

INFORMACIONI SISTEM BANKE

drugi deo projekta

Ime i prezime: **Nikola Stojković**

Broj indeksa: **18000**

Email: nikoladzonii@elfak.rs

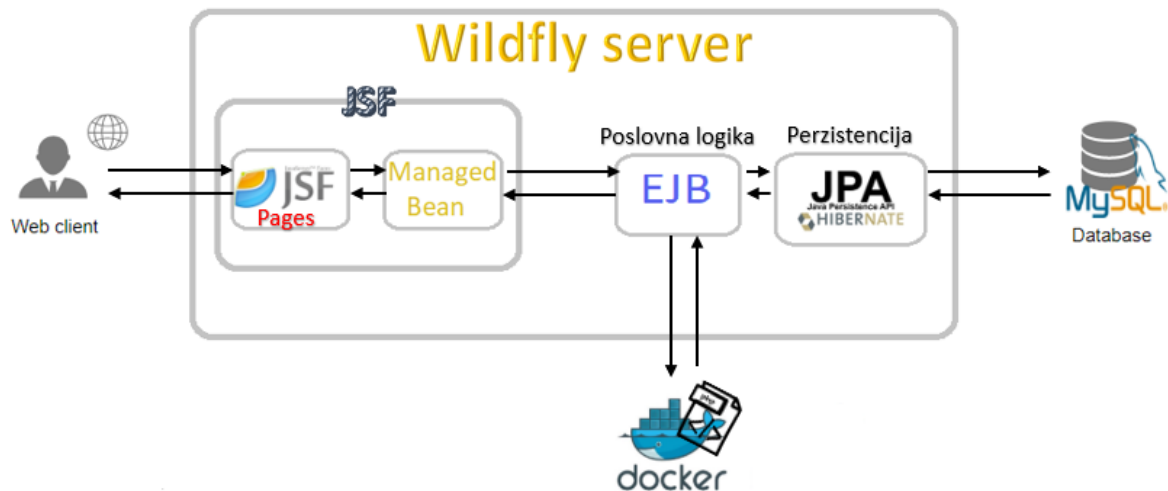
Tekst zadatka

U bazi podataka cuvaju se informacije o racunima klijenata. Svaki klijent može imati jedan ili više racuna. Svaki od racuna može imati dodeljenu kreditnu karticu, koja ima ograniceno trajanje. U sistemu se cuvaju podaci o transakcijama, koje mogu biti priliv ili odliv novca. Novac se može podici u banci (direktno) ili preko bankomata, korišćenjem kartice. Prilikom podizanja veće sume od 3000 eur, neophodno je uneti i razlog podizanja novca. U jednom danu, maksimalno je moguće preuzeti sa bankomata 250 eur, a kraticom u zbiru sa direktnim preuzimanjima i placanjima, nije moguće potrošiti više od 3000 eur. Uzeti u obzir i korišćenje SMS info sistema i mobilne aplikacije, putem kojih se može dobiti obaveštenje o prilivu novca i uvid u trenutno stanje.

Obrazac: *problem poslovnog cilja*

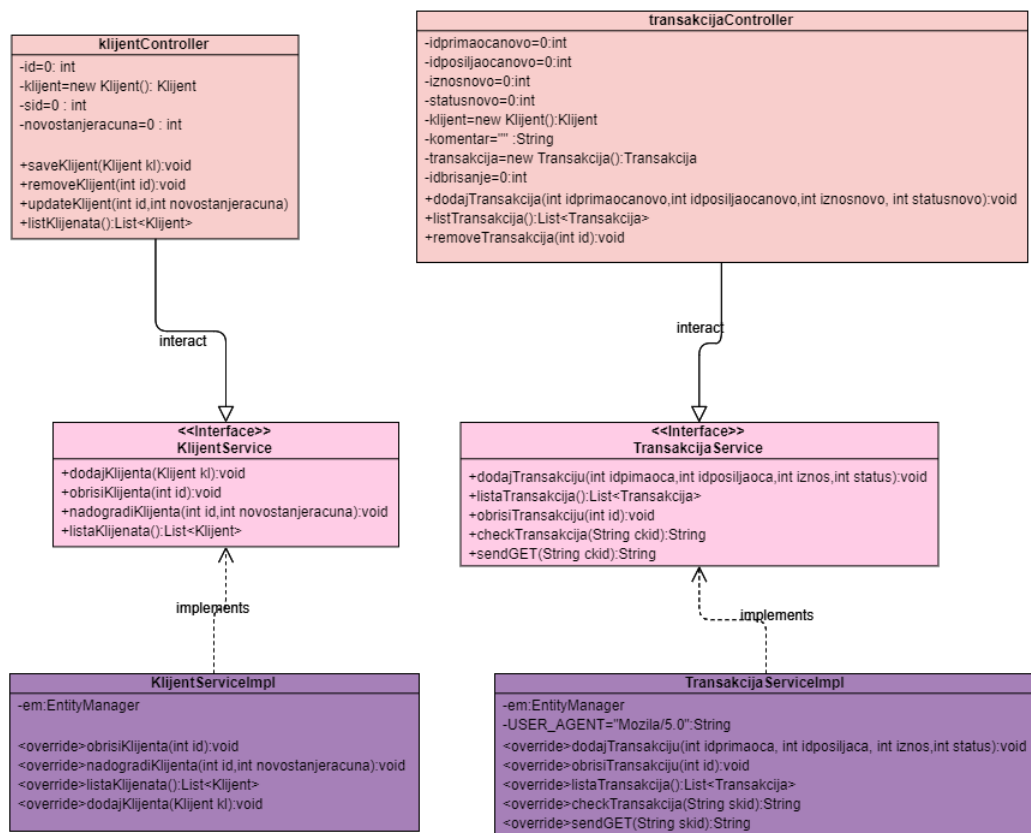
Arhitektura projekta

Arhitektura JSF Web aplikacije:

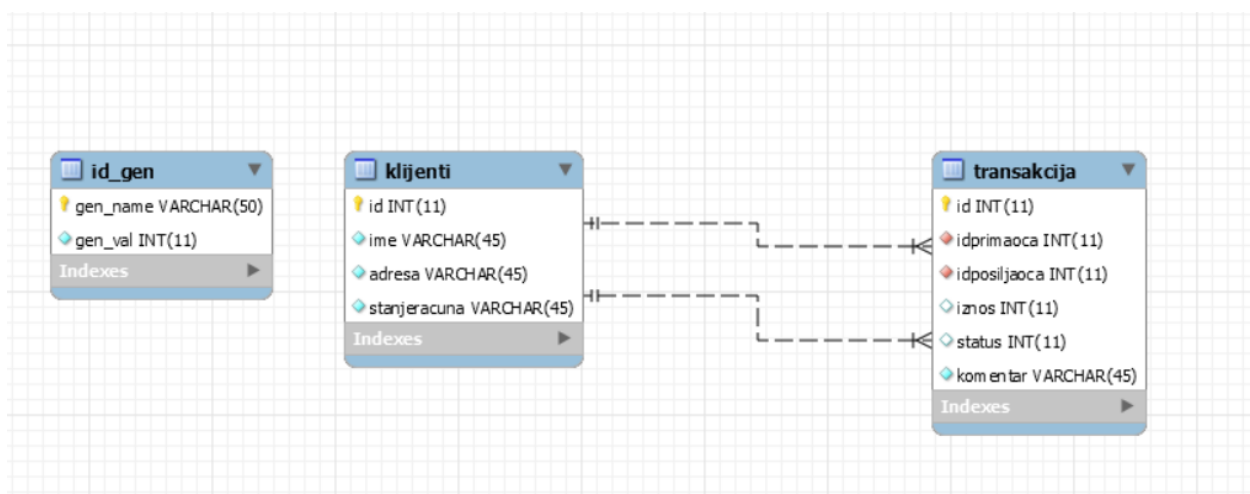


UML klasni dijagram

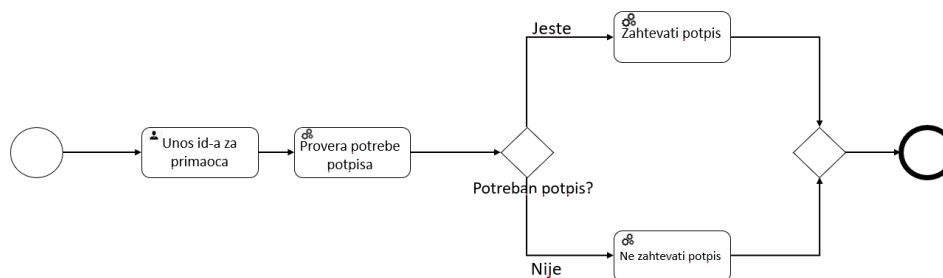
U prilogu se nalazi dijagram na osnovu koga je implementirana aplikacija.



Šema baze podataka



BPMN dijagram



Objašnjenje deployment-a

1. Potrebno je prvo startovati WildFly server – desni klik na „WildFly 8.x at localhost“ u Servers tabu Eclipse-a
Važno: Da bismo koristili Wildfly server potrebno je prvo instalirati Eclipse Enterprise Edition. U datom Eclipse okruženju neophodno je instalirati Jboss Tools plug-in koje moežmo naći u Eclipse Marketplace-u.
2. Nakon pokretanja, dobijamo adresu na kojoj ćemo mu pristupiti u browser-u
3. Nakon što smo otvorili dati link, u kartici Deployment biramo opciju Add

Deployments

Currently deployed application components.

Available Deployments

Filter:

Add

Remove

En/Disable

Replace

4. 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-conne...1.41-bin.jar

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

The screenshot shows the JBoss Configuration console with the 'Configuration' tab selected. The left sidebar shows the 'Subsystems' tree with 'Databases' expanded. The main content area shows the 'JDBC Datasources' configuration page. A table lists available datasources, with 'infsistemi' highlighted. Below the table, the configuration details for 'infsistemi' are displayed.

| Name | JNDI | Enabled? |
|------------|-----------------------------------|----------|
| ExampleDS | java:jboss/datasources/ExampleDS | ✓ |
| infsistemi | java:jboss/datasources/infsistemi | ✓ |

Attributes Connection Pool Security Properties Validation Timeouts

☒ Edit

Name: infsistemi

JNDI: java:jboss/datasources/infsistemi

Is enabled?: true

Statistics enabled?: false

Datasource Class:

Driver: mysql-connector-java-5.1.41-bin.jar_com.mysql.jdbc.Driver_5_1

Na sledećoj slici je odgovarajući prikaz šeme u bazi:



Potrebno je testirati konekciju i ukoliko se dobije informacija da je uspešno, to znači da je povezivanje servera sa bazom podataka uspešno.

6. Da bismo kreirali docker, najpre je potrebno u izabranom folderu kreirati sledeće: -

Dockerfile koji sadrži informacije o serveru na kome će docker biti pokrenut:

```
FROM php:7.2-apache
COPY . /var/www/html/
```

-*index.php* koji će sadržati skriptu koja će se izvršavati prilikom slaja zahteva klijentske aplikacije, konkretno iz biznis logike same aplikacije.

7. U Docker Quickstart Terminal-u navigirati do foldera u kome se nalazi php skripta i DockerFile željenog image-a kako bismo mogli da pokrenemo ono što smo prethodno kreirali

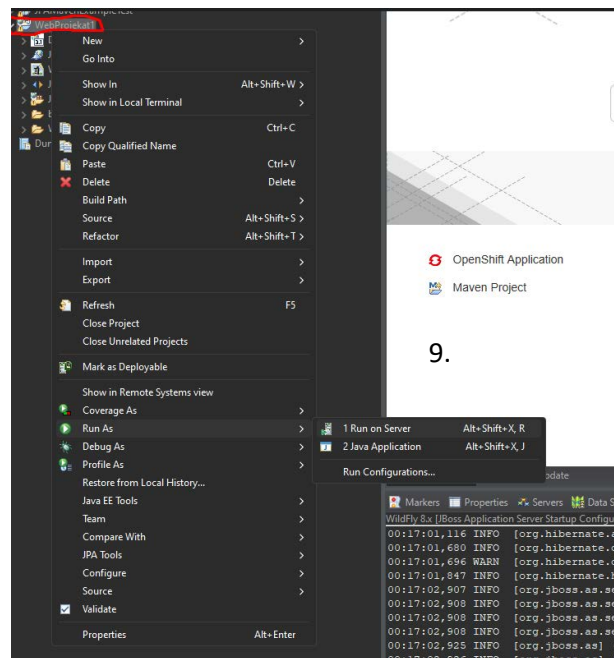
8. Izvršiti komande 'docker build' i 'docker run'

```
dzoni@DESKTOP-5QQ5G8A MINGW64 /c/users/dzoni/desktop/drugi/dockerprojekat
$ docker build -t dockerfile .
Sending build context to Docker daemon 3.072kB
Step 1/2 : FROM php:7.2-apache
----> c61d277263e1
Step 2/2 : COPY . /var/www/html/
----> Using cache
----> ae70788c2095
Successfully built ae70788c2095
Successfully tagged dockerfile:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build con
text will have 'rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
```

```
dzoni@DESKTOP-5QQ5G8A MINGW64 /c/users/dzoni/desktop/drugi/dockerprojekat
$ docker run -d -p 80:80 --name dockerprojekat dockerfile
5f9285ab20dced644d0ce8c3644cb08914942aa19514385514d6fc40c63ccb9a
```

Pristupamo na adresi <http://192.168.99.100/>.

9. Pokrenuti izvršenje aplikacije desnim klikom na projekat u Eclipse-u -> Run As -> Run on Server



10. Biramo već startovan Wildfly server

Nakon što kliknemo na Finish i ukoliko nije bilo problema, možemo videti da je aplikacija deplojovana na Wildfly server.



11. Aplikaciji pristupamo na dobijenoj adresi

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/WebProjekat1/'. The page title is 'BANKA'. Below the title, there is a section 'Dodavanje klijenata i njihovo azuriranje' (Adding clients and their update). This section contains several input fields: 'Ime' (Name), 'Adresa' (Address), 'Stanje racuna' (Account status), 'Id za brisanje' (ID for deletion), 'Id za promenu stanja racuna' (ID for account status change), and 'Novo stanje racuna' (New account status). Each input field has a corresponding button: 'Sacuvaj' (Save), 'Obriši' (Delete), and 'Promeni' (Change). Below the form, there is a table with the following data:

| Id | Ime | Adresa | Stanje racuna |
|----|--------|------------------|---------------|
| 2 | Nikola | Terzovska 2/1 | 304400 |
| 3 | Jana | Ljube Nesica 5 | 922200 |
| 4 | Marko | Kotorska 3 | 16000 |
| 5 | Zarko | B.Radicevica 4/2 | 15000 |
| 6 | Ravi | Textile 6 | 20300 |
| 7 | Milan | Prizrenska 10 | 103400 |
| 8 | Jelena | Cara Dusana 6 | 8000 |
| 9 | Dusan | S.Markovica 50 | 0 |
| 14 | Tra | Cara Miska | 10800 |

Below the table, there is a text input field with the placeholder 'Ukoliko zelite da izvrсите transakciju novca' (If you want to perform a transaction of money) and a button labeled 'Transakcija'.

Test case

Test funkcija da li spisak klijenata vraca neke kupce

1. Testiramo da li funkcija za izlistavanje klijenata ne vraca vrednost NULL. Funkcija bi trebalo da vrati sve klijente iz tabele klijenti.

Preduslovi: Potrebno je da u bazi podataka u tabeli 'klijenti' postoje neki dodati klijenti.

Koraci testa:

1. Dodati nekoliko klijenata u bazu.
2. Pozvati funkciju za pribavljanje klijenata iz baze.
3. Proveriti da li je funkcija vratila vrednost razlicitu od NULL.

Test podaci:

```
INSERT INTO klijenti VALUES ('14','Mina', 'Rade Koncara 1', '9898');
```

Ocekivani rezultati:

Povratna vrednost koja nije NULL.

Post-uslov:

Mozemo vrsiti obradu tih klijenata.

Dobijeni rezultati:

Povratna vrednost funkcije za pribavljanje svih klijenata koji se nalaze u bazi vraca vrednost koja nije NULL.

Status:

Uspesno.

2. Test funkcija da li funkcija dodavanja dodaje klijenta u bazu podataka

Testiramo da li prilikom poziva funkcije dodavanja novog klijenta ona ne vraća NULL vrednost. Rezultat bi trebalo da je klijent ispravno dodat u listu klijenata koja se nalazi u bazi podataka, sa parametrima koji su uneti za njega.

Preduslovi: Postoji funkcija za dodavanje klijenata

Koraci:

1. Uneti attribute potrebne za klijenta.
2. Pozvati funkciju dodavanja klijenta.
3. Proveriti da funkcija nije vratila vrednost NULL.

Test podaci:

```
INSERT INTO klijenti VALUES ('1','Nemanja', 'Rajiceva 1 ', '11100');
```

Ocekivani rezultati:

Povratna vrednost koja nije NULL.

Post-uslov:

Mozemo vrsiti obradu tog klijenta dalje.

Dobijeni rezultati:

Povratna vrednost funkcije za dodavanje klijenta vraca vrednost koja nije NULL.

Status:

Uspesno.

3. Test funkcija za ažuriranje stanja računa klijenta

Testiramo da li funkcija za ažuriranje stanja računa klijenta ne vraca vrednost NULL. Funkcija bi trebalo da promeni stanje računa izabranog klijenta na osnovu id-a.

Preduslovi: Potrebno je da u bazi podataka u tabeli 'klijenti' postoje neki dodati klijenti.

Koraci testa:

- 1.Dodati nekoliko klijenata u bazu.
- 2.Na osnovu id-a izabrati klijenta i uneti novu vrednost stanja racuna.
- 3.Proveriti da li je funkcija vratila vrednost razlicitu od NULL.

Test podaci:

```
UPDATE klijenti SET 'stanjeracuna'='1000' WHERE 'id'='5';
```

Ocekivani rezultati:

Povratna vrednost koja nije NULL.

Post-uslov:

Dalje možemo vršiti obradu sa tim klijentom, obavljati transakciju itd.

Dobijeni rezultati:

Povratna vrednost funkcije za ažuriranje stanja računa klijenta vraca vrednost koja nije NULL.

Status:

Uspesno.

4.Test funkcija da li spisak klijenata vraca neke kupce

Testiramo da li funkcija za izlistavanje transakcija ne vraca vrednost NULL. Funkcija bi trebalo da vrati sve transakcije iz tabele 'transakcija'.

Preduslovi: Potrebno je da u bazi podataka u tabeli 'transakcija' postoje neke dodate transakcije.

Koraci testa:

- 1.Dodati nekoliko transakcija u bazu.
- 2.Pozvati funkciju za pribavljanje transakcija iz baze.
- 3.Proveriti da li je funkcija vratila vrednost razlicitu od NULL.

Test podaci:

```
INSERT INTO transakcija VALUES ('1','4','5', '500','1');
```

Ocekivani rezultati:

Povratna vrednost koja nije NULL.

Post-uslov:

Mozemo vrsiti obradu tih transakcija.

Dobijeni rezultati:

Povratna vrednost funkcije za pribavljanje svih transakcija koji se nalaze u bazi vraca vrednost koja nije NULL.

Status:

Uspesno.