

Examen de Génie logiciel avancé

Note : Vous avez 3 heures. Rédigez chacune des deux parties sur des feuilles indépendantes.

Première partie

Exercice 1 (Spécification informelle des charges)

(1 heure)

Ce sujet porte sur la conception d'un logiciel pour gérer l'emprunt des livres dans une bibliothèque libre-service, nommé *VELIV*. Le système physique s'appuie sur :

- des cartes magnétiques, une pour chaque abonné
- les livres
- une borne d'entrée, située à l'extérieur de la bibliothèque
- une borne de consultation
- une borne d'emprunt
- une boîte à livre pour le retour des livres

Chaque carte magnétique est identifiée par un code unique à 8 chiffres, qui identifie l'abonné. Chaque carte est aussi associée à un code PIN secret et à une période de validité. Pour entrer, les abonnés doivent présenter leurs cartes auprès de la **borne d'entrée** et entrer le PIN. La même procédure est nécessaire pour utiliser les bornes d'emprunt et consultation. Chaque livre est équipé avec un **radio-identificateur RFID**, utilisé à la fois pour identifier le livre et comme dispositifs antivols. Chaque radio-identificateur **RDIF** est associé à un code unique à 13 chiffres correspondant à l'**ISBN** (International Standard Book Number) du livre.

Pour emprunter, l'abonné se rend à la **borne d'emprunt** où, après identification, il pourra valider chacun des livres qui désire emprunter. Pour retourner un livre, l'abonné glisse le livre dans la boîte à livre (sans besoin d'identification). Chaque abonné peut emprunter un maximum de 3 livres à la fois, chacun pour un maximum de 45 jours. Si un livre n'est pas retourné avant le délai maximum, l'accès à tous services de *VELIV* est suspendu jusqu'à la réhabilitation de l'abonnement.

Chaque abonné peut utiliser la **borne de consultation** pour consulter le catalogue de livres. Le catalogue montre les livres de la bibliothèque, leur état (déjà empruntés ou pas), leur collocation physique (si disponibles) et leur meta-données littéraires typiques (titre, auteur, genre, éditeur, etc.).

Votre tâche est de concevoir le système de gestion *VELIV* en respectant les meilleures pratiques de réutilisation de code et d'extensibilité. Le système de gestion sera développé dans un langage de programmation à objets.

Questions

1. Quelle est la frontière du système ? Justifiez.
2. Quelles sont les charges fonctionnelles ? Quelles sont les charges non fonctionnelles ? Quelles sont les charges liées au domaine d'activité ? Justifiez.

3. D'après vous, quels sont les acteurs principaux de ce système ? Justifiez.
4. Donner le diagramme (en syntaxe UML) des cas d'utilisation principaux pour VELIV et expliquer ses différentes composantes.
5. Donner le diagramme (en syntaxe UML) de séquence correspondant à l'emprunt d'un ou plusieurs livres.
6. Spécifier le diagramme d'état qui décrit les différents états des abonnés dans le fonctionnement du système de gestion VELIV.
7. Donner un diagramme relationnel pour les bases de données que vous avez inclus dans l'architecture.

Exercice 2 (Conception à objet)

(1 heure)

1. Qu'est-ce qu'une classe abstraite ? Comment sont représentées les classes abstraites dans un diagramme UML ?
2. Considérez les classes de la Figure 1 comme base pour un diagramme de classes du système de gestion VELIV. Donner le diagramme de classe du système, en syntaxe UML, en ajoutant les attributs principaux et les relations entre classes. Ajoutez de nouvelles classes en expliquant pourquoi elles sont nécessaires pour modéliser le système.

| Nom de la classe | Description |
|----------------------|---|
| Borne | interface avec le pilote d'une borne |
| BorneEntree | interface avec une borne d'entrée |
| BorneConsultation | interface avec une borne de consultation |
| BorneEmprunt | interface avec une borne d'emprunt |
| Abonne | un abonné de la bibliothèque |
| Carte | une carte magnétique d'accès et identification |
| Livre | un livre disponible dans le catalogue |
| Catalogue | l'ensemble de livres disponibles |
| Emprunt | représentation de l'emprunt d'un livre par un abonné |
| Journal | historique d'emprunts |
| JournalPerso | historique d'emprunts d'un abonné |
| JournalGlobal | historique de tous emprunts de la bibliothèque |
| Requete | requête auprès de la borne de consultation pour vérifier la présence des livres et leur disponibilité |
| RequeteAuteur | requête par auteur |
| RequeteEditeur | requête par éditeur |
| RequeteAuteurEditeur | requête par auteur et éditeur |

FIGURE 1 – description des classes.

3. On désire étendre le système dans plusieurs directions énumérées ci-après. Pour chacune de ces extensions, indiquer si elle est facilitée par la modélisation proposée. Dans le cas contraire, proposez une modélisation alternative qui s'appuie sur un patron de conception et qui aurait simplifié cette extension. Dans les deux cas, montrer les classes fondamentales de la modélisation nécessaire à supporter l'extension. Justifiez vos réponses.

- (a) *Langage de requêtes.* Généraliser le langage de requêtes pour permettre requêtes booléennes arbitraires (p.ex. "tel titre OR tel éditeur AND tel auteur").
 - (b) *Réservation via Web.* Permettre d'accéder via Internet, à travers un site web, aux plusieurs services comme le suivi des livres empruntés, le changement de PIN, la consultation du catalogue et la réservation des emprunts. Un abonné pourra réserver un livre (disponible ou pas) et se rendre pour l'emprunter dans les 24h suivantes à la réservation. Un abonné pourra aussi réserver un livre non disponible ou déjà réservé et il aura le droit de l'emprunter.
 - (c) *Plusieurs copies du même livre.* Pour des livres très populaires, la bibliothèque souhaite avoir à disposition plusieurs copies du même livre dans son catalogue.
4. Présenter les patrons de conception Composite et Interpreter. Discuter comment, dans le cadre du système de gestion VELIV et ses extensions, ils peuvent être appliqués.

Deuxième partie

Exercice 3 (Types abstraits)

(1 heure)

Un multiensemble fini de support A est une fonction $M : A \rightarrow \text{Nat}$ telle que $M(x) > 0$ seulement pour un nombre fini d'éléments de A . Ainsi par exemple $\{a, b, a, b, b, d\}$ représente le multiensemble de support $A_1 = \{a, b, c, d\}$ où M_1 est la fonction $M_1(a) = 2$, $M_1(b) = 3$, $M_1(d) = 1$ et $M_1(c) = 0$.

1. Proposer un type abstrait de données ME pour modéliser les multi-ensembles. Pensez à bien spécifier le domaine, les constantes, les constructeurs, les fonctions de test et les fonctions d'accès.
2. Spécifier les fonctions d'appartenance, d'union et d'intersection sous forme d'un codage abstrait en utilisant les opérations définies dans le point précédent.