

Mise en œuvre d'une application de gestion de Vélos en Libre-Service

1 Organisation du projet

Le projet "administration de bases de données" a lieu du 11/03 au 1/04 2019, les lundi après-midi, avec une soutenance le 1/04. Il doit être réalisé par des groupes de quatre personnes. Les salles affectées à cette activité sont indiquées sur le planning ADE (merci de respecter ces salles ou bien d'informer l'enseignement de votre localisation).

Des permanences sont organisées par les enseignants pour chacune des demi-journées consacrées au projet. Un pointage des gens présents sera réalisé pour chaque demi-journée.

L'archive contenant le rapport final et les sources de l'application prêtes à l'emploi (scripts SQL, procédures stockées, déclencheurs et programmes Java, scripts de lancement, tests automatisés) devront être déposés sur Moodle.

L'évaluation portera sur le travail réalisé (code), le rapport et la soutenance: la présentation des documents (clarté, expression, orthographe), l'état d'avancement de l'application réalisée, la qualité de programmation (lisibilité, justification, pertinence et couverture des tests), ainsi que la soutenance (présentation et réponses aux questions). En outre, la prise de recul de chaque groupe par rapport à ses réussites et à ses lacunes sera également appréciée par les enseignants.

Le rapport devra fournir des réponses claires et synthétiques aux questions de l'énoncé, une présentation de la solution aux problèmes à traiter avec des jeux de tests complets le cas échéant.

2 Objectifs du projet

Le projet consiste en la réalisation de deux applications clientes Java accédant via le protocole JDBC à une base de données sur un serveur distant. L'accent est mis sur le maintien de la cohérence des données et en aucun cas sur l'interface homme-machine : une interface en mode texte offrant un minimum d'interactivité (menus, affichage des résultats, trace de la gestion des transactions concurrentes, etc.) est suffisante. L'évaluation ne tiendra pas compte de la présence d'une interface graphique avancée mais uniquement de la facilité à suivre les tâches effectuées et à vérifier le bon fonctionnement de l'application (traces d'exécution).

Les technologies utilisées sont : Java et JDBC pour la partie applicative, Oracle (SQL, PL/SQL) pour la partie base de données.

L'objectif visé est une meilleure compréhension des rôles respectifs de ces technologies lors de leur utilisation dans la perspective de l'implantation d'une application de gestion de données. Un aspect modélisation, inhérent à tout projet de spécification d'un système d'information, fera partie intégrante de ce projet.

3 Position du problème

La ville de Mahingan Falls¹, située entre les villes de Salem et Groville², souhaite se doter d'un service de vélos en libre-service sur le modèle du Vélib Parisien, du Vélo'v Lyonnais ou du VÉCUB bordelais. Ce service, nommé VéPick, sera géré par une entreprise locale qui se lance dans le domaine de la location de vélos. Vous êtes chargés du système de gestion de données de VéPick, basé sur une base de données relationnelle.

Cette base de données doit gérer à la fois le parc de vélos, les stations et les clients. Deux applications doivent permettre d'accéder à la base de données : une pour les clients et une pour les superviseurs du réseau.

Chaque station est identifiée par son adresse. Elle possède plusieurs emplacements, appelés "bornettes", chacune pouvant accueillir un vélo. Chaque bornette d'une station est identifiée par un numéro et son état (OK, HS). Chaque vélo est décrit par son numéro identifiant, son modèle et la date de sa mise en service et son état (OK, HS). A chaque instant, un vélo peut être en station associée à une bornette, loué par un client, en maintenance. L'association d'un vélo à une bornette est automatique grâce à la puce RFID présente dans chaque vélo. Chaque station communique instantanément les changements d'état de ses bornettes à la base de données.

Le cout de location des vélos dépend du modèle, mais ce cout est à la ½ heure (une ½ heure commencée est due). Les modèles de vélo disponible sont laissés à votre convenance.

Les clients peuvent être ou non abonnés au service VéPick. Les abonnés ont renseigné leur nom, prénom, date de naissance, sexe, adresse, numéro de CB, ainsi qu'un code secret leur permettant de s'identifier (en pratique, ces codes sont cryptés dans la base de données, mais nous ne considérerons pas ces détails ici). Plusieurs abonnés différents peuvent avoir le même numéro de CB (une disposition utile pour les familles). Chaque abonnement a une durée d'un an. Chaque abonné peut louer un vélo en le prenant et le ramenant dans un parc après s'être identifié sur la bornette. Les clients non abonnés ne renseignent que leur numéro de carte CB. La transaction bancaire ne sera pas considérée dans ce projet. Il est possible pour un client de louer plusieurs vélos. Pour chaque location par un non-abonné, un code secret est attribué au locataire, qu'il devra mémoriser pour s'identifier quand il rendra son vélo.

Un client peut également renseigner par la borne de l'état d'un vélo: le vélo rendu est en panne ou le vélo qui vient d'être récupéré est en panne (3mn de délais dans ce dernier cas pour déclarer le problème).

Dans tous les cas le système ouvre la première bornette occupée par un vélo dans un état utilisable lors d'une demande de location et la première bornette libre lors d'un retour. L'algorithme doit être simple et laisser à votre convenance.

Pour un usage futur d'analyse de données et de statistiques, on prendra soin d'enregistrer dans la base de données l'ensemble des locations passées afin de mieux connaître le comportement des usagers du service.

Les clients abonnés disposent en plus d'un système de réservation qui leur permet de réserver des vélos dans une station précise pour une période de plusieurs jours. Un client peut avoir plusieurs réservations en cours mais sans recouvrement. Si une réservation est en conflit avec les réservations d'autres clients (plus de vélos disponibles) alors la réservation ne peut aboutir (même pour un seul jour sur une période).

¹ Le Signal, M. Chattam, Ed. Albin Michel.

² Capitale de la Présipauté de Groland.

Certains trajets sont plus courants que d'autres (par exemple, vers le campus universitaire le matin, et depuis le campus universitaire le soir). Afin d'assurer, à chaque station, la présence de vélos disponibles et d'emplacements libres pour y déposer son vélo, Vépik dispose d'un système de régulation citoyen.

Ce système encourage les usagers à assurer eux-mêmes la régulation des vélos en donnant une prime (sous forme d'un crédit-temps de 1mn pour une future location) aux usagers abonnés effectuant certains trajets bénéfiques pour la régulation. Pour bénéficier de la prime, un usager doit emprunter un vélo à une station "Vmoins" et/ou le déposer à une station "Vplus". La définition de ces stations change en fonction des plages horaires, mais chaque station est toujours soit "Vmoins" soit "Vplus" soit "Vnul" (sans effet). La classification en Vmoins/Vplus est décidée par les superviseurs du service.

4 Cahier des charges de l'application

La liste ci-dessous énumère les fonctionnalités essentielles que doit proposer l'application sur les deux interfaces disponibles : Interface Client et interface superviseur.

Interface Client

1 – Abonnement au service. En renseignant les informations nécessaires, un client doit pouvoir s'abonner à l'année.

2 – Emprunt de vélo. Abonné ou non, un client doit pouvoir récupérer un vélo non endommagé à une borne. L'interface doit lui signaler s'il emprunte à une borne Vmoins ou Vplus, le code secret pour rendre le vélo s'il n'est pas abonné. Le client doit pouvoir alerter de l'état d'un vélo en panne.

3 – Rendu de vélo. Un client qui a emprunté un vélo doit pouvoir le ramener à une station s'il y a un emplacement vide et qu'il connaît le code secret. Il faudra vérifier que le vélo ramené est le même que celui emprunté (on a une entrée manuelle du numéro du vélo pour simuler la vérification électronique par la bornette). Le client doit pouvoir informer de l'état d'un vélo rendu en panne. Le client est débité du temps de location.

4 – Programme Vplus. L'application doit signaler à un client abonné qui prend ou rend son vélo s'il bénéficie d'une remise Vplus pour son prochain trajet.

5 – Affichage des bornes Vmoins/Vplus. A tout moment un client doit pouvoir se renseigner sur le statut de bornes privilégiées.

7 – Réservation d'un vélo. Un client peut réaliser une ou plusieurs réservations. Un affichage des disponibilités dans la période de réservation pour une station peut être une fonction intéressante pour aider l'utilisateur.

Interface Superviseur

1 – Consulter le nombre de vélos dans chaque station, le nombre de vélos endommagés, et le nombre de places libres.

3 – Consulter (et modifier) les plages horaires de "Vmoins / Vplus".

5 Modélisation du schéma de base de données

Question 1: Vous donnerez le schéma conceptuel des données (UML ou entité-association), puis le schéma relationnel complet (tables, identifiants, identifiants externes, typage des attributs, contraintes référentiels et de domaines, formes normales des tables).

Les choix conceptuels devront être argumentés au besoin.

Cette question fait l'objet d'un rendu intermédiaire.

6 Analyse des contraintes d'intégrité

Les contraintes énoncées dans les sections précédentes sont de deux types :

1. les contraintes simples sur les données qui sont exprimables dans la description du schéma de la base de données (schéma SQL),
2. les contraintes métiers qui ne sont pas exprimables directement dans le schéma de la base mais dans le code, soit de l'application, soit de procédures stockées,

Question 2: Donner la liste des contraintes métiers, spécifiées et déduites des sections 3, 4 et 5, en précisant pour chaque contrainte comment elle sera traitée.

Question 3: Donner le script SQL (Oracle) qui implémente le schéma de la base de données énoncé en section 5. Vous peuplerez la base à l'aide d'un jeu d'essai judicieusement choisi (et commenté). Vous prendrez un grand soin dans le choix des types pour les dates et les durées.

7 Réalisation de l'application

Vous devrez réaliser une application Java qui met en œuvre les fonctionnalités énoncées dans la section 4.

L'exécution de l'application doit faire apparaître des traces démontrant la prise en compte des différents types de contraintes d'intégrité, les traitements automatiques (trigger), les différentes opérations pour réaliser une tâche, et cela en limitant les saisies manuelles : le jour de la démo on évitera, dans la mesure du possible, de perdre trop de temps à devoir saisir bêtement de nombreuses données pour montrer un point important.

Chaque tâche devra réaliser une trace permettant de faciliter au maximum la visualisation de la bonne marche de l'opération. Au contraire, le développement d'une interface utilisateur riche n'est absolument pas demandé, l'application doit juste proposer le minimum pour faciliter la saisie des informations et fluidifier le déroulement de vos scénarios lors de la soutenance.

Evidemment, plusieurs opérations concurrentes doivent être possibles sans provoquer d'incohérence et en favorisant la concurrence d'accès.

Question 4: Réaliser les applications, procédures stockées, triggers nécessaires à ce SI. Vous n'oublierez pas de préparer des scénarios précis pour la soutenance.

Contrainte: Vous devrez réaliser au moins 2 procédures stockées pour intégrer certaines contraintes avec les modifications de données, et 1 trigger pour gérer certains calcul ou mise à jour automatique d'attributs. Bien sûr vous pouvez faire plus. Le reste pourra se faire exclusivement en JAVA/JDBC.

8 Rapports et soutenance

Les documents (au format PDF) doivent répondre aux différentes questions dans un français ou anglais correct.

Rapport intermédiaire. Le contenu du rapport décrit les points suivants:

- Le schéma conceptuel des données (UML ou entité-association) en argumentant vos choix si plusieurs solutions apparaissent

- Le schéma relationnel complet (tables, identifiants, identifiants externes, typage des attributs, contraintes référentiels et de domaines, formes normales des tables).
- Le document sera fourni sous la forme d'un fichier `rapint_grXX.pdf`

Rapport final. Vous devrez faire l'effort d'être à la fois précis et concis. Le contenu du rapport doit reprendre les points suivants:

- Rappeler les objectifs du projet ainsi que l'organisation du document dans une brève introduction.
- Indiquer les modifications appliquées sur le schéma conceptuel et/ou relationnel depuis le rapport intermédiaire.
- une brève description de l'état d'avancement du projet (étapes réalisées et étapes non réalisées), les principes de conception des solutions proposées, la raison de non réalisation de certaines étapes si tel est le cas.
- La gestion de projet et le travail en groupe seront également décrits dans ce rapport.
- Décrire les opérations de l'application (explications, code commenté, protocoles de tests).
- Fournir le code de l'application
- L'archive nommée `GRXX` sera composé d'un répertoire `src` pour le code, `doc` pour la documentation Javadoc et un fichier `rapport.pdf`

Soutenance. Vous aurez environ 15 minutes à votre disposition pour nous faire une démonstration de votre application, et répondre à nos questions. Tous les membres d'un groupe sont conviés à la soutenance (sous réserve d'absence justifiée). Chaque groupe est convoqué à une heure précise. Les horaires seront transmis au plus tard le 25 mars sur Moodle.

- Chaque groupe dispose de 15 minutes
 - pour préparer la démonstration (configuration de l'environnement, éditeur ouvert avec code et API du projet et toute initialisation nécessaire au lancement de l'application), et la présentation de la gestion de projet
 - pour remplir le document d'autoévaluation qui sera fourni en entrant dans la salle
- Puis il dispose de 15 minutes pour présenter son travail :
 - Sur l'application elle-même. Présentez les différentes étapes réalisées du projet. Veillez à disposer de jeux d'essai convaincants.
 - Sur les modification majeures de la modélisation UML et du schéma relationnel et les contraintes d'intégrité, l'organisation du travail au sein du groupe.
- Les 15 minutes ne pourront être dépassées : si vous n'êtes pas prêts à l'heure, le temps sera décompté.

9 Dates importantes

Les dates clés du planning sont les suivantes :

- **Mercredi 6 mars** : présentation du projet
- **Vendredi 8 mars** : validation des groupes Moodle.
- **Lundi 11 mars** : première séance Projet.
- **Vendredi 15 mars 19h** : rapport intermédiaire (UML + schéma relationnel)
- **Dimanche 29 mars 23h**: remise de l'archive avec le rapport final et le code
- **Lundi 1er avril** : soutenances. durée 15mn.