

Examen Janvier 2009 (durée 2h) 2eme Session

Tous documents autorisés
Barème indicatif sur 20,5 points

Questions de Cours (5 points)

1. Représentez à l'aide d'un diagramme de classe un système de fichiers : un répertoire contient des répertoires et/ou des fichiers. Votre solution devra être aussi flexible que possible.
2. Quelles règles de cohérence doivent être respectées entre les diagrammes de classe et de séquence d'un modèle ?
3. Quel est le lien entre test de validation et test d'intégration ? Et entre test d'intégration et test unitaire ?
4. Quelles distinctions faites-vous entre une méthodologie de développement logiciel et UML?
5. En quoi consiste un *design pattern* ? Citez au moins 3 design pattern fréquents.

Problème : RéZa

Une SSII a décidé de créer une application nommée RéZa dédiée à la gestion d'emploi du temps et à l'allocation de salles.

Le système vise les moyennes et grosses structures gérant des locaux, typiquement les universités ou écoles, centre de conférences etc...

L'application admet trois sortes d'utilisateurs :

- L'administrateur peut créer/modifier/supprimer la description des salles (capacité, présence de vidéoprojecteur...)
- Le gestionnaire valide ou invalide les réservations de salles réalisées par les utilisateurs.
- Les utilisateurs peuvent consulter les emplois du temps et demander la réservation d'une salle.

La consultation des emplois du temps permet aux utilisateurs d'obtenir un calendrier qui correspond à leurs critères de recherche. Il existe deux types de calendrier :

- Par catégorie : chaque réservation contient un champ « catégorie », qui permet de préciser à quel événement cette réservation se rapporte. Par exemple « ue MI017 ». La recherche par catégorie permet d'obtenir un emploi du temps couvrant toutes les réservations qui se rapportent à la catégorie.
- Par salle: il est possible d'obtenir l'emploi du temps d'une salle particulière, sur une période donnée. L'utilisateur n'a qu'à préciser la salle et les dates de début et de fin du calendrier à produire.

Le système de réservation doit aider les utilisateurs à trouver et réserver une salle libre.

Il existe deux types de réservations : les réservations ponctuelles et les réservations périodiques.

Pour toute réservation le demandeur doit préciser son nom (par exemple « Y.Thierry-Mieg »), la catégorie à laquelle la réservation est liée (par exemple « ueMI017 »), un nom pour cette réservation (par exemple « Examen rattrapage »), la salle demandée (par exemple « Amphi B1 »), et la date et plage horaire désirée (par exemple « 22 Janvier, 15h30-17h30 »).

Si la réservation est périodique, l'utilisateur le précisera sur le même formulaire, et fixera la périodicité (quotidienne, hebdomadaire ou mensuelle) et le nombre d'occurrences (par exemple pour « Cours MIO17 », 10 occurrences de périodicité hebdomadaire).

Tout utilisateur du système peut faire une demande de réservation pour une salle. Une demande périodique de n occurrences sera traitée comme si l'utilisateur avait fait n demandes ponctuelles du point de vue de la validation (chaque occurrence pouvant être individuellement validée ou non).

Le gestionnaire peut consulter toutes les demandes de réservation non validées. Il les inspecte alors une par une et décide de les valider ou les invalider. Dans les deux cas, le demandeur de la réservation sera informé par mail de la mise à jour de l'état de sa réservation. Le gestionnaire décide de valider ou pas les demandes en fonction de ce qu'il sait du demandeur, de l'événement concerné et de ses propres contraintes de planning.

PHASE ANALYSE

Question 1 (2 point)

Réalisez le diagramme de cas d'utilisation du système.

Commentez et/ou annotez le diagramme.

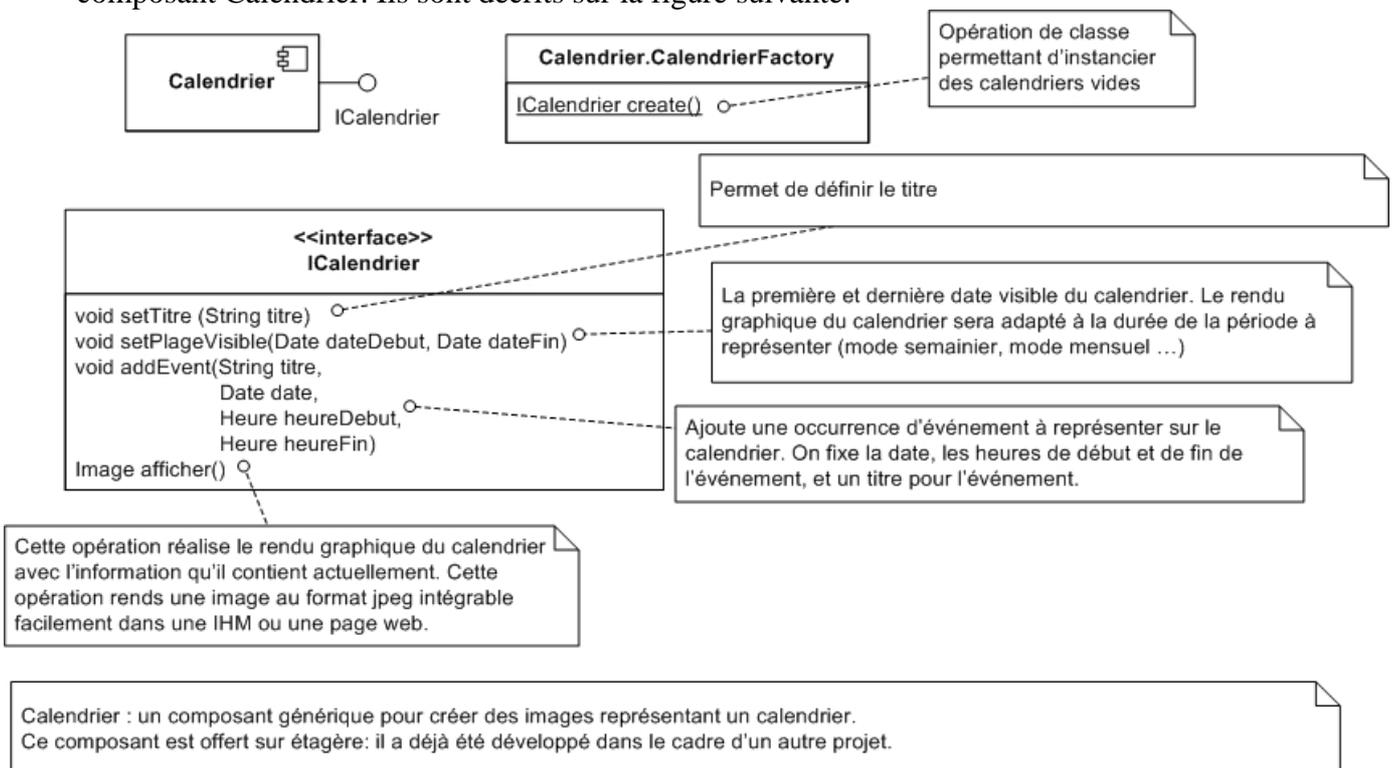
Question 2 (3 point)

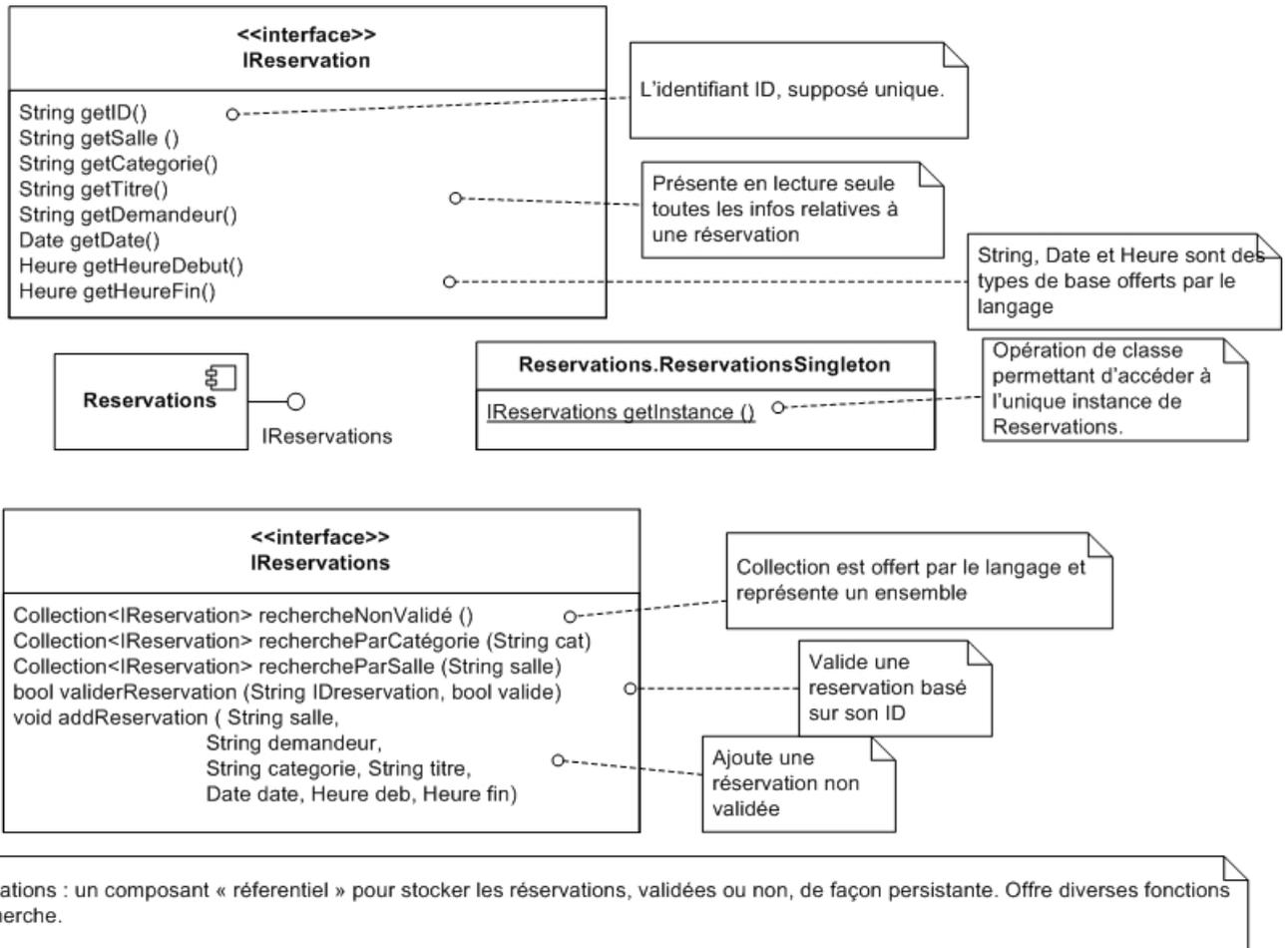
Proposez un diagramme de classes d'analyse pour ce problème.

On ne détaillera pas les opérations de la classe (fictive) représentant le système.

PHASE CONCEPTION

La conception ne part pas de zéro : l'équipe de développement se propose de réutiliser deux composants déjà définis dans le cadre d'un autre projet : le composant réservations et le composant Calendrier. Ils sont décrits sur la figure suivante.





Question 3 (1,5 points)

- Quel est à votre avis le rôle des deux classes « CalendrierFactory » et « ReservationsSingleton » ? Comment s'en servir (en pseudo code) ?
- Le composant Reservations s'appuie en réalité sur une base de données relationnelle. A votre avis, pourquoi utilise-t-on un singleton ?

Question 4 (2 point)

On décide de créer trois composants d'IHM dédiés respectivement à l'administrateur, au gestionnaire et à l'utilisateur. Un composant jouant le rôle de contrôleur RéZaCore et sert les besoins de ces différentes IHM en s'appuyant sur les composants Calendrier et Reservations définis plus haut. Il doit jouer un rôle de façade pour les IHM qui n'interagiront qu'avec lui. Décrivez cette situation à l'aide d'un diagramme de composants présentant les interfaces requises et offertes des divers composants.

Question 5 (4 points)

Modélisez à l'aide d'un ou plusieurs diagrammes de séquence de niveau intégration, c'est-à-dire où les lignes de vie représentent des composants, les interactions nécessaires pour demander une réservation ponctuelle de la salle B1 le 22 janvier de 15h à 17h, puis que le gestionnaire valide cette demande.

On utilisera au besoin des notes en syntaxe libre pour expliquer les points algorithmiques difficiles à représenter sur un diagramme de séquence.

Question 6 (3 point)

Modélisez à l'aide d'un ou plusieurs diagrammes de séquence de niveau intégration, c'est-à-dire où les lignes de vie représentent des composants, les interactions nécessaires pour qu'un utilisateur consulte un calendrier portant sur la salle B22 du 1^{er} au 15 janvier.

On utilisera au besoin des notes en syntaxe libre pour expliquer les points algorithmiques difficiles à représenter sur un diagramme de séquence.

Question 7 (3 point)

Déduisez des questions 4, 5 et 6 la signature des opérations des interfaces liant les IHM de l'utilisateur, du gestionnaire, et le contrôleur RéZaCore.

Question 8 (3 point)

Proposez un test d'intégration du composant Reservations permettant de s'assurer du bon fonctionnement de l'opération addReservation(...). Formulez votre réponse en pseudo-code ou avec un diagramme de séquence.