =, <, <=, >, >=.

Rappel (hors mémento OCL) : l'opération *allInstances()* renvoie la collection des instances du type référencé. Par exemple, *Terrain.allInstances()* renvoie la collection de toutes les instances du type *Terrain*.

- a) Donnez une contrainte OCL qui spécifie que toute équipe contient au moins 11 joueurs.

- b) Donnez une contrainte OCL qui spécifie l'unicité du nom du terrain parmi tous les terrains.

On considère la fonction de la classe Terrain
estLibre(d : Date, m : Boolean) : Boolean
// rend vrai si le Terrain n'est pas le lieu d'un entraînement à la date *d*, le matin si *m* est vrai, l'après-midi sinon.

- c) Donnez les éventuelles pré et postconditions de cette fonction *estLibre*.

On considère la fonction de la classe Terrain
estLibreTouteLaJournée(d : Date) : Boolean
// rend vrai si le Terrain n'est pas le lieu d'un entraînement toute la journée à la date *d*.

- d) Donnez les éventuelles pré et postconditions de cette fonction *estLibreTouteLaJournée*. Vous pouvez bien sûr utiliser d'autres fonctions pour la spécifier.

On considère l'opération de la classe Equipe
programmerEntraînement(t : Terrain, d : Date, m : Boolean)
// réserve le terrain *t* pour l'équipe à la date *d*, le matin si *m* est vrai, l'après-midi sinon (le terrain doit être libre, l'équipe ne doit pas avoir d'autre entraînement à cette date, matin ou après-midi)

- e) Donnez les éventuelles pré et postconditions de cette opération *programmerEntraînement*. Cette opération est censée créer un nouvel objet entraînement correspondant. Vous pouvez bien sûr utiliser d'autres fonctions pour la spécifier.