**PROJEKAT iz predmeta “BAZE PODATAKA 3”**

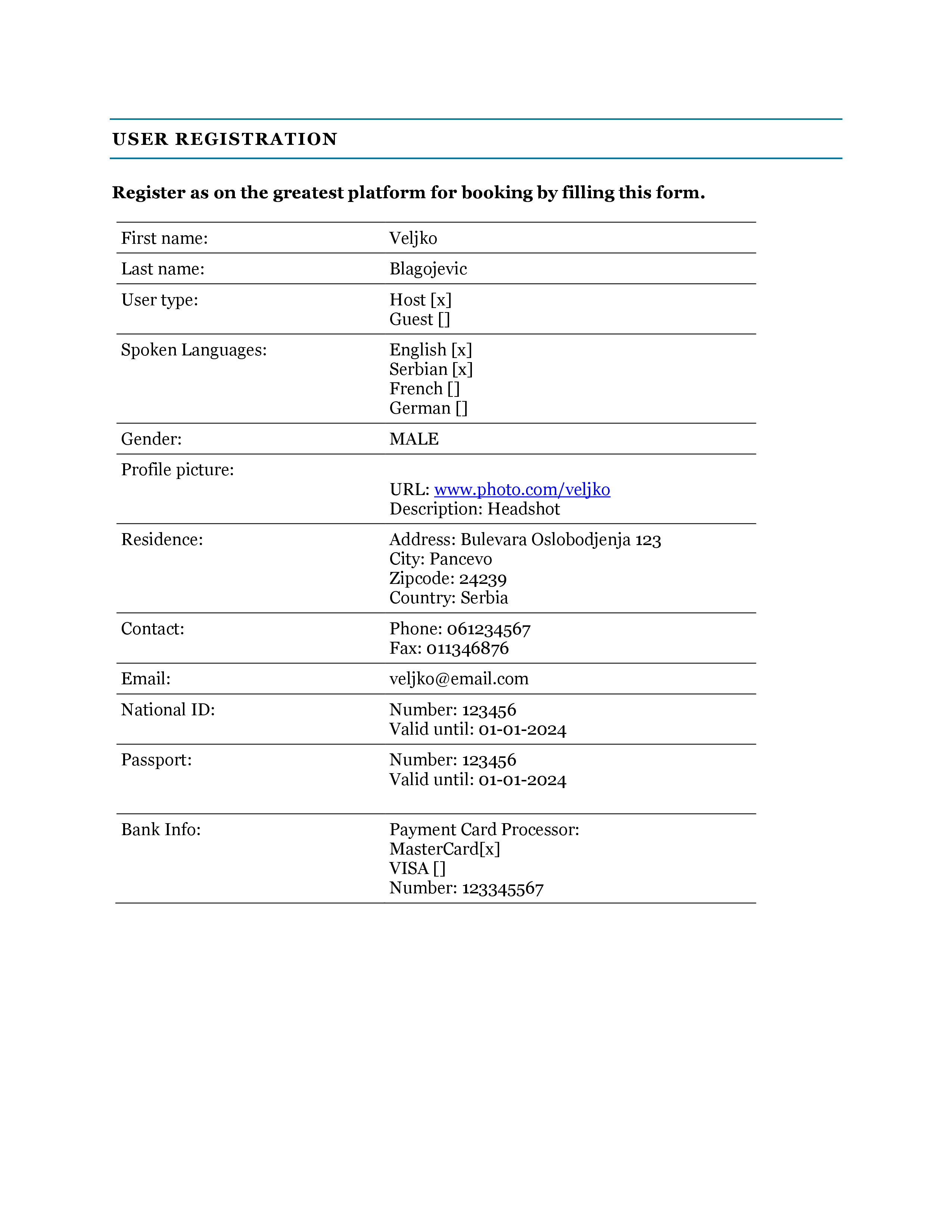
|  |  |  |  |
| --- | --- | --- | --- |
| **IME I PREZIME** | | **Veljko Blagojević** | |
| Broj indeksa | | **2023/3031** | |
| Studijski modul | | Informacioni sistemi | |
| Stečena diploma osnovnih akademskih studija | | Diplomirani inženjer organizacionih nauka | |
| **NAZIV PROJEKTA** | | Sistem za rezervaciju smeštaja | |
| Napomena | |  | |
| *Verzija* | *Datum promene* | | *Opis promene* | |
| 1.0 | 09-Nov-2023 | | **PMOV i relacioni model** | |
| 2.0 | 16-Nov-2023 | | **Denormalizacija relacija** | |
| 3.0 | 23-Nov-2023 | | **Korisnički definisani tipovi i trigeri** | |
| 4.0 | 07-Dec-2023 | | **Logička i fizička optimizacija** | |
| 5.0 | 21-Dec-2023 | | **Implementacija veb aplikacije** | |
|  |  | |  | |

# OPIS PROJEKTA

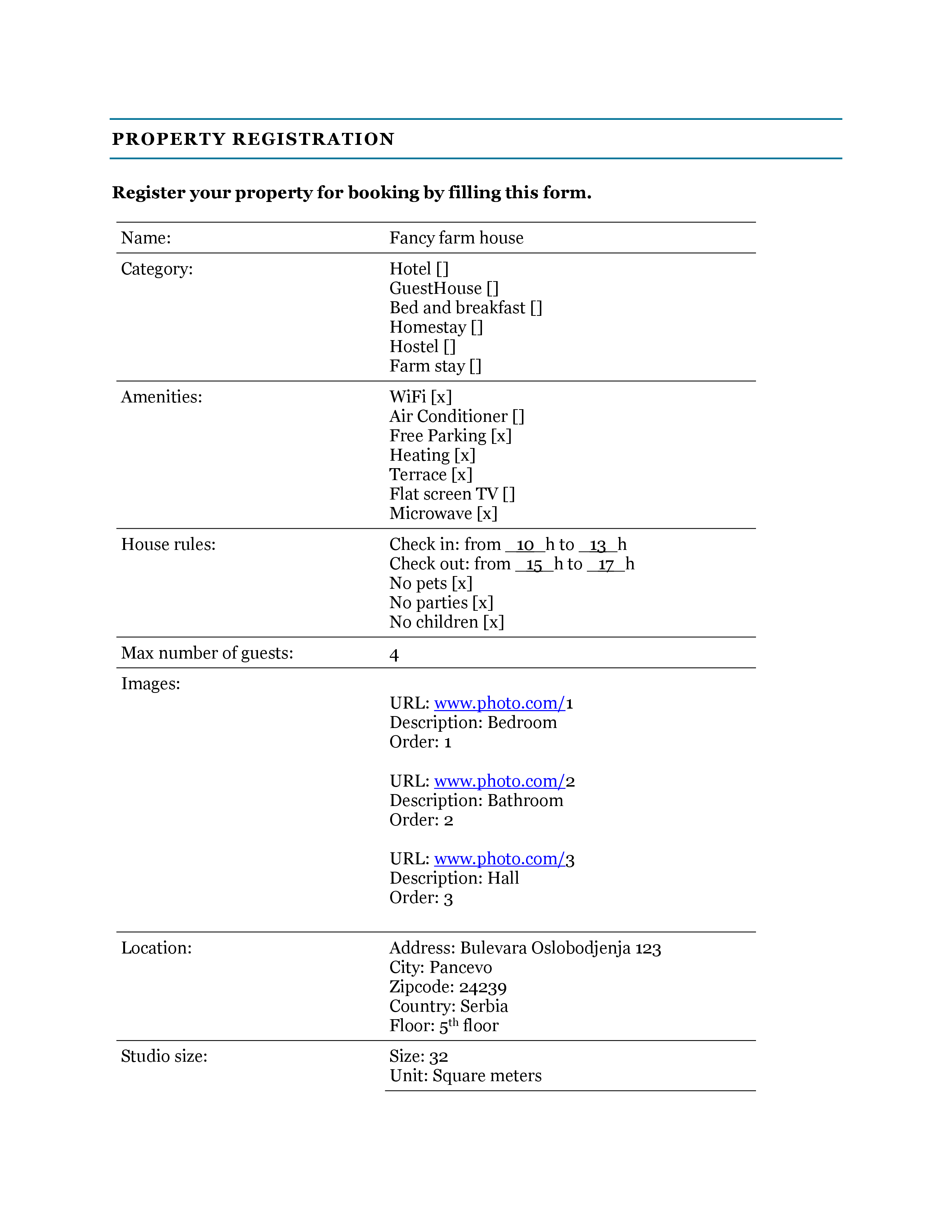
## Opis podsistema za koji se projektuje baza podataka

|  |
| --- |
| Uspešna putovanja zahtevaju kratkotrajno izdavanje smeštaja zarad odmora i prenoćišta. Online prikazivanje i rezervisanje datih objekata je daleko uprostilo ceo postupak pronalaženja kvalitetnog prostora za boravak. Korisnici, kroz informacioni sistem, registruju sebe kao domaćine sa objektima za izdavanje uz pomoć obilja opisnih informacija za svaki. Za svaki objekat je potrebno definisati lokaciju, cenu, fotografije, razne kućne potrepštine kao što su grejanje, WiFi, besplatan parking, mikrotalasna pećnica itd. Korisnik koji je gost može da izabere željeni apartman, i ukoliko je u navedenom terminu slobodan, da rezerviše i plati online karticom. Pri ovom postupku mora da preda informacije o sebi, kao i o broju gostiju za dat vid rezervacije. Nakon odsedanja, gost ima prava da napiše recenziju za dati prostor. |

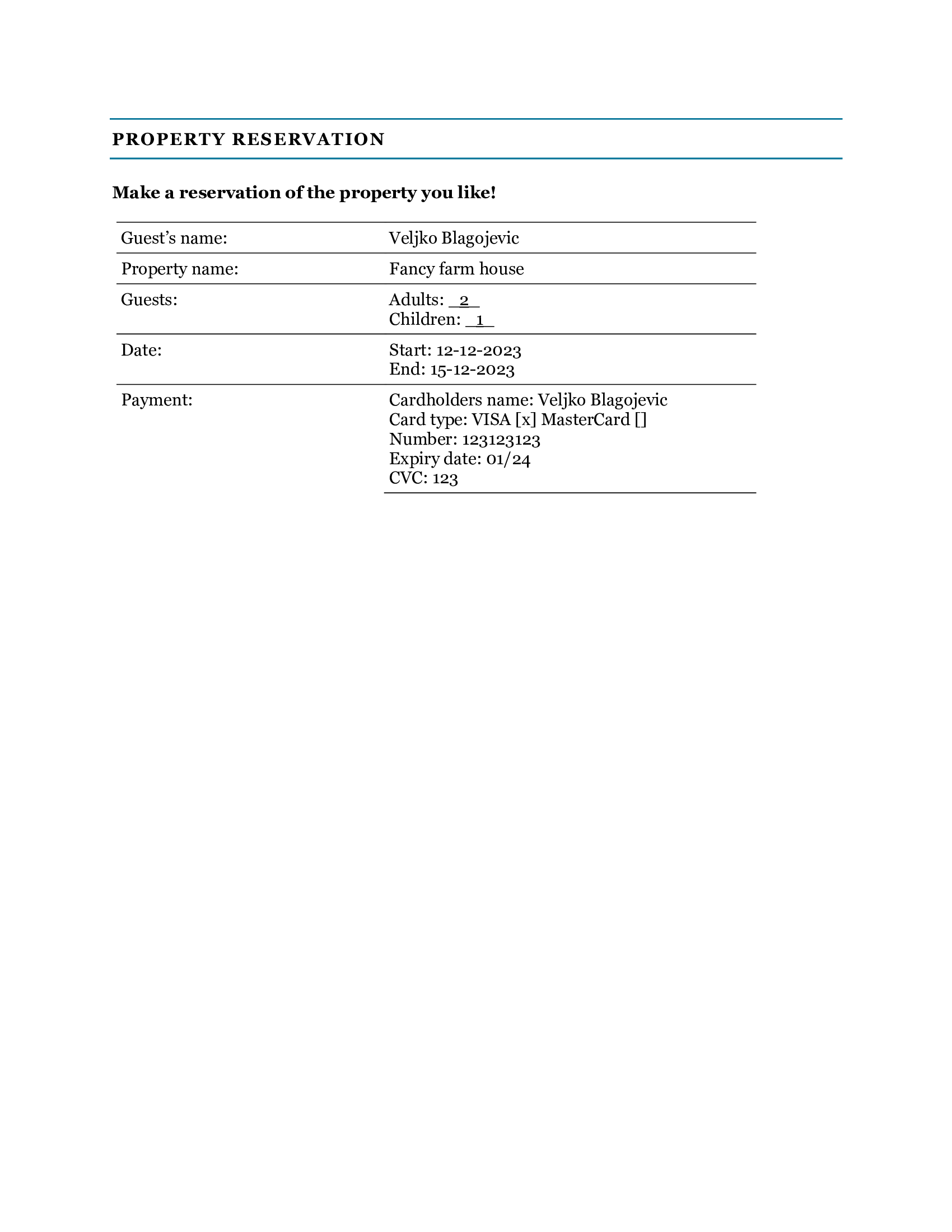
## Poslovna dokumenta koja se koriste u procesu



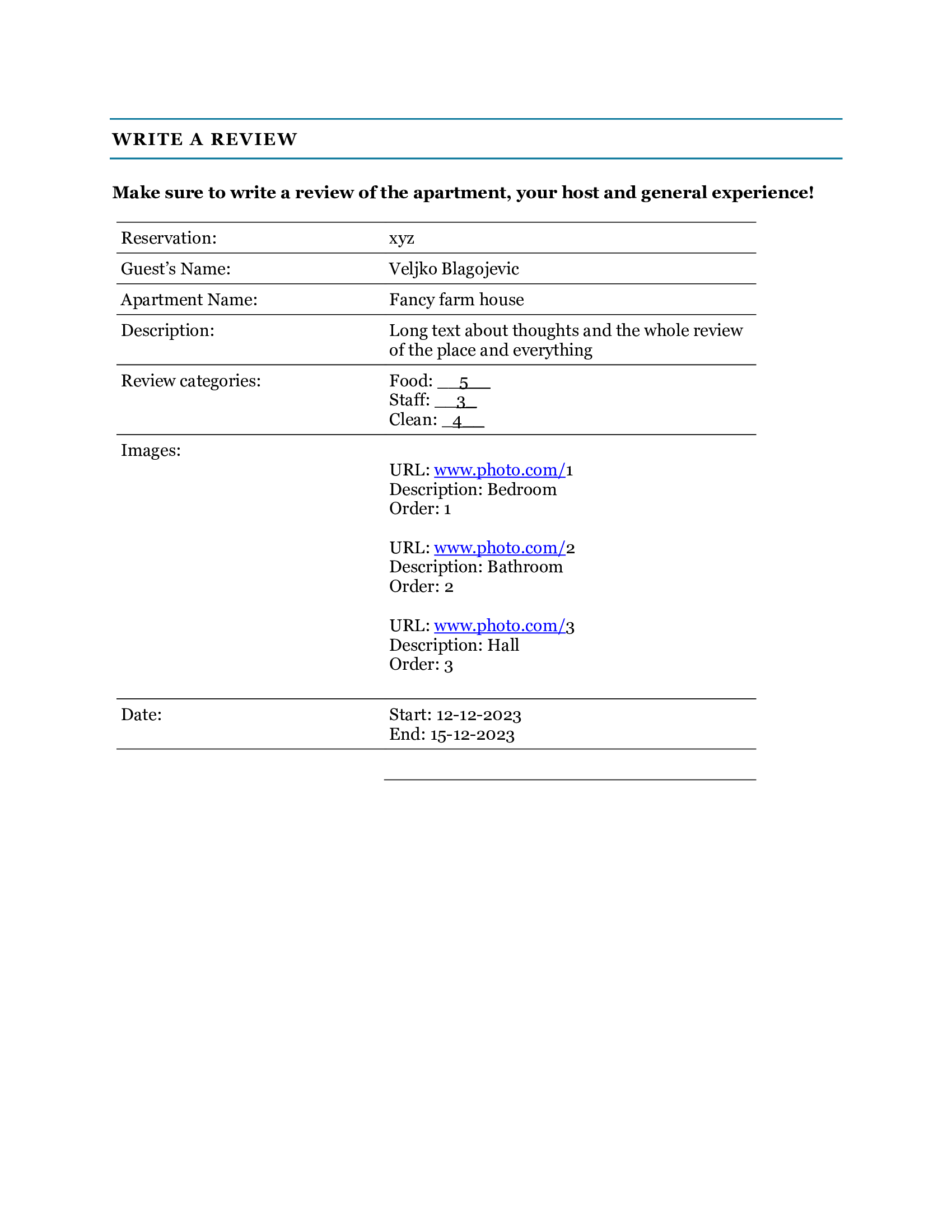
Dokument 1 Registracija korisnika



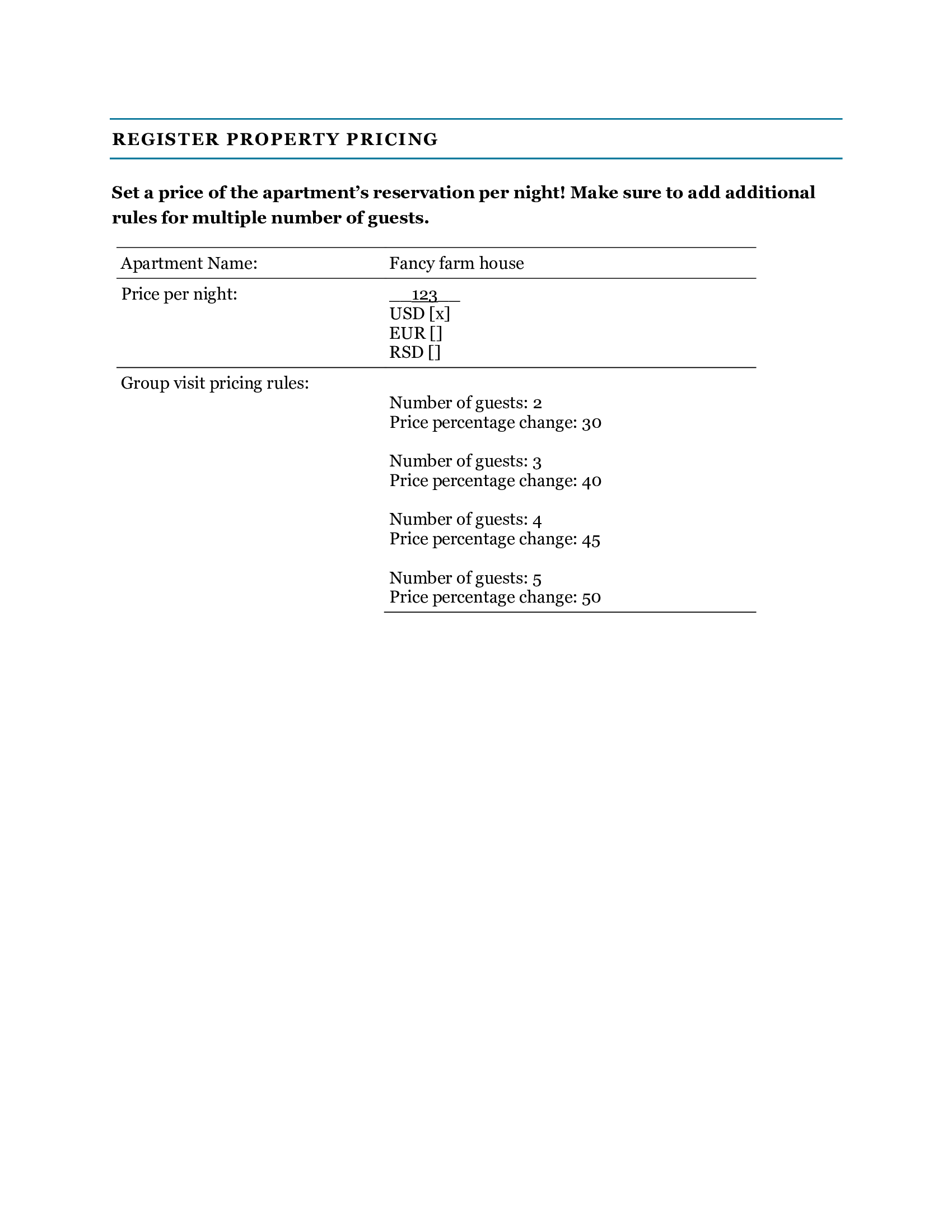
Dokument 2 Registracija smeštaja



Dokument 3 Rezervacija smeštaja



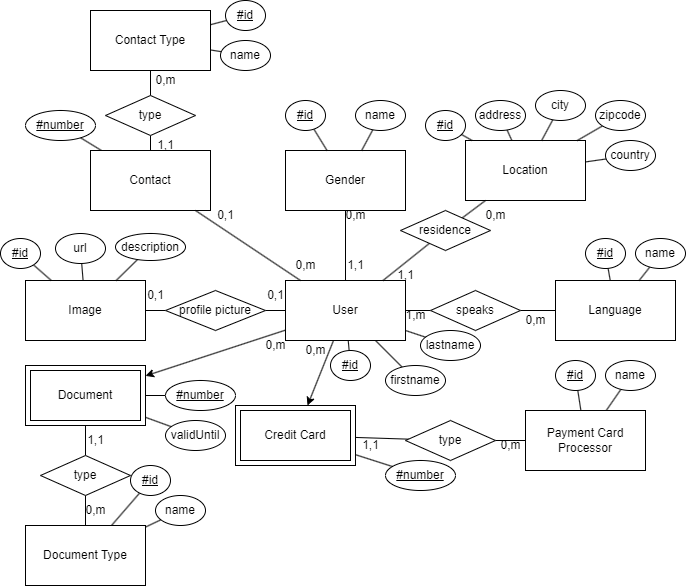
Dokument 4 Pisanje recenzije



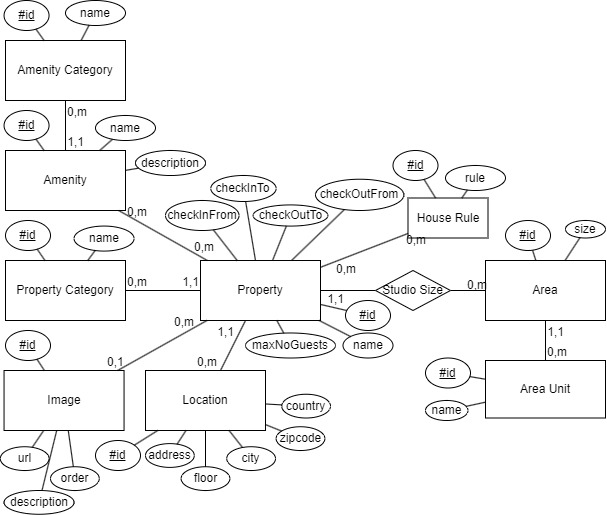
Dokument 5 Definisanje cene smeštaja

# PROJEKTOVANJE BAZE PODATAKA

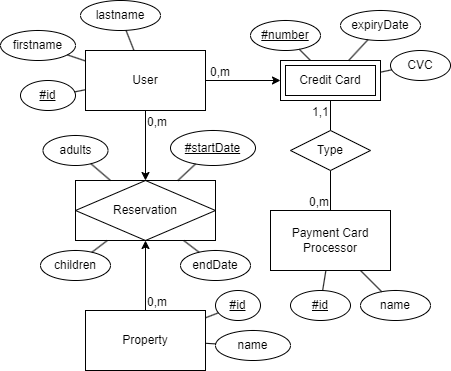
## Model podatka



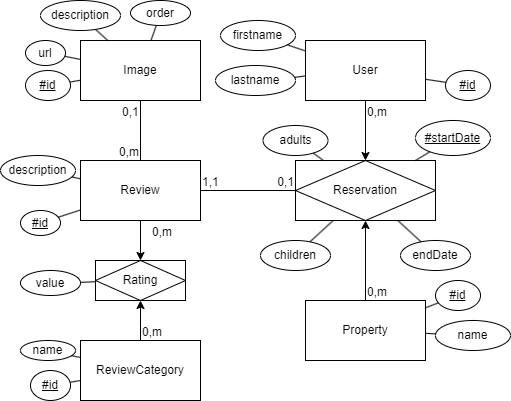
Model podataka Registracija korisnika



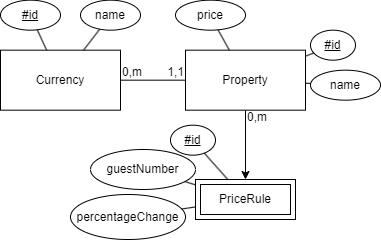
Model podataka Registracija smeštaja



Model podataka Rezervacija smeštaja



Model podataka Pisanje recenzije



Model podataka Definisanje cene smeštaja

## Relacioni model

**AmenityCategory** (amenityCategoryId, name)

**Amenity** (amenityId, name, description, *amenityCategoryId*)

**AmenityProperty** (*amenityId, propertyId*)

**PropertyCategory** (propertyCategoryId, name)

**AreaUnit** (areaUnitId, name)

**Area** (areaId, size, *areaUnitId*)

**Currency** (currencyId, name)

**PriceRule** (*propertyId*, priceRuleId, guestNumber, percentageChange)

**HouseRule** (houseRuleId, rule)

**PropertyHouseRule** (*propertyId, houseRuleId*)

**ContactType** (contactTypeId, name)

**Contact** (number, *userId*, *contactTypeId*)

**Gender** (genderId, name)

**Location** (locationId, country, city, zipcode, address, floor)

**Language** (languageId, name)

**UserSpeaksLanguage** (*userId, languageId*)

**PaymentCardProcessor** (paymentCardProcessorId, name)

**CreditCard** (creditCardNumber, *userId*, expiryDate, CVC)

**DocumentType** (documentTypeId, name)

**Document** (documentNumber, *userId*, validUntil, *documentTypeId*)

**Image** (imageId, url, description, order)

**ProfilePicture** (*userId, imageId*)

**PropertyImage** (*imageId, propertyId*)

**ReviewCategory** (reviewCategoryId, name)

**Property** (propertyId, name, price, *propertyCategoryId, currencyId*, *areaId, hostId, locationId*)

**User** (userId, name, *genderId*, *residenceId*, email)

**Reservation** (startDate*, guestId, propertyId*, adults, children, endDate)

**Review** (reviewId, description, *startDate, guestId, propertyId*)

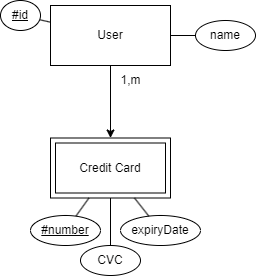
**ReviewImage** (*imageId, reviewId*)

**Rating** (*reviewId, reviewCategoryId*, value)

# DENORMALIZACIJA RELACIJA

## Denormalizacija relacija uz narušavanje 2NF (Pre-joining)

Pre denormalizacije:



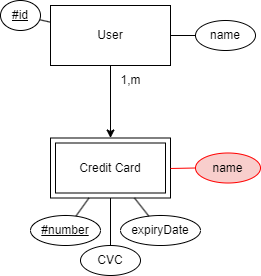
**User** (userId, name, *genderId*, *residenceId*)

**CreditCard** (creditCardNumber, *userId*, expiryDate, CVC)

**Funkcionalna zavisnost:**

userId, creditCardNumber -> CVC, expiryDate

Posle denormalizacije:



**User** (userId, name, *genderId*, *residenceId*)

**CreditCard** (creditCardNumber, *userId*, expiryDate, CVC, name)

**Funkcionalna zavisnost:**

userId, creditCardNumber -> CVC, expiryDate, name

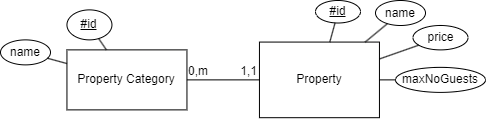
userId -> name

Specifikacija trigera:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tip trigera** | **Kolona** | **Potreban** | **Šta treba da uradi?** |
| **User** | Insert |  | NE |  |
| Update | name | DA | Prilikom izmene vrednosti name u tabeli User, pokreće se triger koji izmenjenu vrednost ažurira i u tabeli CreditCard. |
| Delete |  | NE |  |
| **CreditCard** | Insert |  | DA | Triger ažurira vrednost kolone name na osnovu unete vrednosti userId. |
| Update | userId | DA | Zabraniti direktno ažuriranje ove kolone. |
| name | DA | Zabraniti direktno ažuriranje ove kolone. |
| Delete |  | NE |  |

## Denormalizacija relacija uz narušavanje 3NF (Pre-joining)

Pre denormalizacije:



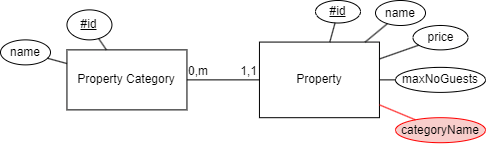
**Property** (propertyId, name, price, *propertyCategoryId, currencyId*, *areaId, hostId, locationId*)

**PropertyCategory** (propertyCategoryId, name)

**Funkcionalna zavistnost:**

propertyId -> name, price, maxNoGuests , categoryId

Posle denormalizacije:



**Property** (propertyId, name, price, *propertyCategoryId, currencyId*, *areaId, hostId, locationId,* categoryName)

**PropertyCategory** (propertyCategoryId, name)

**Funkcionalna zavisnost:**

propertyId -> name, maxNoGuests, categoryName, categoryId

propertyCategoryId -> categoryName

Specifikacija trigera:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tip trigera** | **Kolona** | **Potreban** | **Šta treba da uradi?** |
| **PropertyCategory** | Insert |  | NE |  |
| Update | categoryName | DA | Prilikom izmene vrednosti polja categoryName u tabeli PropertyCategory, pokreće se triger koji izmenjenu vrednost ažurira i u tabeli Property. |
| Delete |  | NE |  |
| **Property** | Insert |  | DA | Triger ažurira vrednost kolone categoryName na osnovu unete vrednosti atributa propertyCategoryId. |
| Update | propertyCategoryId | DA | Triger vrši ažuriranje kolone categoryName na osnovu izmenjene vrednosti kolone propertyCategoryId |
| categoryName | DA | Zabraniti direktno ažuriranje ove kolone. |
| Delete |  | NE |  |

# KORISNIČKI DEFINISANI TIPOVI I TRIGERI

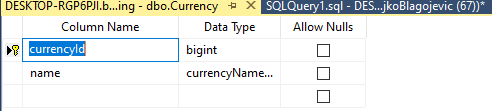
## Definicija korisničkom tipa i njegovo korišćenje

### Distinkt tip

Za primer distinkt tipa kreiran je tip CurrencyName koji se koristi unutar tabele Currency koja je data kroz sledeći relacioni model:

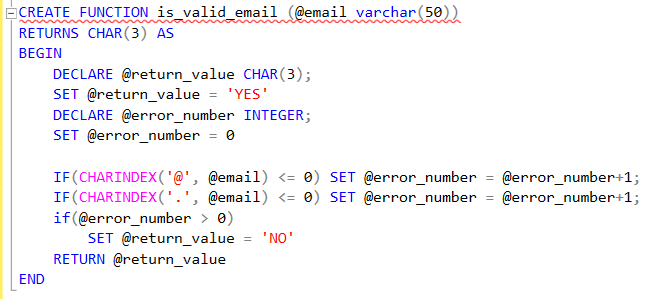
**Currency** (currencyId, name).





Pored toga je i dodata logika za validaciju email adrese koja se nalazi unutar relacije User:

**User** (userId, name, *genderId*, *residenceId*, email)



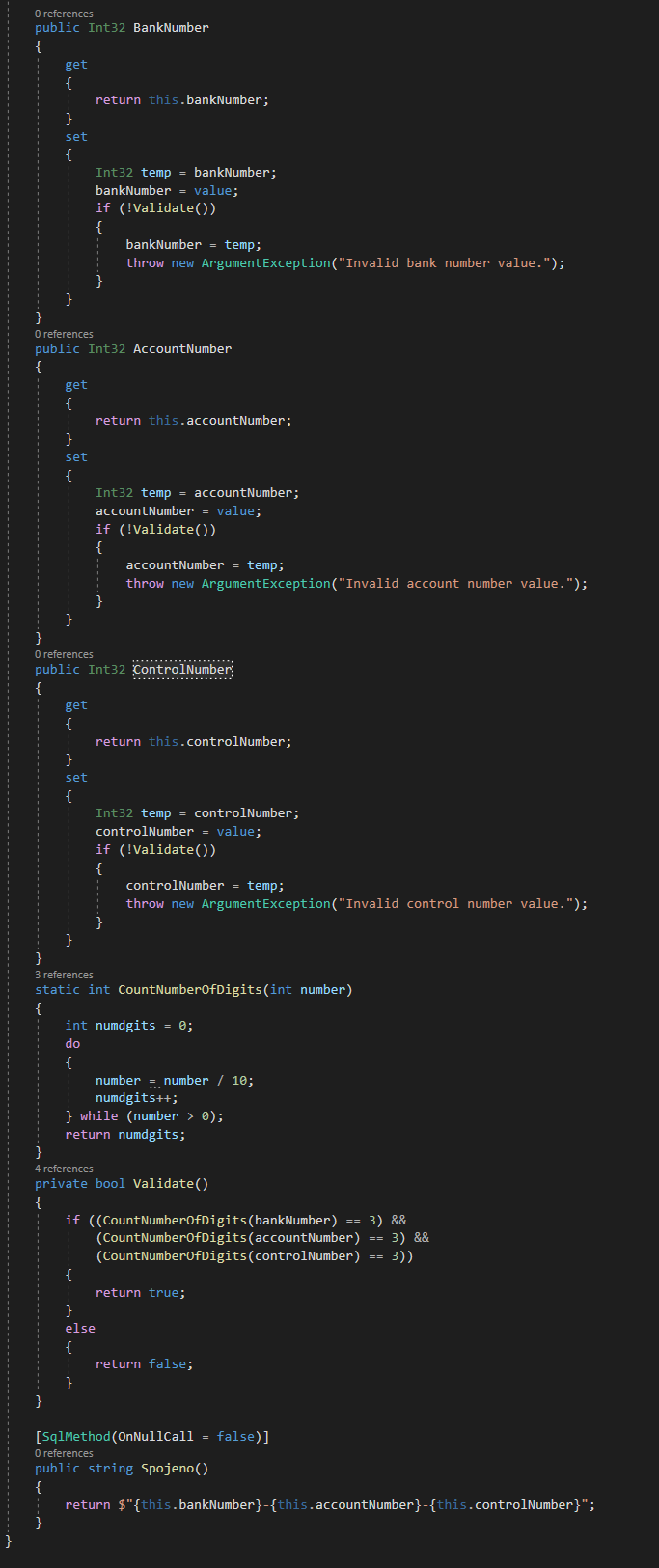
### Struktuirani tip

Za struktuirani tip je odabran broj kreditne kartice iz relacije CreditCard:

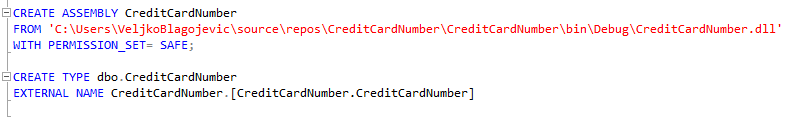
**CreditCard** (creditCardNumber, *userId*, expiryDate, CVC, placeholderName)

Prvo je potrebno kreirati strukturu kao .cs struct unutar .NET okruženja:





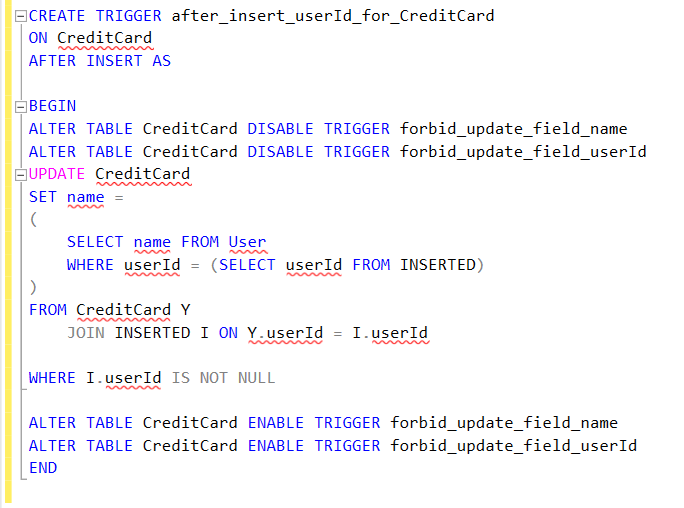
Posle toga se Class Library projekat builduje kako bi se kreirali .dll fajlovi, nakon čega se putem sledeće naredne u Microsoft SQL Server Management Studio programu kreira assembly i uvozi novi tip.

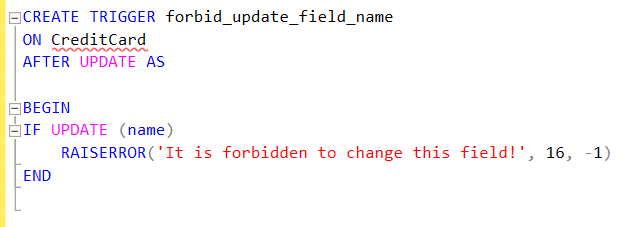


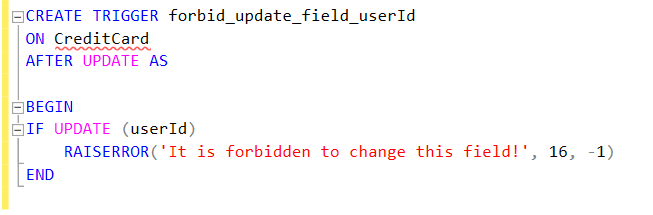
Naime je potrebno voditi računa o konfiguracionim parametrima baze kao što su clr strict security i njihova manipulacija sa sp\_configure.

## Realizacija proceduralne logike uz pomoć trigera

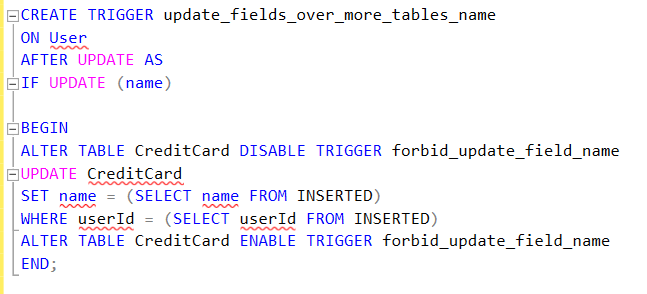
### Tabela CreditCard



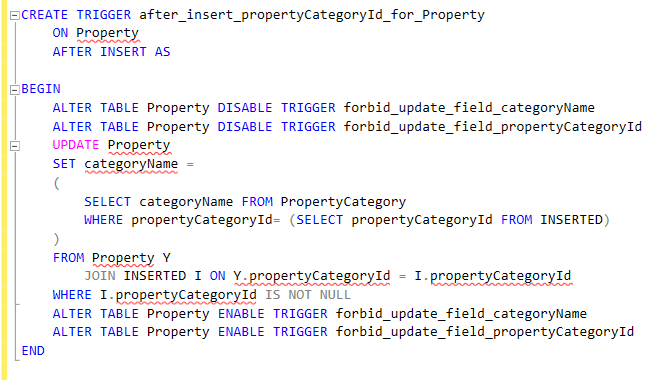


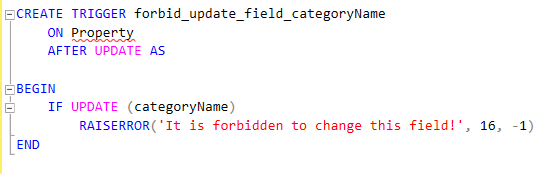


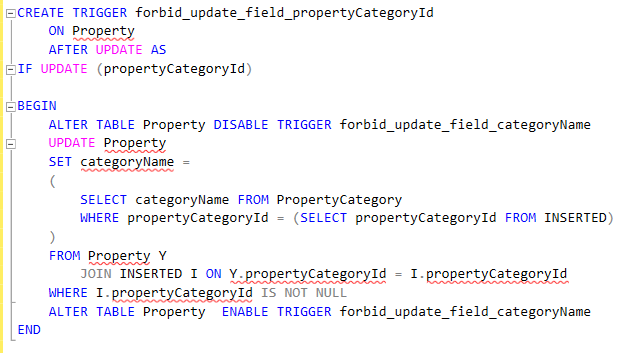
### Tabela User



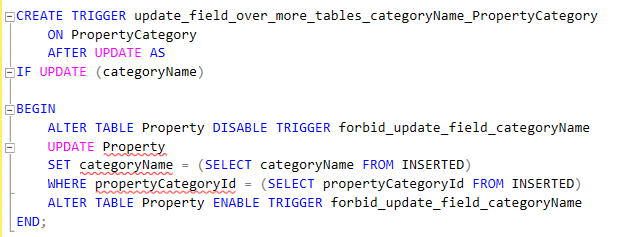
### Tabela Property







### Tabela PropertyCategory

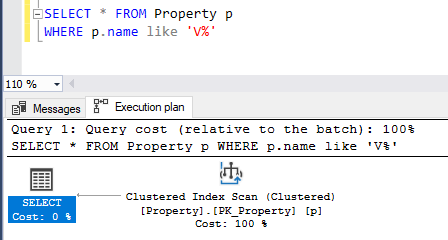


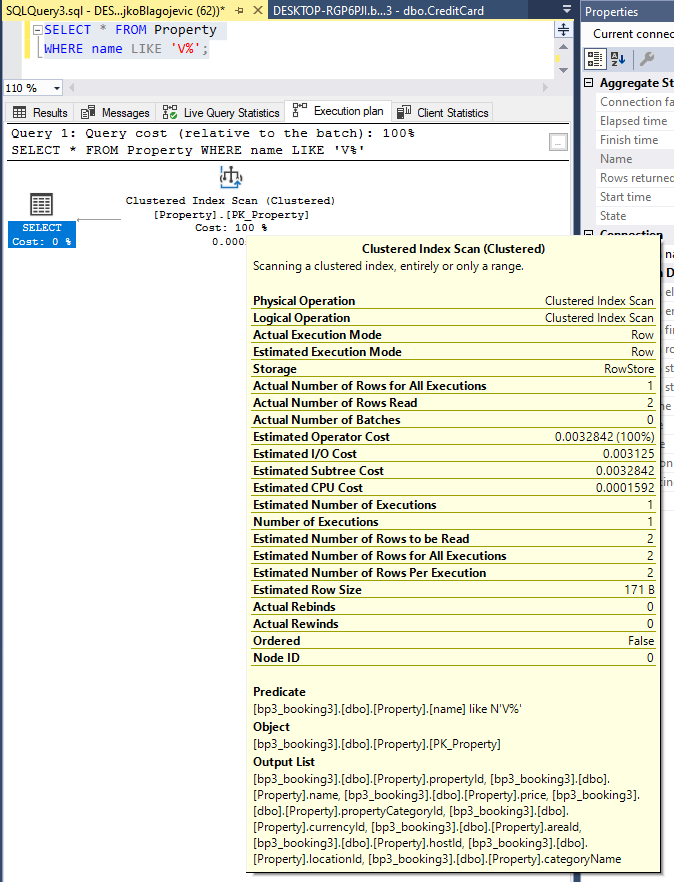
# OPTIMIZACIJA BAZE PODATAKA

## Definisanje i analiza indeksa

Kreiranje indeksa će se izvršiti nad atributom naziva unutar imovine jer će se po tome dosta pretraživati unutar same aplikacije.

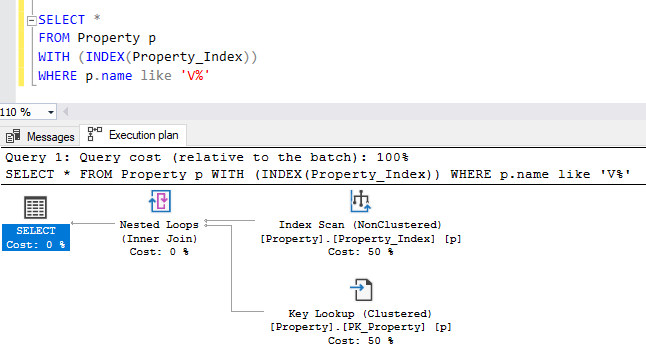
Na sledećoj slici je prikazan plan izvršavanja pre dodavanja bilo kakvog custom indeks rešenja.



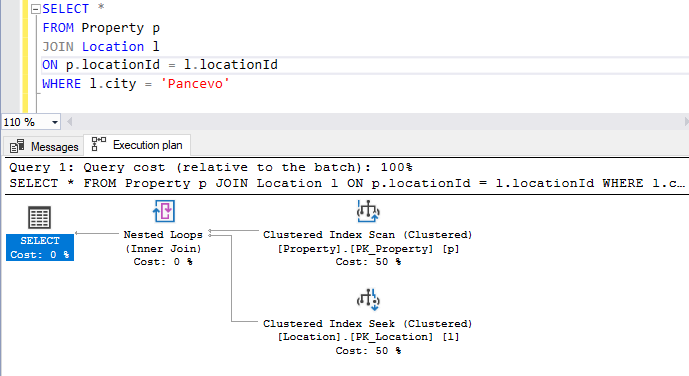




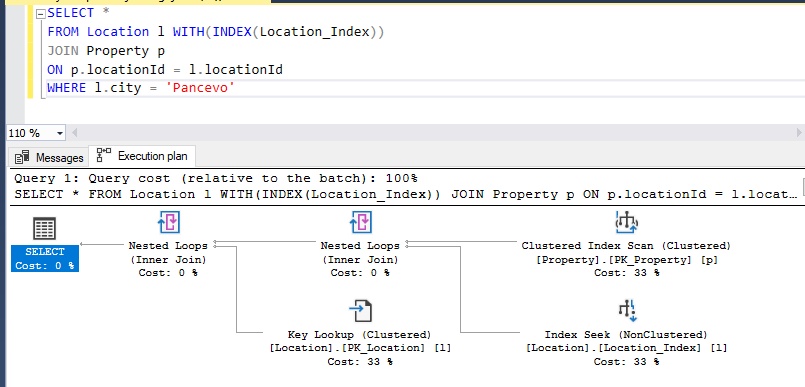
Dok je na sledećoj slici prikazan plan izvršenja nakon dodavanja indeksa.



Takođe, drugo mesto na kome bi bilo korisno da se optimizuje baza podataka putem indeksa je u tabeli za Lokaciju. Naime, ova tabela, sa svojim primarnim ključem locationId, se javlja kao spoljni ključ u velikom broju ostalih često korišćenih tabela. A pogotovo pri pretrazi imovina gde korisnik može da filtrira po željenoj lokaciji.

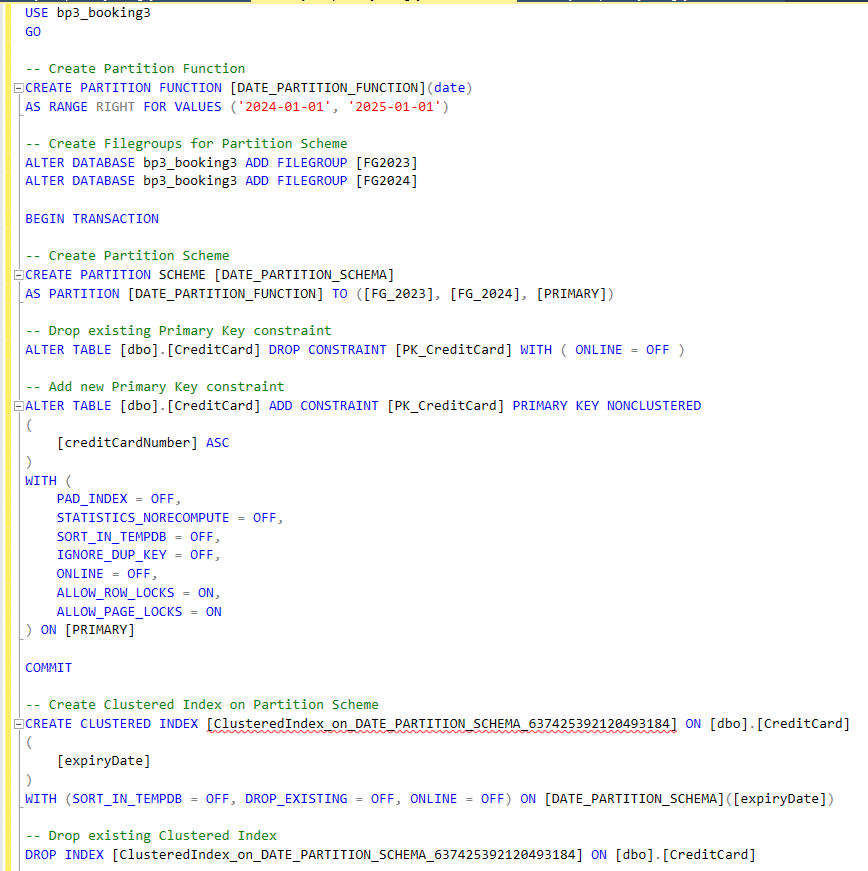




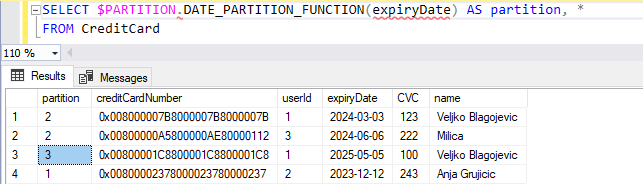


## Horizontalno particionisanje

Izvršava se horizontalno particionisanje nad tabelom CreditCard nad kolonom expiryDate gde se particije račvaju na osnovu godine.



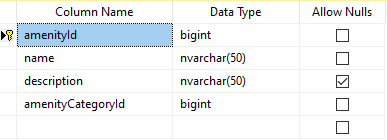
Pregled particija nad tabelom CreditCard se može videti sledećom naredbom



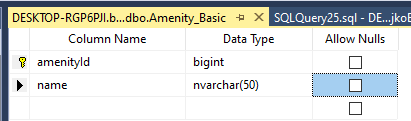
## Vertikalno particionisanje

Izvršava se vertikalno particionisanje nad tabelom Amenity:

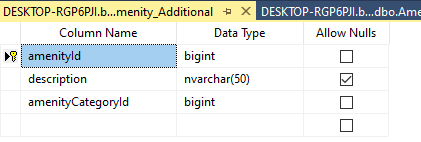
**Amenity** (amenityId, name, description, *amenityCategoryId*)



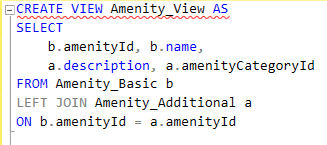
Delimo je na dve tabele, prve Basic sa osnovnim informacijama.



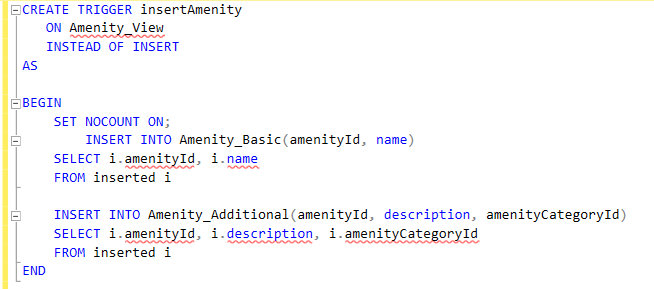
I na drugu Additional sa dodatnim podacima

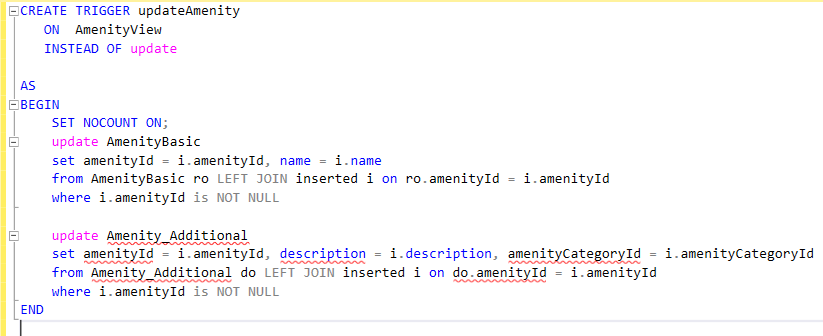


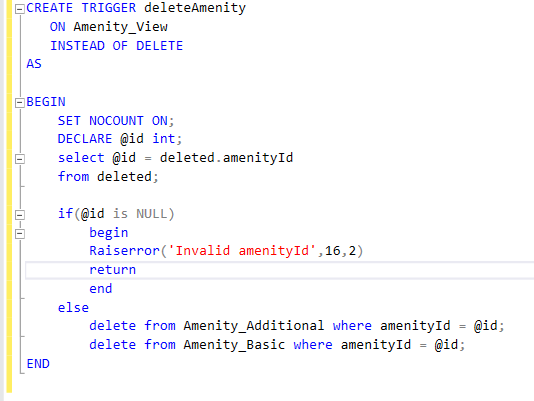
Pogled koji vraća sve podatke iz obe tabele kombinovano se kreira na sledeći način:



U nastavku su dati neophodni trigeri za rad sa tabelama nad kojima je urađen proces podele na vertikalne particije.



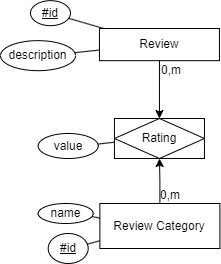




## Primena drugih optimizacionih tehnika

### Storing Derivable Values

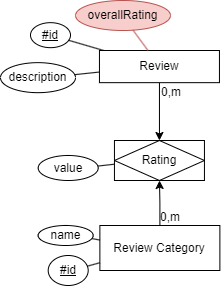
Pre optimizacije:



**Review** (reviewId, description, *startDate, guestId, propertyId*)

**Rating** (*reviewId, reviewCategoryId*, value)

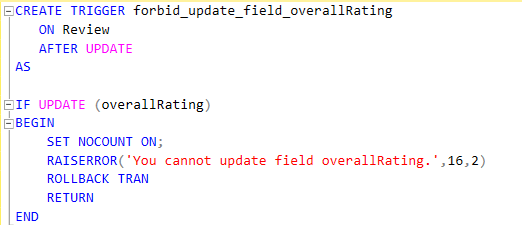
Posle optimizacije:

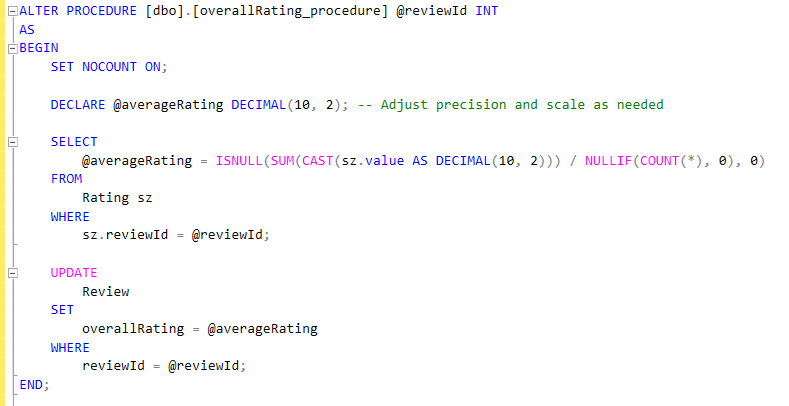


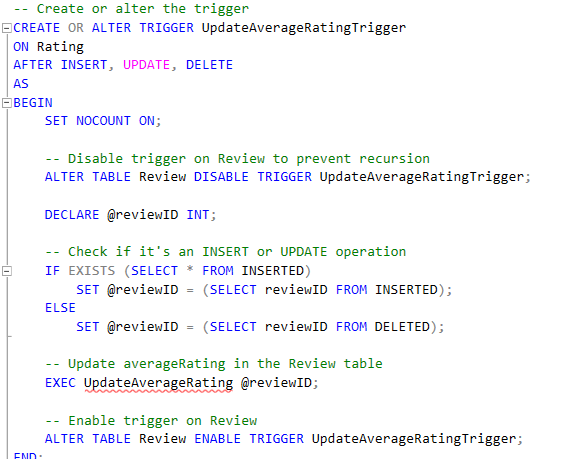
**Review** (reviewId, description, *startDate, guestId, propertyId,* overallRating)

**Rating** (*reviewId, reviewCategoryId*, value)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tip trigera** | **Kolona** | **Potreban** | **Šta treba da uradi?** |
| **Review** | Insert |  | NE |  |
| Update | overallRating | DA | Zabraniti direktno ažuriranje ove kolone. |
| Delete |  | NE |  |
| **Rating** | Insert |  | DA | Prilikom unosa nove stavke okida se triger koji poziva procedure za ažuriranje polja overallRating unutar tabele Review. |
| Update | value | DA | Prilikom ažuriranja kolone value okida se triger koji poziva proceduru za ažuriranje polja overallRating tabele Review. |
| Delete |  | DA | Prilikom brianja stavke okida se triger koji poziva proceduru za ažuriranje polja overallRating tabele Review. |

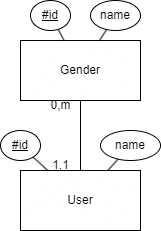






### Hard Coded Values

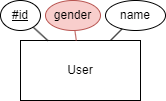
Pre optimizacije:



**Gender** (genderId, name)

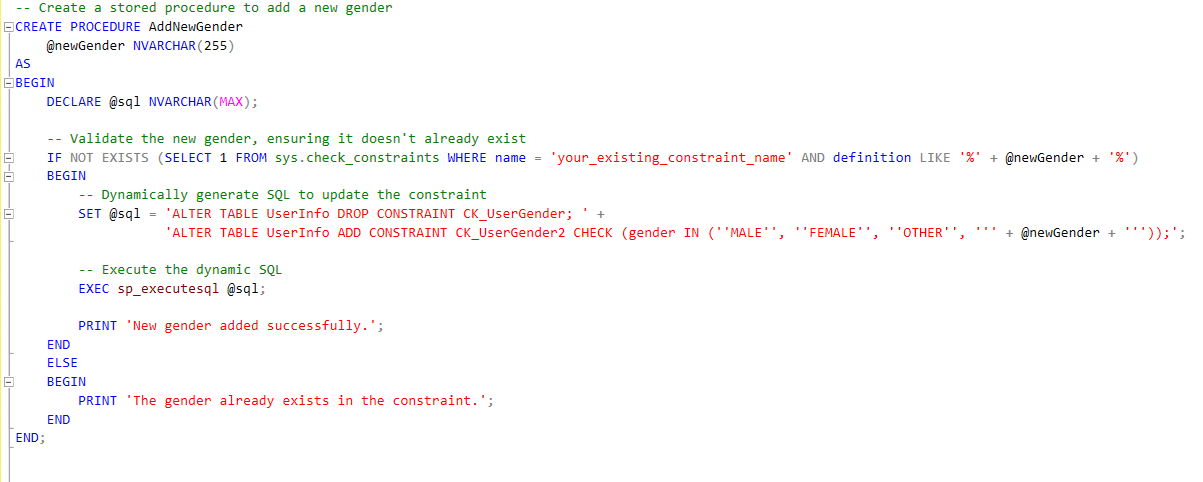
**User** (userId, name, *genderId*, *residenceId*, email)

Posle optimizacije:



**User** (userId, name, *genderId*, *residenceId*, email, gender)

Sa CHECK CONSTRAINT ograničenjem za dozvoljene vrednosti MALE, FEMALE, OTHER. Pri čemu je i napisana uskladištenja procedura čime se dozvoljava korisniku da prošiti ograničenje sa novim polovima.



# TEHNOLOGIJE ZA IMPLEMENTACIJU PROJEKTA

## SUBP korišćen za implementaciju baze podatka (kratak opis)

## Programsko okruženje za razvoj korisničkog interfejsa (kratak opis)