

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 7/03 au 28/03 2022, les lundis après-midi, avec une soutenance le 28/03. Il doit être réalisé par des groupes de **cinq** 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). La présence est obligatoire.

L'évaluation portera sur le travail réalisé (modélisation, qualité du 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.

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 (modèle de données, scripts SQL, projet Java, tests) devront être accessibles sur un dépôt GitLab (<https://gricad-gitlab.univ-grenoble-alpes.fr> de préférence).

Attention, ce projet est prévu pour un travail pendant les créneaux dédiés de l'emploi du temps, c'est-à-dire les lundis après-midi. Vous devez faire attention à ne pas passer trop de temps hors de ces créneaux et à ne surtout pas négliger d'autres cours. L'intérêt pédagogique de ce projet est essentiel, mais il n'est qu'une partie du contrôle continu du cours de BD.

2 Objectifs du projet

Le projet consiste en la réalisation d'une application Java de prêt/réservation accédant 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 certaines opérations, etc.) est demandée. **L'évaluation ne tiendra pas compte de la présence d'une interface graphique avancée mais uniquement de la facilité à suivre les tâches effectuées pour mener à bien une fonctionnalité, à vérifier le bon fonctionnement de l'application et la cohérence des données de la base.**

Les technologies utilisées seront : Java et JDBC pour la partie applicative, Oracle (SQL, PL/SQL) pour la partie base de données, patron DAO et/ou JPA pour l'architecture logicielle. Les technologies SPRING/SPRING BOOT sont autorisées si suffisamment maîtrisées. Une base embarquée (H2) ou une base locale (MariaDB, PostgreSQL) autre qu'Oracle est bien sûr autorisée pendant la phase de développement. Mais l'application devra, au final, pouvoir utiliser Oracle 11g de l'UFR (contrainte forte mais pas éliminatoire).

Un aspect modélisation, inhérent à tout projet de spécification d'un système d'information, fera partie intégrante de ce projet. Les modèles produits seront développés en conjonction avec le cours Systèmes d'Information (SI). Les logiciels StarUML et USE OCL seront utilisés dans le cadre de la méthode ModelScript.

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.

(A1) Chaque station est identifiée par son adresse. (A2) Elle possède plusieurs emplacements, appelés "bornettes", chacune pouvant accueillir un vélo. (A3) Chaque bornette d'une station est identifiée par un numéro et son état (OK, HS). (A4) Chaque vélo est décrit par son numéro identifiant, son modèle et la date de sa mise en service et son état (OK, HS). (A5) A chaque instant, un vélo peut être en station associée à une bornette, loué par un client, ou en maintenance. (A6) L'association d'un vélo à une bornette est automatique grâce à la puce RFID présente dans chaque vélo. (A7) Chaque station communique instantanément les changements d'état de ses bornettes à la base de données. (A8) *L'aspect "communication" sera bien entendu simulé dans ce projet.*

(A9) Le cout de location des vélos dépend du modèle, et ce cout est compté à l'heure (une heure commencée est due). (A10) *Les modèles de vélo disponible sont laissés à votre convenance.*

(A11) Les clients peuvent être ou non abonnés au service VéPick. (A12) 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). (A13) Plusieurs abonnés différents peuvent avoir le même numéro de CB (une disposition utile pour les familles). (A14) Chaque abonnement a une durée d'un an. (A15) Un abonné dispose de 30% de réduction sur le prix de location. (A16) Chaque abonné peut simplement louer un vélo en le prenant et le ramenant dans un parc après s'être identifié sur la bornette à l'aide de son code secret (manipulation réduite). (A17) Les clients non abonnés ne renseignent que leur numéro de carte CB. (A18) *La transaction bancaire ne sera pas considérée dans ce projet.* (A19) Il est possible pour un client de louer plusieurs vélos. (A20) 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. (A21) Un client peut éventuellement louer plusieurs vélos à la fois.

(A22) 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 ((A23) 5mn de délais dans ce dernier cas pour déclarer le problème).

(A24) Dans tous les cas le système ouvre la première bornette occupée par un vélo du modèle demandé dans un état utilisable lors d'une demande de location et la première bornette libre lors d'un retour. (A25) L'algorithme doit être simple et laissé à votre convenance.

(A26) Pour un usage futur d'analyse de données et de statistiques, on prendra soin de conserver dans la base de données l'ensemble des locations passées.

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

² Capitale de la Présipauté de Groland.

(A27) Certains trajets sont plus courants que d'autres (par exemple, vers le campus universitaire le matin, et depuis le campus universitaire le soir). (A28) Afin d'assurer, à chaque station, la présence de vélos disponibles et d'emplacements libres pour y déposer son vélo, VÉPick dispose d'un système de régulation citoyen. (A29) 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. (A30) Pour bénéficier de la prime, un usager doit emprunter un vélo à une station "Vmoins" et/ou le déposer à une station "Vplus". (A31) La définition de ces stations change en fonction des plages horaires, mais chaque station est toujours soit "Vmoins" soit "Vplus" soit "Vnul" (sans effet). (A32) 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 l'Interface Client (**en gras** les fonctionnalités prioritaires).

1 – Emprunt de vélo. (B1) Abonné ou non, un client doit pouvoir récupérer un vélo non endommagé à une borne. (B2) L'interface doit lui signaler le code secret pour rendre le vélo s'il n'est pas abonné. (B3) Le client doit pouvoir alerter de l'état d'un vélo en panne.

2 – Rendu de vélo. (B4) 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. (B5) Le client doit pouvoir informer de l'état d'un vélo rendu en panne. (B6) Le client est débité du temps de location.

4 – Abonnement au service. (B7) En renseignant les informations nécessaires, un client doit pouvoir s'abonner à l'année.

5 – Programme Vplus. (B8) L'application doit gérer le système de prime de ce programme. L'application doit alors signaler à un client abonné qui prend ou rend son vélo s'il bénéficie d'une remise Vplus pour son prochain trajet.

6 – Location de plusieurs vélos. (B9) Un client doit pouvoir louer plusieurs vélos à la fois (pour sa famille ou des amis par exemple).

7 – (B10) Consulter le nombre de vélos dans chaque station, le nombre de vélos endommagés, et le nombre de places libres.

5 Modélisation du schéma de base de données

Vous donnerez un modèle conceptuel (un ou plusieurs diagrammes de classes UML annotés et testés à l'aide d'un ou plusieurs diagrammes d'objets/scénarios), puis le schéma relationnel complet (tables, identifiants, identifiants externes, typage des attributs, contraintes référentielles et de domaines, formes normales des tables) ou le modèle de données avec son mapping JPA (si vous utilisez JPA dans votre projet). Les choix conceptuels devront être argumentés au besoin. **Cette question fait l'objet d'un rendu intermédiaire.**

6 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 le respect des différentes fonctionnalités et des contraintes métiers associées : 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.

Un ensemble de test devra être réalisé et documenté pour démontrer la correction de votre application. Pour cela la base de données devra être peuplée à l'aide d'un jeu de tests cohérents.

7 **Rapports et soutenance**

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

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

- Le modèle conceptuel en argumentant vos choix si plusieurs solutions apparaissent
- Le schéma relationnel complet (tables, identifiants, identifiants externes, typage des attributs, contraintes référentielles et de domaines) avec des explications si nécessaire (traduction non triviale ou choix d'implémentation par soucis d'optimisation) **OU** le modèle de données avec son mapping JPA (selon solution retenue pour le développement)
- Le document sera fourni sous la forme d'un fichier `conception_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 :

- Le rapport doit rappeler les objectifs du projet ainsi que l'organisation du document dans une brève introduction.
- Le rapport doit indiquer les modifications appliquées sur le schéma conceptuel et/ou relationnel et/ou modèle de données/mapping depuis le rapport intermédiaire.
- Le rapport doit indiquer la manière utilisée pour valider le schéma conceptuel.
- Le rapport doit faire une brève description de l'état d'avancement du projet (fonctionnalités réalisées et non réalisées), les principes des solutions proposées, la raison de non réalisation.
- Le rapport doit présenter les scénarios en support des tests et de la démo.
- Le rapport doit lister les tests réalisés et les scripts de peuplement des jeux de tests.
- Le rapport doit présenter la gestion de projet et l'organisation du travail en groupe.
- Le rapport doit contenir le lien GitLab pour accéder aux sources du projet.
- Le rapport sera Fourni dans une archive (nommée GRXX) sur Moodle.
- Le code devra être correctement commenté.
- L'application devra pouvoir être compilée et exécutée facilement sur un environnement quelconque (Windows, Linux, MacOS) et pouvoir accéder si possible au SGBD Oracle 11g de l'UFR (via VPN).

Soutenance. Vous aurez environ 15 minutes à votre disposition pour nous présenter votre application (démo, gestion projet, etc.), et 5mn répondre aux questions. Tous les membres d'un groupe sont conviés à la soutenance. Chaque groupe est convoqué à une heure précise. Les horaires seront transmis au plus tard le 22 mars sur Moodle.

8 **Dates importantes**

Les dates clés du planning sont les suivantes :

- **Lundi 7 mars** : validation des groupes projet (sur Moodle).
- **Lundi 7 mars** : démarrage du projet
- **Vendredi 11 mars 20h** : rapport intermédiaire modélisation
- **Dimanche 27 mars 23h**: Dernier commit & remise du rapport final (Moodle)
- **Lundi 28 mars** : soutenances.