

Informacioni sistemi – Drugi deo projekta

Dokumentacija

Matija Špeletić 18043

matijasp@elfak.rs

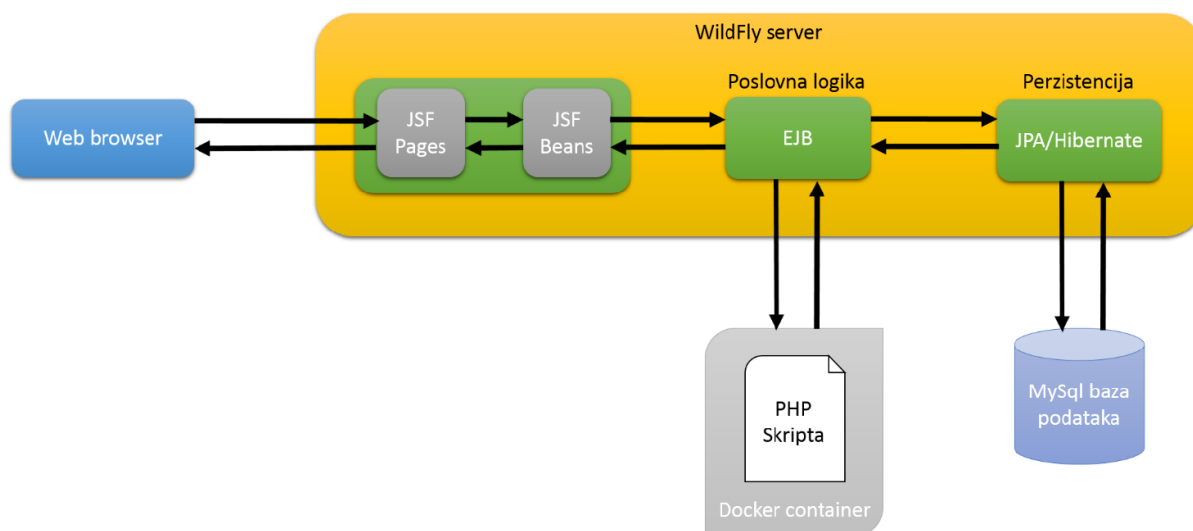
1. Zadatak

Informacioni sistem prodajnog mesta operatera mobilne telefonije. Potrebno je čuvati podatke o zaposlenima, klijentima, brojevima, dostupnim uređajima, tarifama i ugovorima. Brojevi mogu biti prepaid ili tarife sa ugovorom. Za ugovor je neophodno čuvati i podatke o klijentu. Uz ugovor može biti kupljen i mobilni telefon po ceni koja zavisi od tarife. Za prepaid brojeve nije neophodno čuvati podatke o klijentu, ali se vodi evidencija o dopunama i stanju kredita. Predvideti i funkcionalnost elektronske dopune uz pomoć ovog sistema, ali i mogućnost obnove tarife nakon isteka obavezujućeg perioda.

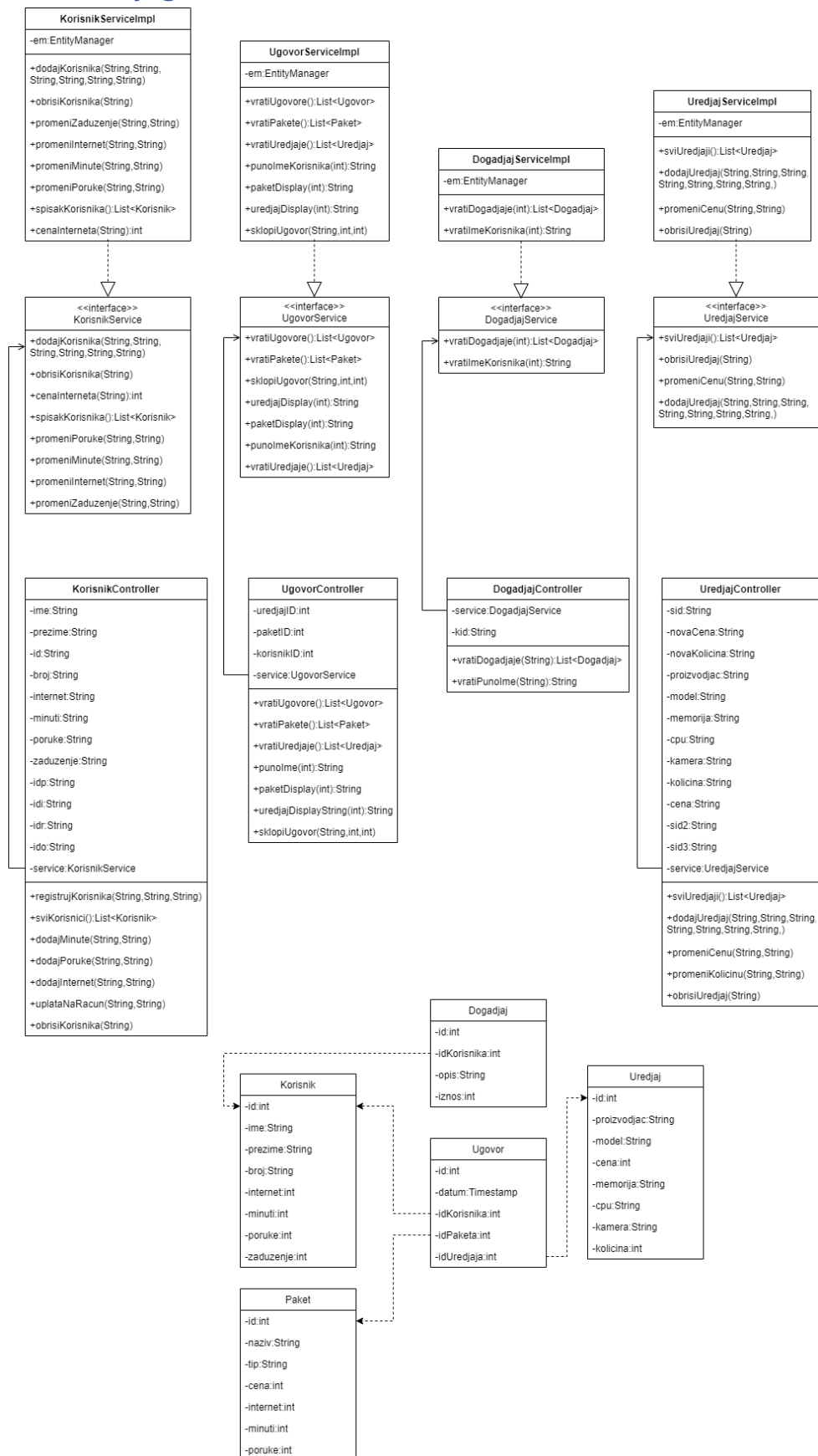
Obrazac: poslovni događaj – istorija rezultata

2. Arhitektura projekta

Arhitektura JSF Web aplikacije:



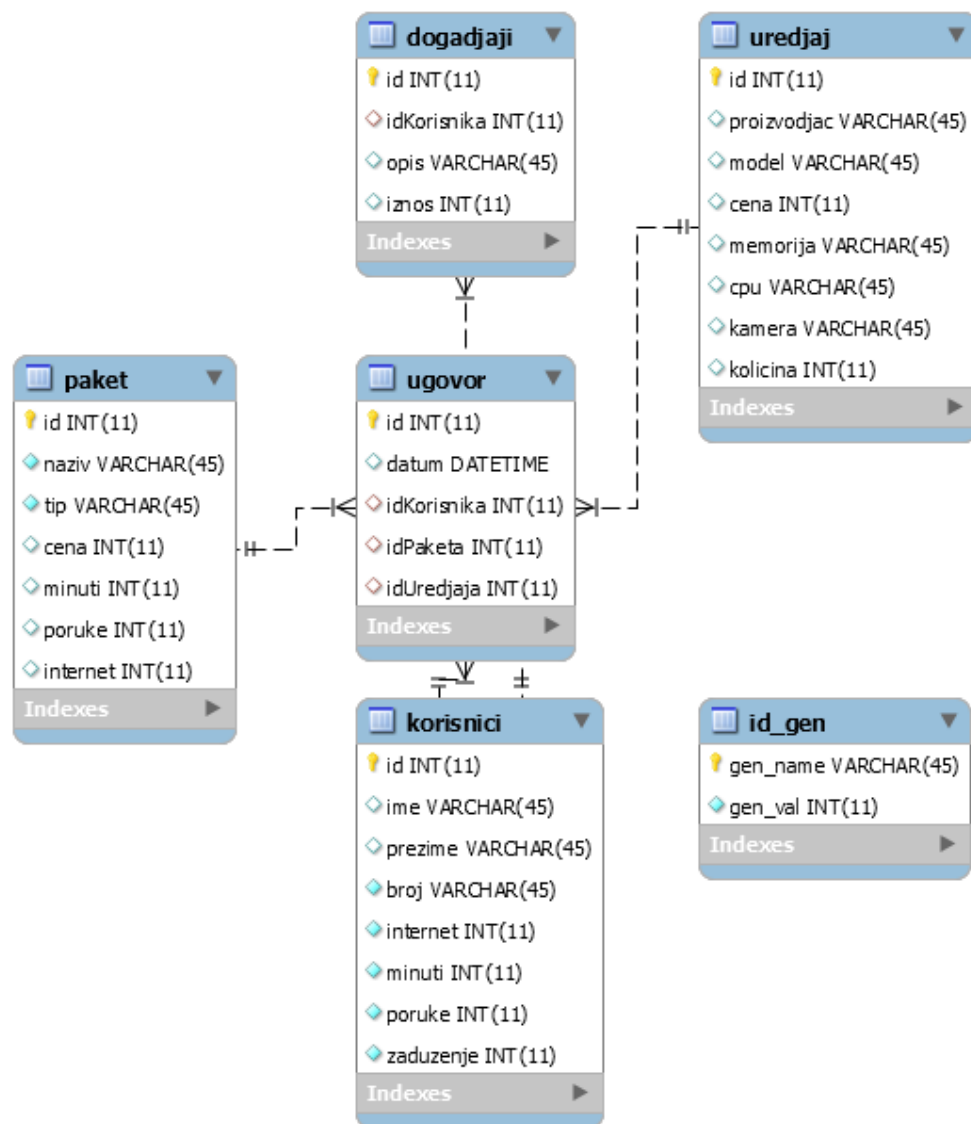
3. UML Klasni dijagram



Dijagram je rađen korišćenjem online alata draw.io

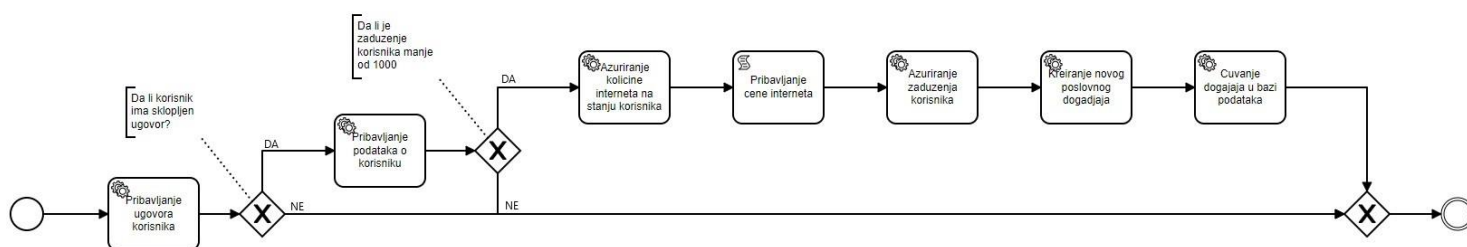
4. Šema baze podataka

Priložene su i naredbe za rekreiranje baze podataka zajedno sa test podacima.



6. BPMN dijagram

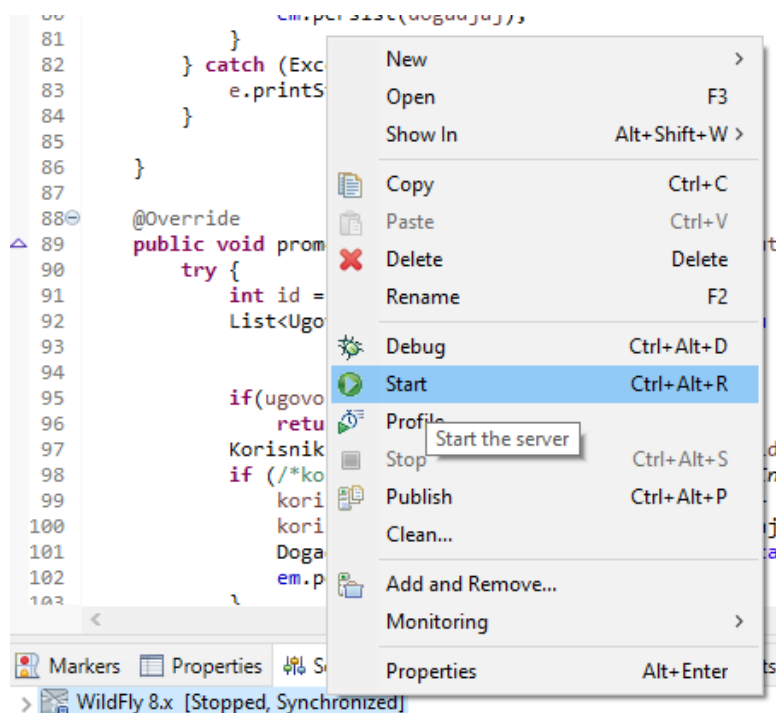
Na sledećem dijagramu je prikazan process uplate dodatnog internet ana stanje korisnika. Cena interneta se određuje putem PHP skripte koja je pokrenuta u Docker kontejneru.



Dijagram je rađen korišćenjem online alata <https://demo.bpmn.io>.

5. Objašnjenje deployment-a

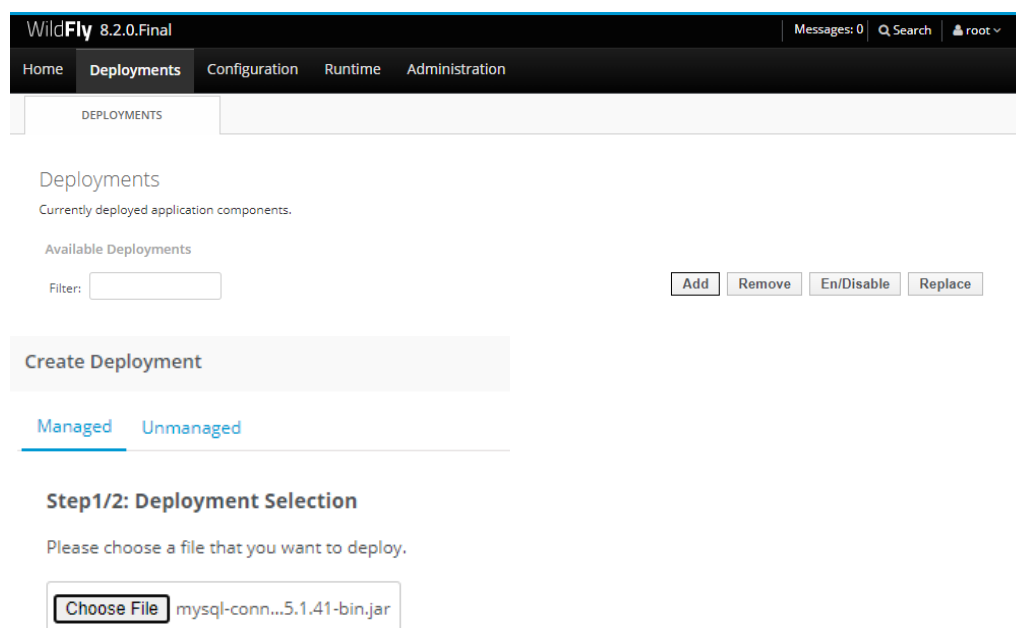
1. Startovati Wildfly server desnim klikom na "Wildfly 8.x" u Servers kartici u okviru Eclipse-a i klikom na opciju Start.



2. Pristupiti server iz browser-a putem adrese koja se prikaže u konzoli nakon pokretanja servera:

```
19:40:49,823 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Http management interface listening on http://127.0.0.1:9990/management
19:40:49,823 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin console listening on http://127.0.0.1:9990
19:40:49,824 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.2.0.Final "Tweek" started in 11243ms - Started 427 of 486 servi
```

3. U kartici Deployments odabrati opciju add i selektovati odgovarajući konektor za mysql iz fajl sistema i sačuvati taj Deployment.



4. U kartici Configuration dodati stavku za bazu podataka koja se koristi u aplikaciji i podesiti Connection URL tako da odgovara bazi podataka

The screenshot shows the WildFly 8.2.0.Final Configuration console. The 'Configuration' tab is active, and the 'Datasources' sub-tab is selected in the left sidebar. The main area displays 'JDBC Datasources' with a table of available datasources. The 'baza' datasource is highlighted, showing its JNDI name as 'java:jboss/datasources/baza?useSSL=false'. Below the table, the 'Attributes' tab is selected, showing the 'Name' as 'baza' and the 'JNDI' as 'java:jboss/datasources/baza?useSSL=false'. At the bottom, the 'Connection URL' is displayed as 'jdbc:mysql://localhost:3306/projekat'.

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

Attributes: Name: baza, JNDI: java:jboss/datasources/baza?useSSL=false

Connection URL: jdbc:mysql://localhost:3306/projekat

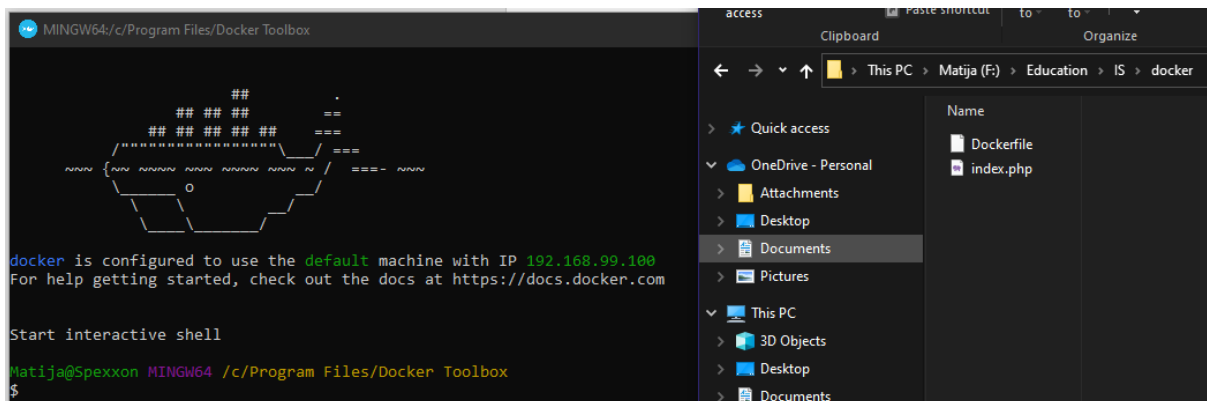
Opcija „?useSSL=false“ (kao u ovom slučaju, označeno zelenom bojom) ili „?useSSL=true“ može biti zahtevana od Wildfly servera kako bi deployment uspešno prošao. Obavezno testirati konekciju. Ukoliko se dobije informacija da je test uspešan, znači da je povezivanje servera sa bazom podataka prošlo uspešno.

Uspešan deployment web aplikacije:

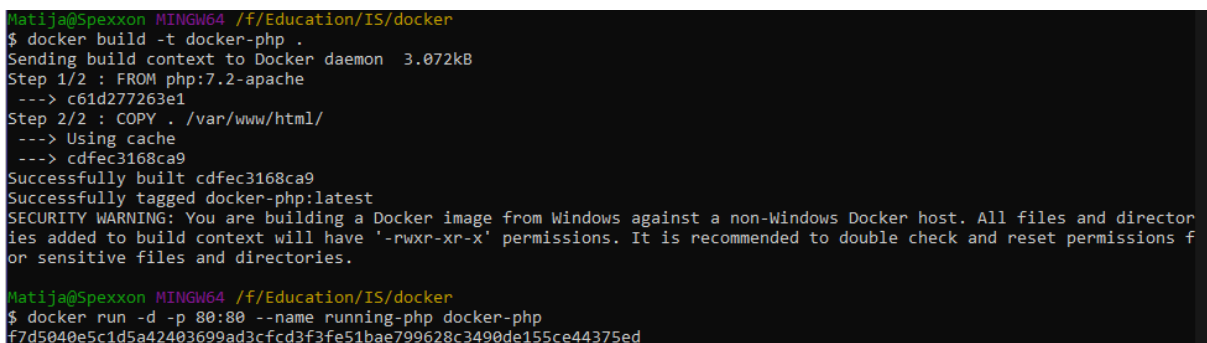
EmployeeApp.war ✓	ejb3	UredjajServiceImpl	
mysql-connector-java-5.1. ... ✓	jpa	UgovorServiceImpl	
	undertow	KorisnikServiceImpl	
		EmployeeServiceImpl	
		DogadjajServiceImpl	

Deployment servisa za računanje cene interneta pomoću Docker-a:

5.1. U folderu docker koji je priložen se nalaze Dockerfile i index.php fajlovi. Pomoću Docker Quickstart Terminala navigirati do ovog foldera:

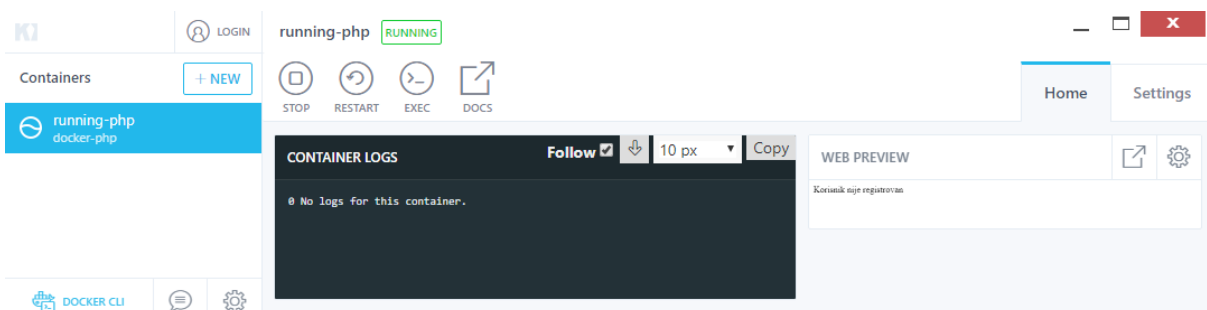


5.2. Izvršiti komande docker build i docker run kao što je prikazano na slici



Docker build će napraviti Docker image, a docker run će pokrenuti instancu kontejnera za taj image i moći ćemo da joj pristupimo preko adrese <http://192.168.99.100/>.

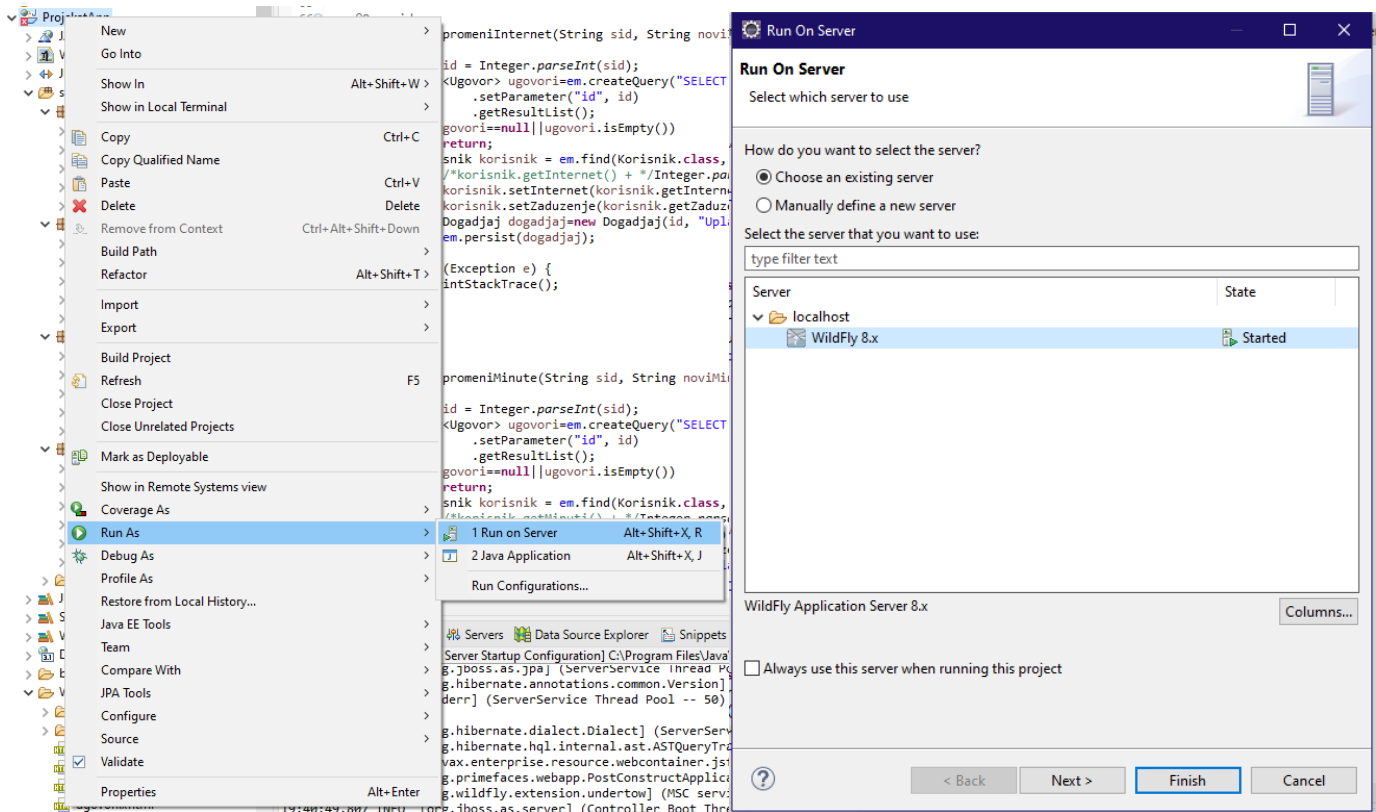
Ovim kontejnerom se može lakše upravljati pomoću Kitematic alata.



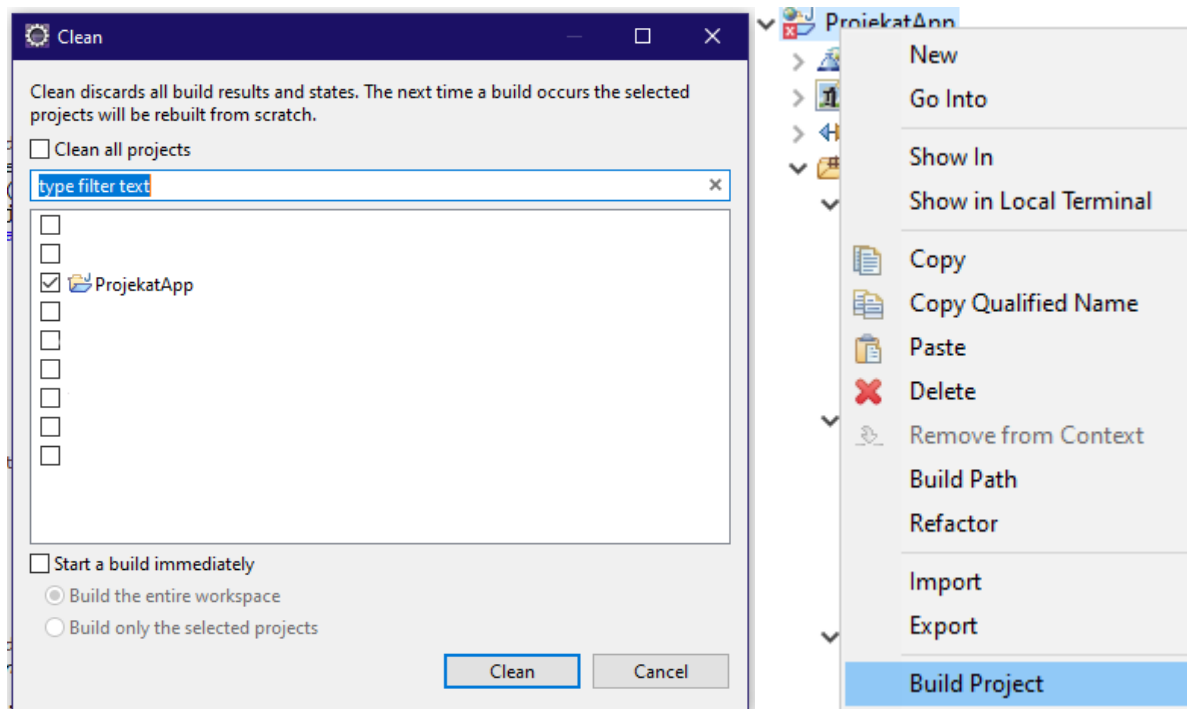
Testiranje servisa izvan aplikacije je moguće direktno iz browser-a putem URL-a, gde se koristi GET metoda kojoj se prosleđuju željeni parametri (npr. <http://192.168.99.100/?registrovano=true&kolicina=2>), a rezultat je broj koji predstavlja cenu interneta. Parametri koji se prosleđuju su logička vrednost koja označava da li je korisnik registrovan i količina interneta koju korisnik želi da kupi.



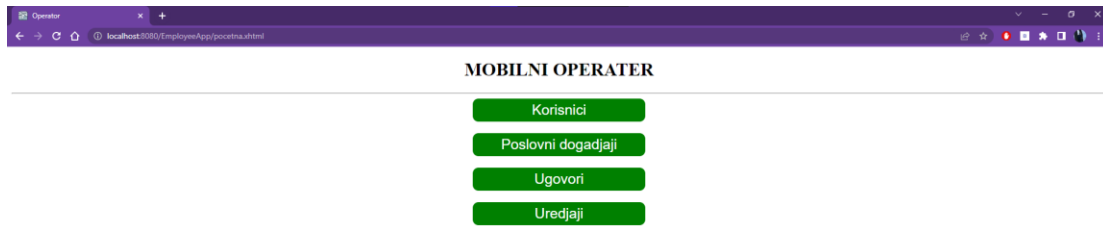
6. Pokretanje aplikacije desnim klikom na projekat u Eclipse-u: Run As -> Run on Server



(NAPOMENA: U koliko deployment na ovaj način ne prođe uspešno potrebno je Clean-ovati projekat iz Eclipse-a i ručno pokrenuti opciju Build project, a zatim ponovo pokrenuti opciju Run on Server)



7. Pristupiti aplikaciji na dobijenoj adresi (<http://localhost:8080/EmployeeApp/pocetna.xhtml>)



7. Test case

1. Testiranje funkcije za sklapanje ugovora

Testiramo da li funkcija za sklapanje ugovora uspešno beleži ugovor u bazi podataka, pri čemu ugovor povezuje korisnika, uređaj i paket, i testiramo da li ova funkcija uspešno zabeleži poslovni događaj sklapanja ugovora.

Preduslovi:

- Ugovor sa ID=999 ne postoji u bazi.
- Korisnik sa ID=2 (npr.) postoji u bazi.
- Paket sa ID=2 (npr.) postoji u bazi.
- Uređaj sa ID=0 postoji u bazi.

Koraci testa:

1. Pozivanje funkcije za sklapanje ugovora
 - a. Kreiranje ugovora i dodavanje u bazu podataka
 - b. Povećavanje zaduženja korisnika (cena ugovora i uređaja)
 - c. Dodavanje minuta, poruka i interneta koje korisnik dobija iz paketa
 - d. Kreiranje i dodavanje poslovnog događaja sklapanja ugovora u bazu

Test podaci:

```
INSERT INTO `paket` (`id`,`naziv`,`tip`,`cena`,`minuti`,`poruke`,`internet`) VALUES  
(2,'naziv1','postpaid',1,1,1,1);
```

```
INSERT INTO `korisnici` (`id`,`ime`,`prezime`,`broj`,`internet`,`minuti`,`poruke`,`zaduzenje`) VALUES  
(2,'ime1','prezime1','061234567',1,1,1,816);
```

```
INSERT INTO `uredjaj` (`id`,`proizvodjac`,`model`,`cena`,`memorija`,`cpu`,`kamera`,`kolicina`) VALUES  
(0,'bez','uredjaja',0,' ',' ',0);
```

Očekivani rezultat: Ugovor je uspešno dodat u bazu podataka, zaduženje korisnika i njegovi minuti, poruke i internet su uspešno postavljeni, a poslovni događaj je uspešno kreiran.

Post-uslov: Ugovor sa ID=999 postoji u bazi, događaj sa ID=999 postoji u bazi, i minuti, poruke i internet korisnika su postavljeni na odgovarajuće vrednosti.

Dobijeni rezultat: Funkcija je uspešno obavila sklapanje ugovora i beleženje poslovnog događaja.

Status: Uspešan.

2. Testiranje funkcije za dodavanje korisnika u bazu podataka i izmene odgovarajućih vrednosti

Testiramo da li funkcija uspešno registruje korisnika i da li izmena zaduženja, poruka, minuta i interneta radi ispravno

Preduslovi: Korisnik sa ID=777 ne postoji u bazi.

Koraci testa:

1. Dodavanje korisnika sa određenim brojem minuta, poruka, interneta i zaduženjem
2. Izmena Interneta
3. Izmena minuta
4. Izmena poruka
5. Izmena zaduženja

Test podaci: Nema.

Očekivani rezultat: Korisnik je uspešno dodat u bazu, a odgovarajuće kolone uspešno izmenjene.

Post-uslov: Korisnik sa ID=777 postoji u bazi, a minuti, poruke i internet odgovaraju novim vrednostima.

Dobijeni rezultat: Funkcija je uspešno dodala korisnika u bazu i izmena odgovarajućih parametara je takođe uspešno izvršena.

Status: Uspešan.

3. Testiranje funkcije za dodavanje novog uređaja i funkcija za izmenu cene i količine

Testiramo da li funkcija uspešno dodaje uređaj u bazu podataka i da li funkcije za izmenu cene i količine uspešno vrše odgovarajuće izmene.

Preduslovi: Uređaj sa ID=999 ne postoji u bazi podataka.

Koraci testa:

1. Dodavanje uređaja sa proizvoljnim parametrima
2. Izmena cene
3. Izmena količine

Test podaci: Nema.

Očekivani rezultat: Uređaj je uspešno dodat u bazu, a odgovarajuće vrednosti su uspešno izmenjene.

Post-uslov: Uređaj sa ID=999 postoji u bazi, a cena i količina odgovaraju novim vrednostima.

Dobijeni rezultat: Funkcija je uspešno dodala uređaj u bazu i izmena cene i količine putem odgovarajućih funkcija je takođe uspešno izvršena.

Status: Uspešan.