

TEHNIČKO VELEUČILIŠTE U ZAGREBU

STRUČNI STUDIJ INFORMATIKE

Seminarski rad iz Objektno orijentiranog programiranja II:

MOJ NOVČANIK

Matko Horvat, 0246047439

Zagreb, lipanj 2014.

Prijavnica – tema za seminar iz Objektno orijentiranog programiranja II

Ime i prezime:	Matko Horvat
JMBAG:	0246047439
Naziv teme/aplikacije:	Moj Novčanik
Opis teme/aplikacije:	<p>Aplikacija „Moj Novčanik“ će biti desktop aplikacija izrađena u programskom jeziku java. Kasnije je moguća i mobilna verzija za Android pametne telefone. Svrha ove aplikacije bi bila vođenje vlastitog budžeta kako bi imali vlastiti i detaljni uvid u svoje troškove. Koristila bi se tako što bi unosili, mijenjali, brisali ili pregledavali svoja primanja i troškove. Nakon toga bi sama aplikacija iz toga obavljala proračune, vadila neke srednje vrijednosti te bi nakon nekog vremena bilo moguće pretpostaviti u kojem vremenskom roku ćemo potrošiti koliko sredstava i pritom nam vratiti neku poruku upozorenja. Npr. Odaberemo interval od tjedan dana i na kraju svakog tjedna se računa prosječni ukupni trošak. Tako će aplikacija pretpostaviti kada ćemo potrošiti iznos koji nam trenutno stoji u novčaniku i izbaciti poruku o tome. Aplikacija će biti u mogućnosti pretraživati sve troškove i zabilježiti na koje stvari najviše trošimo novac i eventualno o tome obavijestiti korisnika kako bi u budućnosti malo pripazio na to. Biti će moguće kreirati više profila kako bi više osoba moglo koristiti ovu aplikaciju, npr. cijelo kućanstvo.</p>
Pokrivene laboratorijske teme:	<p>Aplikacija „Moj Novčanik“ pokriva sljedeće laboratorijske teme:</p> <ol style="list-style-type: none">1. Razvojno okruženje i aplikacija s nekoliko klasa2. Iznimke, datoteke (XML, LOG) i postavke4. Jednostavno grafičko sučelje6. Korištenje vanjskih biblioteka i komponenti8. Mrežne aplikacije, kriptografijai hash funkcije10. Spajanje aplikacije na bazu podataka

Potpis:

Matko Horvat

Odobrio:

SADRŽAJ

1	UVOD	1
2	ARHITEKTURA SUSTAVA	2
2.1	GRAFIČKO SUČELJE	2
2.1.1	Klasa Glavni.....	3
2.1.1.1	Kartica Novčanik	4
2.1.1.2	Kartica Uplata	5
2.1.1.3	Kartica Isplata.....	6
2.1.1.4	Kartica Zadnje transakcije	7
2.1.1.5	Kartica Popis Uplata	7
2.1.1.6	Kartica Popis Isplata.....	8
2.1.1.7	Kartica Statistika	9
2.1.1.8	Kartica Grafovi.....	9
2.1.1.9	Kartica Pomoć	10
2.1.1.10	Kartica Izlazak	10
2.1.2	Klasa Login	11
2.1.3	Klasa Registracija	12
2.1.4	Klasa Info	13
2.2	BAZA PODATAKA.....	13
3	OPIS FUNKCIONALNOSTI	16
4	ZAKLJUČAK.....	25
5	LITERATURA	26

POPIS SLIKA

Slika 1 - Struktura paketa i klasa.....	2
Slika 2 - Struktura klase Glavni	3
Slika 3 - Tablica uplata.....	6
Slika 4 - Tablica isplata.....	7
Slika 5 - tablica korisnik	12
Slika 6 - Struktura tablice korisnik.....	13
Slika 7 - Struktura tablice isplata.....	14
Slika 8 - Struktura tablice uplata.....	14
Slika 9 - UML dijagram klasa	15
Slika 10 - Login prozor.....	16
Slika 11 - Registracija prozor	17
Slika 12 - Info prozor.....	17
Slika 13 - Glavni prozor	18
Slika 14 - Kartica Uplata	18
Slika 15 - Kartica Isplata.....	19
Slika 16 - Kartica Zadnje Transakcije.....	20
Slika 17 - Kartica Popis Uplata	20
Slika 18 - Kartica Popis Isplata.....	21
Slika 19 - Kartica Statistika	22
Slika 20 - Kartica Grafovi.....	22
Slika 21 - Linijski graf koji prikazuje Isplate	23
Slika 22 - Kartica Pomoć	24
Slika 23 - Kartica Izlazak	24

POPIS KODOVA

Isječak koda 1 - podaci za spajanje na bazu.....	4
Isječak koda 2 - spajanje na bazu i provjera korisnika	4
Isječak koda 3 - pretvaranje KN u EURO	5
Isječak koda 4 - unos uplate u bazu podataka.....	6
Isječak koda 5 - ispisivanje podataka iz baze u JLabel	7
Isječak koda 6 - tablica uplata	8
Isječak koda 7 - ispisivanje podataka iz baze u tablicu	9
Isječak koda 8 - primjer korištenja funkcije MAX, MIN i AVG	9
Isječak koda 9 - linijski graf za prikaz isplata.....	10
Isječak koda 10 - uspješna prijava	11
Isječak koda 11 - gumb koji vodi do prozora za registraciju	11
Isječak koda 12 - kriptiranje lozinke.....	12
Isječak koda 13 - upit za kreiranje baze novčanik	13
Isječak koda 14 - upit za kreiranje tablice korisnici	13
Isječak koda 15 - upit za kreiranje tablice isplata	14
Isječak koda 16 - upit za kreiranje tablice uplata.....	14
Isječak koda 17 - primjer spajanja na bazu	15

1 UVOD

Tema ovog seminarskog rada je izrada aplikacije „Moj Novčanik“ za potrebe kolegija Objektno orijentirano programiranje II. Aplikacija je napisana u programskom jeziku Java koristeći razvojno okruženje Eclipse i zamišljena je kao desktop aplikacija.

Svrha ove aplikacije je vođenje vlastitog budžeta kako bi imali detaljan uvid u svoje troškove. Aplikacija ima mogućnost kreiranja novog računa te stoga više korisnika može koristiti istu aplikaciju. Postoji provjera korisničkih podataka prilikom pokretanja aplikacije kako bi se utvrdilo da li korisnik postoji u bazi, te ako postoji proslijeđuje ga se na glavni ekran gdje može vidjeti sve svoje financije. Prilikom registracije lozinka se kriptira i kao takva sprema u bazu podataka tako da korisnik nema straha od krađe lozinke ili samo iščitavanja iz baze. Nakon prijave korisnik može uplaćivati i isplaćivati iznose u kunama. Koriste se razne iznimke kako bi se provjerilo da li je iznos napisan u dobrom formatu, odnosno da li ima slova i ostalih nedopuštenih znakova u polju za unos iznosa i slično. Aplikacija sve te podatke sprema u bazu podataka te prikazuje korisniku sve transakcije koje je imao, pokazuje najveće, najmanje i prosječne uplate i isplate. Moguć je i grafički prikaz uplata i isplata koristeći linijski ili stupčasti graf, na korisniku je da odabere. Aplikacija koristi MySql bazu podataka.

Za potrebe aplikacije pokriveno je nekoliko laboratorijskih tema:

- Razvojno okruženje i aplikacija s nekoliko klasa
- Iznimke
- Jednostavno grafičko sučelje
- Korištenje vanjskih biblioteka i komponenti
- Kriptografija i hash funkcije
- Spajanje aplikacije na bazu podataka

2 ARHITEKTURA SUSTAVA

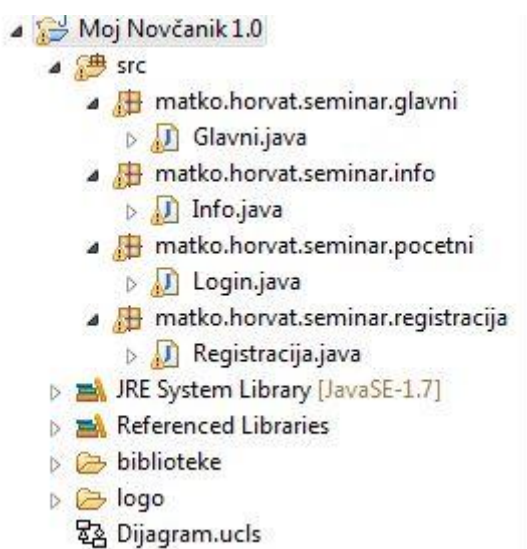
U ovom poglavlju bit će opisane sve tehnologije korištene prilikom izrade aplikacije.

Aplikacija „Moj Novčanik“ pisana je u programskom jeziku Java, jednom od trenutno najpopularnijih programskih jezika. Korišteno je *open source* razvojno okruženje Eclipse. Korišteni su brojni *pluginovi* koje nam nudi Eclipse, „ObjectAid UML Explorer“ za kreiranje UML dijagrama klasa te „Windows Bilder Pro“ za kreiranje grafičkog sučelja pomoću Swing komponenti.

2.1 GRAFIČKO SUČELJE

Prvi korak u stvaranju ove aplikacije bila je izrada grafičkog sučelja. Klase koje definiraju grafičko sučelje nalaze se u paketima

- **matko.horvat.seminar.glavni**
- **matko.horvat.seminar.info**
- **matko.horvat.seminar.pocetni**
- **matko.horvat.seminar.registracija**

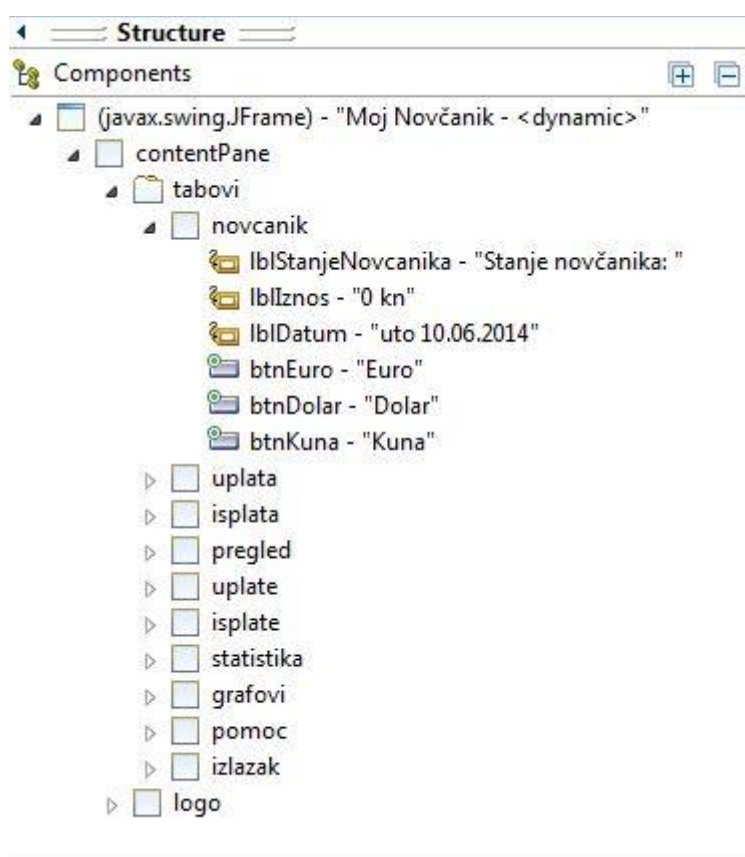


Slika 1 - Struktura paketa i klasa

2.1.1 Klasa Glavni

Klasa Glavni se nalazi u paketu `matko.horvat.seminar.glavni` i ona sadrži glavni prozor u kojemu se nalaze svi ostali tabovi i funkcionalnosti. Grafičko sučelje napravljeno je pomoću plugina „Windows Bilder Pro“ koji nam omogućava jednostavnu izradu sučelja pomoću *drag and drop* funkcije. Korištene su Swing komponente kao što su `JPanel`, `JLabel`, `JBUTTON`, `JTextField`, `JTable`.

Strukturu prozora možemo vidjeti na slijedećoj slici:



Slika 2 - Struktura klase Glavni

Vidimo na slici da je u `contentPane` dodan kontejner `JTabbedPane` u koji su dodavani redom `JPanel` kontejneri koji čine tabove, odnosno kartice. Klikom na svaku tu karticu, dobijem prikaz komponenti koji sadrži dodani `JPanel`. Na slici vidimo primjer kako `JPanel` sadrži 3 `JLabel` komponente i 3 `JBUTTON` komponente.

U slijedećem tekstu će biti opisani kodovi i funkcije pojedinih tabova pošto je velika većina aplikacije sadržana upravo u njima.

2.1.1.1 Kartica Novčanik

Kartica Novčanik sadrži 3JLabela i 3JButtona. Varijabla imena lblIznos pokazuje trenutni iznos novčanika. Ako je korisnik prvi puta pokrenuo aplikaciju nakon registracije, iznos će biti postavljen na 0, a ako je korisnik već koristio aplikaciju, prikazati će iznos izračunat na temelju svih uplata i isplata iz baze. To se izvodi pomoću upita kojima u bazi pretražujemo korisnika.

Primjer podataka za spajanje na bazu možemo vidjeti u slijedećem kodu:

```
// podaci za spajanje na bazu
final static String url = "jdbc:mysql://localhost:3306/";
final static String dbName = "novcanik";
final static String driver = "com.mysql.jdbc.Driver";
final static String userName = "root";
final static String password = "6669";
```

Isječak koda 1 - podaci za spajanje na bazu[3]

Samo spajanje na bazu i provjeravanje pomoću upita da li je korisnik prije uplaćivao išta možemo vidjeti u slijedećem kodu:

```
try {

Class.forName(driver).newInstance();
Connection konekcija = DriverManager.getConnection(url + dbName,
userName, password);

// provjerava da li postoji korisnik u bazi uplata, ako ne, stavlja početni
// iznos na 0
Statement upit = konekcija.createStatement();
ResultSet rezultat = upit
.executeQuery("SELECT COUNT(korisnik) FROM uplata WHERE korisnik LIKE '"
+ username + "'");
rezultat.next();
String broj = rezultat.getString(1);
int ukBroj = (int) Double.parseDouble(broj);

if (ukBroj == 0) {
iznosU = 0;
}

...

}

} catch (Exception a) {
a.printStackTrace();

}
```

Isječak koda 2 - spajanje na bazu i provjera korisnika[3]

Još jedna funkcionalnost kartice Novčanik je ta da pomoću 3 gumba možemo pretvarati naš iznos u druge valute. Treba imati na umu da program i dalje radi sa kunama i prilikom ponovnog pokretanja aplikacije, glavni iznos se ponovno prikazuje u kunama.

Primjer gumba Euro i njegovog koda za pretvorbu:

```
// PRETVARANJE U DRUGE VALUTE-----  
// EURO-----  
JButton btnEuro = new JButton("Euro");  
btnEuro.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
  
        iznosEUR = (float) (iznos * 0.131800);  
  
        if (iznos != iznosEUR) {  
  
            if (iznos < 0) {  
                lblIznos.setForeground(Color.RED);  
            }  
  
            else  
                lblIznos.setForeground(Color.BLACK);  
  
            lblIznos.setText(mojformat.format(iznosEUR) + " €");  
            EUR++;  
            USD = 0;  
        }  
    }  
});  
btnEuro.setBounds(43, 233, 89, 23);  
novcanik.add(btnEuro);
```

Isječak koda 3 - pretvaranje KN u EURO

2.1.1.2 Kartica Uplata

Kartica Uplata sadrži 2 JLabela i 2 JtextFielda te gumb Uplati pomoću kojega uplaćujemo željeni iznos i njegovo ime.

Unos uplate u bazu izgleda ovako:

```
// stavlja uplatu u bazu podataka  
Statement upit = konekcija.createStatement();  
int rezultat = upit  
.executeUpdate("INSERT INTO uplata (korisnik, nazivUplate, iznos, vrijemeUplate)  
VALUES ('"  
+ username  
+ "','"  
+ uplataNaziv.getText()  
+ "','"  
+ uplata1 + ",NOW())");  
  
if (rezultat == 1) {
```

```

JOptionPane.showMessageDialog(null, "Uplatili ste "
+ mojformat.format(uplata1)
+ " kn u svoj novčanik!");
uplataIznos.setText("");
uplataNaziv.setText("");

}

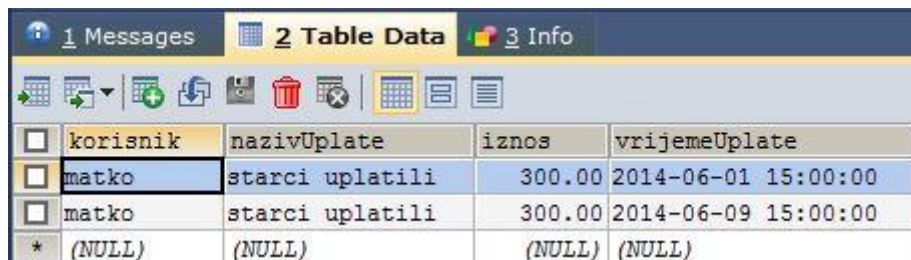
konekcija.close();

}

```

Isječak koda 4 - unos uplate u bazu podataka

Uplata se sprema u bazu podataka imena „novčanik“, u tablicu „uplata“ koja izgleda ovako:



	korisnik	nazivUplate	iznos	vrijemeUplate
<input type="checkbox"/>	matko	starci uplatili	300.00	2014-06-01 15:00:00
<input type="checkbox"/>	matko	starci uplatili	300.00	2014-06-09 15:00:00
*	(NULL)	(NULL)	(NULL)	(NULL)

Slika 3 - Tablica uplata

Također se sprema i datum kada je iznos uplaćen i točno vrijeme.

2.1.1.3 Kartica Isplata

Kartica Isplata sadrži 2 JLabela i 2 JtextFielda te gumb Isplati pomoću kojega uplaćujemo željeni iznos i njegovo ime.

Kod kojime isplaćujemo željeni iznos iz novčanika i bilježimo isplatu u bazu sličan je onome za uplatu.

Tablica „isplata“ izgleda ovako:

1 Messages 2 Table Data 3 Info			
<input type="checkbox"/> korisnik	nazivIsplate	iznos	vrijemeIsplate
<input type="checkbox"/> matko	karta za vlak	39.80	2014-06-01 19:00:00
<input type="checkbox"/> matko	karta za vlak	39.80	2014-06-01 19:00:00
<input type="checkbox"/> matko	lizalica	1.50	2014-06-02 11:34:23
<input type="checkbox"/> matko	platio sobu za 6. mjesec	200.00	2014-06-02 14:00:26
<input type="checkbox"/> matko	sladoled	6.00	2014-06-02 18:00:26
<input type="checkbox"/> matko	piva	10.00	2014-06-03 15:00:26
<input type="checkbox"/> matko	piva	10.00	2014-06-03 15:30:46

Slika 4 - Tablica isplata

2.1.1.4 Kartica Zadnje transakcije

Kartica Zadnje transakcije sadrži 14 JLabela od kojih 8 služi samo za označavanje i imenovanje dok su 6 JLabela dinamički koji prikazuju podatke iz baza i skloni su promjenama. Također postoji i gumb Osvježi koji provjerava eventualne promjene koje su nastale u bazi i shodno tome mijenja sadržaj dinamičkih JLabela.

```
// spajanje na bazu
Class.forName(driver).newInstance();
Connection konekcija = DriverManager.getConnection(url
+ dbName, userName, password);

// ZADNJA UPLATA
Statement upit = konekcija.createStatement();
ResultSet rezultat1 = upit
.executeQuery("SELECT nazivUplate, iznos,
DATE_FORMAT DATE(vrijemeUplate), '%d%.%m%.%Y') FROM uplata WHERE korisnik LIKE '"
+ username
+ "' ORDER BY vrijemeUplate DESC LIMIT 0,1;");
rezultat1.next();

String nazivUplate = rezultat1.getString(1);
String iznosUplate = rezultat1.getString(2);
String datumUplate = rezultat1.getString(3);

lblRezNazivUPLATA.setText(nazivUplate);
lblRezIznosUPLATA.setText(iznosUplate + " kn");
lblRezDatumUPLATA.setText(datumUplate);
```

Isječak koda 5 - ispisivanje podataka iz baze u JLabel

2.1.1.5 Kartica Popis Uplata

Kartica Popis Uplata sadrži tablicu koja prikazuje sve uplate ikad uplaćene od strane trenutno prijavljenog korisnika. To je ostvareno pomoću JScrollPane i JTable.

```
JScrollPane scrollPane1 = new JScrollPane();
```

```

scrollPane1.setViewportBorder(new LineBorder(new Color(0, 0, 0)));
scrollPane1.setToolTipText("");
update.add(scrollPane1);

int red1 = 0;
Object[][] podaci1 = new Object[red1][3];
String[] stupac1 = { "Naziv", "Iznos (kn)", "Datum" };

tablicaUplata = new JTable();
tablicaUplata.setEnabled(false);
tablicaUplata.setModel(new DefaultTableModel(new Object[][] {},
new String[] { "Naziv", "Iznos (kn)", "Datum" }));

tablicaUplata.setBorder(new LineBorder(new Color(0, 0, 0)));
scrollPane1.setViewportView(tablicaUplata);

```

Isječak koda 6 - tablica uplata

Također postoji i gumb Osvježi koji provjerava eventualne promjene koje su nastale u bazi i shodno tome mijenja sadržaj tablice. Provjerava da li ima kakvih promjena, ako nema sve ostaje isto, ako ima, dodaje novi red ili više njih (ovisno o broju novih uplata).

2.1.1.6 Kartica Popis Isplata

Kartica Popis Isplata sadrži tablicu koja prikazuje sve isplate ikad uplaćene od strane trenutno prijavljenog korisnika. To je također ostvareno pomoću JScrollPane i JTable.

Ispisivanje podataka iz baze u tablicu ostvareno je slijedećim kodom:

```

ResultSet rezultat = upit
.executeQuery("SELECT nazivIsplate, iznos,
DATE_FORMAT(DATE(vrijemeIsplate), '%d%.%m%.%Y') FROM isplata WHERE korisnik LIKE '"
+ username
+ "' LIMIT "
+ ukupanBrojIsplata
+ ","
+ razlika
+ ";");

while (rezultat.next()) {

String naziv = "";
String iznos = "";
String datum = "";

naziv = rezultat.getString(1);
iznos = rezultat.getString(2);
datum = rezultat.getString(3);
((DefaultTableModel) tablicaIsplata.getModel())
.addRow(new Object[] { naziv, iznos, datum, });

```

```
}
```

Isječak koda 7 - ispisivanje podataka iz baze u tablicu

U kartici Popis Isplata isto postoji gumb Osvježi koji provjerava eventualne promjene koje su nastale u bazi i shodno tome mijenja sadržaj tablice. Provjerava da li ima kakvih promjena, ako nema sve ostaje isto, ako ima, dodaje novi red ili više njih (ovisno o broju novih uplata).

2.1.1.7 Kartica Statistika

Kartica Statistika sadrži 16 JLabela od kojih je 8 dinamičkih i služe za prikazivanje podataka iz baze koji su podložni promjenama isto kao i u karticama Zadnje transakcije samo se koriste drugi upiti kako bi dobili željene rezultate.

```
ResultSet rezultatMaxUplata = upit
.executeQuery("SELECT MAX(iznos) FROM uplata WHERE korisnik LIKE '"
+ username + "'");

ResultSet rezultatMinUplata = upit
.executeQuery("SELECT MIN(iznos) FROM uplata WHERE korisnik LIKE '"
+ username + "'");

ResultSet rezultatAvgUplata = upit
.executeQuery("SELECT AVG(iznos) FROM uplata WHERE korisnik LIKE '"
+ username + "'");
```

Isječak koda 8 - primjer korištenja funkcije MAX, MIN i AVG

Svim gore navedenim funkcijama uzimamo podatke za isplate. Računaju se najveće i najmanje uplate i isplate. Kao i na svim ostalim karticama imamo gumb Osvježi.

2.1.1.8 Kartica Grafovi

Kartica Grafovi sadrži 2 JLabela i 4 JButtona. JLabeli služe za oznaku, dok gumbima otvaramo novi JFrame sa prikazanim linijskim ili stupčastim grafom za uplate ili isplate. Graf na x osi prikazuje vrijednost iznosa, a na y osi se nalazi datum i vrijeme Uplate/Isplate. Kako bi graf bio u potpunosti funkcionalan bilo je potrebno dodati odgovarajuće vanjske JAR datoteke u „classpath“ Java projekta **jcommon-1.0.21.jar** i **jfreechart-1.0.17.jar**.^[2]

Izgled selektiranja vrijednosti iz baze koje se stavljaju u graf:

```
// Linijski graf za isplate
```

```

JButton btnIsplateLinijski = new JButton("Linijski graf");
btnIsplateLinijski.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {

            Class.forName(driver).newInstance();
            Connection konekcija = DriverManager.getConnection(url
            + dbName, userName, password);

            String upit = ("SELECT vrijemeIsplate, iznos FROM isplata WHERE korisnik LIKE '"
            + username + "'");

            JDBCCategoryDataset dataset = new JDBCCategoryDataset(
            konekcija, upit);

            JFreeChart chart = ChartFactory.createLineChart("Isplate",
            "Datum", "Iznos (kn)", dataset,
            PlotOrientation.VERTICAL, false, true, true);

            BarRenderer renderer = null;
            CategoryPlot plot = null;

            renderer = new BarRenderer();

            ChartFrame frame = new ChartFrame("Graf ", chart);

            frame.setVisible(true);
            frame.setSize(700, 560);

        }

        catch (Exception a) {
            a.printStackTrace();
        }
    }
}

```

Isječak koda 9 - linijski graf za prikaz isplata[2]

2.1.1.9 Kartica Pomoć

Kartica Pomoć sadrži JLabels koji opisuju svaku karticu i njenu funkcionalnost te kako se koristi. Dodan je JScrollPane u koji je dodan JPanel kako bi se moglo pomoću vertikalnog klizača vidjeti sav tekst.

2.1.1.10 Kartica Izlazak

Kartica Izlazak sadrži dva JButtona naziva Izlazak i Promijeni korisnika. Pritiskom na gumb Izlazak pomoću JOptionPanea se prikazuje poruka sadržaja „Jeste li sigurni da želite izaći?“. Odabirom gumba „Da“ aplikacija se zatvara pomoću metode dispose(). Ukoliko se pritisne „Ne“, zatvara se prozor sa porukom te korisnik ostaje raditi u aplikaciji.

Drugi gumb naziva „Promijeni korisnika“ služi za odjavu trenutnog korisnika i prijavu novog. Pritiskom na taj gumb otvara se pomoću JOptionPanea prozor sa porukom sadržaja „Jeste li sigurni da želite promijeniti korisnika?“. Pritiskom na gumb „Da“ Zatvara se trenutni glavni prozor i otvara se početni prozor za prijavu naziva Login. Ukoliko se pritisne gumb „Ne“, zatvara se prozor sa porukom i korisnik ostaje raditi u aplikaciji.

2.1.2 Klasa Login

Ova klasa sadrži izgled grafičkog sučelja za prijavu korisnika u aplikaciju s unosom korisničkog imena.

Klasa Login sadrži main metodu jer se ovo sučelje prvo prikazuje kada pokrenemo aplikaciju. Prozor Login klase sadrži dva JTextFielda naziva username i pass koji služe za upisivanje korisničkog imena i lozinke. Vrš se provjere kojima se utvrđuje da li su polja ispunjena, i da li korisnik uistinu postoji u bazi podataka u odgovarajućoj tablici naziva „korisnici“. Ukoliko se podaci podudaraju, zatvara se trenutni prozor i poziva glavni prozor te se dalje proslijeđuje ime korisnika kako bi se moglo dalje koristiti u svrhu prikupljanja odgovarajućih podataka iz baze podataka.

```
dispose();
String korisnik = username.getText();
Glavni glavni = new Glavni(korisnik);
glavni.glavni();
```

Isječak koda 10 - uspješna prijava

Odgovarajuće navedene provjere i prijava u aplikaciju se događaju pritiskom na gumb imena „Prijava“.

Prozor Login sadrži još gumbe „Napravi račun“, „Izlaz“ i „Info“.

Pritiskom na gumb „Napravi račun“ priziva se prozor za registraciju.

```
JButton btnNapraviRaun = new JButton("Napravi račun");
btnNapraviRaun.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
        Registracija reg = new Registracija();
        reg.registracija();
    }
});
btnNapraviRaun.setBounds(24, 175, 126, 23);
contentPane.add(btnNapraviRaun);
```

Isječak koda 11 - gumb koji vodi do prozora za registraciju

Pritiskom na gumb „Izlaz“ otvara se pomoću JOptionPanea prozor sa porukom sadržaja „Jeste li sigurni da želite izaći iz programa?“. Pritiskom na gumb „Da“

zatvara se aplikacija. Ukoliko se pritisne gumb „Ne“, zatvara se prozor sa porukom i korisnik ostaje raditi u aplikaciji.

Pritiskom na gumb „Info“ otvara se prozor naziva Info koji se sastoji od JLabela koji sadrže ime autora aplikacije, ime aplikacije te kratki opis aplikacije.

2.1.3 Klasa Registracija

Klasa Registracija služi kao prozor za registraciju gdje se korisnik može registrirati te ukoliko je uspješno ispunio polja unijeti svoje podatke u bazu podataka te kasnije koristiti aplikaciju sa svojim podacima koji se pamte. Prozor sadrži JTextField koji se zove username, te 2 JPasswordFielda pass1 i pass2, a služe za unos željenog korisničkog imena, te lozinke koja se mora unijeti dva puta.

Pritiskom na gumb „Napravi račun“ vrše se provjere da li su polja popunjena, da li je korisničko ime već zauzeto te da li je ista lozinka unesena dva puta. Program ispisi prigradnu poruku ovisno o neispunjenom uvjetu. Nakon uspješne registracije prikaže se JOptionPane.showMessageDialog sa porukom da je registracija uspješno izvršena te kojim korisničkim imenom smo se registrirali. Lozinka se kriptira[1] te se sprema u takvom obliku u tablicu čime je zagarantirana korisnikova sigurnost i privatnost. Također se pohranjuje i točno vrijeme kada se korisnik registrirao. Kako bi kriptiranje lozinke bilo moguće dodana je odgovarajuća biblioteka u „classpath“ Java projekta **commons-codec-1.7.jar**. [1]

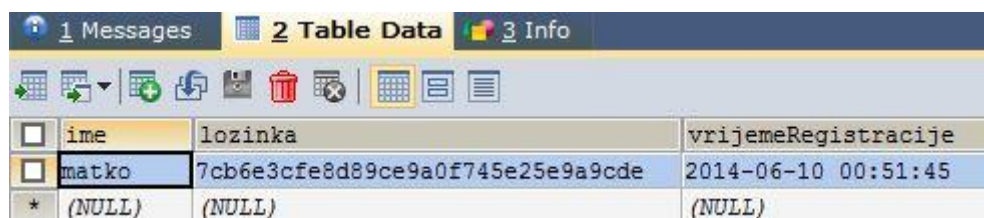
```
char[] unos1 = pass1.getPassword();
String sifra1 = new String(unos1);

//kriptirana lozinka se unosi u bazu

DigestUtils.md5Hex( sifra1 )
```

Isječak koda 12 - kriptiranje lozinke[1]

Podaci se spremaju u tablicu naziva „korisnici“.



ime	lozinka	vrijemeRegistracije
matko	7cb6e3cfe8d89ce9a0f745e25e9a9cde	2014-06-10 00:51:45
* (NULL)	(NULL)	(NULL)

Slika 5 - tablica korisnik

2.1.4 Klasa Info

Klasa Info sadrži JLabels koji sadrže ime autora, ime aplikacije i kratki opis aplikacije. Pritiskom na JButton „Zatvori“ Info prozor se zatvara i otvoren je opet početni Login prozor.

2.2 BAZA PODATAKA

Baza podataka je ključni dio ove aplikacije i bez nje aplikacija ne bi bila u mogućnosti funkcionirati. Korištena baza podataka je MySQL te se prvo trebala napraviti baza podataka naredbom:

```
CREATE DATABASE novcanik;
```

Isječak koda 13 - upit za kreiranje baze novcanik

Nakon toga je bilo potrebno stvoriti tablice u koje će se spremati podaci. Tablice se zovu „korisnici“, „isplata“ i „uplata“ i one su kreirani slijedećim upitom:

```
CREATE TABLE korisnici(  
ime VARCHAR(50),  
lozinka VARCHAR(50),  
vrijemeRegistracije DATETIME  
);
```

Isječak koda 14 - upit za kreiranje tablice korisnici

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	ime	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	lozinka	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	vrijemeRegistracije	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 6 - Struktura tablice korisnik

```
CREATE TABLE isplata(  
korisnik VARCHAR(50),  
nazivIsplate VARCHAR(100),  
iznos DECIMAL(7,2),
```

vrijemelsplate DATETIME

);

Isječak koda 15 - upit za kreiranje tablice isplata

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	korisnik	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nazivIsplate	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	iznos	decimal	7,2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	vrijemeIsplate	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 7 - Struktura tablice isplata

```
CREATE TABLE uplata(  
    korisnik VARCHAR(50),  
    nazivUplate VARCHAR(100),  
    iznos DECIMAL(7,2),  
    vrijemeUplate DATETIME  
);
```

Isječak koda 16 - upit za kreiranje tablice uplata

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	korisnik	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nazivUplate	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	iznos	decimal	7,2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	vrijemeUplate	datetime			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 8 - Struktura tablice uplata

Veza između tablica je ostvarena zahvaljujući korisniku čije se ime, odnosno username upisuje u tablicu uplata i tablicu isplata prilikom svake uplate ili isplate i zahvaljujući tome je moguće u aplikaciji dohvaćati podatke samo za korisnika koji je trenutno prijavljen. Nakon kreiranja baze podataka dodana je odgovarajuća vanjska JAR datoteka u „classpath“ Java projekta **mysql-connector-java-5.1.6-bin.jar**. [3]

```
// podaci za bazu  
final static String url = "jdbc:mysql://localhost:3306/";  
final static String dbName = "novcanik";  
final static String driver = "com.mysql.jdbc.Driver";  
final static String userName = "root";  
final static String password = "6669";
```

```
try {
```

```

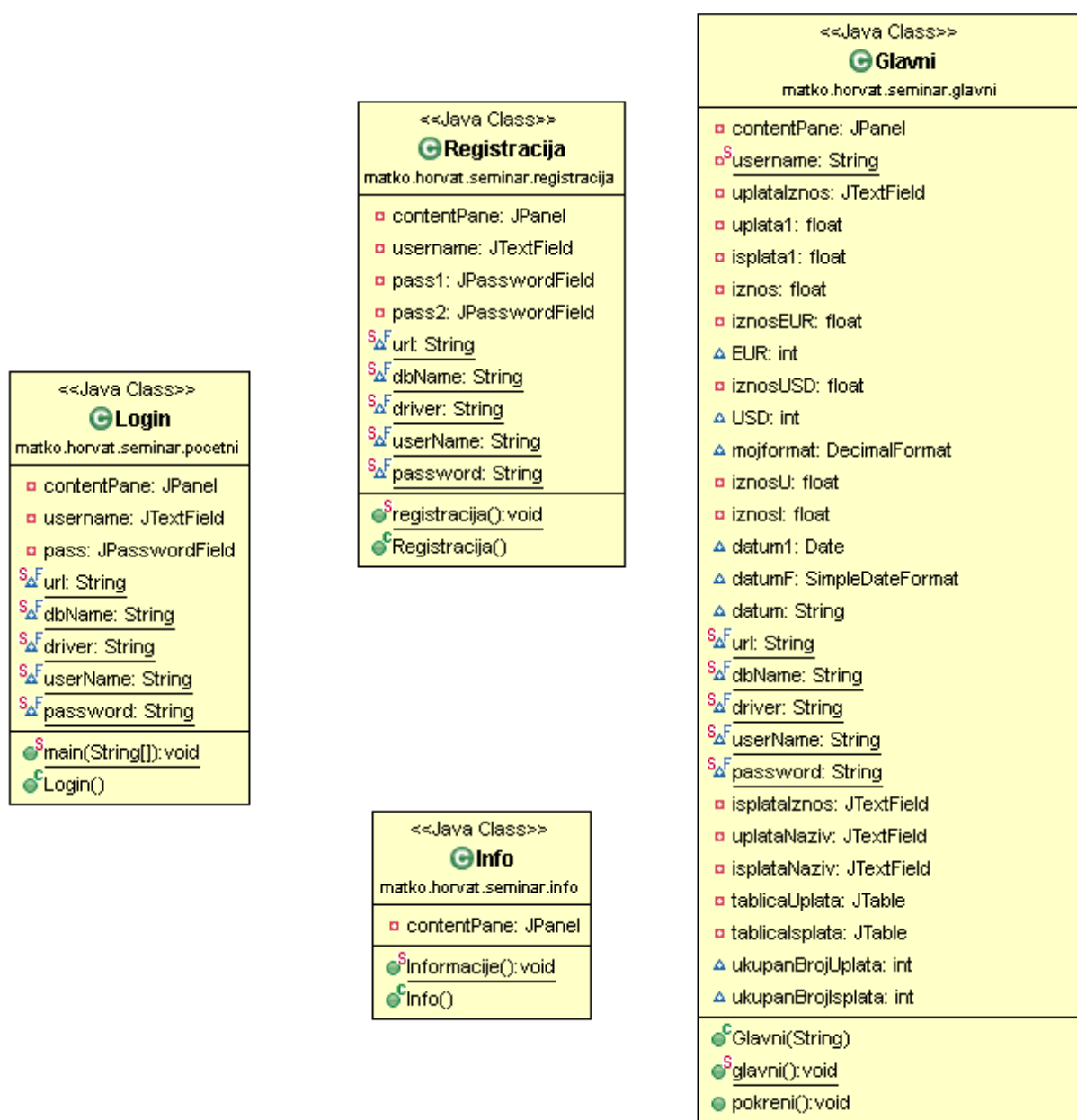
Class.forName(driver).newInstance();
Connection konekcija = DriverManager.getConnection(url
+ dbName, userName, password);
konekcija.close();

} catch (Exception e) {
e.printStackTrace();
}

```

Isječak koda 17 - primjer spajanja na bazu

Na kraju poglavlja u kojem smo opisali arhitekturu sustava na slici je prikazan UML dijagram klasa korištenih u ovome radu.

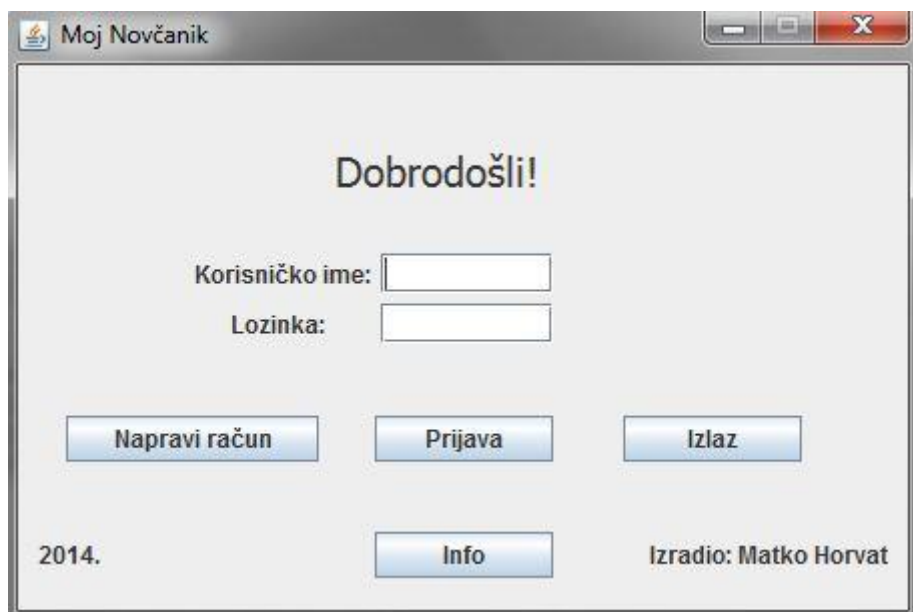


Slika 9 - UML dijagram klasa

3 OPIS FUNKCIONALNOSTI

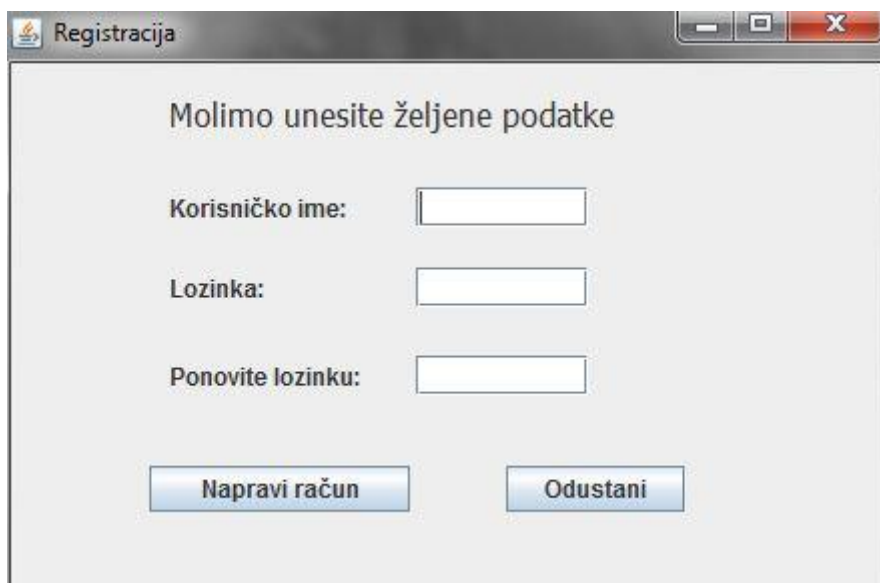
U ovom poglavlju bit će opisane sve funkcionalnosti ove aplikacije, prikaz grafičkoga sučelja iz korisničke perspektive.

Prvo što korisnik vidi kad pokrene aplikaciju je okvir za prijavu u kojem se unosi korisničko ime i lozinka (Slika 10).



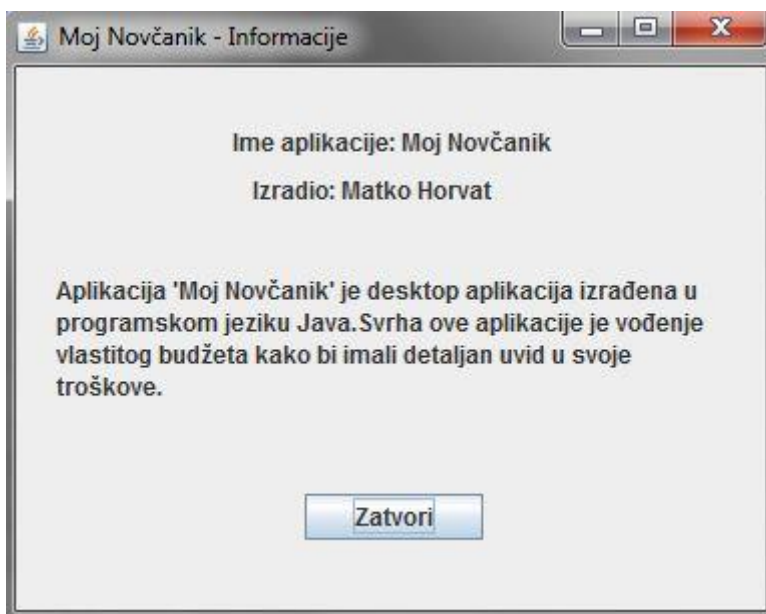
Slika 10 - Login prozor

Klikom na gumb „Napravi račun“ otvara se prozor za registraciju u kojemu se korisnik može registrirati unoseći željeno korisničko ime te dva puta lozinku u polja predviđena za to (Slika 11).



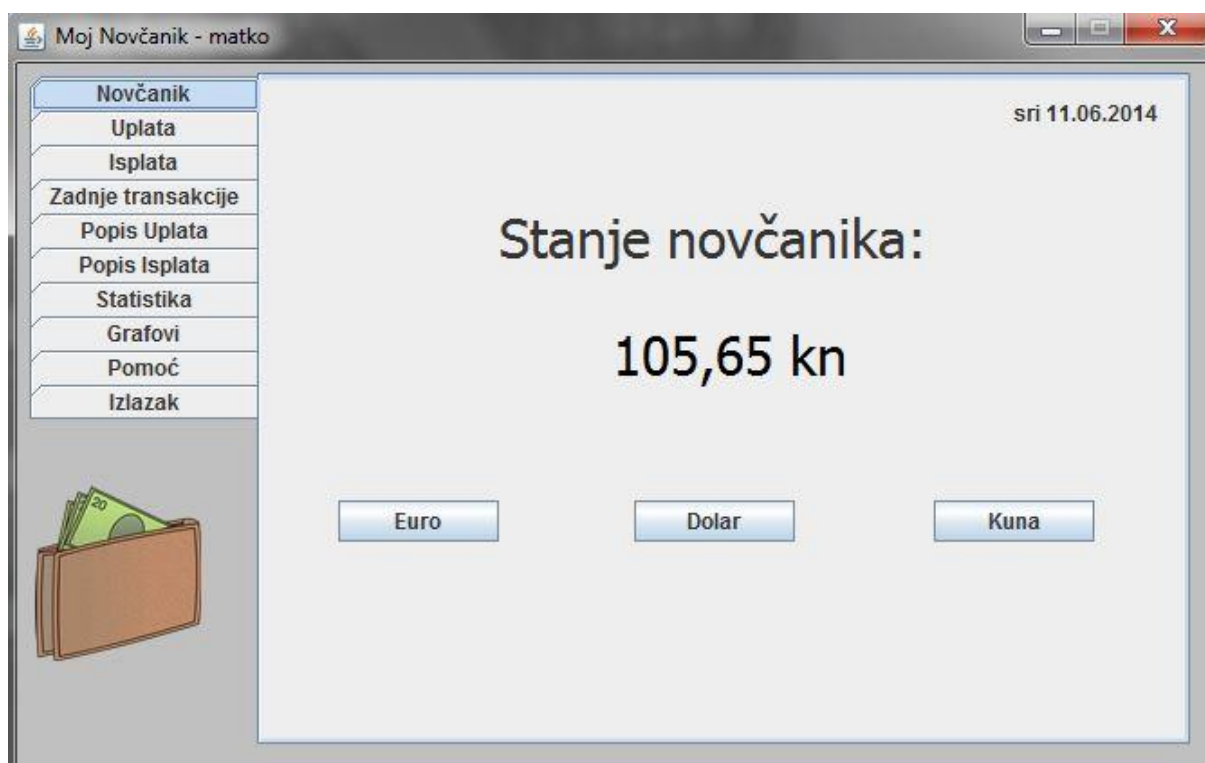
Slika 11 - Registracija prozor

Klikom na gumb „Info“ otvara se prozor sa nekim općenitim informacijama kao što su ime autora aplikacije, ime aplikacije te kratki opis aplikacije (Slika 12).



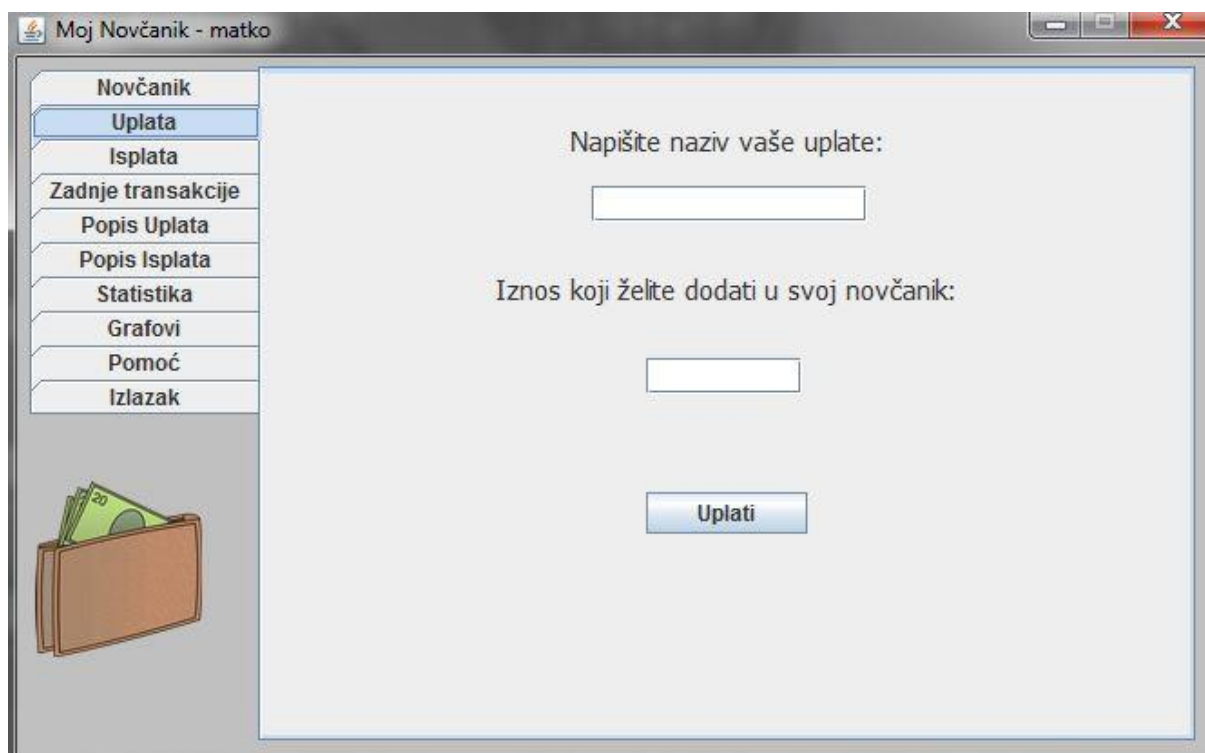
Slika 12 - Info prozor

Ukoliko korisnik već postoji u bazi, odnosno ako se uspješno registrirao te pravilno unio podatke u početni prozor i pritisnuo gumb „Prijava“, korisniku je omogućen pristup glavnom prozoru u kojem se nalaze sve važne funkcije aplikacije. Glavni prozor (Slika 13) se sastoji od 10 kartica od kojih je prva naziva „Novčanik“. U njoj se prikazuje trenutni iznos u korisnikovom novčaniku. Moguća je pretvorba u druge valute.



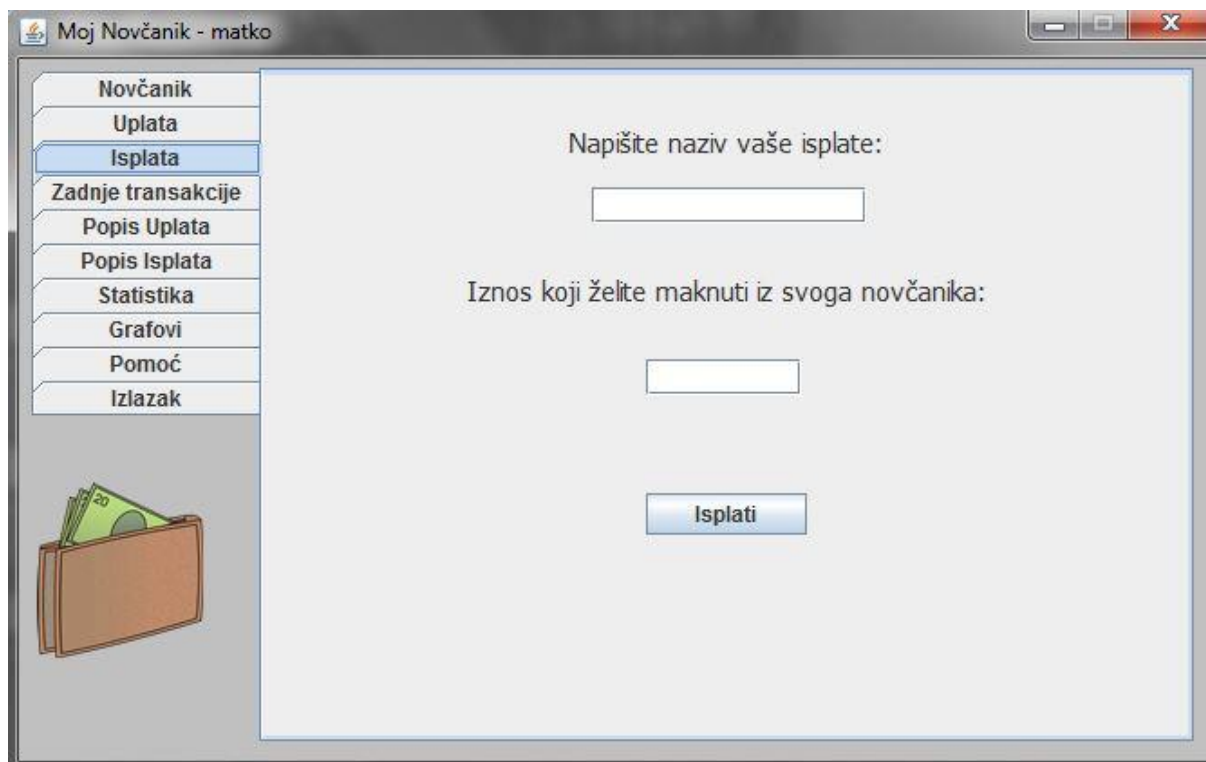
Slika 13 - Glavni prozor

Slijedeća kartica je „Uplata“ u kojoj korisnik može uplatiti željeni iznos i ime tog iznosa u svoj novčanik.



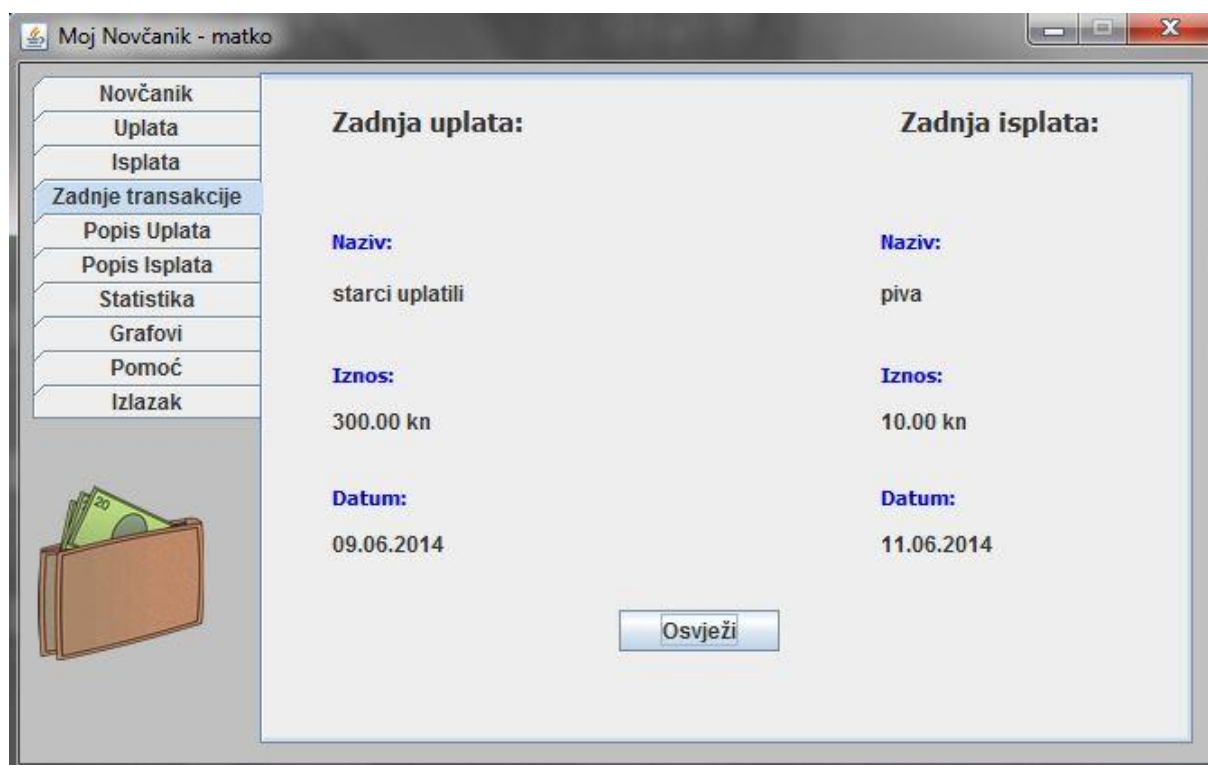
Slika 14 - Kartica Uplata

Kartica „Isplata“ je sličnog sadržaja kao i kartica „Uplata“ jedino što služi za unos isplate, odnosno korisnik oduzima taj iznos od ukupnog iznosa u svom novčaniku.



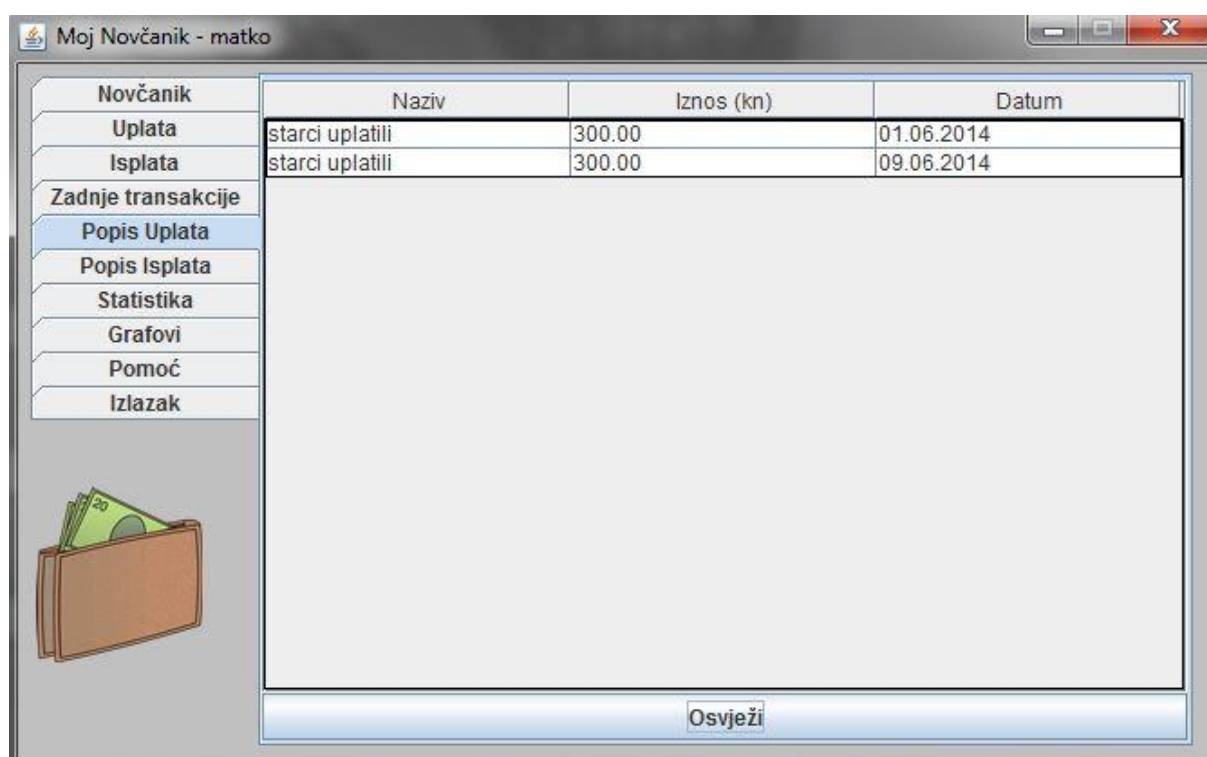
Slika 15 - Kartica Isplata

Kartica naziva „Zadnje transakcije“ sadrži informacije vezane uz zadnju uplatu i zadnju isplatu. Dobro je pritisnuti gumb „Osvježi“ ukoliko je u međuvremenu od pokretanja aplikacije bilo kakvih transakcija kako bi bili sigurni da su informacije ažurirane.



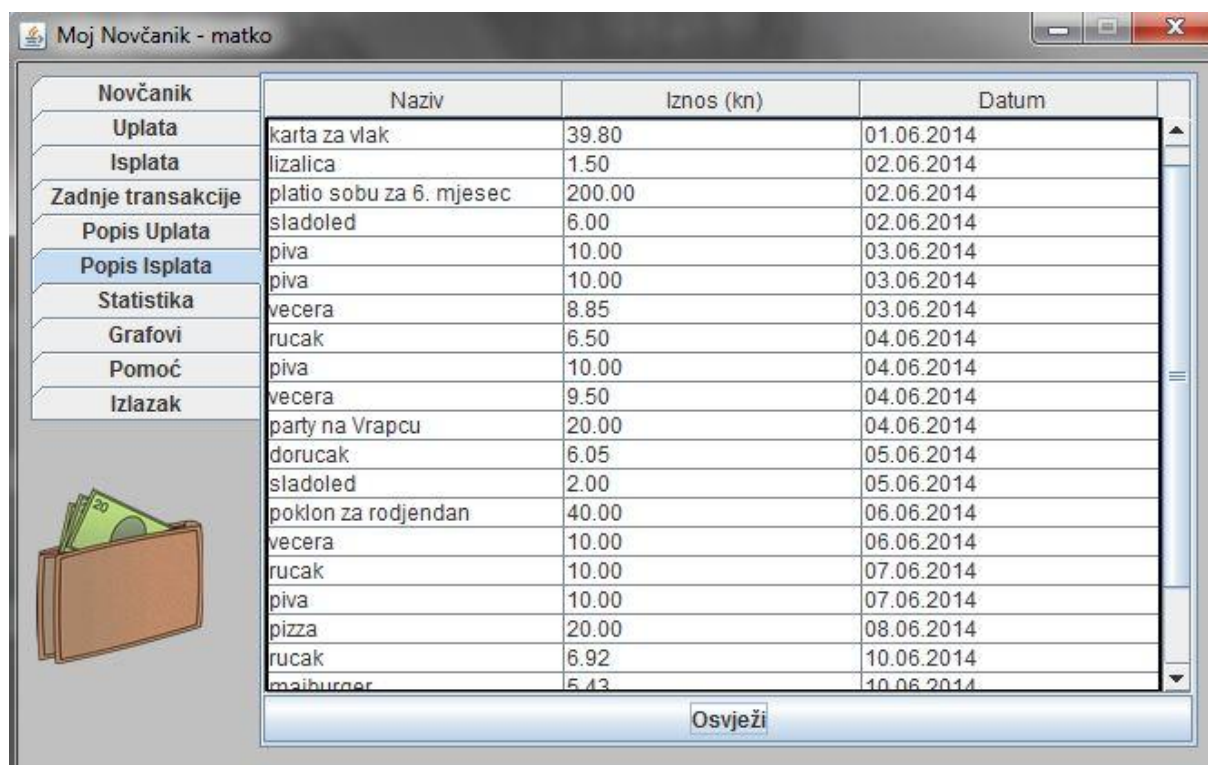
Slika 16 - Kartica Zadnje Transakcije

Kartica „Popis Uplata“ sadrži tablicu u kojoj su navedene sve uplate koje je korisnik imao do sada.



Slika 17 - Kartica Popis Uplata

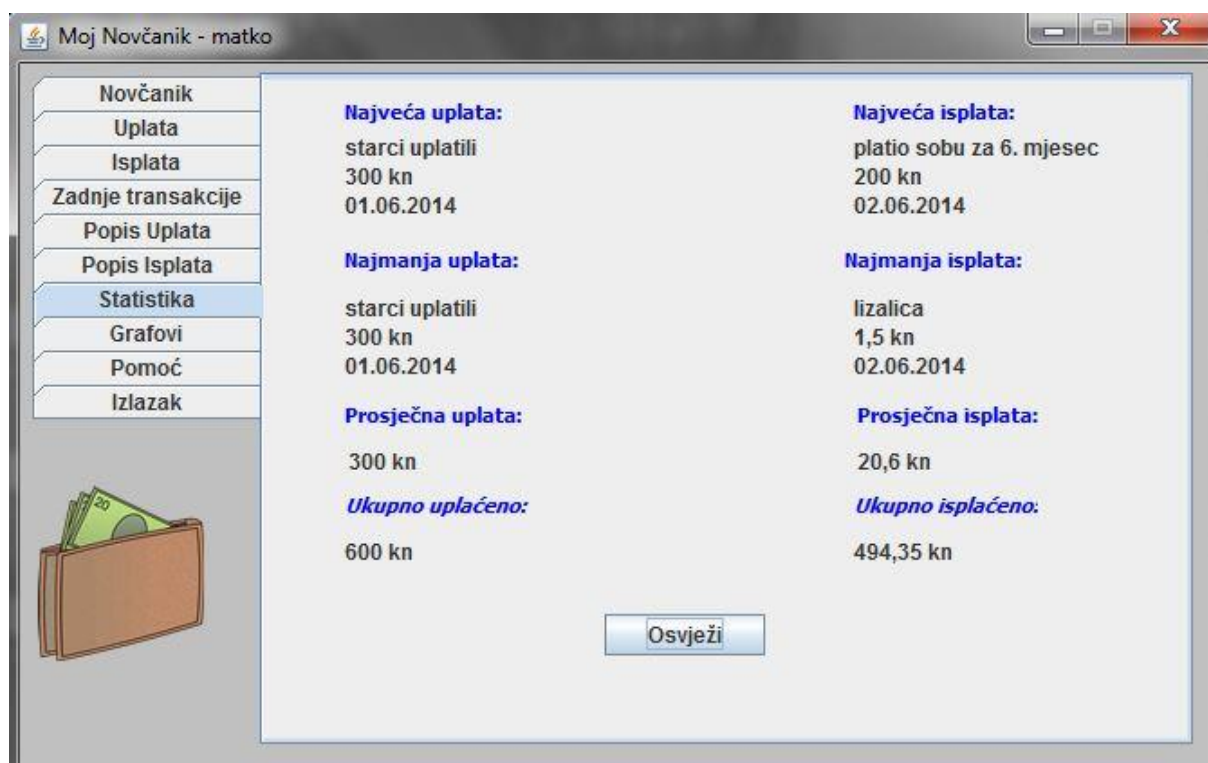
Kartica „Popis Isplata“ sadrži tablicu u kojoj su navedene sve isplate koje je korisnik imao do sada.



Naziv	Iznos (kn)	Datum
karta za vlak	39.80	01.06.2014
lizalica	1.50	02.06.2014
platio sobu za 6. mjesec	200.00	02.06.2014
sladoled	6.00	02.06.2014
piva	10.00	03.06.2014
piva	10.00	03.06.2014
vecera	8.85	03.06.2014
rucak	6.50	04.06.2014
piva	10.00	04.06.2014
vecera	9.50	04.06.2014
party na Vrapcu	20.00	04.06.2014
dorucak	6.05	05.06.2014
sladoled	2.00	05.06.2014
poklon za rođendan	40.00	06.06.2014
vecera	10.00	06.06.2014
rucak	10.00	07.06.2014
piva	10.00	07.06.2014
pizza	20.00	08.06.2014
rucak	6.92	10.06.2014
maihurger	5.43	10.06.2014

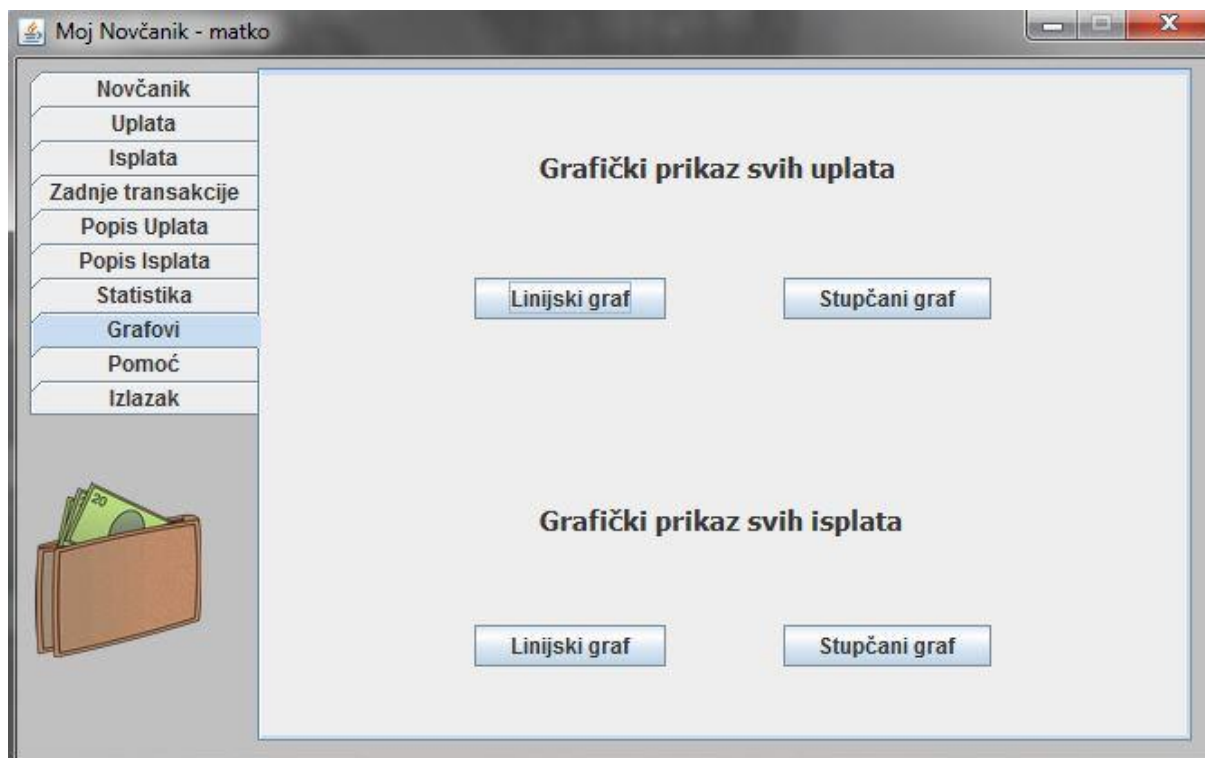
Slika 18 - Kartica Popis Isplata

Kartica „Statistika“ sadrži detaljne informacije o svim transakcijama koje je korisnik imao do sada. Možemo vidjeti koja nam je uplata bila najveća, koja najmanja, prosječnu uplatu i ukupno koliko smo uplatili u novčanik do sada. Također je to sve moguće vidjeti i za isplate.

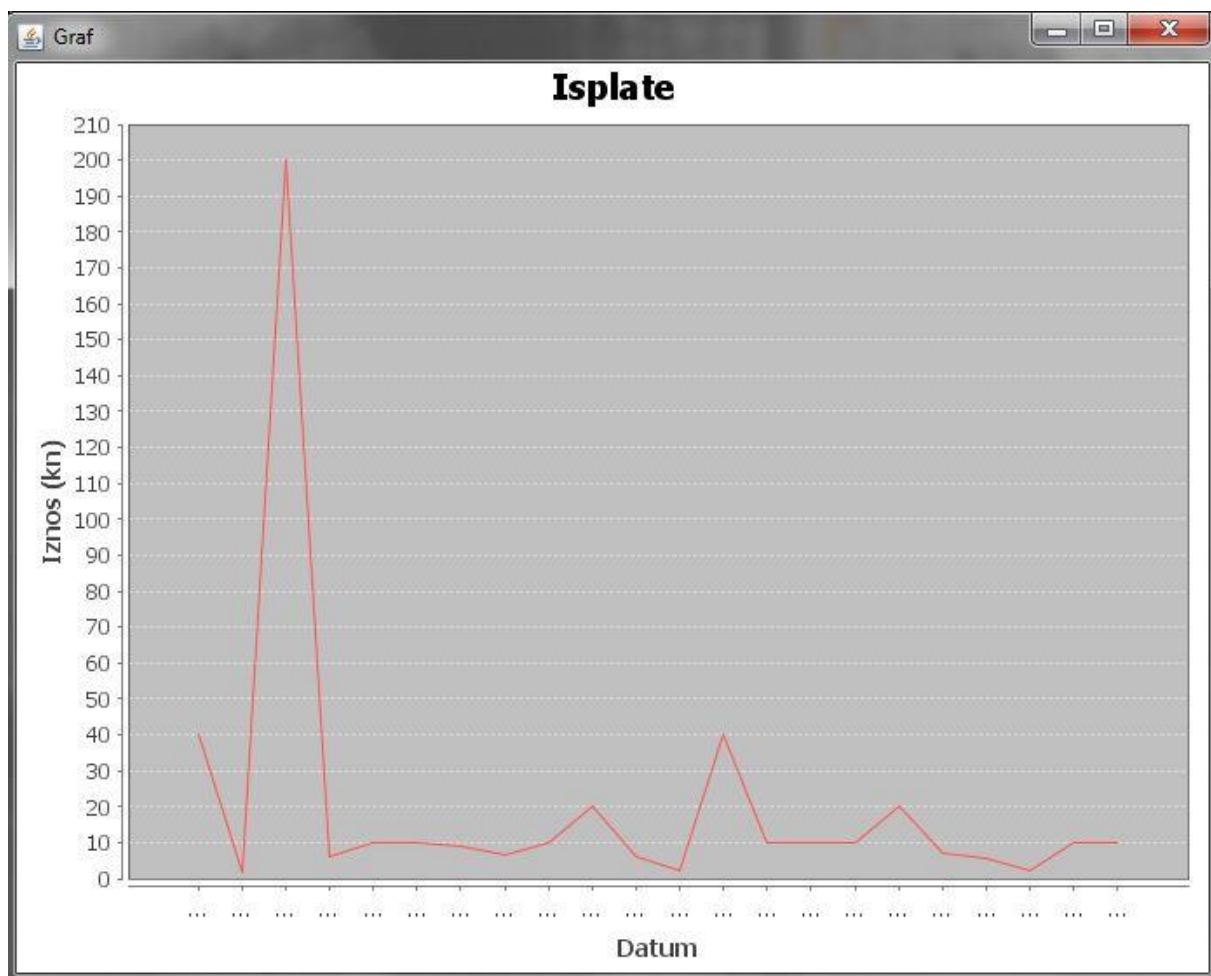


Slika 19 - Kartica Statistika

Kartica „Grafovi“ sadrži gumbe kojima korisnik bira kakav graf želi da mu prikaže uplate ili isplate koje je imao do sada.

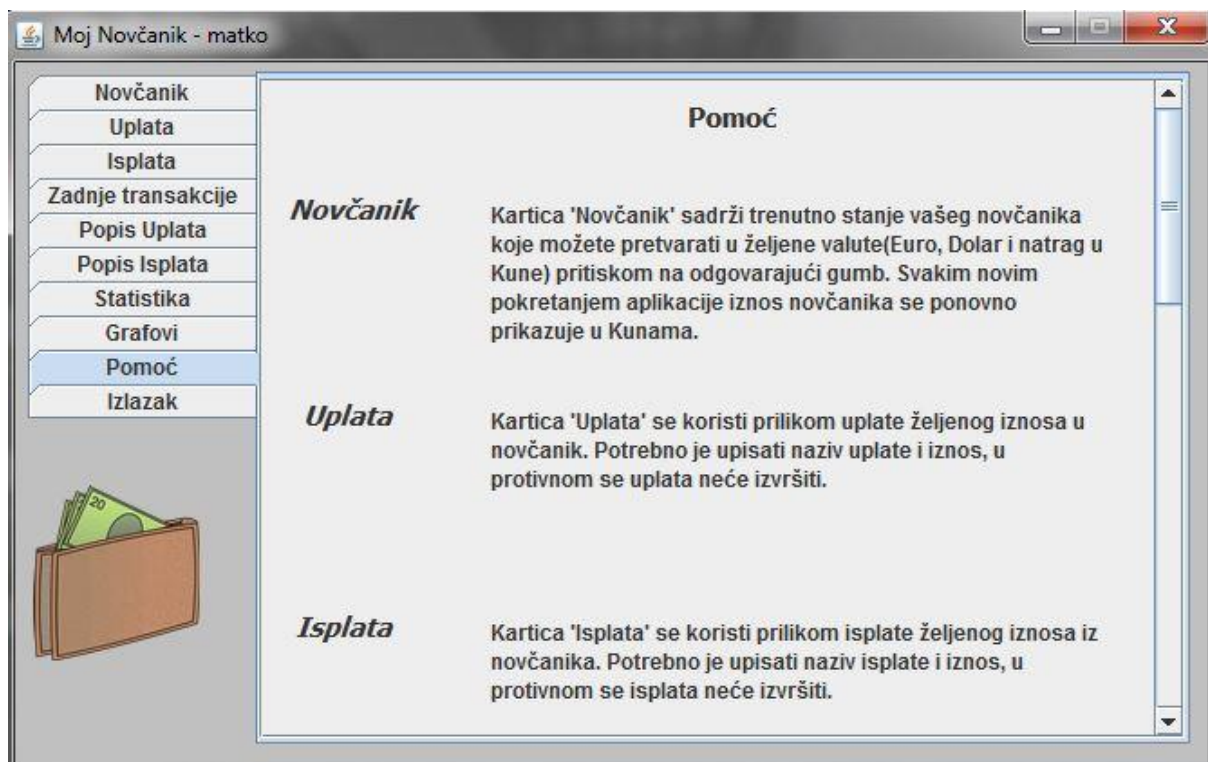


Slika 20 - Kartica Grafovi



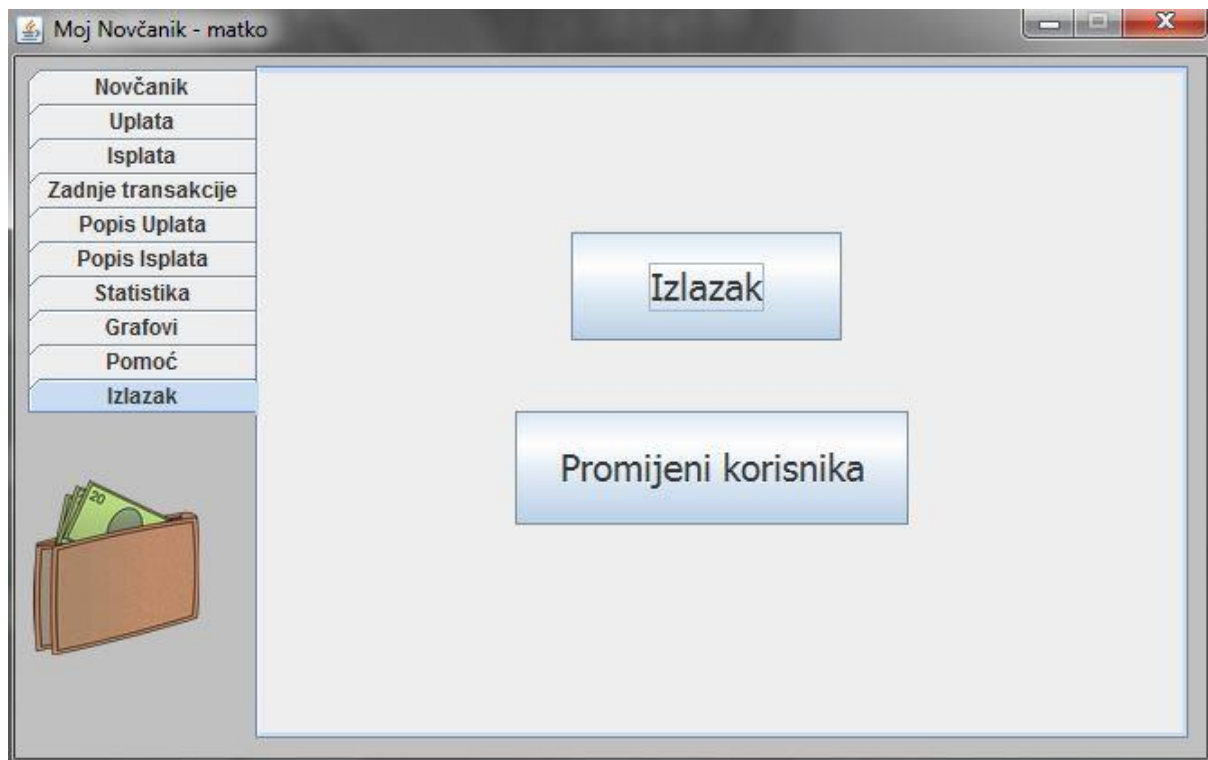
Slika 21 - Linijski graf koji prikazuje Isplate[2]

Kartica „Pomoć“ sadrži sve informacije gdje se što nalazi na kojoj kartici i kako se ta kartica koristi.



Slika 22 - Kartica Pomoć

Kartica „Izlazak“ sadrži gumb za izlazak iz aplikacije i gumb kojim se može odjaviti te ponovno prijaviti kao neki drugi korisnik.



Slika 23 - Kartica Izlazak

4 ZAKLJUČAK

Izrađena aplikacija je vrlo jednostavna, lagana za koristiti a opet dovoljno detaljna i korisna. Savršena je za nekoga tko želi voditi računa o svojim financijama. Savršena je za studente kako bi lakše pratili svoje novce kojima raspolažu i kako bi vidjeli na što najviše troše novce i koliko često. Aplikacija „Moj Novčanik“ je potpuno funkcionalna i uporabljiva.

Moguće je poboljšati aplikaciju i time možda povećati broj korisnika i time ju popularizirati. Aplikaciju bi moglo obogatiti još nekim dodatnim mogućnostima kao što su personalizacija sučelja, izgled prozora i slično. Mogle bi se dodati nove opcije kao što su timeri koji bi odbrojavali vrijeme do uplate plaće koje bi korisnik zadao datumski. Prvobitno bi bilo korisnije i praktičnije aplikaciju „Moj Novčanik“ napraviti kao android verziju te bi tada korisnik mogao na licu mjesta upisati isplate kako ih obavlja dok se još sjeća što je potrošio na što jer postoje velike šanse da do povratka doma za svoje računalo može zaboraviti podatke o transakcijama koje je imao tokom dana.

Matko Horvat

5 LITERATURA

[1] Good Hash Function for Strings [Mrežno] Available:
<http://stackoverflow.com/questions/2624192/good-hash-function-for-strings>. [Pokušaj pristupa 28.5.2014.]

[2] JFreeChart: Line Chart Demo. [Mrežno]. Available:
<http://www.java2s.com/Code/Java/Chart/JFreeChartLineChartDemo6.htm>. [Pokušaj pristupa 8.6.2014.]

[3] How to connect with MySQL database using Java.[Mrežno]. Available:
<http://mrbool.com/how-to-connect-with-mysql-database-using-java/25440>. [Pokušaj pristupa 1.6.2014.]