

PANEVROPSKI UNIVERZITET APEIRON
FAKULTET INFORMACIONIH TEHNOLOGIJA
INŽENJERING INFORMACIONIH TEHNOLOGIJA
Programiranje i softversko inženjerstvo
Banja Luka

Seminarski rad - projektni zadatak iz predmeta
OBJEKTNO-ORJENTISANO PROGRAMIRANJE
(sa primjenom na programskom jeziku Java)

Naslov rada:
Elektronska registar kasa -
aplikativno rješenje u Java programskom jeziku

Mentor:
prof. dr Saša Salapura

Student:
Goran Sičanović 197-22/FIT

Banja Luka, novembar, 2022. godine

SADRŽAJ

1. UVOD	3
2. PRIMJENJENE TEHNOLOGIJE	3
3. KONCEPT APLIKACIJE "Ca\$h"	3
3.1. PROGRAMSKI SEGMENT	4
3.2. KORISNIČKE FORME.....	6
4. "Ca\$h" APLIKACIJA - ZAKLJČAK.....	8
5. LITERATURA.....	8

1. UVOD

Polazeći od zahtjeva prirode posla "Registracija prometa roba i usluga", aplikacijom ćemo omogućiti obavljanje osnovnih aktivnosti koje podrazumijevaju ovaj vid posla: formiranje i ažuriranje kataloga roba/usluga kao i dodatnih kataloga koji su u relacijama s njima. Zatim, evidentiranje obavljenog prometa i izdavanje/formiranje fiskalnog (ili po potrebi drugog oblika) računa kao i Izvještaji po različitim kriterijumima.

Imajući u vidu dinamičnost u zakonodavnoj regulativi, potrebnim novčanim ulaganjima u materijalno-tehnička sredstva, različita uvjerenja i licence težimo da razvijemo "višenamjensku i proširivu" aplikaciju koja se uz manje korekcije može prilagoditi različitim zahtjevima.

2. PRIMJENJENE TEHNOLOGIJE

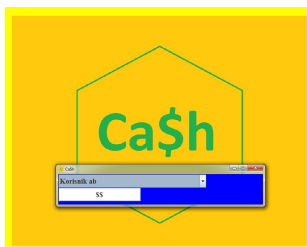
Uzimajući u obzir definisane zahtjeve iz uvoda, koristimo Java platformu za razvoj i dizajn aplikacije, odnosno programski dio pravila i logike aplikacije implementiramo u Java programskom jeziku.

Zbog svoje multiplatformske osobine (relativno jednostavna prenosivost java-aplikacija na različite operativne sisteme), velikim izborom klasa za rad sa grafičkim interfejsom kao činjenice da se Java razvojno okruženje ne naplaćuje u odnosu na druge razvojne platforme, Java platforma će biti adekvatna za izradu aplikacije kojom ćemo obuhvatiti definisane zahtjeve. Aplikativno rješenje u daljem radu predstavljamo imenom "Ca\$h"

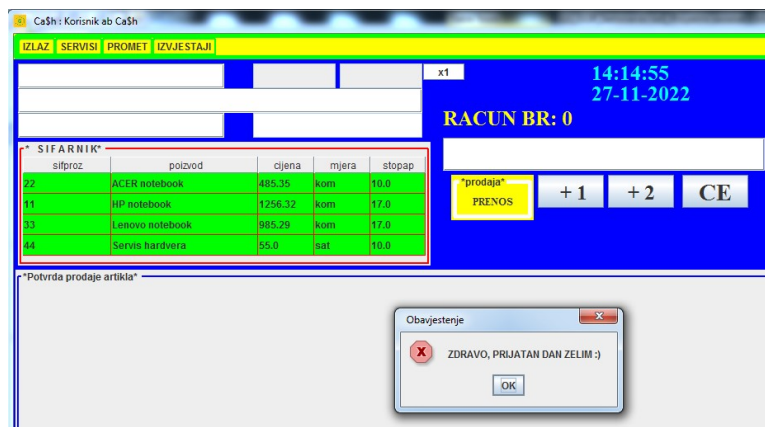
Radi čuvanja obrađenih podataka korist ćemo sistem relacione baze podataka, u ovom slučaju to je MS ACCESS.

3. KONCEPT APLIKACIJE "Ca\$h"

Rad "Ca\$h"-om počinje prijavom korisnika u aplikaciju unošenjem svojih pristupnih podataka, gdje nakon uspješne prijave korisnik dobija pozdravni odziv. Nakon ovog korisniku se omogućuje izbor da počne registrovanje prometa, radi sa katalogima ili pretražuje evidentirane podatke putem formi izvještaja.



sl. 1. Forma za prijavu



sl. 2. Radna korisnička forma

3.1. PROGRAMSKI SEGMENT

Klasa "pr" sadrži "main" metodu koja generiše korisničku formu, interfejs za autentifikaciju korisnika. Java kôd kojim kreiramo formu gdje odmah učitavamo i drajver za upravljanje bazom podataka. Koristimo *open source* Java JDBC drajver "UcanaccessDriver" koji će omogućiti povezivanje na Microsoft Access® desktop bazu podataka (link za preuzimanje naveden u Literatura, a nakon preuzimanja potrebno je raspakovati preuzet sadržaj u "ext" folder java virtuelne mašine: C:\Program Files\Java\jre1.8.0_101\lib\ext).

```
public static void main (String args [] ){
    try{
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");    } /* java drajver, nekoristi
se ODBC */
    catch( ClassNotFoundException cnfe ){ System.out.println(cnfe)    ;
    } // kraj catch
        pr epr = new pr("Ca$h"    );
    } //kraj main
```

Java kôd za autentifikaciju korisnika i kreiranje zapisa, sa podacima o pristupu aplikaciji, i upis u datoteku "osxacs.htn"

```
lka = new File ("c:/kasaStr/osxacs.htn");
pslog = new FileWriter(lka,true);

krs = new link();
kbxs = new JComboBox(krs.uzmp (new String ( "SELECT ime FROM kori " )));

if ( 10 == sl.getKeyCode()){
    String valid = new String (krs.getus(String.valueOf(kbxs.getSelectedItem()) )) ;
    if ( valid.equals( new String(pas.getPassword() ) ) ){
        pslog.write('\n');
        pslog.write(">>OPND AT TM:"+tak.toString()+" DT
"+dak.toString()+"<US>"+String.valueOf(kbxs.getSelectedItem())+"<>"+ new
String(pas.getPassword() )+"<<"    );
        pslog.flush();
        pslog.close();

        rac racuni = new rac(" Ca$h : "+ String.valueOf(kbxs.getSelectedItem())+" Ca$h");
        racuni.kaser = String.valueOf(kbxs.getSelectedItem());
        String asd = System.setProperty("kaser", String.valueOf(kbxs.getSelectedItem()));
        wi.dispose();
        dispose();
    } //

    else{ //ako autentifikacija ne prolazi
        for (int j = 0;j <= 2 ; j++){
            System.out.println('\7'); }
        JOptionPane.showMessageDialog(null, "NEVAZECA PRISTUPNA SIFRA ", " IZLAZ",0);
        wi.dispose();
        dispose();
        System.exit(0);
    }
}
```

Odabir artikala je omogućen iz objekta "tabela" klase "stamp". Klasa "stamp" nesleđuje klasu JTable (iz javax.swing paketa) i implementira interfejs Printable (ovaj interfejs je koristan ako bismo htjeli omogućiti štampanje kataloga artikala).

```
public class stamp extends JTable implements Printable {

stamp(){

    }
```

```

public void odstampaj () {
    Paper papir = new Paper();
    papir.setImageableArea(0.5,0.5,623.622,2000 );
    PrinterJob priob = PrinterJob.getPrinterJob();

    PageFormat fmt = new PageFormat();
        fmt.setOrientation(1);
        fmt.setPaper(papir);
    priob.setPrintable(this,fmt);
        try {

            if (priob.printDialog())

                priob.print();
        }
        catch(PrinterException pe) {
            System.out.println("Error printing: " + pe);
        }

        this.setShowHorizontalLines(true) ;
        this.setShowVerticalLines(true) ;

    } //kraj metode odstampaj

```

Objekat "tabela" popunjavamo podacima iz baze podataka pomoću instrukcije:

```

tabela.setModel(new mdl ("SELECT rbs ,sifproz ,proiz,prodkol ,cijena FROM temst ORDER BY
rbs ASC"));

```

Pozivamo metod `setModel()` koji je naslijeđen iz `JTable` klase, i koji za argument traži interfejs `TableModel`. Prosleđujemo objekat "mdl" nastao od istoimene klase tj. "mdl" koja nasleđuje klasu `DefaultTableModel` (a `DefaultTableModel` ima već sistemski implementiran interfejs `TableModel`).

Java kôd konstruktora klase "mdl":

```

mdl (String esqkom){ //za prikaz modela

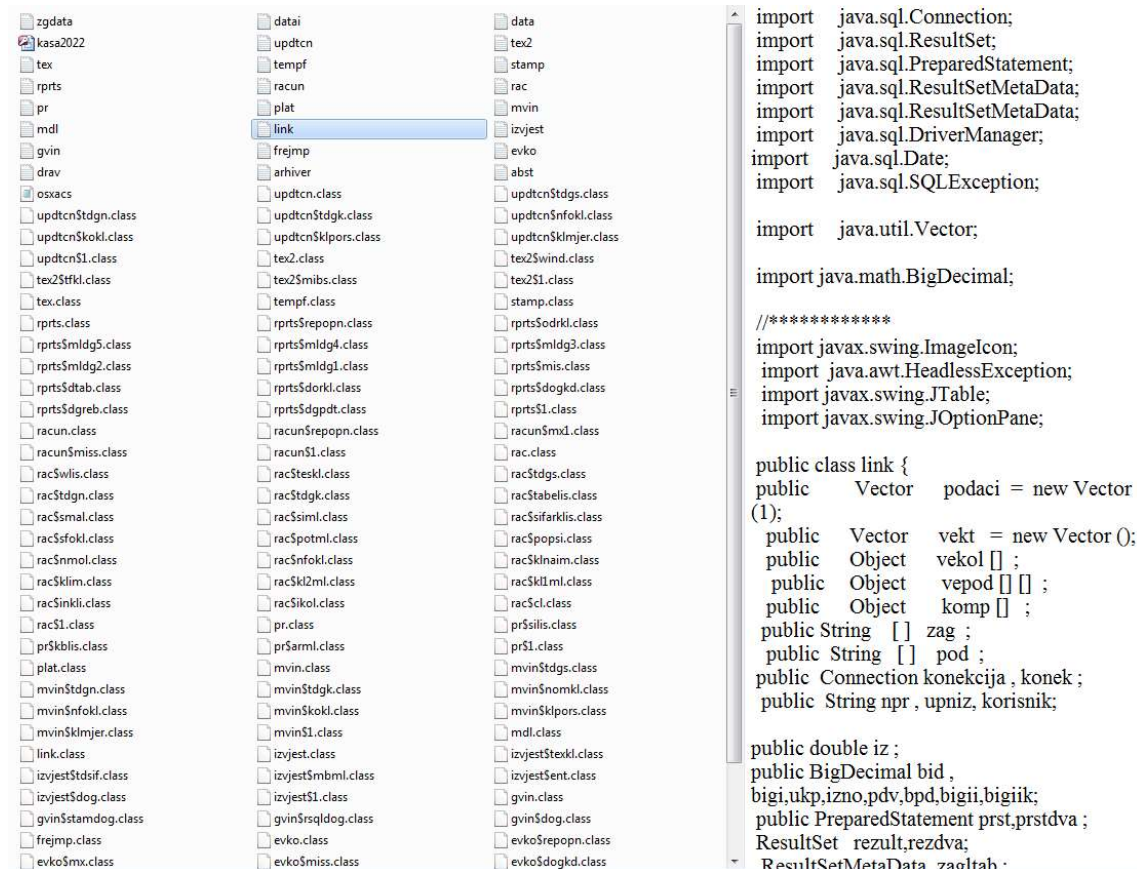
    String [] identkol = {"RB","SIFRA","NAZIV","KOLICINA","CIJENA","IZNOS"};
    setColumnIdentifiers(identkol);
    try {
        konekcija = DriverManager.getConnection
("jdbc:ucanaccess://C:/kasaStr/kasa2022.accdb");
        prst = konekcija.prepareStatement ( esqkom );
        rezultat = prst.executeQuery();
        zagltab = rezultat.getMetaData ();
        int brk = zagltab.getColumnCount();
        while ( rezultat.next() ) {
            Object [] podt = new Object [brk + 1 ] ;
            podt [0] = rezultat.getString("rbs");
            podt [1] = rezultat.getString ("sifproz");
            podt [2] = rezultat.getString("proiz");
            BigDecimal bdco = new BigDecimal (rezultat.getDouble("prodkol") );
            podt [3] = bdco.setScale(3,4) ;
            BigDecimal cebd = new BigDecimal (rezultat.getDouble("cijena"));
            podt [4] = cebd.setScale(3,4) ;
            BigDecimal bgd = new BigDecimal (rezultat.getDouble("prodkol") *
rezultat.getDouble("cijena"));
            podt [5] = bgd.setScale(2,4) ;
           .addRow(podt);
        } // k while
        rezultat.close();
    } // try

```

```

catch ( SQLException sq ) {
} // kr catch
} //k konst

```



sl. 3. Prikaz svih Java klasa projekta

Pošto smo sve klase preveli u java-bajtkood, sada, radi kompaktnosti i lakše portabilnosti aplikacije prema korisniku, koristimo komandu "jar" iz konzole "command prompt" (jar cf *jar-file input-file(s)*) :

C:\> jar cfm Cash.jar MA.mf *.class

cfm -> parametri: c - da se radi o kreiranju paketa (kompresovane datoteke) ; f - zahtjev da se kreira datoteka ;

m -> zahtjev da se uključi naša manifest datoteka koja će java virtuelnoj mašini "govoriti" koja je to glavna klasa tj. ona klasa koja sadrži "main" funkciju

Cash.jar -> naziv naše aplikacije

MA.mf -> naša manifest datoteka koja indikatorom "Main-Class:" pokazuje na glavnu klasu (Main-Class: pr)

*.class -> zahtjev gdje tražimo da jar datoteka sadrži samo klase **Java bytecode** koje će JVM interpretirati i izvršavati

3.2. KORISNIČKE FORME

Korisničke forme kojim se obavlja interakcija korisnika sa aplikacijom su:

IZLAZ

1

n

sifmjere

mjera

idstopa

stopap

1

kom

1

17.0

2

Kg

2

10.0

3

pak

3

0.0

4

sat

sifproiz

poizvod

cijena

mjera

22

ACER notebook

485.35

kom

11

HP notebook

1256.32

kom

33

Lenovo notebook

985.29

kom

44

Servis hardvera

55.0

sat

Stampaj PROMET

27-11-2022

27-11-2022

UKUPNO PROIZVODI (KM)

UKUPNO USLUGA (KM)

UKUPNO PDV 17% PROIZVODI (KM)

UKUPNO PDV 17% USLUGE (KM)

sl. 4. Unos i izmjena podataka o artiklima

UPIS NOVOG KASERA

KORISNIK IME I PREZIME

PRISTUPNA SIFRA

sl.5. Pregled prometa

sl.6. Dodavanje novog korisnika

IZVJEŠTAJI

27-11-2022

27-11-2022

Promet kasera OD-DO

Ukupne količine OD-DO F5

Artikal promet OD-DOjenter

Korisnik ab

Korisnik ab

Korisnik CD

Korisnik Ef

stam.PROMET KASER

sl.7. Izvještaji

4. "Ca\$h" APLIKACIJA - ZAKLJČAK

Aplikacija "Ca\$h" je kompletno kreirana preko Java platforme, odnosno sve funkcionalnosti aplikacije su realizovane bez potrebe za korištenjem drugih tehnologija. Stoga, aplikaciju možemo koristiti na svim platformama koje mogu da pokreću Java virtualnu mašinu. Izuzetak od ovog bi bila baza podataka koja bi tražila izvjesna prilagođavanja. Aplikacija se može nadograditi novim funkcionalnostima zahvaljujući Java konceptu nasljeđivanja te tako dobiti novu upotrebnu vrijednost.

5. LITERATURA

1. <https://salapura.com/java/> (22.11.2022.)
2. <https://docs.oracle.com/en/java/index.html> (22.11.2022.)
3. <https://ucanaccess.sourceforge.net/site.html> (22.11.2022.) "*(Home page of UCanAccess, an open-source Java JDBC driver implementation that allows Java developers and JDBC client programs (e.g., DBeaver, NetBeans, SQLeo, OpenOffice Base, LibreOffice Base, Squirrel SQL) to read/write Microsoft Access databases. Because it is a pure Java implementation it runs on both Windows and non-Windows operating systems (e.g., Linux/unix).)*"