Informacioni sistemi

II deo

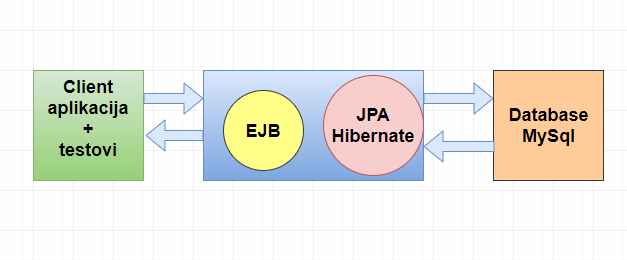
*Pavle Marinkovic 15705*

**Tekst zadatka:**

96. Informacioni sistem službe za registraciju vozila.   
Pretpostaviti da služba za registraciju vozila predstavlja posebnu organizaciju. Sistem vodi evidenciju o svim registrovanim vozilima, njihovim vlasnicima i pristiglim zahtevima za registraciju. Zahtevi za registraciju se podnose na šalteru. Nakon toga se prisleduju službama za dalju obradu tj. odobravanje ili odbijanje zahteva. Placanje takse za registraciju vozila moguce je izvršiti elektronskim putem ili uplatom na bankovni racun službe. Svi relevantni podaci cuvaju se u centralizovanoj bazi podataka.

Obrazac: Upravljanje podacima proizvoda

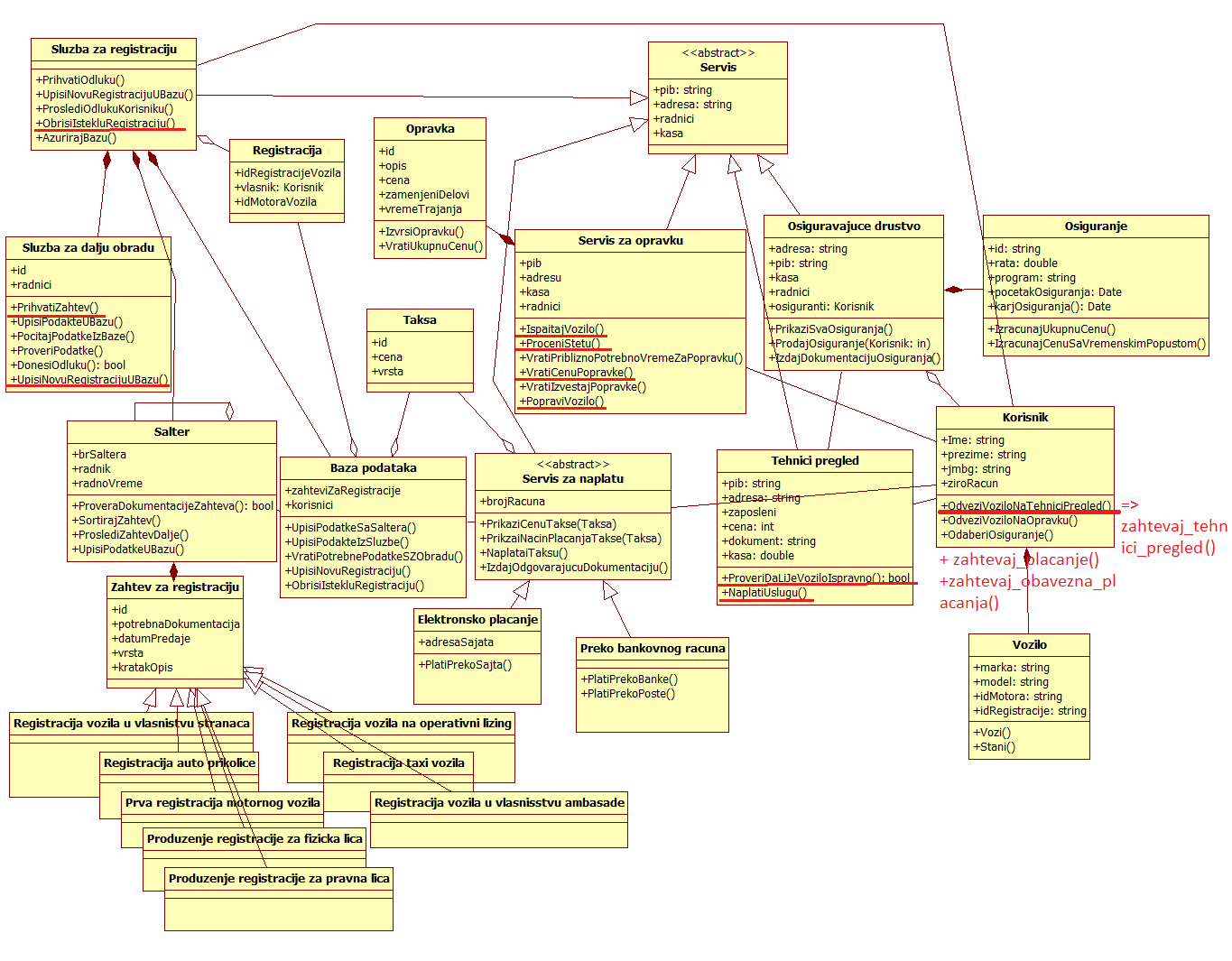
1. *Arhitektura aplikacije*



*-Cilj je bio da arhitektura aplikacije bude kao na slici.*

Osnovna struktura aplikacije je Maven projekat koji omogućava kreiranje,razvoj i testiranje aplikacije.Sve ovo vrši se uz pomoć Wildfly servera i EJB poslovne logike,odnosno Session i Entity Bean-ova.Baza podataka je MySql baza kreirana u MySqlWorkbenchu.Bazi pristupamo pomoću Hibernate i JPA alata.Bazu testiramo tako što testiramo osnovne funkcionalnosti za rad sa bazom,kreiranje,dodavanje,brisanje i modifikaciju odredjenih podataka.Testiranje ovih funkcionalnosti smo izvršili sa serverske strane pre deploymenta na Wildfly server.Nakon deploymenta sa klijentske se izvršava aplikaciona logika.

1. *Dijagram klasa*

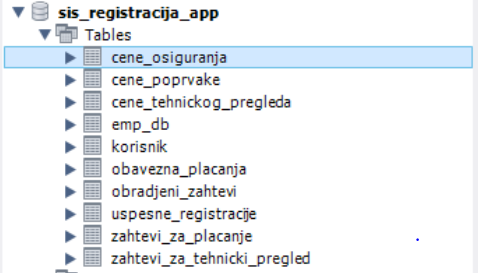
**

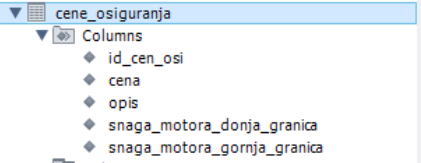
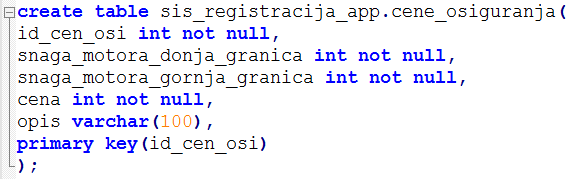
Poslovni obrazac: Upravljanje podacima proizovda (Product Data Managment)

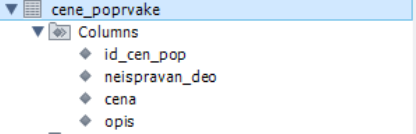
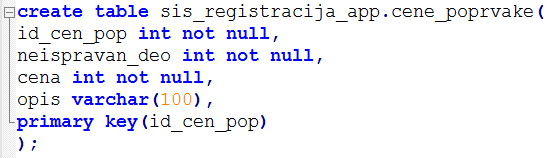
Ovaj poslovni obrazac služi za struktiranje proizvoda i ili dokumnta u poslovnim organizacijama. Zbog fizičke strukture sistema službe za registraciju vozila obrazac nije mogao direktno da se uklopi u klasni dijagram. Ali na apstraktnom nivou se skroz poklapa sa strukturom u ondnosu na finalnu ragistraciju kao porozvod prema njenim delovima: plaćene takse, tehnički pregled i osiguranje, jer svaki deo sam za sebe predstavlja proizvod, a oni zajedno omogućavaju da se dodje do registracije vozila koa finalnog proizvoda. I naravno ceo taj proces i u celom tom sistemu neizbežna je dokumentacija, koja je usko povezana sa svakim proizvodom.

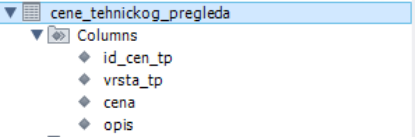
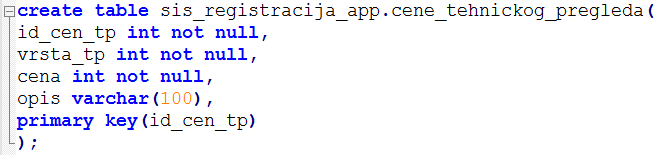
Sve funkcije koje su realizovane u drugom delu projekta su podvučene crvenom bojom na klasnom dijagramu.

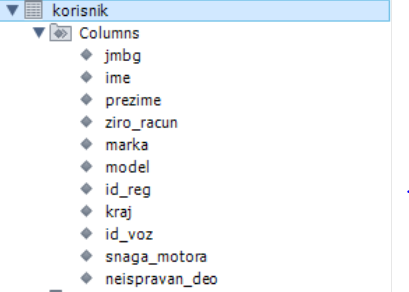
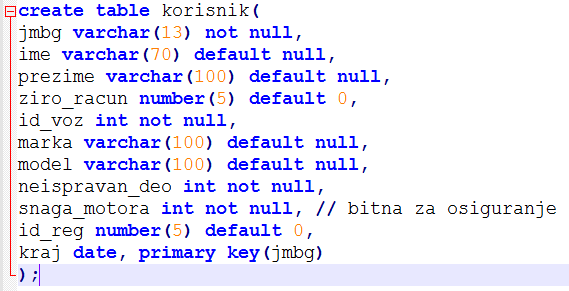
1. *Šema baze podataka*

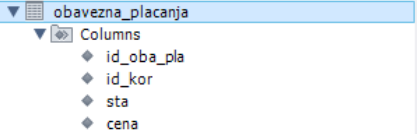
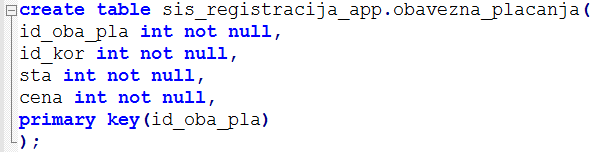


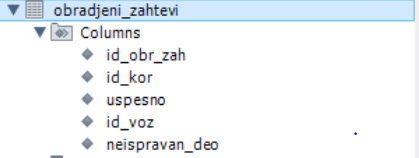
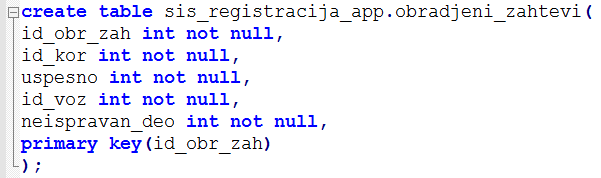


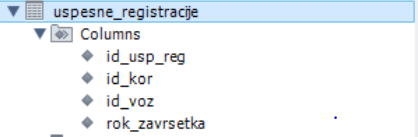
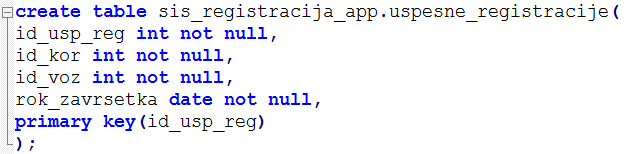


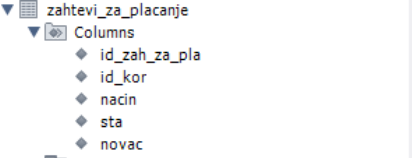
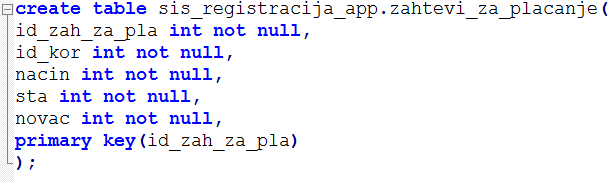


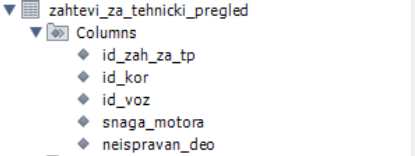
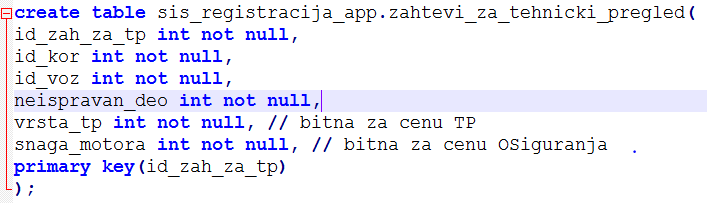






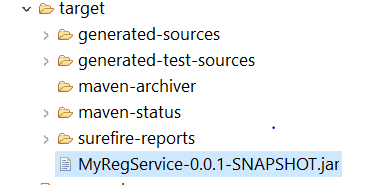






1. *Deployment aplikacije*

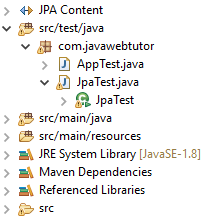
Najpre moramo startovati odgovarajući Wildfly server.Nakon toga idemo desni klik na naš projekat,Run as -> Maven build i kao Goals unesemo eclipse:eclipse da bismo buildovali projekat.Nakon toga idemo opet Run as -> Maven build sada kao goals unesemo clean install,tada u target folderu u nasem project solution-u dobijamo Snapshot odnosno EJB JAR fajl našeg projekta.



Nakon toga, desnim klikom na taj Snapshot Run as -> Run on Server,izvršavamo Deployment naše aplikacije na WildFly server. Nažalost, ja nisam opciju Run on Server...Za proveru da li je Deployment bio uspešan otvorili bi smo odgovarajuću WildFly konzolu na adresi 127.0.0.1:9991/console a zatim i karticu Deployments gde bi trebalo da se pojave odgovarajući jar i xml fajlovi za naš projekat.

*5.Testiranje*

Sva testiranja radili smo na serverskoj strani pre Deploymenta aplikacije na WildFly server,i to u test klasi „ JpaTest“ na putanji *src/test/java* u paketu *com.webtutor* prikazanoj na slici:



Svrha svakog od ovih testova bila je proveravanje funkcionalnosti svih kreiranih Session bean-ova pre njihovog Deploymenta na server.Provera funkcionalnosti se sastojala u njihovom ispravnom radu sa MySql bazom,odnosno dodavanju,čitanju,modifikaciji ili brisanju određenih podataka iz iste.