

SEMINARSKI RAD

PANEVROPSKI UNIVERZITET APEIRON
FAKULTET INFORMACIONIH TEHNOLOGIJA

Redovne studije

Smijer „Programiranje i softversko inženjerstvo”

Predmet

Objektno orijentisano programiranje

„Restoran - Java projekt sa bazom podataka”

(seminarski rad)

Predmetni nastavnik

Prof.dr Saša Salapura

Student

Ognjen Vukić

Index br. 14-21-FITIIT1-PRO-240

SADRŽAJ

1. Uvod	1
2. Korišteni alati u projektu	2
2.1 Java programski jezik.....	2
2.2 Eclipse IDE za Javu.....	3
2.3 XAMPP i MySQL	4
3. Struktura, kod i izgled aplikacije	5
3.1 Struktura aplikacije	5
3.2 Admin meni.....	6
3.3 Ažuriraj hranu	9
3.4 Unesi hranu	12
3.5 Informacija o naplati	15
3.6 Hrana info.....	18
3.7 Izbriši hranu.....	20
3.8 Baza podataka “Restoran”	23
3.9 Konvertovanje Java file u .exe	26
4. Zaključak	27
5. Literatura	28

1. Uvod

Java je jedan od najpopularnijih programskih jezika na svijetu, poznat po svojoj prenosivosti, sigurnosti i robusnosti. Razvijen od strane kompanije Sun Microsystems (sada u vlasništvu Oracle Corporation), Java omogućava razvoj aplikacija koje mogu raditi na bilo kojoj platformi koja ima Java Virtual Machine (JVM). Zbog svoje sintakse slične C++ jeziku i bogate standardne biblioteke, Java je često prvi izbor za razvoj raznovrsnih aplikacija, uključujući web aplikacije, mobilne aplikacije, enterprise sisteme i desktop softver.

Projekat "Restoran" je sveobuhvatan sistem za upravljanje restoranom koji uključuje nekoliko ključnih funkcionalnosti kroz administrativni panel. Ovaj projekat je razvijen korišćenjem Java programskog jezika, XAMPP platforme (koja obuhvata Apache server, MySQL bazu podataka i PHP), i dodatnog softvera za konvertovanje Java fajlova u izvršne (.exe) fajlove.

Admin meni je srce sistema, gdje menadžeri restorana mogu pristupiti različitim opcijama za upravljanje resursima i informacijama o restoranu. Ažuriranje hrane omogućava menadžerima da ažuriraju informacije o postojećim jelima u ponudi. Menadžeri mogu menjati cijene, opise, i dostupnost jela. Unos nove hrane u ponudu restorana je jednostavan zahvaljujući ovoj funkcionalnosti. Korisnici mogu unijeti naziv jela, cijenu, sastojke i druge relevantne informacije. Informacija o naplati omogućava praćenje i upravljanje informacijama o naplati, što uključuje evidentiranje računa, praćenje uplata i generisanje finansijskih izveštaja. Brisanje hrane je funkcija koja omogućava da se određeno jelo trajno ukloni iz ponude ukoliko više nije dostupno ili se menadžer odluči da ga ukloni iz sistema. Informacije o hrani pružaju pregled svih jela koja su trenutno u ponudi restorana, uključujući detalje kao što su sastojci, cijena, i dostupnost.

2. Korišteni alati u projektu

2.1 Java programski jezik

Java je visoko nivoan, objektno-orijentisan programski jezik koji je razvijen od strane kompanije Sun Microsystems, sada u vlasništvu Oracle Corporation. Lansirana 1995. godine, Java je dizajnirana da bude prenosiva, sigurna i robusna, što je čini idealnim izborom za razvoj raznovrsnih aplikacija. Jedna od ključnih karakteristika Jave je njena prenosivost, koja se ostvaruje kroz Java Virtual Machine (JVM). JVM omogućava izvršavanje Java programa na bilo kojoj platformi bez potrebe za promjenama u kodu, što znači da programi napisani u Javi mogu raditi na Windows, Mac OS, Linux, i mnogim drugim operativnim sistemima.

Java podržava objektno-orijentisani pristup programiranju, što znači da se koncepti kao što su objekti, klase, nasleđivanje, polimorfizam i enkapsulacija koriste za organizaciju koda. Ovo olakšava razvoj i održavanje velikih i složenih sistema. Java ima ugrađene sigurnosne mehanizme koji štite od virusa i drugih zlonamernih programa. Sigurnosni model Jave uključuje proveru tipova u vreme kompajliranja i izvođenja, kao i sandbox model koji ograničava operacije koje program može izvršavati. Java je dizajnirana da bude otporna na greške. Ima strogu proveru grešaka tokom kompajliranja i runtime-a, što smanjuje šanse za pojavu grešaka u programima. Java takođe koristi automatsko upravljanje memorijom i garbage collection, što pomaže u sprečavanju curenja memorije.



Slika 1. Logo Java programskog jezika

2.2 Eclipse IDE za Javu

Eclipse IDE za Javu je jedan od najpopularnijih integrisanih razvojnih okruženja (IDE) koje koriste Java programeri širom svijeta. Razvijen od strane Eclipse Foundation, ovaj alat pruža sveobuhvatnu podršku za razvoj, testiranje i debugging Java aplikacija. Eclipse IDE je poznat po svojoj modularnosti i proširivosti, omogućavajući programerima da dodaju različite pluginove za dodatne funkcionalnosti.

Jedna od glavnih prednosti Eclipse-a je njegov moćan uređivač koda, koji nudi funkcionalnosti kao što su automatsko dovršavanje koda, refaktorisanje, i integrisanu pomoć pri pisanju koda. Takođe, Eclipse podržava različite verzije Jave i omogućava lako prebacivanje između njih. Integracija sa alatima za verzionisanje koda poput Git-a i SVN-a, kao i sa alatima za gradnju kao što su Maven i Gradle, čini Eclipse sveobuhvatnim alatom za razvoj softvera.

Eclipse takođe nudi vizualne alate za razvoj GUI aplikacija, kao što su WindowBuilder, što olakšava kreiranje korisničkih interfejsa. Sa svojom širokom zajednicom i bogatim ekosistemom pluginova, Eclipse IDE ostaje jedan od vodećih izbora za Java programere, omogućavajući im da efikasno razvijaju i održavaju svoje aplikacije.



Slika 2. Logo od Eclipse IDE za Javu

2.3 XAMPP i MySQL

XAMPP je besplatna i jednostavna distribucija koja omogućava lako postavljanje i upravljanje lokalnim web serverom. Sadrži Apache HTTP Server, MySQL bazu podataka, te interpretere za skriptne jezike PHP i Perl. XAMPP je izuzetno koristan za razvoj i testiranje web aplikacija na lokalnom računaru pre nego što se one objave na produkcijskom serveru. Njegova jednostavna instalacija i korisnički interfejs čine ga idealnim za programere koji žele brzo postaviti razvojno okruženje.

MySQL, uključen u XAMPP paket, je jedan od najpopularnijih sistema za upravljanje relacijskim bazama podataka. Poznat po svojoj brzini, pouzdanosti i lakoći korišćenja, MySQL je široko korišćen u raznim aplikacijama, od malih web stranica do velikih enterprise sistema. MySQL pruža robustne funkcionalnosti za upravljanje podacima, uključujući složene upite, transakcije i podršku za različite vrste podataka.

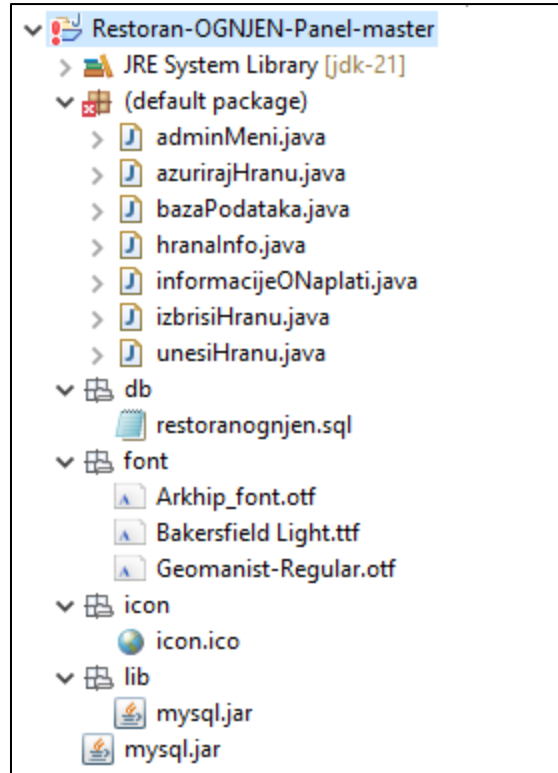
Kombinacija XAMPP-a i MySQL-a omogućava programerima da brzo postave i upravljaju lokalnim razvojnim okruženjem, što olakšava razvoj, testiranje i debugovanje web aplikacija. XAMPP-ov kontrolni panel pruža jednostavan način za pokretanje i zaustavljanje servera, kao i pristup alatima poput phpMyAdmin-a za administraciju MySQL baza podataka. Zajedno, XAMPP i MySQL čine moćan alat za svakog web programera koji želi efikasno raditi na razvoju i testiranju svojih aplikacija.



Slika 3. XAMPP i MySQL logo

3. Struktura, kod i izgled aplikacije

3.1 Struktura aplikacije



Slika 4. Struktura aplikacije

a) Admin meni

Administrativni meni omogućava menadžerima pristup raznim funkcionalnostima sistema, olakšavajući upravljanje restoranom i njegovim resursima.

b) Unesi hranu

Korisnici mogu dodavati nova jela u ponudu restorana. Ova funkcionalnost omogućava unos naziva jela, cijene, sastojaka i drugih relevantnih informacija.

c) Ažuriraj hranu

Menadžeri mogu menjati informacije o postojećim jelima, kao što su cijena, opis i dostupnost. Ova funkcionalnost pomaže u održavanju ažurnih podataka o jelima u ponudi.

d) Izbriši hranu

Funkcionalnost koja omogućava uklanjanje jela iz sistema kada više nisu dostupna ili kada menadžer odluči da ih ukloni iz ponude.

e) Hrana Info

Pruža pregled svih jela u ponudi restorana, uključujući detalje kao što su sastojci, cijena i dostupnost. Ova funkcionalnost omogućava korisnicima da lako pretražuju i pregledaju informacije o jelima.

f) Informacija o naplati

Ova opcija omogućava praćenje i upravljanje transakcijama, osiguravajući precizne finansijske izveštaje. Uključuje evidentiranje računa i praćenje uplata.

g) Baza podataka

Aplikacija koristi bazu podataka za skladištenje svih informacija o jelima, transakcijama, osiguravajući da su svi podaci bezbedni i lako dostupni za pretragu i izmjenu.

3.2 Admin meni



Slika 5. Admin meni


```

import java.awt.*;
import java.awt.event.*;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import java.awt.Color;

public class adminMeni {
    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;

    public adminMeni(){
        prepareGUI();
    }
    public static void main(String[] args){
        adminMeni swingControlDemo = new adminMeni();
        swingControlDemo.showButtonDemo();
    }
    private void prepareGUI(){
        mainFrame = new JFrame("RESTORAN Ognjen Panel");
        mainFrame.setBounds(100,100,700,400);
        mainFrame.setLayout(new GridLayout(3,1));
        mainFrame.getContentPane().setBackground(Color.orange);

        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                System.exit(0);
            }
        });
        headerLabel = new JLabel("", JLabel.CENTER);
        statusLabel = new JLabel("",JLabel.CENTER);
        statusLabel.setSize(350,300);
        controlPanel = new JPanel();
        controlPanel.setLayout(new GridLayout(1,5));

        mainFrame.add(headerLabel);
        mainFrame.add(controlPanel);
        mainFrame.add(statusLabel);
        mainFrame.setVisible(true);

    }
    public void showButtonDemo(){

        headerLabel.setText("RESTORAN Ognjen");
        this.headerLabel.setFont(new Font(null, Font.BOLD, 27));
        headerLabel.setForeground(Color.white);

        JButton fkButton = new JButton("Hrana Info");
        JButton billButton = new JButton("Naplata Info");
    }

```

```

        JButton afButton = new JButton("Unesi Hranu");
        JButton ufButton = new JButton("Ažuriraj Hranu");
        //izbrisiHranu dugme
        JButton dlButton = new JButton("Izbriši Hranu");

        fkButton.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e) {

                hranaInfo ii=new hranaInfo();
                try {
                    ii.showButtonDemo();
                } catch (SQLException ex) {
                    Logger.getLogger(adminMeni.class.getName()).log(Level.SEVERE,
null, ex);
                }
            }
        });

        billButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                informacijeONaplati gb=new informacijeONaplati();}
        });

        afButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                unesiHranu ef=new unesiHranu();
                ef.showButtonDemo();
            }
        });

        ufButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                azurirajHranu uf=new azurirajHranu();
                uf.showButtonDemo();

            }
        });

        //izbrisiHranu dugme
        dlButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                izbrisiHranu dl=new izbrisiHranu();
                dl.showButtonDemo();
            }
        });

        controlPanel.add(ufButton);
        controlPanel.add(afButton);
        controlPanel.add(billButton);

```

```

        controlPanel.add(fkButton);
        controlPanel.add(dlButton);

        mainFrame.setVisible(true);
        mainFrame.setLocationRelativeTo(null);
    }
}

```

3.3 Ažuriraj hranu

Slika 6. Ažuriraj hranu

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class azurirajHranu {

    private JFrame mainFrame;

```

```

private JLabel headerLabel;
private JLabel statusLabel;
private JPanel controlPanel;
private JLabel id,name,price,quantity;
private static int count = 0;
GridLayout experimentLayout = new GridLayout(0,2);
ResultSet rs;

azurirajHranu(){

    prepareGUI();
}

public static void main(String[] args){
    azurirajHranu swingControlDemo = new azurirajHranu();
    swingControlDemo.showButtonDemo();
}

private void prepareGUI(){
    mainFrame = new JFrame("Ažuriraj!");
    mainFrame.setSize(700,400);
    mainFrame.setLayout(new GridLayout(3, 1));
    mainFrame.getContentPane().setBackground(Color.lightGray);
    mainFrame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent windowEvent){
            mainFrame.setVisible(false);
        }
    });
    headerLabel = new JLabel("", JLabel.CENTER);
    statusLabel = new JLabel("",JLabel.CENTER);

    statusLabel.setSize(350,400);

    controlPanel = new JPanel();
    controlPanel.setLayout(new FlowLayout());

    mainFrame.add(headerLabel);
    mainFrame.add(controlPanel);
    mainFrame.add(statusLabel);
    mainFrame.setVisible(true);
}

public void showButtonDemo(){

    headerLabel.setText("RESTORAN Ognjen");
    headerLabel.setFont(new Font(null, Font.BOLD, 27));

    name = new JLabel("Unesi Ime");
    JTextField tf2=new JTextField();
    tf2.setSize(100,30);

    price = new JLabel("Unesi Vrijednost");
    JTextField tf3=new JTextField();

```

```

        tf3.setSize(100,30);

        quantity = new JLabel("Unesi Količinu");
        JTextField tf4=new JTextField();
        tf4.setSize(100,30);

        JButton okButton = new JButton("AŽURIRAJ");

        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                PreparedStatement pst;
                bazaPodataka con = new bazaPodataka();
                try{
                    pst = con.mkDataBase().prepareStatement("UPDATE
restoranognjen.hrana SET f_kolicina= ?, f_vrijednost=? where f_ime = ?");
                    pst.setString(3, tf2.getText());
                    pst.setDouble(2, Double.parseDouble(tf3.getText()));
                    pst.setInt(1, Integer.parseInt(tf4.getText()));
                    pst.execute();

                    JOptionPane.showMessageDialog(null, "Ažuriranje Uspješno " +
tf2.getText());
                    mainFrame.setVisible(false);

                }catch(Exception ex){
                    System.out.println(ex);
                    System.out.println("Error");
                    JOptionPane.showMessageDialog(null, "Unošenje pogrešno : " +
tf2.getText());
                }finally{
                }
            }
        });

        JPanel jp = new JPanel();
        jp.add(name);
        jp.add(tf2);
        jp.add(price);
        jp.add(tf3);
        jp.add(quantity);
        jp.add(tf4);

        jp.setSize(200,200);
        jp.setLayout(experimentLayout);
        controlPanel.add(jp);
        jp.add(okButton);

        mainFrame.setVisible(true);

        mainFrame.setLocationRelativeTo(null);
    }
}

```

3.4 Unesi hranu



The screenshot shows a Java Swing window titled "Unesi novu hranu!". The window has a dark gray header bar with the text "RESTORAN Ognjen" in white. Below the header, there is a light gray panel containing three input fields and an "OK" button. The input fields are labeled "Unesi Ime", "Unesi Vrijednost", and "Unesi Količinu". The "OK" button is a blue button with white text.

Slika 7. Unesi hranu

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class unesiHranu {

    private JFrame mainFrame;
    private JLabel headerLabel;
    private JPanel controlPanel;
    private JLabel id,name,price,quantity;
    private static int count = 0;
    GridLayout experimentLayout = new GridLayout(0,2);
    ResultSet rs;

    unesiHranu(){

        prepareGUI();
    }
}
```

```

public static void main(String[] args){
    unesiHranu swingControlDemo = new unesiHranu();
    swingControlDemo.showButtonDemo();
}

private void prepareGUI(){
    mainFrame = new JFrame("Unesi novu hranu!");
    mainFrame.setSize(700,400);
    mainFrame.setLayout(new GridLayout(3, 1));

    mainFrame.getContentPane().setBackground(Color.gray);

    mainFrame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent windowEvent){
            mainFrame.setVisible(false);
        }
    });
    headerLabel = new JLabel("", JLabel.CENTER);

    controlPanel = new JPanel();
    controlPanel.setLayout(new FlowLayout());

    mainFrame.add(headerLabel);
    mainFrame.add(controlPanel);
    mainFrame.setVisible(true);
}

public void showButtonDemo(){

    headerLabel.setText("RESTORAN Ognjen");
    headerLabel.setFont(new Font(null, Font.BOLD, 27));

    name = new JLabel("Unesi Ime");
    JTextField tf2=new JTextField();
    tf2.setSize(100,40);

    price = new JLabel("Unesi Vrijednost");
    JTextField tf3=new JTextField();
    tf3.setSize(100,40);

    quantity = new JLabel("Unesi Količinu");
    JTextField tf4=new JTextField();
    tf4.setSize(100,40);

    JButton okButton = new JButton("OK");

    okButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            PreparedStatement pst;
            bazaPodataka con = new bazaPodataka();
            try{

```

```

        pst = con.mkDataBase().prepareStatement("insert into
restoranognjen.hrana(f_ime,f_vrijednost,f_kolicina) values (?, ?, ?)");
        pst.setString(1, tf2.getText());
        pst.setDouble(2, Double.parseDouble(tf3.getText()));
        pst.setInt(3, Integer.parseInt(tf4.getText()));
        pst.execute();

        JOptionPane.showMessageDialog(null, "Gotovo unošenje " +
tf2.getText());
        mainFrame.setVisible(false);

        }catch(Exception ex){
            System.out.println(ex);
            System.out.println("EEEE");
            JOptionPane.showMessageDialog(null, "Unošenje pogrešno : " +
tf2.getText());
        }finally{
        }
    }
});

JPanel jp = new JPanel(null);
jp.add(name);
jp.add(tf2);
jp.add(price);
jp.add(tf3);
jp.add(quantity);
jp.add(tf4);

jp.setSize(500, 500);
jp.setLayout(experimentLayout);
controlPanel.add(jp);
jp.add(okButton);

        mainFrame.setLocationRelativeTo(null);
        mainFrame.setVisible(true);
    }
}

```


3.5 Informacija o naplati

Ime Hrane	Količina	Vrijednost

Slika 8. Informacija o naplati

```
import static com.sun.java.accessibility.util.AWTEventMonitor.addActionListener;
import java.awt.*;
import java.awt.event.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class informacijeONaplati extends JFrame{

    JTextField food,quantity;
    String[] columnNames = {"Ime Hrane",
                           "Količina",
                           "Vrijednost"
    };

    JTable cart;
```

```

JLabel totalP = new JLabel();
Object data[][] = new Object[100][3];
int i = 0;
double totalprice = 0;
ArrayList<foodCart> foodList = new ArrayList<>();

informacijeONaplati(){
    JPanel jp1 = new JPanel();

        setBackground(Color.red);

        this.setLayout(new GridLayout(2,2));

        JLabel a = new JLabel("Ime Hrane : ");
        jp1.add(a);
        food = new JTextField(50);
        jp1.add(food);
        JLabel b = new JLabel("Količina : ");
        jp1.add(b);
        quantity = new JTextField(50);
        jp1.add(quantity);

        JButton ok = new JButton("OK");

        JPanel jp2 = new JPanel();
        jp2.setSize(700, 400);
        jp1.add(ok);
        ok.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                PreparedStatement pst;
                bazaPodataka con = new bazaPodataka();
                ResultSet rs;
                try{
                    pst = con.mkDataBase().prepareStatement("select f_kolicina from
restoranognjen.hrana where f_ime = ?");
                    pst.setString(1, food.getText());
                    rs = pst.executeQuery();

                    while (rs.next()){
                        foodCart f = new foodCart();
                        f.name = food.getText();
                        f.quantity = Integer.parseInt(quantity.getText());
                        f.totalPer = f.quantity*rs.getDouble("f_kolicina");
                        totalprice += f.quantity*rs.getDouble("f_kolicina");

                        foodList.add(f);
                        data[i][0] = f.name;
                        data[i][1] = Integer.parseInt(quantity.getText());
                        data[i][2] = f.quantity*rs.getDouble("f_kolicina");
                        i++;
                        food.setText("");
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

        quantity.setText("");
        DefaultTableModel model = (DefaultTableModel) cart.getModel();
        model.setRowCount(0);
        cart = new JTable(data, columnNames);
        System.out.println(totalprice);
        removeAll();
        totalP.revalidate();
        totalP.repaint();
        revalidate();
        repaint();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
catch(Exception ex){
    System.out.println(ex);
}
});
cart = new JTable(data, columnNames);
cart.setSize(300, 450);
jp2.setLayout(new FlowLayout());
jp2.add(totalP);
jp2.add(new JScrollPane(cart, JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED,
JScrollPane.HORIZONTAL_SCROLLBAR_NEVER));
JButton checkOut = new JButton("Provjera");
checkOut.setSize(40, 50);
jp2.add(checkOut);
checkOut.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int count = 1;
        for(foodCart fc : foodList){
            System.out.println(count + ": Ime Hrane : " + fc.name + "
Količina : "+ fc.quantity + " Vrijednost : " + fc.totalPer + "KM");

        }
        double vat = 15;
        System.out.println("Ukupna Vrijednost : " + (totalprice+totalprice*vat)
+ "KM");

        JOptionPane.showMessageDialog(null, "Ukupna Vrijednost : " +
(totalprice+totalprice*vat/100) + "KM " + vat+"%");
        hide();
    }
});

this.add(jp1);
this.add(jp2);
this.setSize(600,550);
this.setLocationRelativeTo(null);
this.setVisible(true);
}

```

```

class foodCart{
    String name;
    Double totalPer;
    int quantity;
}
}

```

3.6 Hrana info



Slika 9. Hrana info

```

import java.awt.*;
import java.awt.event.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.*;

public class hranaInfo {

    private JFrame mainFrame;
    private JLabel headerLabel;
    private JPanel controlPanel;

    hranaInfo () {

```

```

        prepareGUI();
    }

    public static void main(String[] args) throws SQLException{
        hranaInfo swingControlDemo = new hranaInfo();
        swingControlDemo.showButtonDemo();
    }

    private void prepareGUI(){
        mainFrame = new JFrame("Prikaz svih stavki");
        mainFrame.setSize(700,400);
        mainFrame.getContentPane().setBackground(Color.lightGray);
        mainFrame.setLayout(new FlowLayout());
        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
            }
        });
        headerLabel = new JLabel("", JLabel.CENTER);

        controlPanel = new JPanel();
        controlPanel.setSize(700,400);
        controlPanel.setLayout(new FlowLayout());

        mainFrame.add(headerLabel);
        mainFrame.add(controlPanel);
        mainFrame.setVisible(true);
    }

    public void showButtonDemo() throws SQLException{

        headerLabel.setText("RESTORAN Ognjen Panel");
        headerLabel.setFont(new Font(null, Font.BOLD, 25));

        String[] columnNames = {"ID", "Ime Hrane",
                                "Vrijednost",
                                "Količina"
                                };
        Object[][] data = new Object[100][4];

        PreparedStatement pst;
        ResultSet rs;
        bazaPodataka con = new bazaPodataka();
        try{
            pst = con.mkDataBase().prepareStatement("select * from
            restoranognjen.hrana");

            rs = pst.executeQuery();
            int i=0;
            while(rs.next()){
                data[i][0] = rs.getInt("f_id");
                data[i][1] = rs.getString("f_ime");
            }
        }
    }

```

```

        data[i][2] = rs.getDouble("f_vrijednost");
        data[i][3] = rs.getInt("f_kolicina");
        i++;
    }
    mainFrame.setVisible(false);

    }catch(Exception ex){
        System.out.println(ex);
        System.out.println("Greška");
        JOptionPane.showMessageDialog(null, "Greška!");
    }

    JTable table = new JTable(data, columnNames);
    table.setSize(400, 400);
    table.setVisible(true);
    controlPanel.add(new JScrollPane(table,
JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED, JScrollPane.HORIZONTAL_SCROLLBAR_NEVER));

    mainFrame.setVisible(true);

    mainFrame.setLocationRelativeTo(null);
}
}

```

3.7 Izbriši hranu



Slika 10. Izbriši hranu

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class izbrisiHranu {

    private JFrame mainFrame;
    private JLabel headerLabel;
    private JPanel controlPanel;
    private JLabel id,name,price,quantity;
    private static int count = 0;
    GridLayout experimentLayout = new GridLayout(1,1);
    ResultSet rs;

    izbrisiHranu(){

        prepareGUI();
    }

    public static void main(String[] args){
        azurirajHranu swingControlDemo = new azurirajHranu();
        swingControlDemo.showButtonDemo();
    }

    private void prepareGUI(){
        mainFrame = new JFrame("BRISANJE!");
        mainFrame.setSize(700,400);
        mainFrame.getContentPane().setBackground(Color.gray);

        mainFrame.setLayout(new GridLayout(3, 1));
        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                mainFrame.setVisible(false);
            }
        });
        headerLabel = new JLabel("", JLabel.CENTER);

        controlPanel = new JPanel();
        controlPanel.setLayout(new FlowLayout());

        mainFrame.add(headerLabel);
        mainFrame.add(controlPanel);
        mainFrame.setVisible(true);
    }

    public void showButtonDemo(){

```

```

        headerLabel.setText("RESTORAN Ognjen");
        headerLabel.setFont(new Font(null, Font.BOLD, 30));
        headerLabel.setForeground(Color.white);

        name = new JLabel("Unesi IME");
        JTextField tf2=new JTextField();
        tf2.setSize(100,30);

        JButton okButton = new JButton("IZBRIŠI");

        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                PreparedStatement pst;
                bazaPodataka con = new bazaPodataka();
                try{
                    pst = con.mkDataBase().prepareStatement("DELETE FROM
restoranognjen.hrana where f_ime = ?");
                    pst.setString(1,tf2.getText());

                    pst.execute();

                    JOptionPane.showMessageDialog(null, "Predmet Izbrisan " +
tf2.getText());
                    mainFrame.setVisible(false);

                }catch(Exception ex){
                    System.out.println(ex);
                    System.out.println("EEEE");
                    JOptionPane.showMessageDialog(null, "Predmet Nije Pronadjen : " +
tf2.getText());
                }finally{

                }
            }
        });

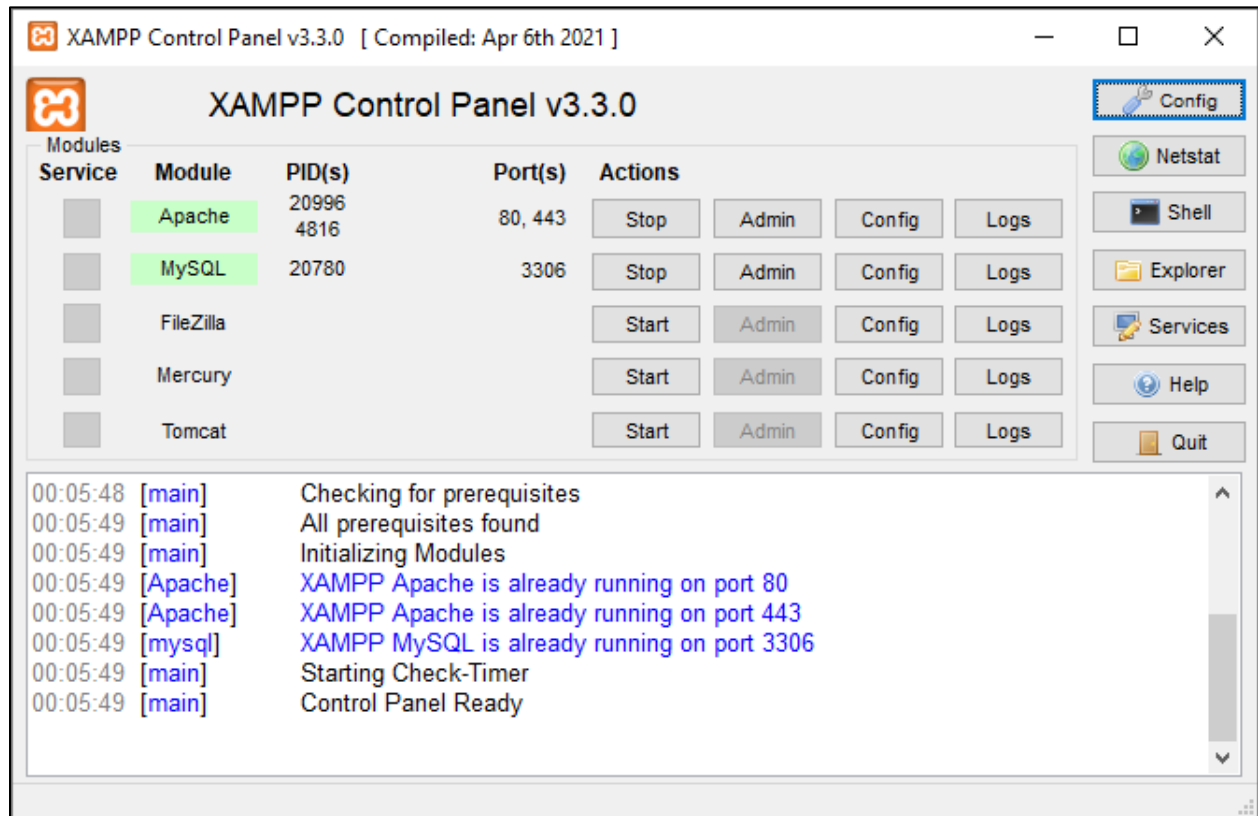
        JPanel jp = new JPanel();
        jp.add(name);
        jp.add(tf2);

        jp.setSize(700,400);
        jp.setLayout(experimentLayout);
        controlPanel.add(jp);
        jp.add(okButton);

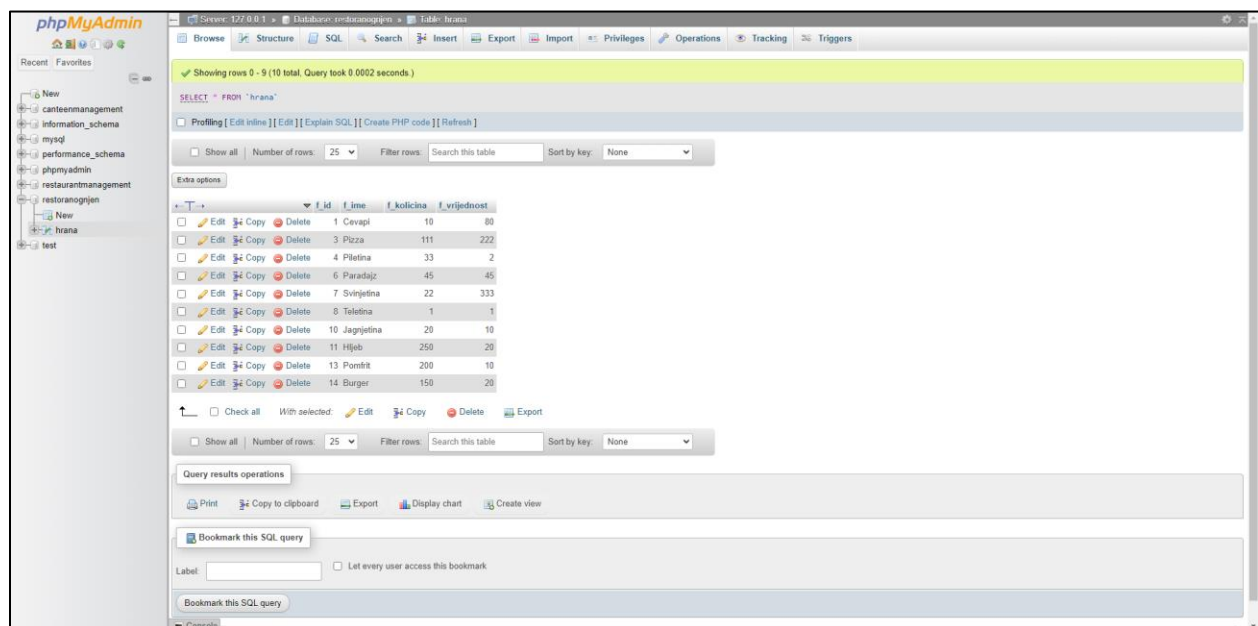
        mainFrame.setVisible(true);
        mainFrame.setLocationRelativeTo(null);
    }
}

```


3.8 Baza podataka “Restoran”



Slika 11. XAMPP i MySQL



Slika 12. phpMyAdmin

Za postavljanje lokalnog servera i baze podataka, koristićemo XAMPP, besplatnu i jednostavnu distribuciju koja sadrži Apache HTTP Server, MySQL bazu podataka i PHP interpretator. Instalacija XAMPP-a je jednostavna i dostupna za različite operativne sisteme.

Nakon instalacije XAMPP-a, pokrenite MySQL i Apache module putem XAMPP kontrolnog panela. Otvorite phpMyAdmin u vašem pretraživaču (obično dostupno na <http://localhost/phpmyadmin>) i kreirajte novu bazu podataka sa odgovarajućim tabelama koje će skladištiti podatke o jelima.

Dodavanje MySQL konektora: Preuzmite MySQL JDBC driver (MySQL Connector/J) sa zvanične MySQL stranice. U Eclipse IDE-u, desni klik na vaš projekat u "Project Explorer" i izaberite **Build Path > Configure Build Path**. U prozoru koji se otvori, idite na karticu **Libraries** i kliknite na **Add External JARs**. Pronađite i izaberite preuzeti MySQL JDBC driver JAR fajl, a zatim kliknite **Apply and Close**.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

CREATE TABLE `hrana` (
  `f_id` int(11) NOT NULL,
  `f_ime` varchar(20) NOT NULL,
  `f_kolicina` int(11) NOT NULL,
  `f_vrijednost` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `hrana` (`f_id`, `f_ime`, `f_kolicina`, `f_vrijednost`) VALUES
(1, 'Cevapi', 10, 80),
(3, 'Pizza', 111, 222),
(4, 'Piletina', 33, 2),
(5, 'Luk', 21, 22),
(6, 'Paradajz', 45, 45),
(7, 'Svinjetina', 22, 333),
(8, 'Teletina', 1, 1),
(10, 'Jagnjetina', 20, 10),
(11, 'Hljeb', 250, 20),
(13, 'Pomfrit', 200, 10),
(14, 'Burger', 150, 20);

ALTER TABLE `hrana`
  ADD PRIMARY KEY (`f_id`),
  ADD UNIQUE KEY `f_ime` (`f_ime`);

ALTER TABLE `hrana`
  MODIFY `f_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

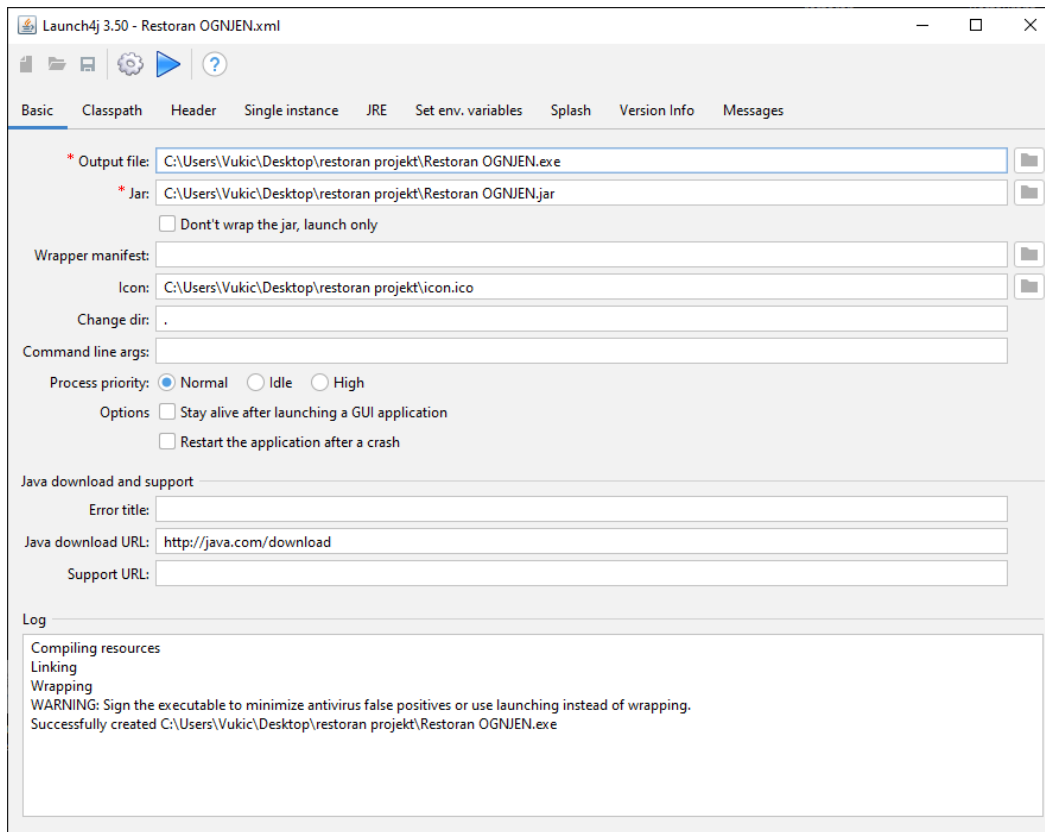
public class bazaPodataka {
    public Connection con;
    String url ="jdbc:mysql://localhost:3306/restoranognjen";
    String db = "restoranognjen";
    String user = "root";
    String pass = "";

    public Connection mkDataBase() throws SQLException{
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(url, user, pass);

        } catch (ClassNotFoundException ex) {
            Logger.getLogger(bazaPodataka.class.getName()).log(Level.SEVERE, null,
ex);
        }
        return con;
    }
}

```

3.9 Konvertovanje Java file u .exe



Slika 13. Konvertovanje Java file u .exe

Za korišćenje Launch4j-a, prvo je potrebno preuzeti i instalirati softver sa zvanične web stranice ili iz distribucije kao što je SourceForge. Nakon instalacije, otvorite Launch4j i konfigurišite ga prema potrebama vaše Java aplikacije.

1. **Osnovne Postavke:** U Launch4j-u, prvo treba odabrati .exe ime izlazne datoteke i putanju gdje će se datoteka generisati. Takođe se može postaviti ikona za .exe datoteku.
2. **JRE Postavke:** Postavite minimalnu verziju JRE koja je potrebna za pokretanje vaše aplikacije. Takođe se može uključiti ili isključiti podrška za 32-bitne ili 64-bitne verzije JRE.
3. **Klasa Glavnog Programa:** Odaberite klasu koja sadrži main metod vaše Java aplikacije.
4. **JVM Opcije:** Definišite JVM opcije kao što su Xmx, Xms, Classpath i slično, ako je potrebno.
5. **Dodatne Opcije:** Ovde možete dodati dodatne parametre za konfiguraciju .exe datoteke, kao što su verzija proizvoda, opis, kompanija i slično.

Nakon što ste konfigurali sve potrebne parametre, možete pokrenuti proces generisanja .exe datoteke pritiskom na dugme za kompiliranje ili generisanje. Launch4j će automatski kreirati .exe izvršnu datoteku u navedenoj putanji.

4. Zaključak

Zaključak ovog projekta ukazuje na važnost alata poput Launch4j za efikasnu distribuciju Java aplikacija na Windows platformi. Integracija baze podataka u Java aplikaciju putem Eclipse IDE omogućava strukturiranje i efikasno upravljanje podacima, što je ključno za aplikacije poput sistema za upravljanje restoranom. Implementacija MySQL baze podataka omogućava skladištenje informacija o jelima, transakcijama i korisnicima, pružajući programerima robustan okvir za razvoj i održavanje aplikacije.

Upotreba Launch4j-a omogućava konverziju Java aplikacije u .exe datoteku, čime se olakšava distribucija aplikacije korisnicima bez potrebe za instalacijom Java Runtime Environment (JRE). Ovaj proces čini aplikaciju pristupačnijom i jednostavnijom za krajnje korisnike, poboljšavajući njihovo korisničko iskustvo.

Zaključno, projekat "Restoran" demonstrira kako Java programski jezik, uz pomoć Eclipse IDE-a za razvoj i Launch4j alata za konvertovanje u .exe format, može efikasno podržati kompleksne aplikacije poput sistema za upravljanje restoranom. Kombinacija ovih tehnologija pruža programerima snažne alate za razvoj, distribuciju i održavanje aplikacija, čime se postiže visok nivo funkcionalnosti, sigurnosti i korisničkog zadovoljstva.

5. Literatura

- [1] Knjiga: Eclipse: Programming Java Applications 1st Edition by Steve Holzner
- [2] salapura.com/java
- [3] youtube.com
- [4] [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- [5] java.com
- [6] scribd.com