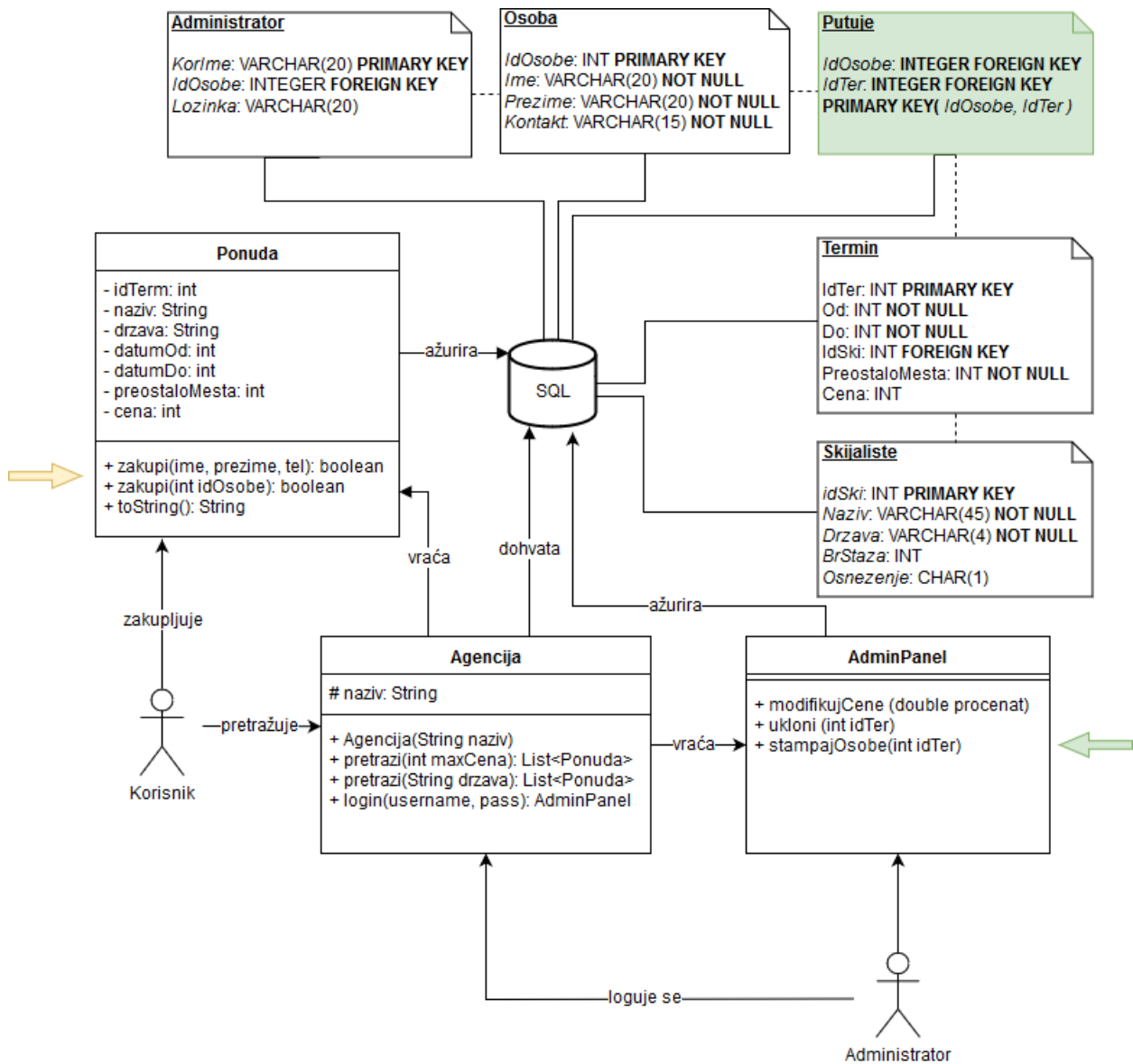


Agencija v0.3

Napravite turističku agenciju koristeći bazu podatka nalik onoj iz prethodnog domaćeg. Dat je dijagram implementacije kako bi rešenje lakše vizuelizovali kao i detaljan tekst implementacije.



Agencija

Agencija je opisana imenom i povezana je sa lokalnom bazom podataka. Agencija nudi mogućnost pretrage kreiranjem liste ponuda shodno rezultatu pretrage baze po parametru imena države ili po maksimalnoj ceni. Administratorski korisnik se može ulogovati pozivom metode login sa validnim parametrima pristupa pri čemu dobija administratorski panel sa kojim može dalje interagovati. Polje naziva može se dohvatiti.

Ponuda

Ponuda predstavlja jedinstveni aranžman agencije dobijen spajanjem skijališta i termina iz baze. Opisana identifikacionim brojem termina, nazivom skijališta, državom u kojoj se nalazi skijalište, vremenskim intervalom održavanja, preostalim broj mesta i cenom. Termin ponude se može zakupiti pozivom metode zakupi koja vraća boolean vrednost u zavisnosti da li je zakup moguće izvršiti i ažurira vrednost u bazi

ukoliko je izvršen zakup. Ponuda ima metodu toString koja vraća string u formatu (idTerm) naziv(drzava) datumOd - datumDo / cena : preostaloMesta.

AdminPanel

Administratorski panel je klasa koja predstavlja skup metoda koje administrator može da izvrši. Modifikacija cene uvećava/umanjuje sve cene za prosleđeni udeo. Administrator preko panela može i ukloniti termin iz baze prosleđivanjem njegovog identifikacionog koda, a takođe može zatražati pregled svih osoba koji su zakupili prosleđeni termin gde je ispis svake osobe u formatu: (IdOsobe) Ime Prezime - Kontakt

Kod metoda za potrebe testiranja koda i dobijanja ispisa koji ne treba dirati osim za potrebe razrešavanja dodatnih izuzetaka ili drugačije veze sa bazom podataka:

```
import java.sql.SQLException;
import java.util.List;

public class Program {
    public static void main(String[] args) {
        try {
            Agencija agencija = new Agencija("ZimoTurs");

            List<Ponuda> p = agencija.pretrazi("Bugarska");

            p.get(0).zakupi("Marko", "Kraljevic", "064987678");
            p.get(0).zakupi("Jovanka", "Orleanka", "066555666");
            p.get(0).zakupi("Jug", "Bogdan", "063853657");

            AdminPanel panel = agencija.login("admin", "admin");
            panel.stampajOsobe(p.get(0).idTerm);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Ispis koji se dobija:

```
(4) Marko Kraljevic - 064987678
(5) Jovanka Orleanka - 066555666
(6) Jug Bogdan - 063853657
```

SQL za generaciju baze:

```
DROP TABLE IF EXISTS Skijaliste;

CREATE TABLE Skijaliste (IdSki INTEGER PRIMARY KEY,
    Naziv VARCHAR(45) NOT NULL,
    Drzava VARCHAR(45) NOT NULL,
    BrStaza INTEGER,
    Osnezenje CHAR(1)) ;

INSERT INTO Skijaliste(IdSki,Drzava,Naziv,BrStaza,Osnezenje)
VALUES (1, 'Srbija', 'Kopaonik', 25, 'D'),
    (2, 'Srbija', 'Zlatibor', 4, 'N'),
    (3, 'Francuska', 'Les 2 Alpes', 96, 'D'),
    (4, 'Francuska', 'Paradiski', 253, 'D'),
    (5, 'Francuska', 'Chatel', 49, NULL),
    (6, 'Bugarska', 'Bansko', 18, 'D');

DROP TABLE IF EXISTS Termin;

CREATE TABLE Termin (IdTer INTEGER PRIMARY KEY,
    Od INTEGER NOT NULL,
    Do INTEGER NOT NULL,
    IdSki INTEGER,
    Cena INTEGER,
    PreostaloMesta INTEGER NOT NULL) ;

INSERT INTO Termin(IdTer,Od,Do,IdSki,Cena, PreostaloMesta)
VALUES (1, 20181229, 20190105, 1, 350, 4),
    (2, 20190105, 20190112, 1, 230, 0),
    (3, 20190126, 20190202, 6, 165, 3),
    (4, 20190125, 20190203, 3, 396, 2),
    (5, 20190201, 20190210, 3, 410, 5),
    (6, 20190308, 20190317, 3, 379 , 6);

DROP TABLE IF EXISTS Administrator;
DROP TABLE IF EXISTS Osoba;

CREATE TABLE Osoba (
    IdOsobe INTEGER PRIMARY KEY AUTOINCREMENT,
    Ime VARCHAR(20),
    Prezime VARCHAR(20),
    Kontakt VARCHAR(15) UNIQUE
);

INSERT INTO Osoba (Ime, Prezime, Kontakt) VALUES
("Boban", "Dundjer", "+38163990938"),
("Jovo", "Denda", "+381673993830"),
("Milan", "Mladenovic", "+38109494829");
```

```
CREATE TABLE Administrator(  
    KorIme VARCHAR(20) PRIMARY KEY,  
    IdOsobe INTEGER,  
    Lozinka VARCHAR(20),  
    FOREIGN KEY(`IdOsobe`) REFERENCES `Osoba`(`IdOsobe`) ON UPDATE CASCADE ON DELETE RESTRICT  
);  
  
INSERT INTO Administrator(IdOsobe, KorIme, Lozinka)  
VALUES (1, "admin", "admin");  
  
DROP TABLE IF EXISTS Putuje;  
CREATE TABLE Putuje(  
    IdOsobe INTEGER,  
    IdTer INTEGER,  
    FOREIGN KEY(`IdOsobe`) REFERENCES `Osoba`(`IdOsobe`) ON UPDATE CASCADE ON DELETE CASCADE  
    FOREIGN KEY(`IdTer`) REFERENCES `Termin`(`IdTer`) ON UPDATE CASCADE ON DELETE CASCADE  
    PRIMARY KEY (IdOsobe, IdTer)  
);
```