BDR - Projet

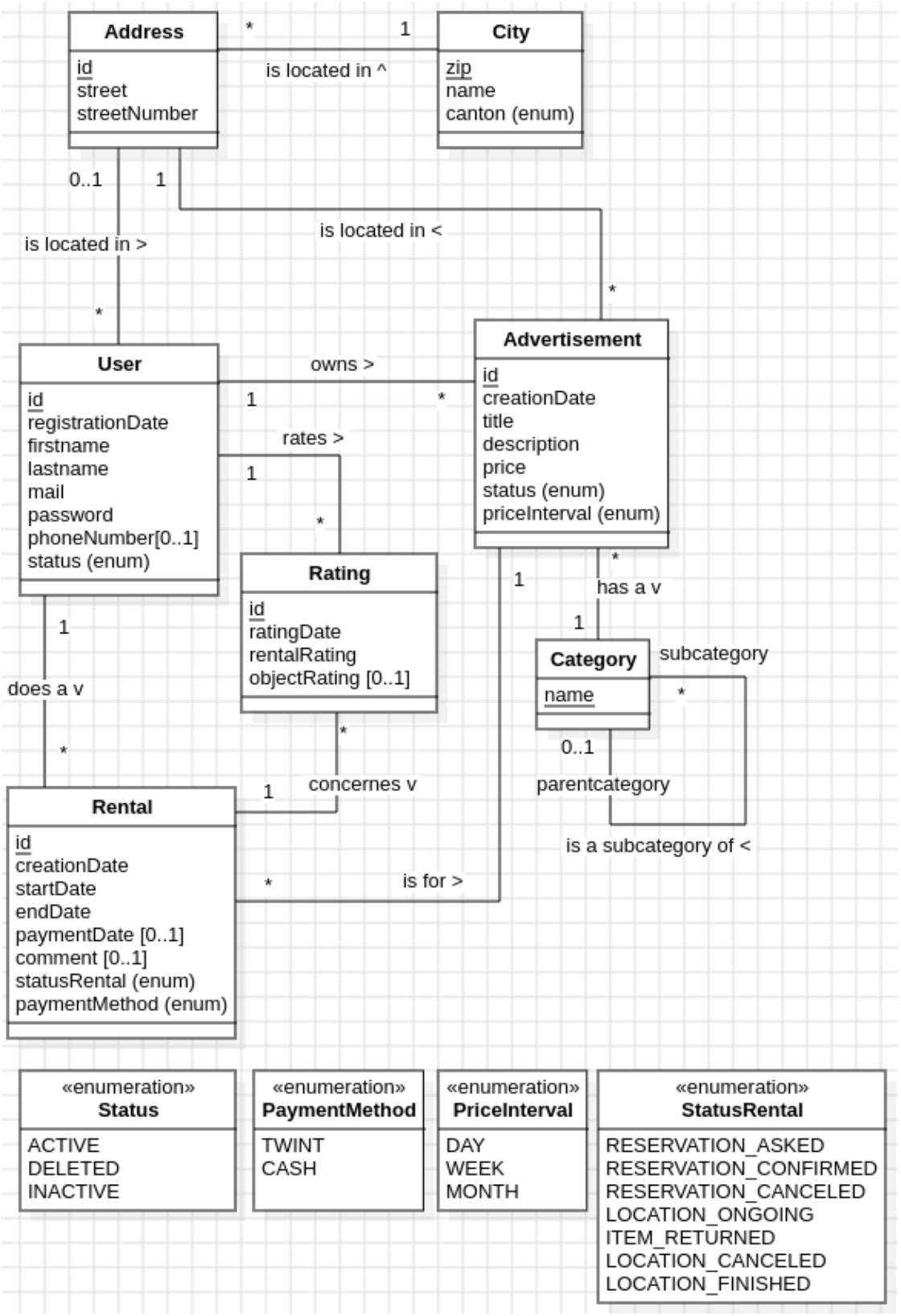
Dominik Saul, Glodi Domingos & Hugo Germano

Table des matières

1. Introduction

Nous allons réaliser un site web pour mettre en contact des utilisateurs voulant louer leurs objets et louer les objets d'autres utilisateurs. La page d'accueil se compose d'une liste des objets les plus proches de la localisation actuelle de l'utilisateur.
Les objets peuvent être des différents types comme voiture, vélo, moto, livres, dvd, bateaux, outils. Les locations sont gratuites ou payantes. On peut payer via plusieurs méthodes de payments tel quel carte de crédit, Twint et etc....

2. Modèle EA


Address
id
street
streetNumber
City
zip
name
canton (enum)
is located in ^
is located in >
is located in <
User
id
registrationDate
firstname
lastname
mail
password
phoneNumber[0..1]
status (enum)
owns >
Advertisement
id
creationDate
title
description
price
status (enum)
priceInterval (enum)
rates >
Rating
id
ratingDate
rentalRating
objectRating [0..1]
has a v
Category
name
subcategory
parentcategory
is a subcategory of <
does a v
concernes v
Rental
id
creationDate
startDate
endDate
paymentDate [0..1]
comment [0..1]
statusRental (enum)
paymentMethod (enum)
is for >
«enumeration» Status
ACTIVE
DELETED
INACTIVE
«enumeration» PaymentMethod
TWINT
CASH
«enumeration» PriceInterval
DAY
WEEK
MONTH
«enumeration» StatusRental
RESERVATION_ASKED
RESERVATION_CONFIRMED
RESERVATION_CANCELED
LOCATION_ONGOING
ITEM_RETURNED
LOCATION_CANCELED
LOCATION_FINISHED

2.1. Contraintes d'intégrités

- Un User ne peut pas louer ses propres Advertisements.
- On ne peut pas créer un Rental avec le startDate dans le passé (donc après la creationDate). Le startDate d'un Rental doit être avant le endDate. Le paymentDate doit être après la creationDate.
- Category ne peut pas être une sub-category de lui-même.
- La creationDate de Rental ne peut pas se situer avant la creationDate de Advertisement
- La creationDate de Rental ne peut pas se situer avant la registrationDate du User
- La registrationDate d'un User doit être avant le creationDate d'un Rental d'un User
- La registrationDate d'un User doit être avant le creationDate d'un Advertisement d'un User
- Un User peut seulement faire un Rating pour un Rental pour lequel il est le locataire ou le propriétaire de l'Advertisement.
- La date d'un Rating doit être après la creationDate d'un Rental
- Un Rating est un smallint et peut seulement avoir une value de 1-5
- Le User propriétaire d'un Advertisement peut seulement noter rentalRating dans le Rating
- Le User locataire d'un Rental doit noter le rentalRating et le objectRating dans le Rating
- Le streetNumber dans Address doit être strictement positif
- Le price dans Advertisement ne peut pas être négatif
- Le zip dans City ne peut pas être négatif et doit être 4 chiffres
- Le phoneNumber dans User doit être dans un format valide ('+41%[0-9]{9}')
- Le mail dans User doit être dans un format valide ('%@%.%')

2.2. Choix de conceptions

Nous avons créé l'enum Status pour pouvoir donner un Status (p.ex DELETED) à User et Advertisement au lieu de les supprimer. Parce que si on supprimerait un de ces tuples dans la base de données les relations avec Rental pourraient être invalidés.

Quand un User fait une demande de location (Rental) il peut ajouter un message initial en tant que commentaire. Dès que le propriétaire accepte le Rental un mail sera envoyé aux deux Users avec les coordonnées (email et numéro tél si existant), pour qu'ils puissent échanger plus en détail.

On va mettre à disposition pour les utilisateurs dans l'application un outil de recherche pour pouvoir filtrer les annonces selon les localisations. Ce filtrage se fait en sélectionnant les Cantons et/ou Villes souhaitées. Puisqu'on ne met pas à disposition une fonctionnalité pour filtrer les selon la distance depuis la localisation de l'utilisateur, nous enregistrons pas les localisations géographiques exactes d'une ville dans la base de données (pour calculer la distance entre deux villes).

2.3. Hypothèses

La liste des Cantons de l'enum Canton est définie selon l'art. 1 de la Constitution fédérale de la Confédération suisse.

2.4. Nice to have

Les fonctionnalités de rating (la table UserRatesUser et UserRatesObjects, l'enum Rating) que nous avons modélisé dans l'UML serait seulement implémentée si le temps le permet.

3. Modèle relationnel

City(zip, name, canton)
Address(id, zipCity, street, streetNumber)
    Address.zipCity référence City.zip NOT NULL
Category(name, parentCategory)
    Category.ParentCategory référence Category.name
User(id, idAddress, registrationDate, firstname, lastname, mail, password, phoneNumber, status)
    User.idAddress référence Address.id
Advertisement(id, idAddress, idUser, nameCategory, creationDate, title, description, price, priceInterval, status)
    Advertisement.idAddress référence Address.id NOT NULL
    Advertisement.idUser référence User.id NOT NULL
    Advertisement.nameCategory référence Category.name NOT NULL
Rental(id, idUser, idAdvertisement, creationDate, startDate, endDate, paymentDate, comment, statusRental, paymentMethod)
    Rental.idUser référence User.id NOT NULL
    Rental.idAdvertisement référence Advertisement.id NOT NULL
Rating(id, idUser, idRental, ratingDate, rentalRating, objectRating)
    Rating.idUser référence User.id NOT NULL
    Rating.idRental référence Rental.id NOT NULL

4. Application

5. Bugs

6. Conclusion