

Informacioni sistemi

Drugi deo projekta

Jovana Golijanin 16038

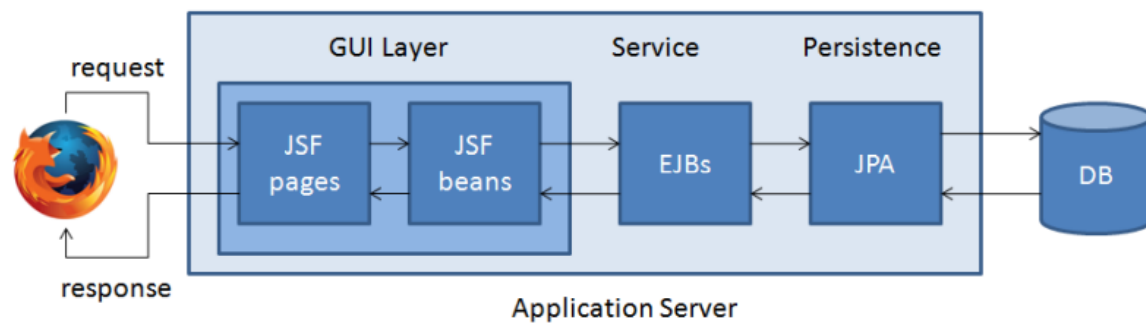
Tekst zadatka:

Informacioni sistem pametne elektricne mreže. Cuvaju se podaci o domacinstvu: ime i prezime vlasnika, adresa, prosečna potrošnja i broj neplacenih racuna. Osim toga, cuvaju se informacije o racunima svakog meseca – kolicina potrošene elektricne energije, iznos koji treba platiti i da li je placen ili ne. Omoguciti mehanizam za ažuriranje prosečne potrošnje, brojanje neplacenih racuna i obracunavanje cene.

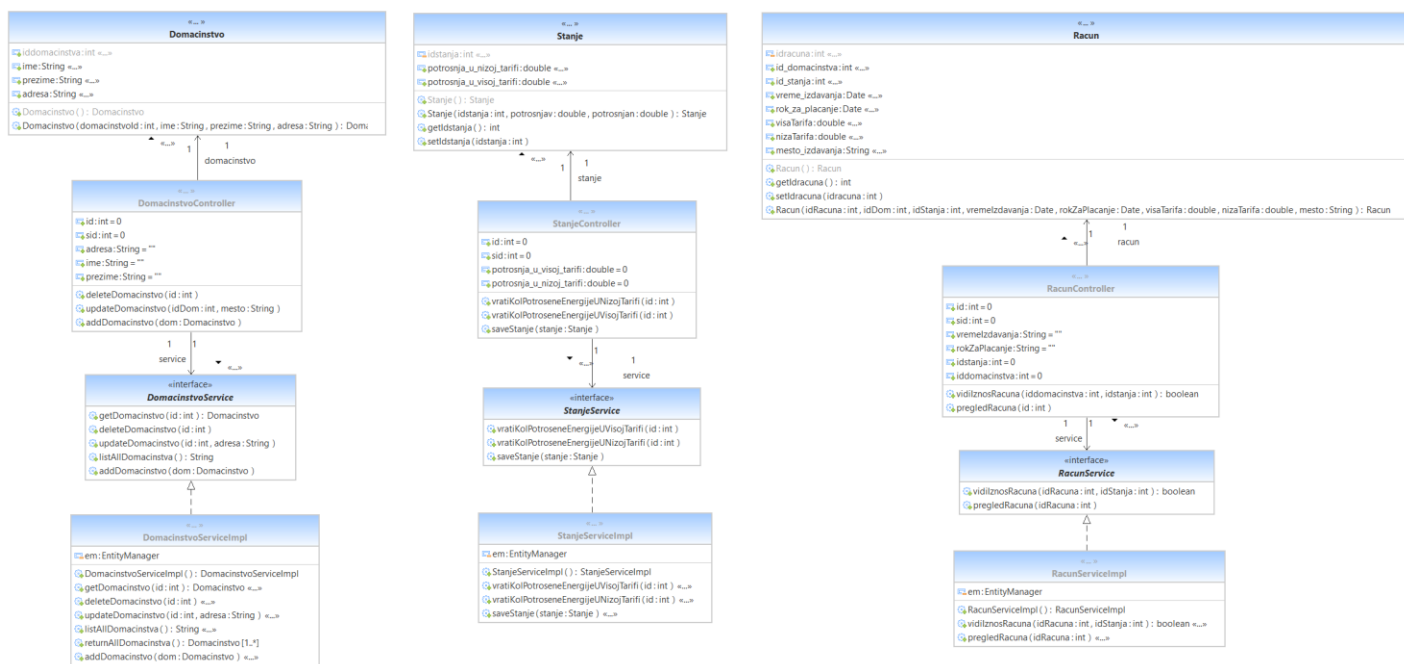
Obrazac: Dekompozicija poslovnog cilja

1. Arhitektura projekta:

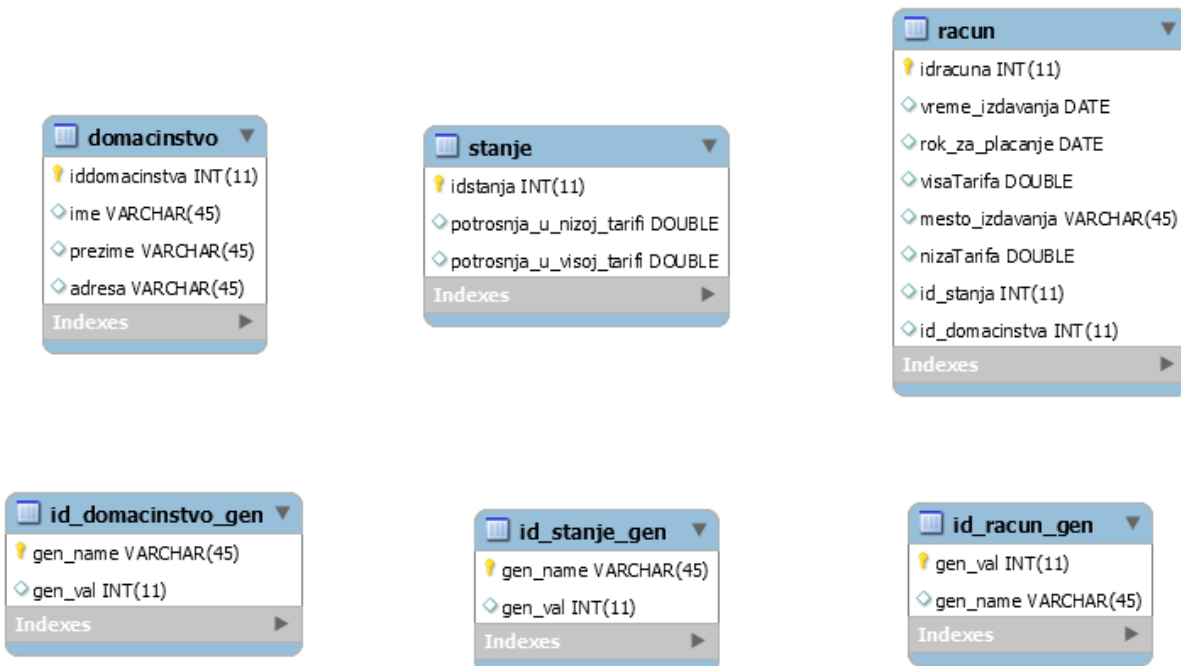
Arhitektura JSF Web aplikacije:



2. UML dijagram klase:

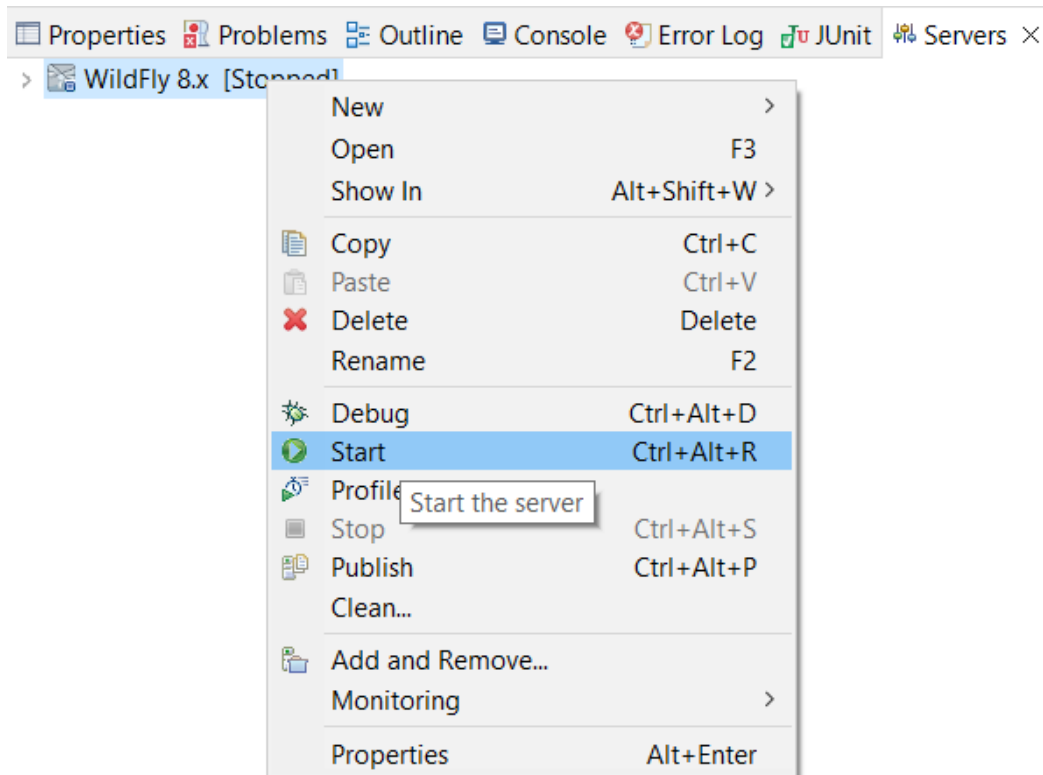


3. Šema baze podataka:



4. Objašnjenje deployment-a:

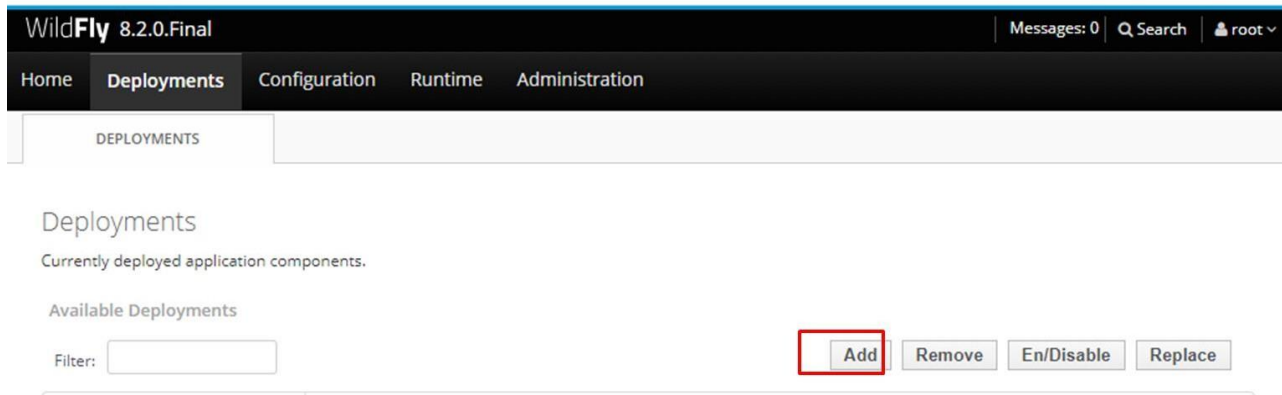
1. Startovati WildFly server – desni klik na „WildFly 8.x at localhost“ u Servers tabu Eclipse-a



2. Pristupiti mu iz browser-a na adresi koja se prikaže u konzoli nakon pokretanja servera

```
Properties Problems Outline Console Error Log JUnit Servers
WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_311\bin\javaw.exe (Jun 4, 2022, 12:33:20 PM)
12:33:26,192 INFO [org.jboss.weld.deployer] (MSC service thread 1-6) JBAS016005: Starting Services for CDI deployment: ElektricnaMrezaWeb.war
12:33:26,422 INFO [org.jboss.weld.Version] (MSC service thread 1-6) WELD-000900: 2.2.6 (Final)
12:33:26,446 INFO [org.jboss.weld.deployer] (MSC service thread 1-4) JBAS016008: Starting weld service for deployment ElektricnaMrezaWeb.war
12:33:26,553 INFO [org.jboss.as.jpa] (ServerService Thread Pool -- 50) JBAS011409: Starting Persistence Unit (phase 2 of 2) Service 'ElektricnaMrezaWeb.war#Elektricna
12:33:26,614 INFO [org.hibernate.annotations.common.Version] (ServerService Thread Pool -- 50) HCANN000001: Hibernate Commons Annotations {4.0.4.Final}
12:33:26,780 ERROR [stderr] (ServerService Thread Pool -- 50) Sat Jun 04 12:33:26 CEST 2022 WARN: Establishing SSL connection without server's identity verification is
12:33:26,912 INFO [org.hibernate.dialect.Dialect] (ServerService Thread Pool -- 50) HH0000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
12:33:27,019 INFO [org.hibernate.hql.internal.ast.ASTQueryTranslatorFactory] (ServerService Thread Pool -- 50) HH0000397: Using ASTQueryTranslatorFactory
12:33:28,076 INFO [javax.enterprise.resource.webcontainer.jsf.config] (MSC service thread 1-3) Initializing Mojarra 2.2.8-jbossorg-1 20140822-1131 for context '/Elekt
12:33:28,938 INFO [org.primefaces.webapp.PostConstructApplicationEventListener] (MSC service thread 1-3) Running on PrimeFaces 5.1
12:33:28,964 INFO [org.wildfly.extension.undertow] (MSC service thread 1-3) JBAS017534: Registered web context: /ElektricnaMrezaWeb
12:33:29,023 INFO [org.jboss.as.server] (Controller Boot Thread) JBAS018559: Deployed "mysql-connector-java-5.1.41-bin.jar" (runtime-name : "mysql-connector-java-5.1.
12:33:29,024 INFO [org.jboss.as.server] (ServerService Thread Pool -- 28) JBAS018559: Deployed "ElektricnaMrezaWeb.war" (runtime-name : "ElektricnaMrezaWeb.war")
12:33:29,053 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Http management interface listening on http://127.0.0.1:9990/management
12:33:29,053 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin console listening on http://127.0.0.1:9990/
12:33:29,054 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.2.0.Final "Tweek" started in 8303ms - Started 384 of 443 services (100 services are la
Go to Settings to activate Windows
```

3. U kartici Deployments odabrati opciju Add



1. Selektovati odgovarajući jar fajl za mysql konektor iz fajl sistema i sačuvati taj Deployment

Create Deployment



Managed Unmanaged

Step1/2: Deployment Selection

Please choose a file that you want to deploy.

Choose File

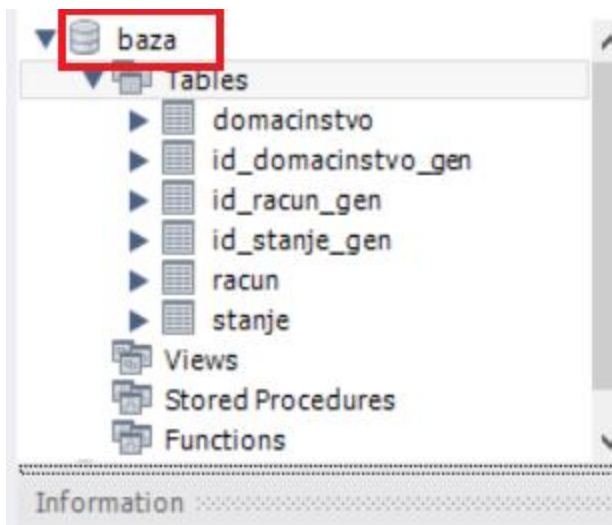
mysql-connec...1.41-bin.jar

mysql-connector-java-5.1.41-bin.jar

Cancel

Next »

5.U kartici Configuration dodati stavku za bazu podataka koja se koristi u aplikaciji



WildFly 8.2.0.Final Messages: 0 Search root

Home Deployments **Configuration** Runtime Administration

Subsystems << DATASOURCES XA DATASOURCES

▼ Connector
JCA
Databases
Resource Adapters
Mail
► Container
► Core
► Infinispan
► Security
► Web

General Configuration
Interfaces
Socket Binding
Paths
System Properties

JDBC Datasources

JDBC datasource configurations.

Available Datasources

Add Remove Disable

Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	✓
baza	java:jboss/datasources/baza	✓

<< < 1-2 of 2 > >>

Attributes Connection Pool Security Properties Validation Timeouts

[Edit](#)

Name: baza

JNDI: java:jboss/datasources/baza

Is enabled?: true

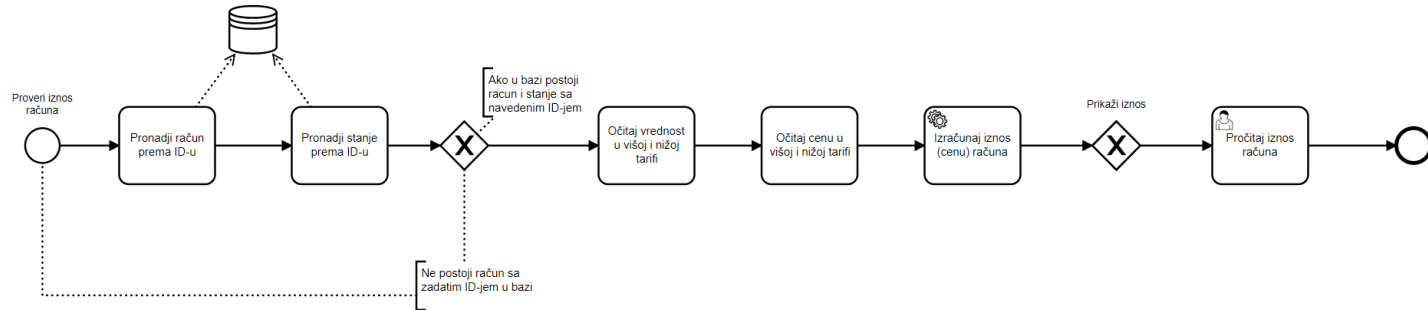
Statistics enabled?: false

Datasource Class:

[Need Help?](#)

Testirati konekciju, ukoliko se dobije informacija da je uspešno znači da je povezivanje servera sa bazom podataka uspešno.

4. BPMN dijagram



BPMN dijagram je rađen u sledećem online alatu: <https://demo.bpmn.io/s/start>

5. Test Case

1. Testiranje funkcije za ažuriranje adrese domaćinstva čiji smo ID naveli

Naziv testa: TestCase

Opis testa: Testiramo da li funkcija za ažuriranje adrese domaćinstva iz baze podataka radi adekvatno time što proveravamo da li je adresa domaćinstva promenjena nakon izvršenja f-je

Preduslovi: U bazi postoji domacinstvo sa datim id-jem

Koraci testa:

1. Dodati nekoliko domacinstava u bazu sa odredjenom adresom
2. Pozvati funkciju za pribavljanje domacinstva iz baze
3. Proveriti da li je funkcija vraća vrednost različitu od NULL

Test podaci:

```
INSERT INTO `baza`.`domacinstvo` (`iddomacinstva`, `ime`, `prezime`, `adresa`) VALUES ('9', 'Mika', 'Mikic', 'Pobedina 2/9');
```

Očekivani rezultati:

Ažuriranje adresa domaćinstva koje smo pribavili

Post uslovi:

Adresa je ažurirana, što proveravamo uz pomoć assert-a

Dobijeni rezultat:

Domaćinstvo sa novom adresom

Status:

Uspešan

2. Testiranje funkcije za izračunavanje cene računa domaćinstva na osnovu potrošnje koja se obračuvana u klasi Stanje

Naziv testa: TestRacunFunkcije

Opis testa: Testiramo da li funkcija za izračunavanje cene računa domaćinstva na osnovu potrošnje vraća cenu tj. Iznos za plaćanje

Preduslovi: U bazi postoji račun sa datim id-jem

Koraci testa:

1. Dodati račun u bazu
2. Pozvati funkciju za pribavljanje računa iz baze
3. Proveriti da li je funkcija vraća vrednost različitu od NULL

Test podaci:

```
INSERT INTO `baza`.`racun` (`idracuna`, `vreme_izdavanja`, `rok_za_placanje`, `visaTarifa`,  
`mesto_izdavanja`, `nizaTarifa`, `id_stanja`, `id_domacinstva`) VALUES ('2', '2008-10-03', '2008-  
10-03', '5', 'nis', '2', '1', '1');
```

Očekivani rezultati:

Funkcija vraća cenu tj. Iznos za plaćanje

Post uslovi:

Možemo odštampati neku poruku

Dobijeni rezultat:

Iznos za plaćanje

Status:

Uspešan

3. Testiranje funkcije koja vraća količinu potrošene električne energije u višoj tj. nižoj tarifi iz klase Stanje

Naziv testa: TestStanjeFunkcije

Opis testa: Testiramo da li funkcija za vraćanje količine potrošene električne energije u višoj tj. nižoj tarifi vraća količinu potrošene električne energije za datu tarifu

Preduslovi: U bazi postoji stanje sa datim id-jem

Koraci testa:

1. Dodati stanje u bazu
2. Pozvati funkciju za pribavljanje stanja iz baze
3. Proveriti da li je funkcija vraća vrednost različitu od NULL

Test podaci:

```
INSERT INTO `baza`.`stanje` (`idstanja`, `potrosnja_u_nizoj_tarifi`, `potrosnja_u_visoj_tarifi`)  
VALUES ('1', '3', '5');
```

Očekivani rezultati:

Funkcija vraća količinu potrošene električne energije u višoj tj. nižoj tarifi

Post uslovi:

Možemo odštampati neku poruku

Dobijeni rezultat:

Količinu potrošene električne energije u višoj tj. nižoj tarifi

Status:

Uspešan