ORDRE DES INGÉNIEURS DU QUÉBEC

SESSION DE MAI 2013

Toute documentation permise

Calculatrices : modèles autorisés seulement

Durée de l'examen : 3 heures

04-Log-A3 Conception logicielle

Mise en contexte: Le texte qui suit décrit les principales fonctionnalités du logiciel *ReservDBus*.

Le logiciel *ReservDBus* est une application de gestion de flottes d'autobus et du personnel associé, s'adressant à des compagnie de transport du secteur du tourisme voulant offrir des services de transport de passagers par autobus pour des excursions privées par la location d'autobus nolisés. Cette application s'exécute localement sur un ordinateur équipé du système d'exploitation Windows 7, et communique avec un serveur de base de données s'exécutant sur un ordinateur distinct pour sauvegarder toutes les informations de l'application. L'application est principalement utilisée par les employés des compagnies de transport par autobus qui sont responsables de prendre les réservations de clients au téléphone. L'application est également utilisée par les gestionnaires des compagnies de transport.

L'application *ReservDBus* offre les fonctionnalités suivantes à tous les usagers. L'application permet :

1- à chaque utilisateur d'ouvrir une session en utilisant un identificateur et un mot-de-passe unique,
2- d'enregistrer une réservation pour un certain nombre d'autobus. Une réservation inclut la date et le lieu du départ, la ou les destinations, la durée d'attente à chaque destination et l'heure et le lieu du retour. Lors d'une réservation, le client peut également réserver les services de guides accompagnateurs, en précisant la ou les langues dans lesquelles les guides doivent pouvoir s'exprimer.
3- d'assigner, pour une réservation, un chauffeur à chacun des autobus réservés,
4- d'assigner, pour une réservation, un guide touristique à chacun des autobus réservés,
5- d'afficher la liste des réservations enregistrées pour une période de calendrier donnée. Une période de calendrier est spécifiée par une date et une heure de début et une date et une heure de fin,
6- d'afficher la liste des autobus disponibles et la capacité de chacun pour une période de calendrier donnée,
7- d'afficher la liste des chauffeurs disponibles pour une période de calendrier donnée,
8- d'afficher la liste des guides disponibles qui sont capables de s'exprimer dans une langue donnée pour une période de calendrier donnée,
9- d'imprimer les informations liées à une réservation afin de la remettre à chacun des chauffeurs et des guides accompagnateurs,
10- d'enregistrer la date et l'heure de départ des autobus associés à une réservation,
11- d'enregistrer la date, l'heure de retour et la distance parcourue par chaque autobus associé à une réservation,
12- de calculer et d'imprimer une facture pour le client, une fois la prestation de service de transport complétée.

En plus des fonctions ci-dessus, disponibles à tous les employés de la compagnie de transport, les gestionnaires de la compagnie ont également accès à certaines fonctions d'administration.

Les fonctions d'administration du système *ReservDBus* permettent :

1- de créer, de modifier les paramètres ou d'éliminer un compte usager,
2- de gérer la liste et la capacité des autobus,
3- de gérer la liste des chauffeurs,
4- de gérer la liste des guides accompagnateurs et de préciser la liste des langues maîtrisées par chacun.

Question 1 - Processus de conception et cas d'utilisation (20 points)

a) Les processus modernes de développement de logiciels orientés-objet s'appuient sur l'élaboration de cas d'utilisation du logiciel à concevoir. Dans le contexte des cas d'utilisation, définissez en vos propres mots et fournissez un exemple tiré de la description du logiciel *ReservDBus* :
   i. d'un cas d'utilisation (2 points),
   ii. d'un acteur (2 points),
   iii. d'une précondition (2 points),
   iv. d'une association entre deux cas d'utilisation décorée du stéréotype <<extend>> (2 points).

b) Tracez le diagramme UML des cas d'utilisation du logiciel *ReservDBus* (12 points)

Question 2 - Architecture logique (20 points)

Les logiciels modernes sont développés selon des architectures logiques comportant plusieurs couches ou niveaux (*multi-tier architecture*).

- Indiquez deux avantages importants de la décomposition d'un logiciel en une architecture multi-niveaux (4 points),
- Lors de la conception d'une architecture multi-niveaux, Larman suggère l'emploi du patron Contrôleur (*Controller*). Expliquer le rôle du Contrôleur et discutez des avantages liés à son utilisation (2 points),
- Au sein d'un logiciel conçu selon une architecture multi-niveaux, certains appels de fonctions ou de méthodes entre des modules appartenant à des couches distinctes de l'architecture sont permis alors que d'autres sont interdits. Décrivez et expliquez les règles permettant de déterminer quels appels sont permis et quels appels sont interdits (4 points).
- Proposez une décomposition architecturale multi-niveaux du logiciel *ReservDBus*. Fournissez votre réponse sous la forme d'un diagramme UML de paquetages (10 points).

Question 3 – Analyse et conception orientées-objet (30 points)

a) Dans le contexte de la conception orienté-objet, définissez en vos propres mots chacun des termes suivants:
   i. Classe de base (1 point),
   ii. Interface (1 point),
   iii. Type (1 point).

b) Quelle distinction faites-vous entre les trois éléments que vous venez de définir ci-dessus ? (2 points)

c) En vous limitant uniquement aux fonctionnalités et concepts décrits dans le texte de mise en contexte, proposez un diagramme UML des classes du logiciel *ReservDBus* (25 points).

Question 4 - Patrons de conception (30 points)

Durant la phase d'analyse et conception du logiciel *ReservDBus*, voici trois problèmes qui ont été identifiés et que l'on vous demande de résoudre en utilisant les patrons de conception :

1. Il existe, dans l'application *ReservDBus*, une relation de collaboration complexe entre les objets représentant les `chauffeurs` et les objets représentant les `réservations`. On

désire représenter la relation entre les objets `chauffeurs` et les objets `réservations` tout en conservant un couplage faible entre les objets.

2. Il existe un très grand nombre de types possibles d'`autobus`, chacun avec leurs caractéristiques propres, et on veut fournir un mécanisme de création des objets représentant les `autobus` qui isole l'application *ReservDBus* des types concrets d'`autobus` qui seront manipulés dans l'application.
3. Lorsqu'un `chauffeur` ou un `guide touristique` est assigné à une réservation par un employé de la compagnie, l'application *ReservDBus* doit émettre un avertissement automatique en envoyant un courriel ou un message texte (SMS) à l'employé, en lui fournissant toutes les informations pertinentes sur la réservation qui vient d'être effectuée. Si une modification est faite par la suite à une réservation, tous les employés (chauffeurs et guides) concernés doivent être avertis automatiquement.

Pour chacun des trois problèmes de conception, on vous demande :

a) d'identifier le patron de conception approprié (tiré du livre de Gamma et al.) qui doit être utilisé pour résoudre le problème et de justifier votre choix (2 points),
b) d'appliquer le patron en fournissant un diagramme de classes annoté, où les rôles identifiés dans le patron sont explicitement associés aux classes appropriées (4 points),
c) de discuter des avantages, des inconvénients et des conséquences liés à l'utilisation du patron choisi pour résoudre le problème de conception (4 points).