

PRIMER - JDBC

SISTEM MOBILNE TELEFONIJE

Sistem mobilne telefonije opslužuje svoje korisnike od kojih svaki može da ima jednu ili više SIM kartica u postpejd režimu (plaćanje je putem mesečnih računa). Obračun se vrši po sekundama razgovora odnosno broju znakova poruka. Svaki račun osim opštih podataka ima i stavke, od kojih svaka odgovara jednoj pruženoj usluzi.

Korisnicima su na raspolaganju usluge poziva za razgovor i slanja SMS poruka. Slanje SMS poruke uvek uspeva - poruka se zapisuje u bazi podataka i trajno čuva, a prosleđuje se odredištu čim je to moguće. Svaki poziv za razgovor se trajno evidentira i može da ima tri ishoda:

- Uspešno uspostavljena veza sa pozvanim brojem ; na kraju razgovora, evidentira se njegovo trajanje;
- Zauzetost pozvanog broja: automatski se generiše poruka o zauzeću koju će sistem poslati pozivaru čim pozvani prestane da bude zauzet;
- Nedostupnost pozvanog broja: automatski se generiše poruka o dostupnosti koju će poslati pozivaru čim pozvani postane dostupan.

Automatski generisane poruke se raspoznaju po tome što je identifikacija pošiljaoca NULL. Aktor u ovom sistemu nije čovek nego uređaj - kontroler koji na početku poziva daje interfejsu odgovarajuće podatke o pozivu a posle toga, ako je razgovor obavljen, odgovarajuće dodatne podatke.

Tabele u bazi podataka su:

```
KORISNIK (IDKor, PrezimeIme, Adresa)
SIM_KARTICA (IDSIm, TelBroj, Status, IDKor)
USLUGA (IDUsl, IDSImOd, IDSImDo, DatumVremeOd, Status, Iznos)
POZIV (IDUsl, DatumVremeDo)
PORUKA (IDUsl, TekstPoruke)
CENA (IDCen, CenaPoziva, CenaPoruke, Datum)
RACUN (IDRac, ZaMesec, IDKor)
STAVKA (IDRac, RedBr, Iznos, IDUsl)
```

Sastaviti Java metodu koja koristeći JDBC implementira funkcionalnost ObradaPocetkaPoziva.

```

public boolean ObradaPocetkaPoziva (Connection konekcija,

                                     int idUsluge,

                                     int idSimOd,

                                     String telBrojDo,

                                     Timestamp datumVremeOd){

//odredjivanje broja pozivajuceg

String telBrojOd;

Statement upitTelBrojOd = konekcija.createStatement();

ResultSet rezultatTelBrojOd = upitTelBrojOd.executeQuery("
SELECT TelBroj FROM SIM_KARTICA WHERE IDSimOd =" + idSimOd);

while(rezultatTelBrojOd.next()){
    telBrojOd = rezultatTermin.getString(1);
}

upitTelBrojOd.close();
rezultatTelBrojOd.close();

//provera dostupnosti pozvanog

int idSimDo;
Boolean statusPozvanog;

Statement upitSimStatus = konekcija.createStatement();

ResultSet rezultatSimStatus = upitSimStatus.executeQuery("
SELECT IDSim, Status FROM SIM_KARTICA WHERE telBroj =" + telBrojOd);

while(rezultatSimStatus.next()){
    idSimDo = rezultatSimStatus.getInt(1);
    statusPozvanog = rezultatSimStatus.getBoolean(2);
}

upitSimStatus.close();
rezultatSimStatus.close();

if(statusPozvanog){

//evidentiranje pocetka poziva

Statement dodajUslugu = konekcija.createStatement();

dodajUslugu.executeUpdate("INSERT INTO USLUGA VALUES (" + idUsluge + "," + idSimOd + "," +
idSimDo + "," + datumVremeOd + "," + " 'N', 0");

dodajUslugu.close();

Statement dodajPoziv = konekcija.createStatement();

dodajPoziv.executeUpdate("INSERT INTO POZIV VALUES (" + idUsluge + ", NULL");

dodajPoziv.close();

} else {

```

```
//obrada propustenog poziva
Statement dodajUslugu = konekcija.createStatement();

dodajUslugu.executeUpdate("INSERT INTO USLUGA VALUES (" + idUsluge + ", NULL," + idSimDo +
", " + datumVremeOd + ", " + " `N`, 0");

dodajUslugu.close();

String tekstPoruke = "Imate propusten poziv od broja" + telBrojOd + "u " + datumVremeOd;
Statement dodajPoruku = konekcija.createStatement();

dodajPoruku.executeUpdate("INSERT INTO PORUKA VALUES (" + idUsluge + ", " + tekstPoruke);

dodajPoruku.close();

}

return true;
}
```