

# Informacioni sistemi

## II deo

*Pavle Marinkovic 15705*

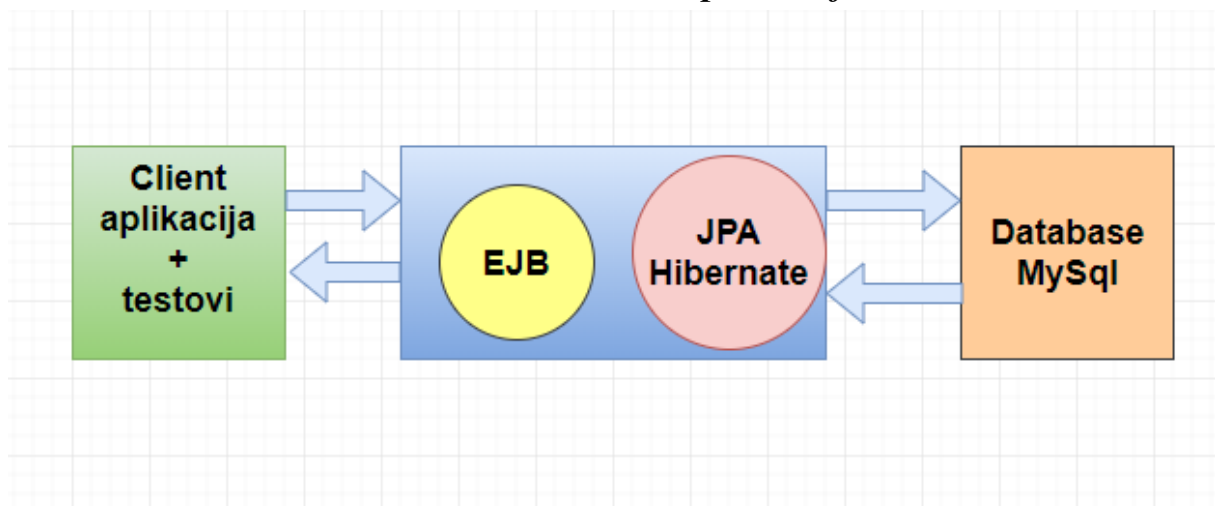
### **Tekst zadatka:**

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 prisleđuju službama za dalju obradu tj. odobravanje ili odbijanje zahteva. Plaćanje takse za registraciju vozila moguće je izvršiti elektronskim putem ili uplatom na bankovni račun službe. Svi relevantni podaci čuvaju se u centralizovanoj bazi podataka.

Obrazac: Upravljanje podacima proizvoda

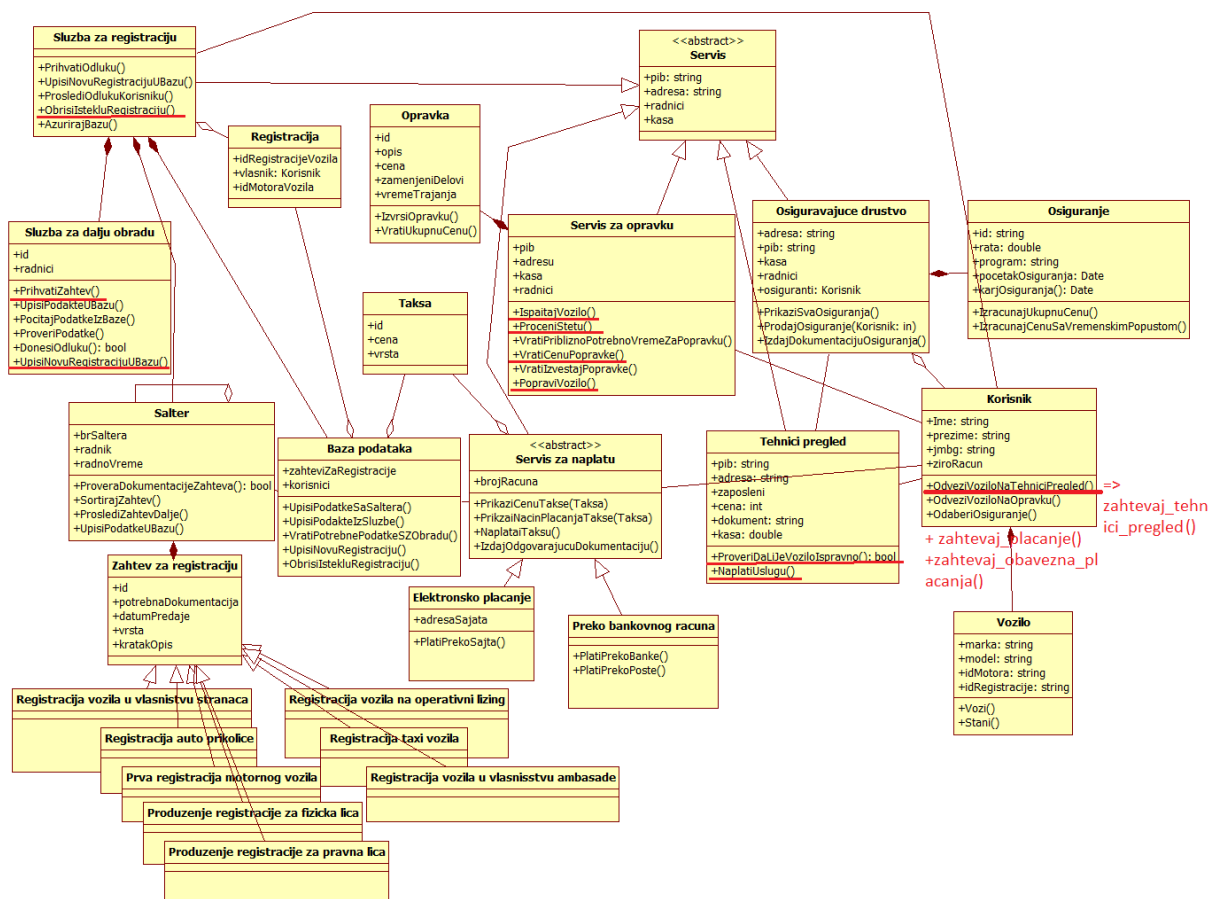
### *1. Arhitektura aplikacije*



*-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 MySql Workbenchu. Bazi pristupamo pomoću Hibernate i JPA alata. Bazu testiramo tako što testiramo osnovne funkcionalnosti za rad sa bazom, kreiranje, dodavanje, brisanje i modifikaciju određenih podataka. Testiranje ovih funkcionalnosti smo izvršili sa serverske strane pre deploymenta na Wildfly server. Nakon deploymenta sa klijentske se izvršava aplikaciona logika.

### *2. Dijagram klasa*



Poslovni obrazac: Upravljanje podacima proizvoda (Product Data Managment)

Ovaj poslovni obrazac služi za strukturiranje 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.

### 3. Šema baze podataka

▼ <b>sis_registracija_app</b>
▼ Tables
▶ cene_osiguranja
▶ cene_poprvake
▶ cene_tehnickog_pregleda
▶ emp_db
▶ korisnik
▶ obavezna_placanja
▶ obradjeni_zahtevi
▶ uspesne_registracije
▶ zahtevi_za_placanje
▶ zahtevi_za_tehnicki_pregled

```
create table sis_registracija_app.cene_osiguranja(
  id_cen_osi int not null,
  snaga_motora_donja_granica int not null,
  snaga_motora_gornja_granica int not null,
  cena int not null,
  opis varchar(100),
  primary key(id_cen_osi)
);
```

▼ <b>cene_osiguranja</b>
▼ Columns
◆ id_cen_osi
◆ cena
◆ opis
◆ snaga_motora_donja_granica
◆ snaga_motora_gornja_granica

```
create table sis_registracija_app.cene_poprvake(
  id_cen_pop int not null,
  neispravan_deo int not null,
  cena int not null,
  opis varchar(100),
  primary key(id_cen_pop)
);
```

▼ <b>cene_poprvake</b>
▼ Columns
◆ id_cen_pop
◆ neispravan_deo
◆ cena
◆ opis

```
create table sis_registracija_app.cene_tehnickog_pregleda(
  id_cen_tp int not null,
  vrsta_tp int not null,
  cena int not null,
  opis varchar(100),
  primary key(id_cen_tp)
);
```

▼ <b>cene_tehnickog_pregleda</b>
▼ Columns
◆ id_cen_tp
◆ vrsta_tp
◆ cena
◆ opis

```
create table korisnik(
  jmbg varchar(13) not null,
  ime varchar(70) default null,
  prezime varchar(100) default null,
  ziro_racun number(5) default 0,
  id_voz int not null,
  marka varchar(100) default null,
  model varchar(100) default null,
  neispravan_deo int not null,
  snaga_motora int not null, // bitna za osiguranje
  id_reg number(5) default 0,
  kraj date, primary key(jmbg)
);
```

▼ <b>korisnik</b>
▼ Columns
◆ jmbg
◆ ime
◆ prezime
◆ ziro_racun
◆ marka
◆ model
◆ id_reg
◆ kraj
◆ id_voz
◆ snaga_motora
◆ neispravan_deo

```
create table sis_registracija_app.obavezna_placanja(
  id_ob_pla int not null,
  id_kor int not null,
  sta int not null,
  cena int not null,
  primary key(id_ob_pla)
);
```

▼ <b>obavezna_placanja</b>
▼ Columns
◆ id_ob_pla
◆ id_kor
◆ sta
◆ cena

```
create table sis_registracija_app.obradjeni_zahtevi(
  id_obr_zah int not null,
  id_kor int not null,
  uspesno int not null,
  id_voz int not null,
  neispravan_deo int not null,
  primary key(id_obr_zah)
);
```

▼ <b>obradjeni_zahtevi</b>
▼ Columns
◆ id_obr_zah
◆ id_kor
◆ uspesno
◆ id_voz
◆ neispravan_deo

<pre> create table sis_registracija_app.uspesne_registracije( id_osp_reg int not null, id_kor int not null, id_voz int not null, rok_zavrsetka date not null, primary key(id_osp_reg) ); </pre>	<div>uspesne_registracije</div> <div>Columns</div> <ul style="list-style-type: none"> <li>id_osp_reg</li> <li>id_kor</li> <li>id_voz</li> <li>rok_zavrsetka</li> </ul>
<pre> create table sis_registracija_app.zahtevi_za_placanje( id_zah_za_pla int not null, id_kor int not null, nacin int not null, sta int not null, novac int not null, primary key(id_zah_za_pla) ); </pre>	<div>zahtevi_za_placanje</div> <div>Columns</div> <ul style="list-style-type: none"> <li>id_zah_za_pla</li> <li>id_kor</li> <li>nacin</li> <li>sta</li> <li>novac</li> </ul>
<pre> create table sis_registracija_app.zahtevi_za_tehnicki_pregled( id_zah_za_tp int not null, id_kor int not null, id_voz int not null, neispravan_deo int not null, vrsta_tp int not null, // bitna za cenu TP snaga_motora int not null, // bitna za cenu OSiguranja primary key(id_zah_za_tp) ); </pre>	<div>zahtevi_za_tehnicki_pregled</div> <div>Columns</div> <ul style="list-style-type: none"> <li>id_zah_za_tp</li> <li>id_kor</li> <li>id_voz</li> <li>snaga_motora</li> <li>neispravan_deo</li> </ul>

## 4. Deployment aplikacije

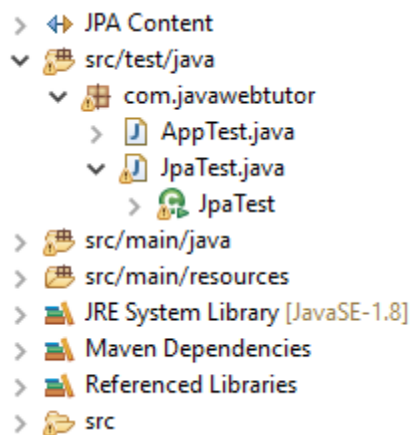
Najpre moramo startovati odgovarajući Wildfly server. Nakon toga idemo desni klik na naš projekat, Run as -> Maven build i kao Goals unesemo eclipse:eclipse da bismo buildovali projekat. Nakon toga idemo opet Run as -> Maven build sada kao goals unesemo clean install, tada u target folderu u našem project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

- ▼ target
  - > generated-sources
  - > generated-test-sources
  - maven-archiver
  - > maven-status
  - > surefire-reports
  - MyRegService-0.0.1-SNAPSHOT.jar

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.