Informacioni sistemi
2. Deo

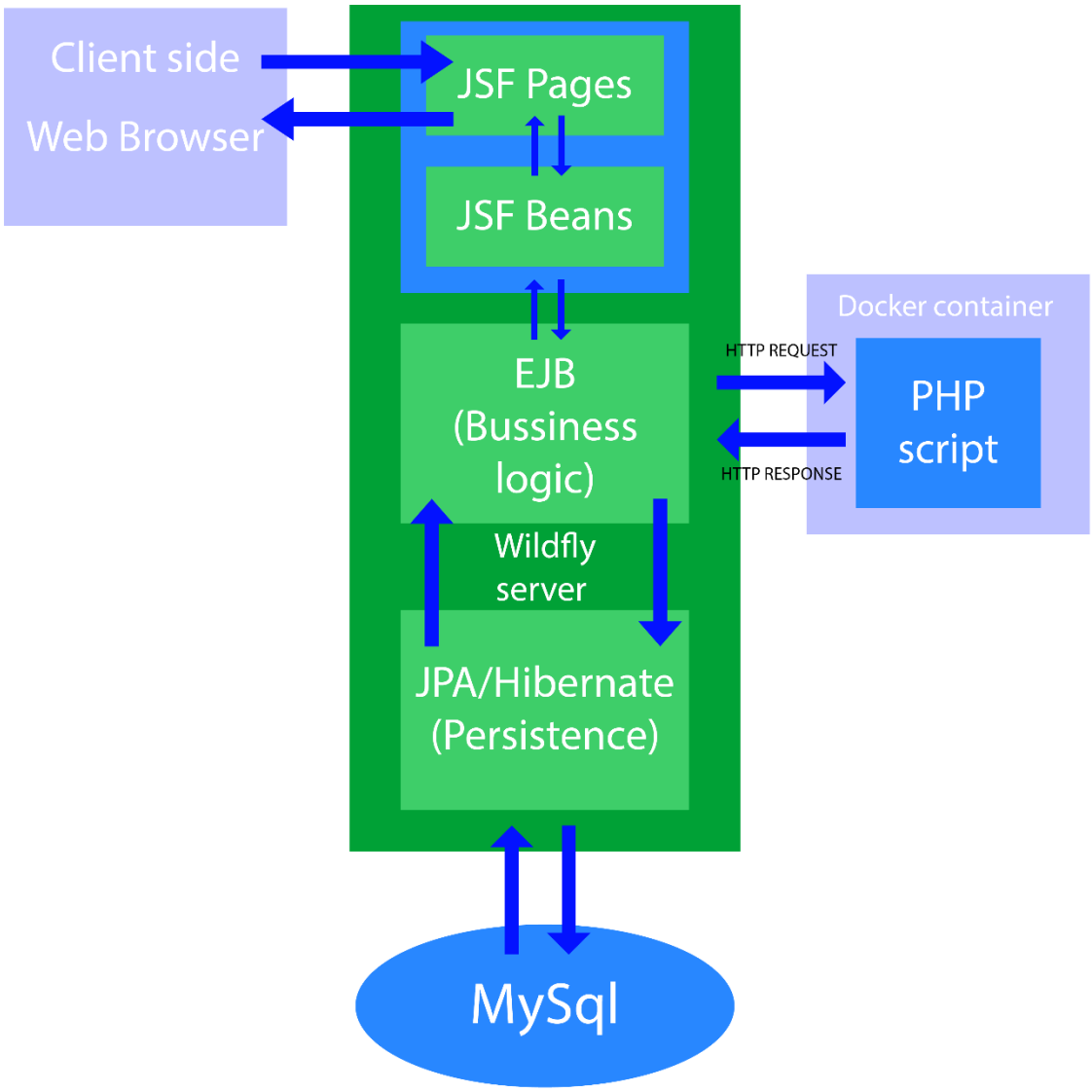
Ime I prezime: Stefan Stojanović
Broj indeksa: 16929
Kontakt: stefan.stojanovic@elfak.rs, 065/8780826

Tekst zadatka:

193. Informacioni sistem supermarketa

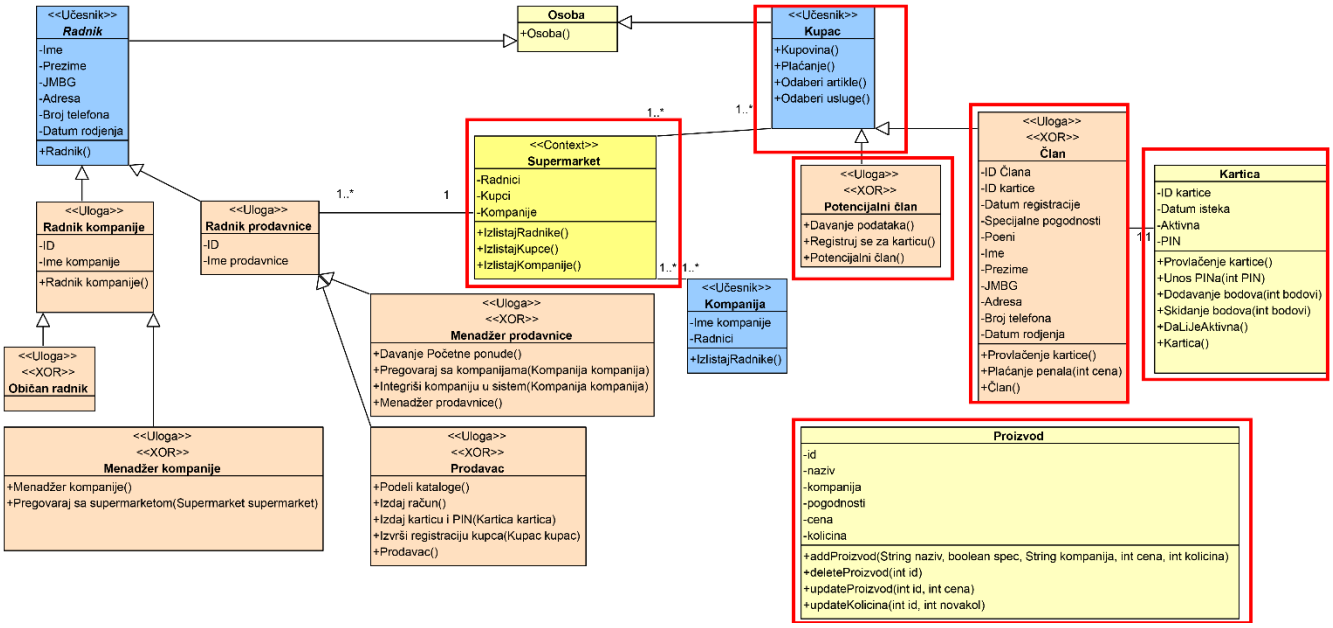
Supermarket svojim kupcima omogucava da se registruju i dobiju clansku karticu, kojom se mogu ostvariti
razni benefiti. Kupovinom u supermarketu, zavisno od velicine placenog racuna, vrši se skaliranje i dobijaju
bodovi koji su akumulirani na kartici. Ovim bodovima, verni kupci mogu ostvariti razne benefite – dobiti specijalne proizvode, rezervisane samo za clanove, iz posebnog kataloga besplatno (zavisno od broja akumuliranih poena), pri cemu se broj bodova umanjuje za cenu proizvoda izraženu u bodovima, ili kupovati neke od redovnih proizvoda iz marketa po promotivnim cenama. Osim toga, supermarket, u saradnji sa drugim kompanijama nudi popuste i pogodnosti za clanove sa velikim brojem poena – specijalne tarife za SIM kartice kod nekog od mobilnih operatera, kursevi jezika po redukovanim cenama i slicno.
Obrazac: UCESNIK – ULOGA

1.Arhitektura Sistema:



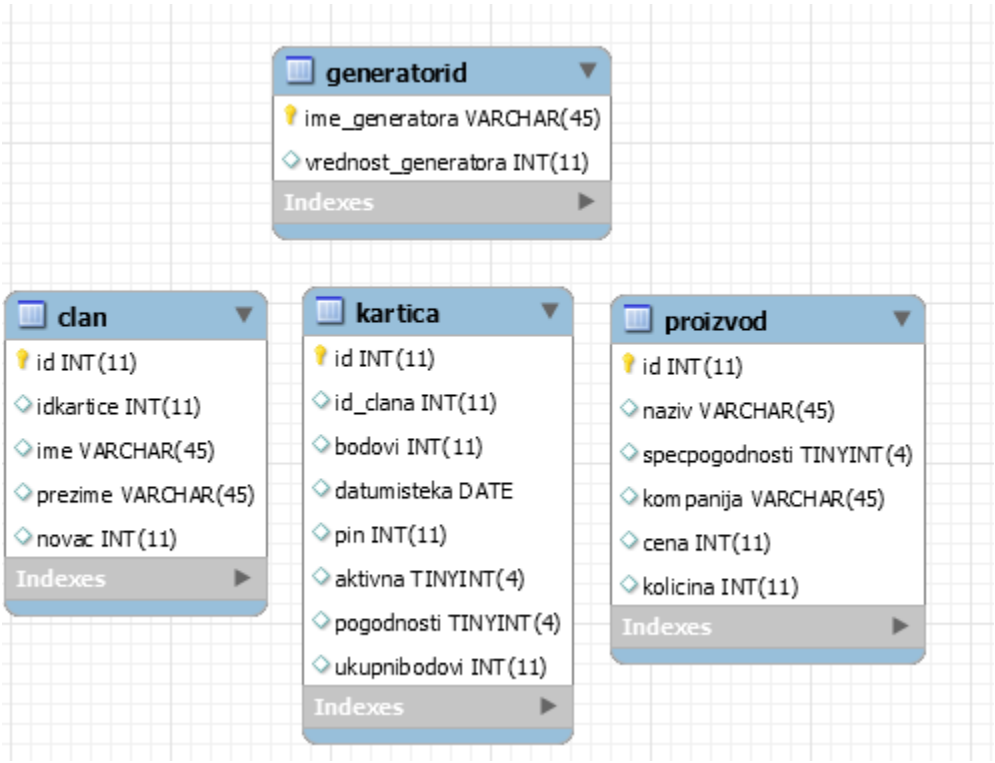
Da bi se izvele sve moguće operacije i omogućio pristup podacima u bazi, kreirana je JSF web aplikacija koja koristi Jboss WildFly server na kome je deployovana, EJB za biznis logiku, JPA i Hibernate za perzistenciju objekata i Docker container u kome je smeštena PHP skripta koja se koristi za proveru nekih podataka. Aplikacija se pokreće otvaranjem njene adrese u browseru(localhost:8080/ImeAplikacije). Kao baza podataka koristi se MySQL.

2. Klasni dijagram

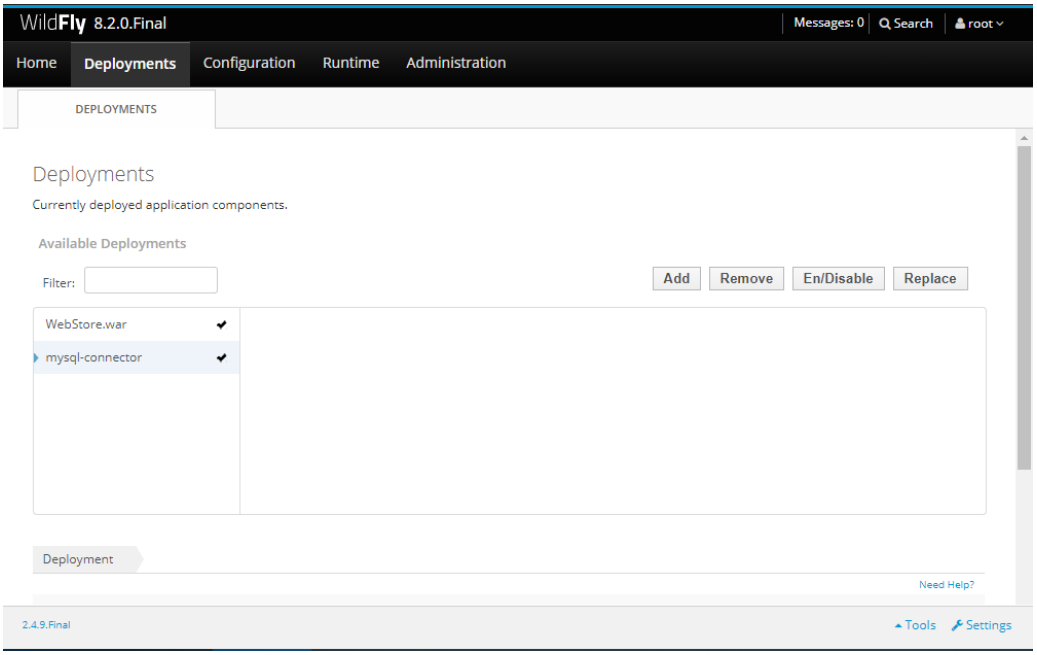


Na slici je prikazan modifikovan dijagram klasa u odnosu na dijagram klasa prvog dela projekta. Kako bi se bolje prikazala simulacija informacionog sistema dodata je klasa Proizvod, i pažnja je bila usmerena ka kreiranju funkcionalne prodavnice, dok je obrazac Učesnik-uloga iskorišćen za apstrakciju mogućih uloga u realnom životu. Na slici su označeni delovi dijagrama koji su u potpunosti ili sa odredjenim modifikacijama implementirani. Npr. Član i Kupac su jedan entitet Član, nema apstrakcije. Kontekstna klasa Supermarket predstavlja bazu podataka.

3. Šema baze podataka



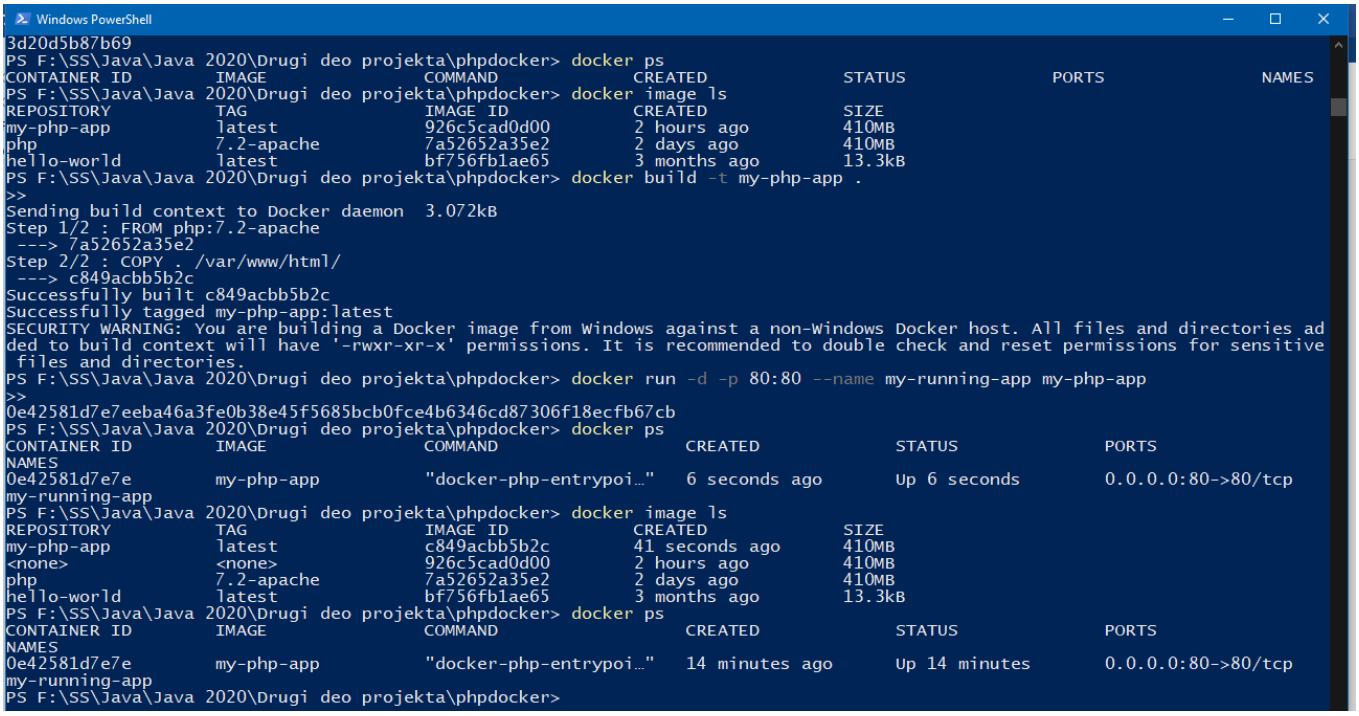
4. Deployment aplikacije



WildFly server:

Na početku se startuje WildFly 8 server.Nakon toga se vrši deployment aplikacije na server (desni klik na projekat, run on Server). Kada se pristupi strani servera (adresa: 127.0.0:9990/console), u tabu Deployments se može videti da je aplikacija postavljena na server. Aplikacija je dostupna na strani localhost:8080/WebStore, na kojoj se otvara default-na stranica (početna stranica je definisana u okviru projekta u fajlu web.xml).

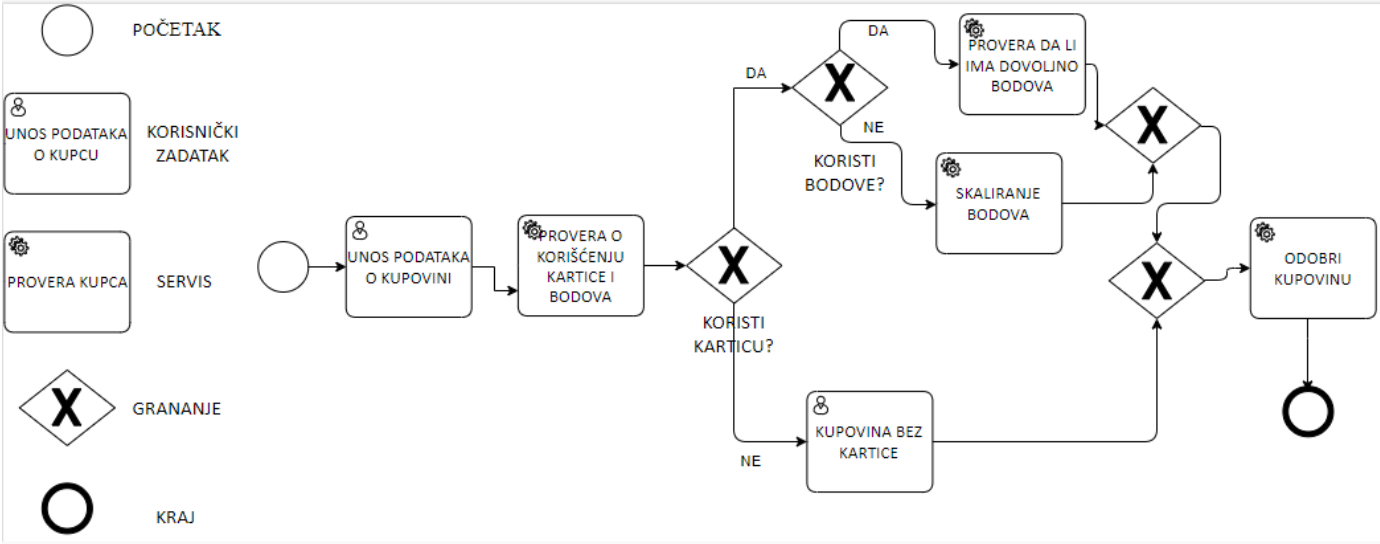
Docker:



Prvo se kreira poseban folder sa fajlom pod nazivom dockerfile koji se koristi za kreiranje specifikacije image-a I izgradnju image-a, I php skriptom koja ce se izvrsavati u docker containeru. Nakon ovoga, pokreće se container na definisanom portu, a njemu se moze pristupiti preko adrese localhost, a tim pristupom se i pokrece php skripta. Za korišćenje skripte koristi se adresa oblika localhost/kartica=vrednost1&bodovi=vrednost2.

5.BPMN Dijagram

Na slici je prikazan BPMN dijagram za kupovinu u supermarketu. Izvršenje ove funkcije poziva php skriptu koja vrši proveru o korišćenju kartice i bodova.



6.Testiranje

Implementirani testovi:

6.1 Pribavljanje podataka o svim karticama

Preduslov: U bazi postoje podaci o bar jednoj kartici.

Opis: Bazi se šalje zahtev za pribavljanje podataka o svim karticama. Baza vraća podatke i podaci se prikazuju.

Očekivani rezultati: Dodat je novi klijent u bazu podataka.

Akteri: Kartica,Baza

6.2 Promena pina kartice

Preduslov: Član poseduje karticu i zna pin kartice.

Opis: Član unosi Id kartice, stari pin i novi pin. Vrši se provera pina.

Očekivani rezultati: Pin kartice je promenjen.

Akteri: Kartica,Član

6.3 Dodavanje kartice

Preduslov: U bazi postoji registrovan član sa kojim će se povezati kartica

Opis: Korisnik unosi podatke o kartici i svoj id i submituje ih.

Očekivani rezultati: Dodata je nova kartica u bazu podataka.

Akteri: Kartica,Član

6.4 Ažuriranje stanja na računu člana

Preduslov: U bazi postoji registrovan član

Opis: Korisnik unosi podatke svoj id i koliko novca želi da uplati.

Očekivani rezultati: Stanje na računu je povećano.

Akteri: Član

Potencijalni testovi:

6.5 Brisanje kartice

Preduslov: U bazi postoji kartica sa unetim id-om

Opis: Korisnik unosi id kartice.

Očekivani rezultati: Kartica je obrisana

Akteri: Kartica

6.6 Brisanje člana

Preduslov: U bazi postoji član sa unetim id-om

Opis: Korisnik unosi svoj id.

Očekivani rezultati: Član je obrisan.

Posledice: Ukoliko je član imao karticu, podaci o toj kartici se takodje brišu.

Akteri: Član, Kartica

6.7 Dodavanje novog proizvoda

Preduslov: Nema.

Opis: Unose se osnovni podaci o proizvodu.

Očekivani rezultati: Proizvod je uspešno dodat u bazu podataka.

Akteri: Proizvod.

6.8 Ažuriranje cene proizvoda

Preduslov: Proizvod sa unetim id-om postoji u bazi.

Opis: Unosi se id proizvoda i nova cena proizvoda.

Očekivani rezultati: Proizvod je ažuriran.

Akteri: Proizvod.

6.9 Ažuriranje količine proizvoda

Preduslov: Proizvod sa unetim id-om postoji u bazi.

Opis: Unosi se id proizvoda i nova količina proizvoda u magacinu.

Očekivani rezultati: Količina proizvoda je ažurirana.

Akteri: Proizvod.