

RAPPORT INTERMEDIAIRE

Introduction

Le but du projet est la réalisation d'une application Java de prêt/réservation de vélo accédant à une base de données sur un serveur distant. Nous devons, dans un premier temps, nous accorder sur une modélisation conceptuelle du problème ainsi que sur un modèle de données avec son mapping JPA. Nous allons argumenter nos choix et hypothèses dans ce rapport intermédiaire.

Dans le but de réaliser une première analyse et de comprendre les enjeux de l'application Vépick nous avons décidé de mettre en place un Glossaire ainsi qu'un document répertoriant nos hypothèses. Ces deux documents se trouvent en annexe 1 et 2.

Enfin nous avons mise en place un prototype de mapping JPA implémentant presque directement notre modèle conceptuel de donnée. Nous joindrons aussi le schéma relationnel complet.

Glossaire

A

Abonné

Cf. [A12] [A15]

Un abonné est un usager ayant acheté un abonnement valable. Ses informations personnelles sont enregistrées, et il bénéficie d'une réduction de location et s'identifie grâce à son mot de passe.

Abonnement

Cf. [A14]

Un abonnement permet à un usager d'effectuer des locations avec plus de facilités et avec un tarif réduit.

B

Borne

Cf. [A22]

Une borne est un équipement qui permet à l'utilisateur d'interagir avec VéPick afin d'effectuer la location d'un vélo. C'est l'intermédiaire entre l'utilisateur et la station.

Bornette

Cf. [A2] [A3]

Emplacement d'une station pouvant héberger un vélo.

C

Code

Cf. [A16] [A20]

Un code est un identifiant unique qui permet à chaque usager non abonné de s'identifier afin de remettre une location à une station.

Crédit-temps

Cf. [A28] [A29]

Le crédit-temps est un compte de minutes déductibles du temps de location. Il est crédité d'une minute à chaque fois qu'un abonné a effectué une location qui participe au système de régulation citoyenne

E

État

Cf. [A4] [H8]

Un vélo et une bornette ont forcément un état, OK s'ils sont en état de fonctionner ou HS en cas de problème technique signalé. Ils peuvent également avoir comme état "maintenance" s'ils sont en maintenance (parce qu'ils étaient HS / pour révision).

L

Location

Cf. [A20] [A21] [A26]

Une location est définie par une date de départ, un temps de location et un prix. Elle permet à un usager de louer un vélo.

Location Non Abonné

Cf. [A20]

Une location "non abonné" est définie par une date de départ, un temps de location, un prix, un code de carte bleu et un code secret. Elle permet à un usager non abonné de louer un vélo. Lorsqu'un usager non abonné loue un vélo, il doit renseigner son code de carte bleu. Une fois la transaction effectuée, le code est délivré.

Location Abonné

Cf. [A15] [A16] [A20] [A28] [A29]

Une location "abonné" est définie par une date de départ un temps de location et un prix. Elle permet à un usager abonné de louer un vélo plus facilement. Ainsi que de bénéficier d'une réduction de location, et d'un crédit-temps grâce au système de régulation citoyenne.

M

Modèle

Cf. [A4] [A9] [H6]

Le modèle d'un vélo correspond à un type de vélo du parc VéPick, il est associé à un nom qui l'identifie [H6] et à un prix.

Mot De Passe

Cf. [A12]

Le mot de passe permet à un abonné de s'identifier.

N

Non Abonné

Cf.[A20]

Un non abonné est un usager qui n'a pas pris d'abonnement.

R

Régulation Citoyenne

Cf. [A28] [A29]

Le système de régulation citoyenne permet à VePick d'assurer une disponibilité des vélos en remplissant les stations sur lesquelles la demande est forte et inversement de laisser des bornettes libres sur les stations de destination. Ainsi chaque station se voit attribuer un statut VStatut correspondant à l'importance que l'algorithme de VePick devra apporter à chaque station. Ce système participatif se base sur un crédit-temps accordé à chaque abonné lorsqu'il rend un vélo sur une station définie par l'algorithme plutôt que sur sa station habituelle.

Station

Cf. [A1] [A2] [A3]

Emplacement composé d'une borne elle-même constituée de plusieurs bornettes, chacune pouvant héberger un vélo. La station est située à une adresse précise.

U

Usager

Cf. [A12] [A15] [A16] [A20] [A21] [A26] [A28] [A29]

Un Usager est une personne qui utilise le service VéPick. Il peut être un abonné ou un non abonné.

V

Vélo

Cf. [A4]

Un vélo (abréviation du mot vélodipède), est un véhicule terrestre à propulsion humaine entrant dans la catégorie des cycles et composé de deux roues alignées, qui lui donnent son nom. La force motrice est fournie par son conducteur (appelé « cycliste »), en position le plus souvent assise, par l'intermédiaire de deux pédales entraînant la roue arrière par une chaîne à rouleaux. Ici un vélo a un numéro unique, une date de mise en circulation, ainsi qu'un état et un modèle.

VStatut

Cf. [A27] [A28] [A29] [A30] [A31] [A32]

Le VStatut est le statut d'une station. Il est défini manuellement par l'administrateur selon un planning. Il existe 3 statuts :

- VPlus : la station a besoin de plus de vélos
- VMoins : la station a besoin de moins de vélos
- VNul : la station n'a pas besoin spécifiquement de vélos

L'algorithme de VéPick va prioriser les stations ayant un VStatut VPlus, en proposant à l'abonné un crédit-temps s'il dépose son vélo sur une station ayant le VStatut VPlus. De même l'algorithme de VéPick va vider les stations ayant un VStatut VMoins, en proposant aux abonnés un crédit-temps s'ils récupèrent leurs vélos sur une station ayant le VStatut VMoins.

Hypothèse

Ce fichier répertorie nos hypothèses discutées en réunion et utilisées pour générer notre modèle conceptuel.

[H1] Hypothèse 1

Cf. [A14]

On considère qu'un abonné est automatiquement lié à son abonnement. Le renouvellement de l'abonnement est automatique. Un abonné peut toutefois choisir de désactiver le renouvellement automatique. Cela veut dire que l'on considèrera ici une date d'expiration de l'abonnement directement dans l'abonné. Si l'abonné a choisi de désactiver son renouvellement, une fois la date de son abonnement expirée, l'abonné ne pourra plus utiliser son abonnement.

[H2] Hypothèse 2

Comme aucun exemple de numéro de vélo nous a été donné, on considère ici que le numéro de vélo est un entier naturel qui s'incrémente de 1 à chaque nouveau vélo.

[H3] Hypothèse 3

Comme aucun exemple d'identifiant de bornette nous a été donné, on considère ici que l'identifiant de bornette est un entier naturel qui s'incrémente de 1 à chaque nouvelle bornette. De plus comme on considère qu'une bornette appartient à une station alors la bornette sera aussi identifiée par sa station. Autrement dit on introduit ici une identification relative à la station.

[H4] Hypothèse 4

On considère ici que le temps de trajet d'une location est exprimé en minutes, car cela permet éventuellement d'effectuer une déduction de temps grâce au crédit-temps qui lui est clairement exprimé en minutes.

[H5] Hypothèse 5

Cf. [A19] [A21]

On considère qu'un usager peut louer plusieurs vélos en reprenant une location à chaque vélo. Cela veut dire que

l'utilisateur peut prendre autant de location qu'il veut sans aucune limite. Cela veut aussi dire qu'il devra effectuer autant de location que de vélo qu'il souhaite louer.

On considère ici qu'une interface graphique bien conçue pour l'utilisateur permet de sélectionner le nombre de vélo qu'il souhaite louer. Et générera dans le SI autant de locations que de vélos loués.

[H6] Hypothèse 6

Cf. [A4] [A9]

On suppose ici qu'un modèle de vélo dispose d'un nom qui fait office d'identifiant unique.

[H7] Hypothèse 7

Cf. [A2]

On utilise la cardinalité 2..* car il est précisé dans le sujet qu'une station contient plusieurs bornettes, donc au moins deux.

[H8] Hypothèse 8

Cf. [A5]

On suppose ici qu'un vélo et une bornette pourront avoir un nouvel état différent d'OK ou HS : maintenance. Le sujet ne laisse pas de doute concernant la possibilité d'effectuer une maintenance sur un vélo, cependant nous avons décidé d'ajouter la possibilité de maintenance pour une bornette HS.

[H9] Hypothèse 9

Cf. [A14]

On suppose ici qu'un abonnement d'un an est automatiquement renouvelé à la fin de l'année, toutefois les abonnés pourront désactiver le renouvellement automatique s'ils ne souhaitent pas renouveler leurs abonnements.

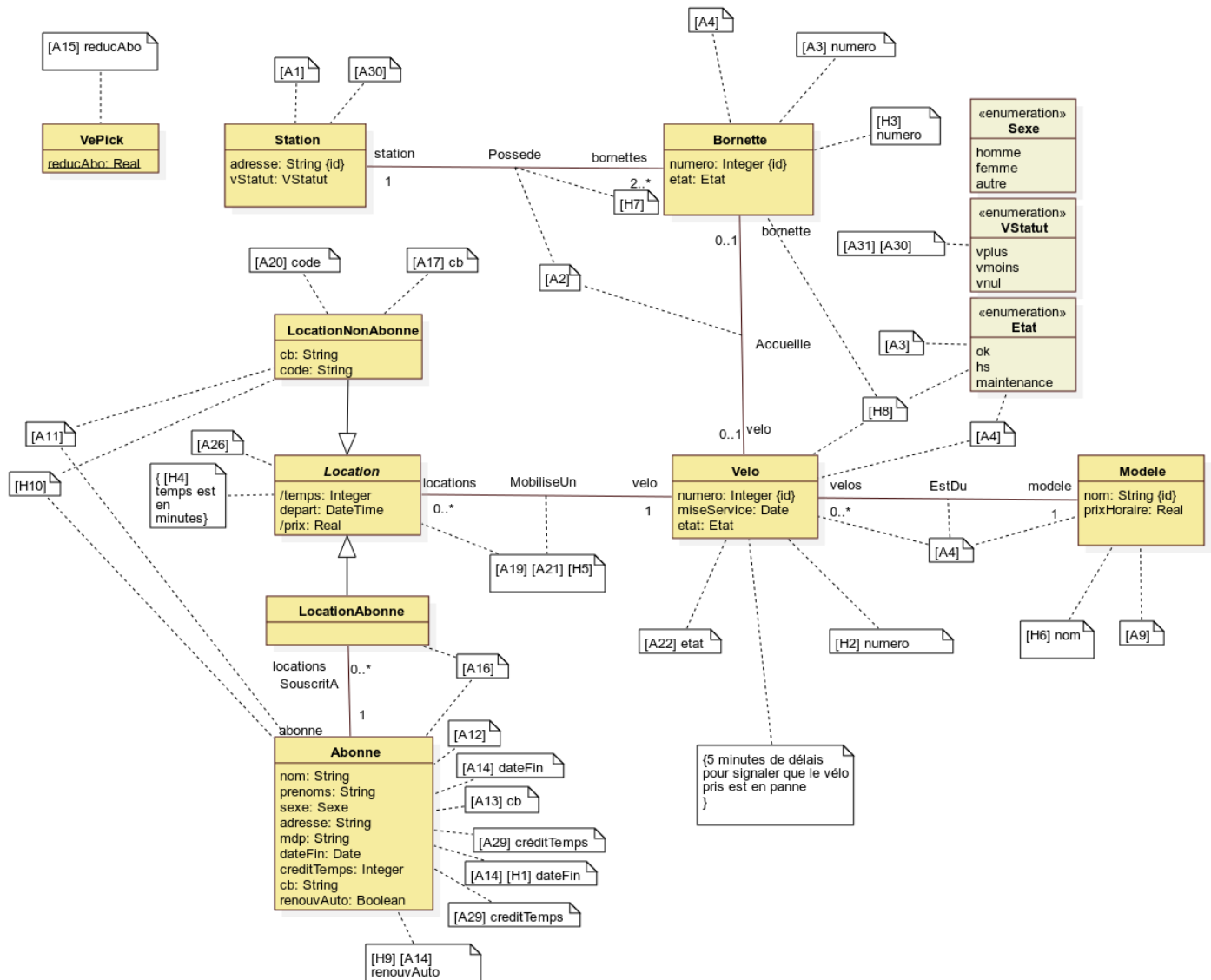
[H10] Hypothèse 10

Cf. [A16] [A20]

On estime ici que la taille maximale autorisée pour un mot de passe ou un code est de 20 caractères.

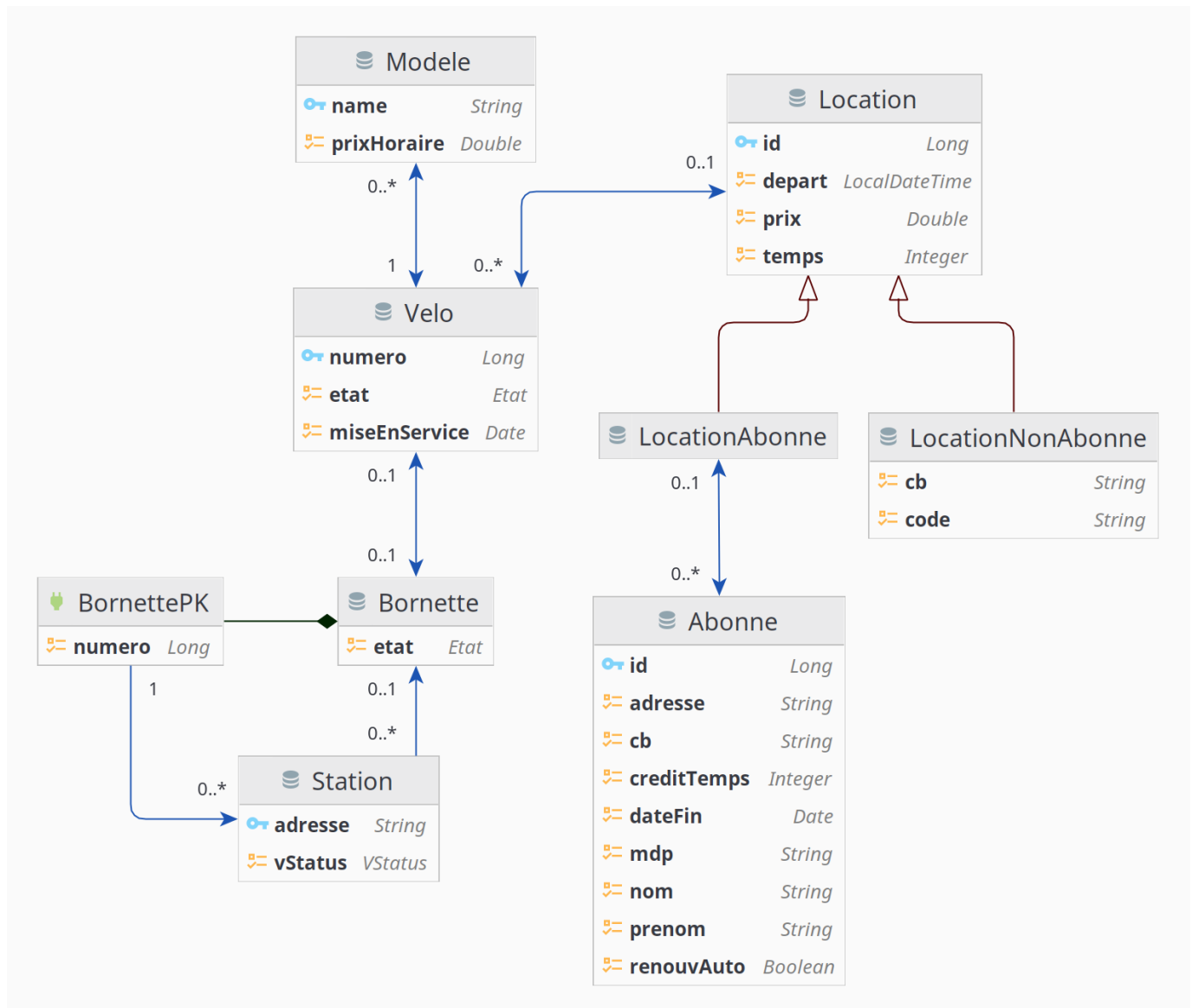
Modèle Conceptuelle de données

À noter que chaque [XX] fait référence à une phrase du texte ou à une de nos hypothèses ci-dessus.



Implémentation JPA

Nous avons fait le choix de réimplémenter presque directement notre modèle de donnée dans JPA. Pour se faire nous avons décidé d'utiliser Gradle avec la dernière version de Hibernate d'installée Voici donc ci-dessous le mapping JPA :



Choix d'implémentation

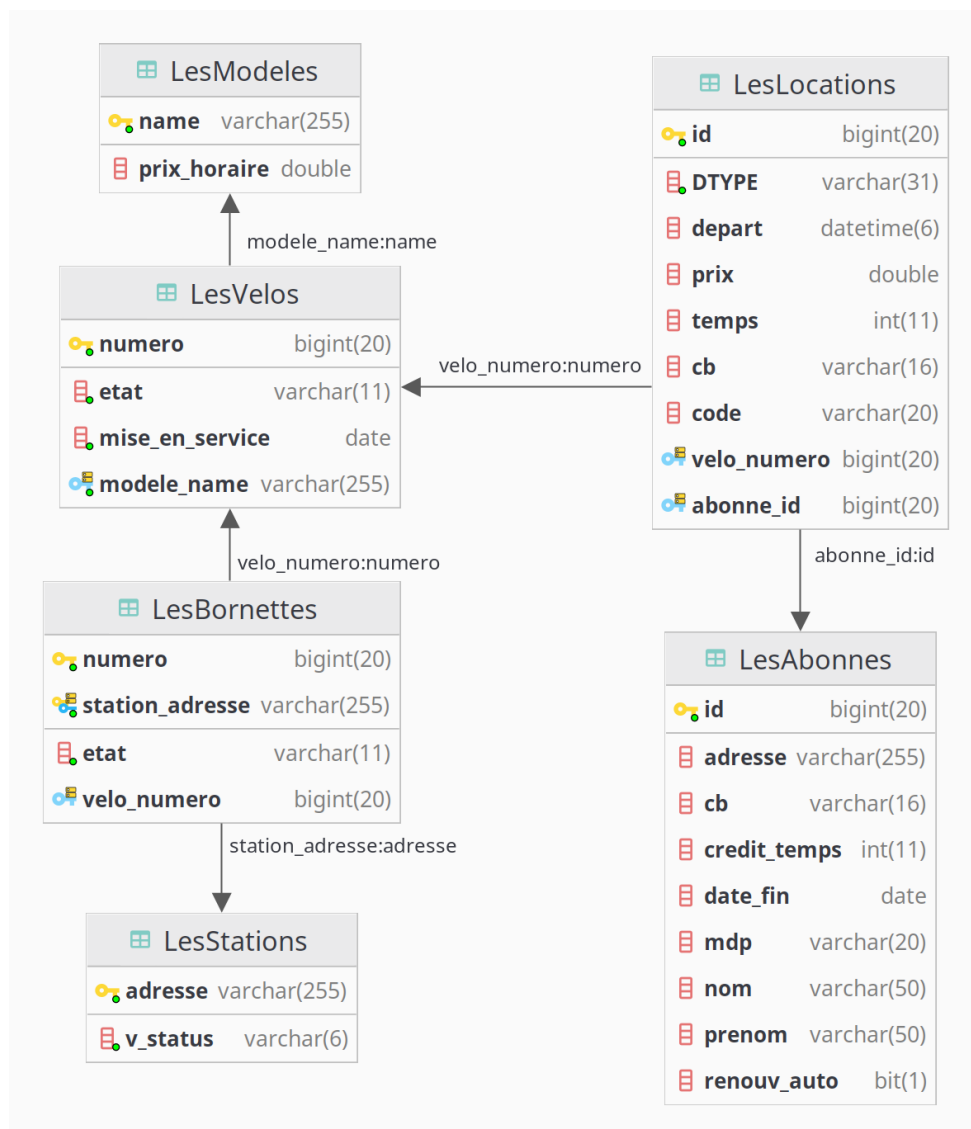
Pour implémenter notre modèle conceptuel dans JPA nous avons décidé de mettre en place quelques subtilités supplémentaires telles qu'une identification relative entre station et bornette. Elle se justifie par le fait que l'on ne veut qu'une et une seule borne par station et que lorsque l'on détruit une station, la bornette doit être obligatoirement détruite. Cette identification est représentée ici par la classe embarquée et sérialisable `BornettePK`. Cette classe a pour rôle de faire la relation entre la bornette et la station dans une unique clé primaire.

Nous avons aussi décidé de ne pas faire apparaître la classe `VéPick` dans le mapping JPA car elle ne contient qu'un seul attribut statique et unique.

Pour ce qui est de l'héritage des locations, nous avons décidé d'utiliser la stratégie JPA `SINGLE_TABLE`, qui générera une seule table généralisant les attributs des 3 classes en y ajoutant une colonne indiquant le type de la ligne.

Correspondance BDD

Une fois le mapping JPA fait nous avons généré la BDD afin d'observer le schéma généré et vérifié qu'il soit bien fondé. Voici donc ci-dessous le schéma relationnel complet de notre base de données.



Nous en sommes satisfaits car on y retrouve bien notre identifiant relatif (numero, station_adresse) sur la bornette, notre généralisation des locations avec l'introduction d'une nouvelle colonne DTYPE pour indiquer le type de location. Les énumérations se sont ici bien transformées en string. Le schéma semble être en bonne adéquation avec notre modèle conceptuel de données.

Conclusion

Cette modélisation actuelle ainsi que son début d'implémentation risquent probablement d'évoluer à mesure que notre groupe se trouvera confronté à des réalités techniques. Cette modélisation constitue cependant pour l'heure une réponse satisfaisante au besoin exprimé par le client. C'est pourquoi il serait nécessaire de valider avec lui notre compréhension de son problème, ainsi que notre modélisation.