



**Tehnička škola Čakovec**

# **ELABORAT ZAVRŠNOG RADA**

## **MALOPRODAJNA BLAGAJNA**

**Mentor:**

**Mirko Jambrošić, mag. ing. inf. et comm. techn.**

**Učenik**

**Noa Turk, 4.RT**

**Čakovec, svibanj 2024.**



**Tehnička škola Čakovec**  
**Prosudbeni odbor za završni rad**

Učenik: **Noa Turk**

Razred: **4.RT**

Školska godina: **2023./2024.**

Obrazovno područje: **računalstvo**

Zanimanje: **Tehničar za računalstvo**

Naziv zadatka: **MALOPRODAJNA BLAGAJNA**

Opis zadatka: Učenik će napraviti rješenje blagajne. Rješenje mora omogućiti upravljanje skladištem (dodavanje na skladište), automatizirano naručivanje robe, ispis računa, storniranje računa.

Učenik će se za konzultacije obratiti svojem mentoru.

Zadatak zadan:

Rok predaje pisanog rada:

Predviđeni datum obrane:

Mentor:

Mirko Jambrošić, mag. ing. inf. et comm. techn.



## Sadržaj

<b>1. UVOD.....</b>	<b>6</b>
<b>2. GLAVNI DIO .....</b>	<b>7</b>
2.1. Teorijski dio .....	7
2.1.1. Korišteni alati .....	7
2.1.2. Zahtjevi aplikacije .....	8
2.1.3. Sigurnost u aplikaciji .....	8
2.1.4. Baza podataka .....	9
2.1.5. Budućnost aplikacije .....	10
2.2. Praktični dio .....	11
2.2.1. Prijava u aplikaciju .....	11
2.2.2. Glavno sučelje-blagajna .....	12
2.2.3. Glavno sučelje-administracija .....	12
2.2.3.1. Administracija zaposlenika.....	13
2.2.3.2. Administracija dobavljača .....	14
2.2.3.3. Administracija proizvoda .....	15
2.2.3.4. Administracija računa .....	17
<b>3. ZAKLJUČAK .....</b>	<b>21</b>
<b>4. LITERATURA.....</b>	<b>22</b>
<b>5. POPIS SLIKA .....</b>	<b>23</b>
<b>6. KONZULTACIJSKI LIST .....</b>	<b>24</b>



## 1. UVOD

Softversko rješenje maloprodajna blagajna pruža vlasnicima tvrtki koje se koriste maloprodajnom blagajnom jednostavno i učinkovito upravljanje. Sve što je potrebno jest prijaviti se kao administrator pomoću sigurnosnog ključa, odnosno lozinke koju dobiva od distributera softvera.

Administrator ima potpunu kontrolu nad sustavom i može obavljati različite zadatke kako bi osigurao glatko funkcioniranje tvrtke. To uključuje dodavanje novih zaposlenika u sustav, upravljanje dobavljačima, izdavanje i storniranje računa te upravljanje stavkama na skladištu. Također, administrator ima mogućnost mijenjati informacije o proizvodima u sustavu.

Kada je riječ o izdavanju računa, korisnik ima pregledan popis dostupnih proizvoda. Odabir se vrši jednostavno, putem selekcije odgovarajućih identifikatora proizvoda, što olakšava proces naplate i smanjuje mogućnost grešaka.

U skladu s postavljenim zadatkom, izrađen je sustav za upravljanje, pregled i naplatu artikala za potrebe određenog poduzeća. Odlučeno je implementirati ovaj sustav koristeći Microsoft Visual Studio 2019, C# jezik, .NET Framework, XAMPP i MySQL bazu podataka.

Aplikacija je dizajnirana s više prozora (stranica), pružajući različite funkcionalnosti ovisno o korisničkoj ulozi. Glavne uloge u sustavu su administrator i zaposlenik.

Administrator ima mogućnost dodavanja novih artikala u sustav te im pridodaje ključne informacije poput naziva, opisa, cijene, barkoda i minimalno stanje na skladištu... Osim toga, administrator može upravljati postojećim artiklima, ažurirati njihove podatke te pratiti stanje zaliha.

Zaposlenik ima ovlasti za dodavanje količine artikala na zalihi te izdavanje računa kupcima. Prijavom putem podataka koje su dobili od administratora, zaposlenici mogu pristupiti funkcionalnostima aplikacije, uključujući izdavanje i pregledavanje računa, storniranje računa te označavanje statusa računa kao na čekanju, dovršene ili stornirano.

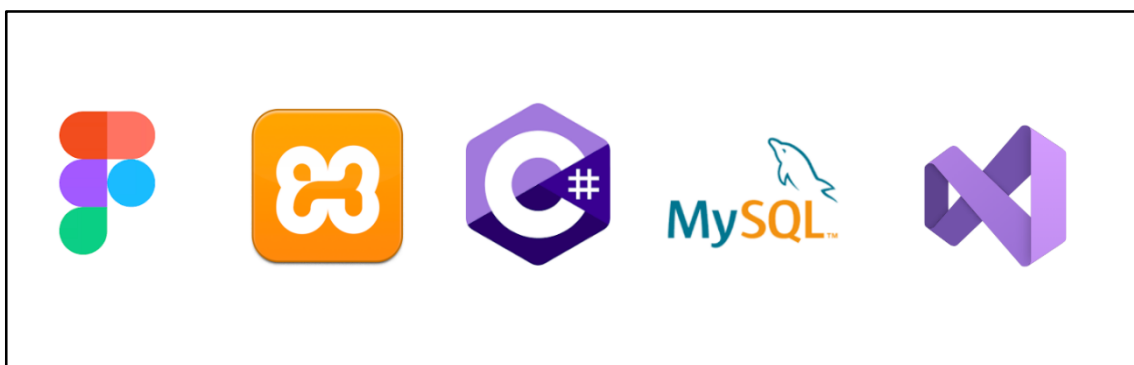
## 2. GLAVNI DIO

### 2.1. Teorijski dio

#### 2.1.1. Korišteni alati

Pri izradi ovog softverskog rješenja korišteni su različiti alati i tehnologije kako bi osigurali njezinu funkcionalnost i učinkovitost. Kao glavno integrirano razvojno okruženje (IDE) korišteni je Microsoft Visual Studio 2019, koje je pružio bogat skup alata i podršku za razvoj aplikacija u C# jeziku. Za izgradnju aplikacije korišteni je C# jezik i .NET Framework kao osnovnu platformu. Za lokalno postavljanje servera i bazu podataka korišten je XAMPP platforma. XAMPP je besplatan i otvoren izvor koji omogućuje jednostavno postavljanje lokalnog servera z. Uz pomoć XAMPP-a pokrenuta je MySQL baza podataka i Apache web server, što je omogućilo pohranu podataka vezanih uz artikle, zalihe, korisnike i ostale relevantne informacije za funkcioniranje aplikacije. Za dizajniranje korisničkih sučelja i prototipiranje aplikacije korišten je alat Figma. Figma je omogućila izradu skica i prototipa korisničkog sučelja aplikacije "Blagajna", pružajući jasnu vizualizaciju funkcionalnosti i rasporeda elemenata sučelja.

Korištenje ovih alata omogućilo je učinkovit razvoj aplikacije "Blagajna", uz osiguranje stabilnosti, sigurnosti i intuitivnog korisničkog iskustva. Integracija alata omogućila je glatko funkcioniranje aplikacije i olakšala proces razvoja, testiranja i implementacije..



*Slika 1. Alati korišteni u izradi aplikacije*



### **2.1.2. Zahtjevi aplikacije**

Pri izradi aplikacije "Blagajna" identificirani su određeni zahtjevi koji korisnicima omogućuju nesmetano korištenje aplikacije. Administrator za pokretanje aplikacije treba imati sigurnosni ključ (lozinku) koju dobiva od distributera softvera. Ostali zaposlenici dobivaju početnu lozinku od administratora. Ova lozinka osigurava pristup aplikaciji te omogućava identifikaciju i autorizaciju korisnika. Bez valjane lozinke, korisnik neće moći pristupiti funkcionalnostima aplikacije.

Osim toga, korisnik treba imati uspostavljenu vezu s bazom podataka. Za ispravno funkcioniranje aplikacije, potrebno je da korisnik ima aktivnu i stabilnu vezu s MySQL bazom podataka koja sadrži sve potrebne informacije o artiklima, zalihama, korisnicima i ostalim relevantnim podacima.

### **2.1.3. Sigurnost u aplikaciji**

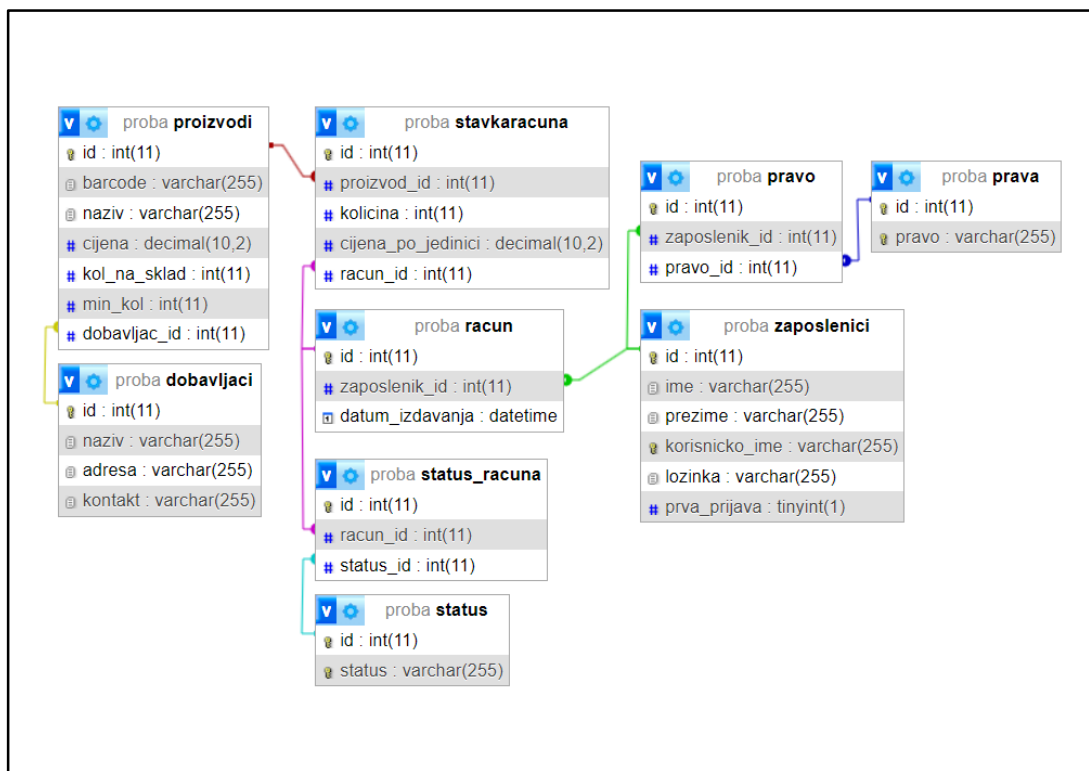
Kako bi se zaštitila sigurnost i stabilnost aplikacije, radi se provjera unosa podataka. Prilikom unosa podataka u određena polja provjerava se točnost podataka, npr. kad se unosi cijena proizvoda aplikacija ne prihvća uneseni podatak koji nije broj te upozorava korisnika. Korisnik kao cijenu može unijeti samo broj (cijeli ili s decimalnim zarezom) bez drugih znakova. Prilikom prijave unesena se lozinka uspoređuje sa lozinkom koja je zapisana u bazi, ako nije, aplikacija će izbaciti grešku te upozoriti korisnika. Kako bi se zaštitio integritet aplikacije provodi se kriptiranje lozinke i taj se zapis upisuje u bazu.

## 2.1.4. Baza podataka

Za bazu je korišten MySQL tablice su kreirana prema uputama struke. To su:

- prava → sadrži vrste prava
- pravo → spaja prava sa zaposlenikom
- zaposlenici → sadrži podatke o zaposleniku
- status → sadrži vrste statusa
- status\_racuna → spaja račun sa statusom
- racun → sadrži podatke o računu
- stavkaracuna → sadrži podatke o proizvodu unutar jednog račun
- proizvodi → sadrži podatke o proizvodu
- dobavljac → sadrži podatke dobavljača proizvoda

Baza je dizajnirana da jedan zaposlenik može imati više prava. Jedan račun može imati točno jednog izdavača (zaposlenika) te više stavka računa, jedan račun može imati jedan status jer je racun\_id u tablici status\_racuna postavljen da bude jedinstveni. Jedan dobavljač može imati više proizvoda a jedan proizvod može imati točno jednog dobavljača.



Slika 2. Baza podataka napravljena u MySQL

### 2.1.5. Budućnost aplikacije

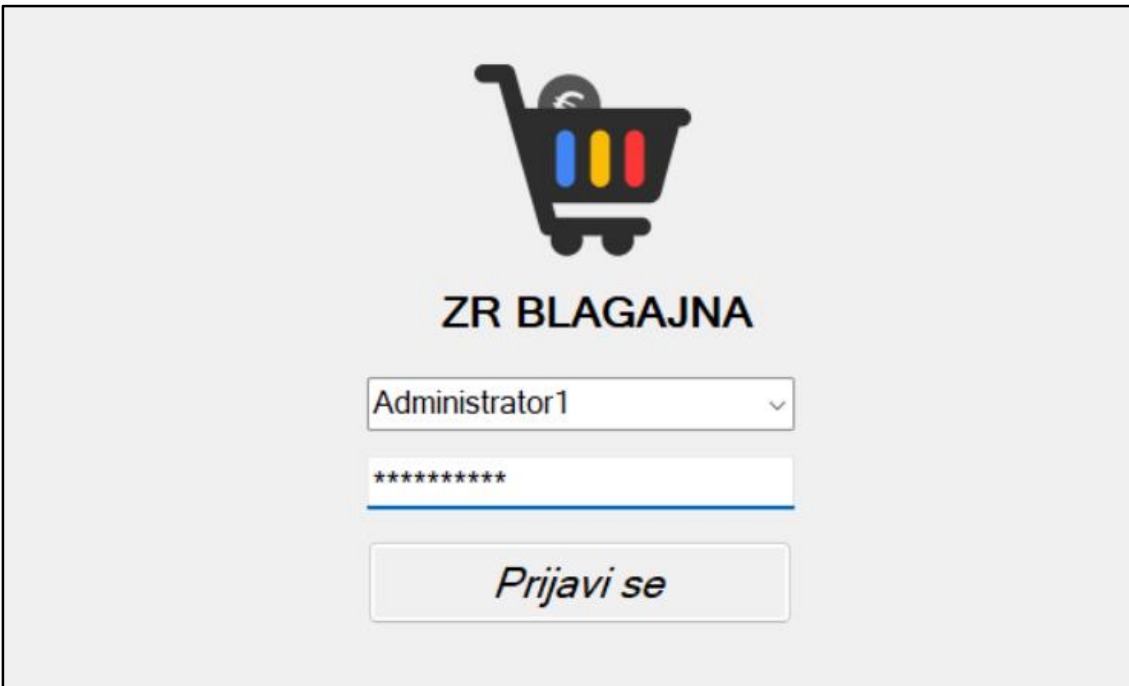
Aplikacija "Blagajna" otvara mnoge mogućnosti za daljnji razvoj i unaprjeđenje funkcionalnosti kako bi se još bolje prilagodila potrebama korisnika i povećala učinkovitost poslovanja. Neki od potencijalnih smjerova razvoja uključuju:

- Integracija printera za automatsko ispisivanje računa: Dodavanje mogućnosti spajanja printera omogućilo bi automatizirano ispisivanje računa nakon što su izdani.
- Mogućnost izdavanja R1 računa: Uvođenje mogućnosti izdavanja R1 računa korisnicima omogućilo bi prilagodbu aplikacije specifičnim zakonskim zahtjevima i potrebama određenih industrija. R1 računi su često potrebni za pravilno evidentiranje financijskih transakcija u skladu s poreznim propisima.
- Integracija skenera za učitavanje barkodova: Dodavanje mogućnosti spajanja skenera omogućilo bi korisnicima brzo i precizno učitavanje barkodova s proizvoda. Ovo bi olakšalo procese evidentiranja proizvoda u sustavu, smanjilo vrijeme potrebno za naplatu i smanjilo mogućnost pogrešaka pri unosu podataka.
- Korištenje kartica s popustima: Dodavanje podrške za korištenje kartica s popustima omogućilo bi korisnicima ostvarivanje različitih pogodnosti prilikom kupnje. Korisnici bi mogli koristiti svoje vjerne kupce kartice ili kupone za ostvarivanje popusta na određene proizvode ili usluge.
- Generiranje izvještaja o prodaji: Automatizirano generiranje izvještaja o prodaji omogućuje vlasnicima tvrtki brz pristup ključnim podacima o prodaji, što im pomaže u donošenju informiranih poslovnih odluka.

## 2.2. Praktični dio

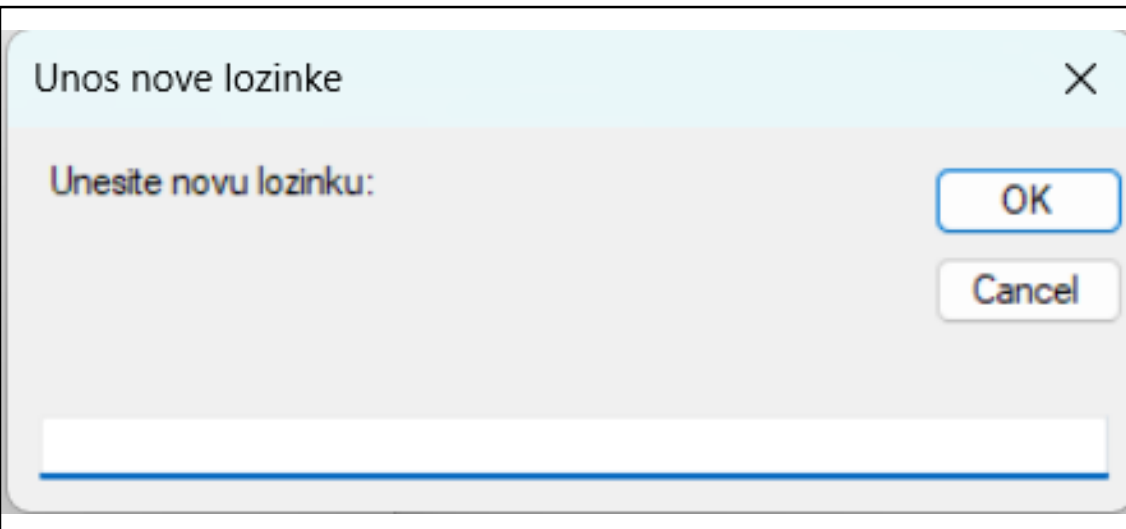
### 2.2.1. Prijava u aplikaciju

Administrator i zaposlenici se prvi puta u sustav prijavljuju s dobivenom lozinkom od distributera softvera ili administratora. Kada se prvi puta prijave sustav ih pita za unos nove lozinke po želji te se tada lozinka kriptira i unosi u bazu. Potrebno se povezati s bazom podataka, postavke povezivanja se postavljaju u App.config datoteci.



The screenshot shows a login window titled "ZR BLAGAJNA". At the top is a shopping cart icon with a coin. Below the title, there is a dropdown menu with "Administrator1" selected. Underneath is a password field filled with asterisks. At the bottom is a button labeled "Prijavi se".

Slika 3. Prikaz prijave u sustav



The screenshot shows a dialog box titled "Unos nove lozinke" with a close button (X) in the top right corner. The text "Unesite novu lozinku:" is displayed. Below the text is a large empty text input field. To the right of the input field are two buttons: "OK" and "Cancel".

Slika 4. Promjena lozinke ako je prva prijava

```
<appSettings>
  <add key="Server" value="localhost" />
  <add key="Baza" value="blagajna" />
  <add key="Korisnik" value="korisnickoime" />
  <add key="Lozinka" value="lozinka" />
</appSettings>
```

Primjer 1. Postavke za spajanje na bazu koje se trebaju konfigurirati u App.config

### 2.2.2. Glavno sučelje-blagajna

Korisnik kao prijavljeni zaposlenik može dodavati proizvode u račun prema potrebi te dodavati ili oduzeti količinu po potrebi.

The screenshot shows the 'BLAGAJNA ADMINISTRACIJA' interface. At the top, there are tabs for 'BLAGAJNA' and 'ADMINISTRACIJA'. Below the tabs, there are input fields for 'Id' (5), 'Barcode' (567890123), 'Naziv proizvoda' (Maslac), 'Kol' (2), and 'Cijena po jedinici' (5,99). There is also a 'Broj računa' field with the value 19 and a 'DODAJ' button. Below these fields is a table with columns 'Proizvod ID', 'Količina', and 'Cijena po jedinici'. The table contains one row with values 1, 2, and 2,99. A large display shows the total amount '5,98'. At the bottom, there are buttons for 'Očisti' and 'IZDAJ RAČUN', and a status bar indicating 'Prijavljen/a: Zaposlenik2'.

ID	Naziv	Barkod
1	Kruh	123456789
2	Mlijeko	234567890
5	Maslac	567890123
9	Čaj	901234567
3	Jaja	345678901
4	Jogurt	456789012
7	Sok	789012345
8	Kava	890123456
6	Kolač	678901234
10	Voće	012345678

Slika 5. Sučelje za izdavanje računa

### 2.2.3. Glavno sučelje-administracija

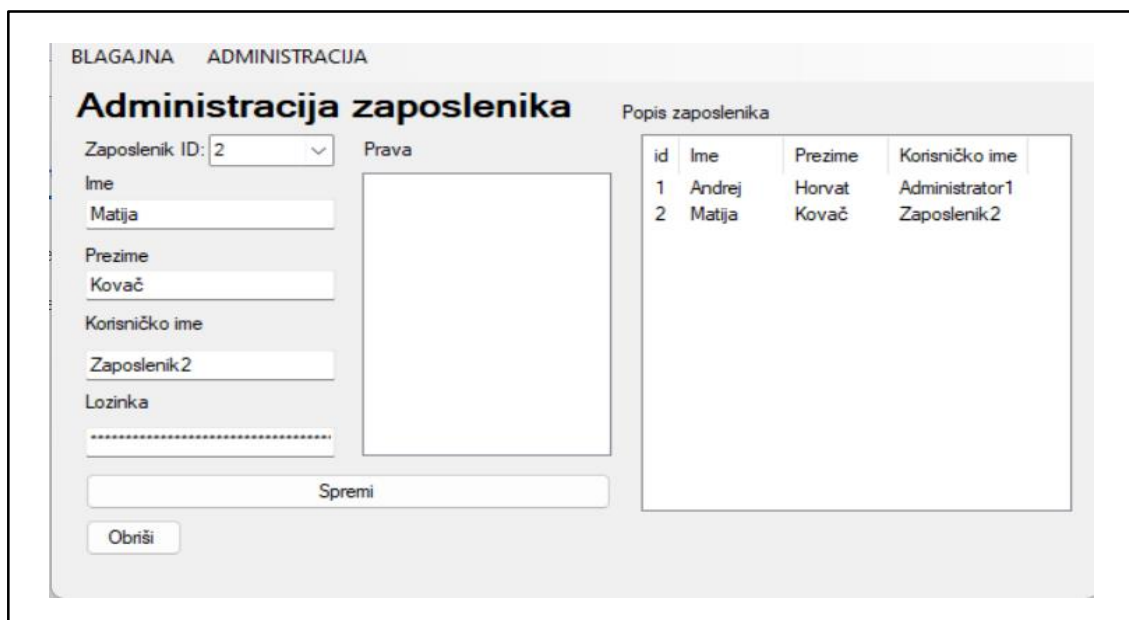
Korisnik ima li ovlasti ima pristup administraciji te dodavati i uređivati zaposlenike, dobavljače i proizvode te upravljati računima prema potrebi ili želji.

The screenshot shows the 'ADMINISTRACIJA' menu in the application interface. The menu is open, showing options: 'Zaposlenici', 'Dobavljači', 'Proizvodi', and 'Računi'. The background shows the 'BLAGAJNA' tab and some input fields.

Slika 6. Izbornik administracijskih opcija

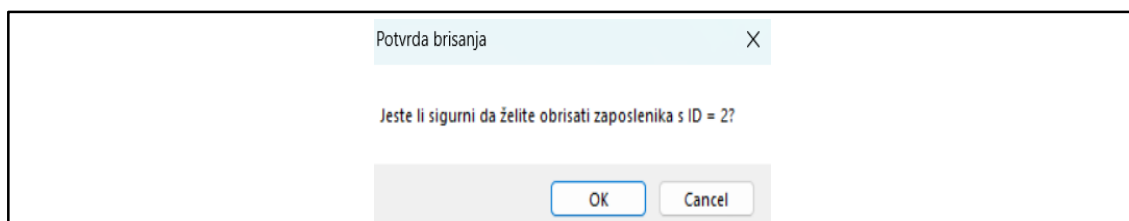
### 2.2.3.1. Administracija zaposlenika

Zaposlenik prema potrebi može pregledati sve zaposlenike i odabrati zaposlenika kojeg želi urediti ili kreirati novog. Ako su uneseni potrebni podatci oni će se upisati u bazu potrebno je unijeti ime, prezime, korisničko ime i početnu lozinku... Također moguće je obrisati svakog zaposlenika ali ne i samog sebe (uvijek mora ostati barem jedan zaposlenik u bazi).

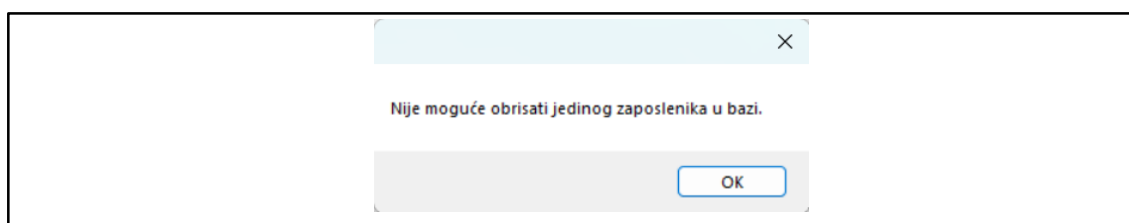


id	Ime	Prezime	Korisničko ime
1	Andrej	Horvat	Administrator1
2	Matija	Kovač	Zaposlenik2

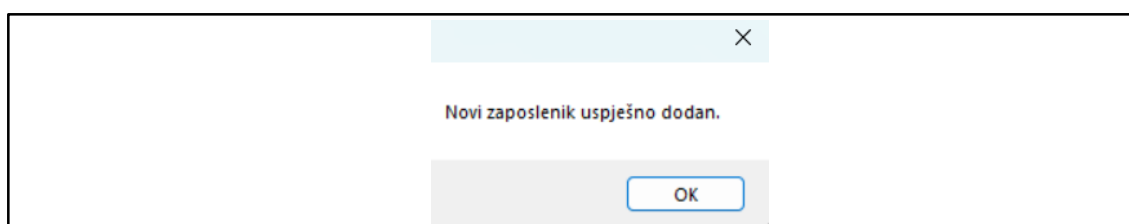
Slika 7. Sučelje za administraciju zaposlenika



Slika 8. Potvrda brisanja zaposlenika



Slika 9. Poruka o odbijenom brisanju



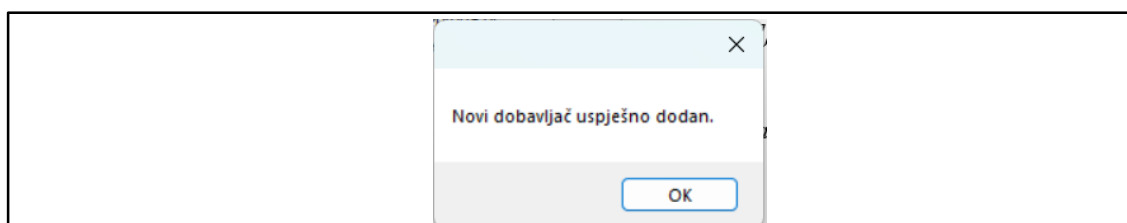
Slika 10. Poruka o uspješnom kreiranju zaposlenika

### 2.2.3.2. Administracija dobavljača

Zaposlenik prema potrebi može pregledati sve dobavljače i odabrati dobavljača kojeg želi urediti ili kreirati novog. Administrator može dodati dobavljače ili ih urediti, ako administrator želi obrisati određenog dobavljača dolazi poruka koja ga pita je li siguran da želi obrisati dobavljača, ako ga obriše tada se proizvodi više neće moći naručiti od tog dobavljača.

id	naziv	adresa	kontakt
1	Dobavljač 1	Primjer adrese 1	mail@mail.com
2	Dobavljač 2	Primjer adrese 2	proba@proba.hr
3	Dobavljač 3	Primjer adrese 3	test@test.net

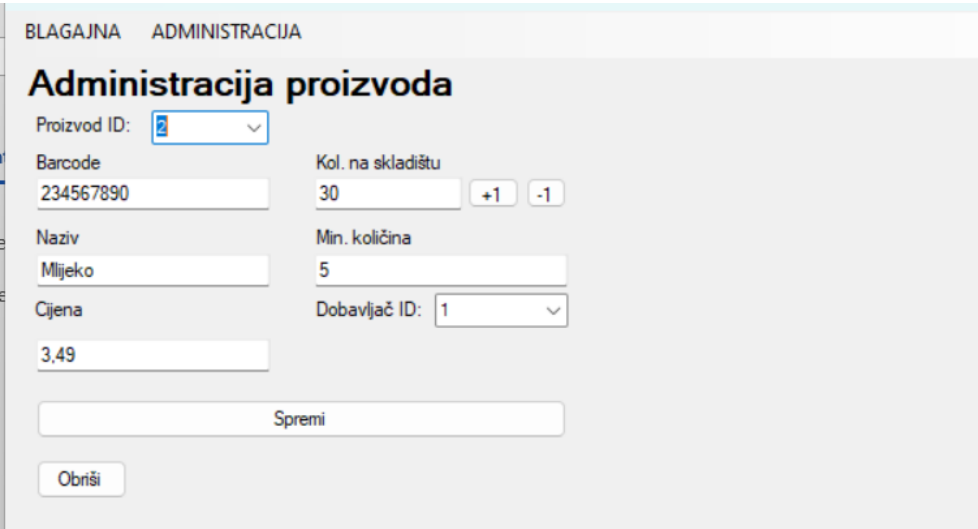
Slika 11. Sučelje za administraciju dobavljača



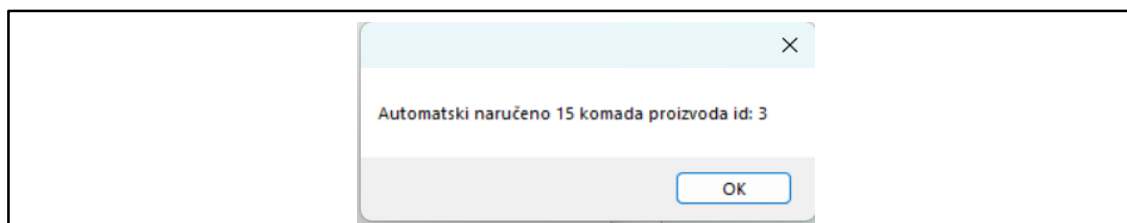
Slika 12. Poruka o uspješno dodanom dobavljaču

### 2.2.3.3. Administracija proizvoda

Zaposlenik prema potrebi može odabrati te urediti ili obrisati odabrani proizvod ili kreirati novi proizvod. Zaposlenik na administraciji proizvoda može postaviti novo stanje na zalihi ili smanjiti ili povećati količinu za jedan. Proizvodi se naručuju automatski kada zaposlenik doda određeni proizvod u neki račun i stanje na zalihi padne ispod minimalne količine ili se u administraciji proizvoda postavi količina na skladištu manja od minimalne količine. Za automatsko naručivanje koristi se logika u metodi `NaruciProizvod` (Primjer 2).



Slika 13. Sučelje za administraciju proizvoda



Slika 14. Poruka o automatsko naručenim proizvodima



```

private void NaruciProizvod(int proizvodId){
    try{
        int razlika = 0;
        int novoStanje = 0;

        using (MySqlConnection veza = new
        MySqlConnection(upraviteljBazom.VezaNaBazu))
        {
            veza.Open();

            string sqlDohvatiKolicine = "SELECT kol_na_sklad, min_kol FROM
        Proizvodi WHERE id = @proizvodId";
            using (MySqlCommand dohvatiKolicine = new
        MySqlCommand(sqlDohvatiKolicine, veza))
            {
                dohvatiKolicine.Parameters.AddWithValue("@proizvodId",
        proizvodId);
                using (MySqlDataReader reader =
        dohvatiKolicine.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        int trenutnaKolicina =
        reader.GetInt32("kol_na_sklad");
                        int minKolicina = reader.GetInt32("min_kol");

                        if (trenutnaKolicina < minKolicina)
                        {
                            razlika = (int)(2 * minKolicina) -
        trenutnaKolicina;
                            novoStanje = trenutnaKolicina + razlika;
                        }
                    }
                }

                if (razlika > 0)
                {
                    MessageBox.Show($"Automatski naručeno {razlika} komada
        proizvoda id: {proizvodId}");

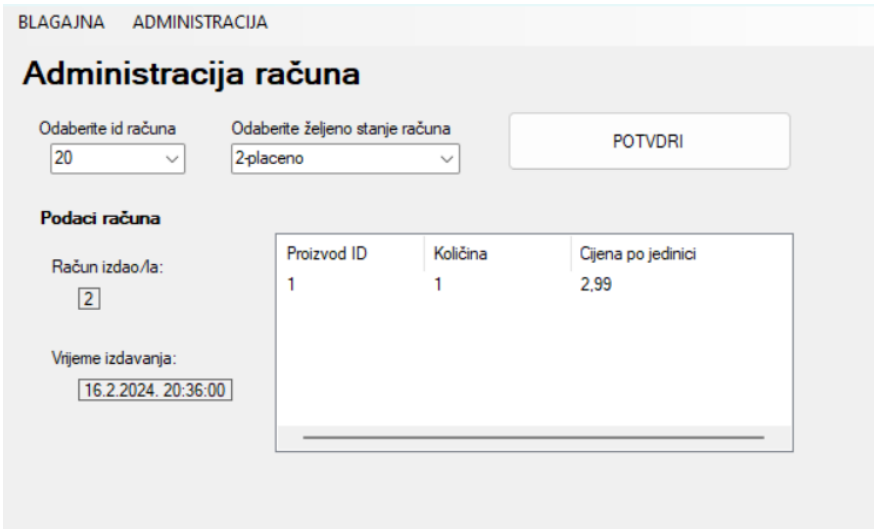
                    string sqlAzurirajSkladiste = "UPDATE Proizvodi SET
        kol_na_sklad = @novoStanje WHERE id = @proizvodId";
                    using (MySqlCommand azurirajSkladiste = new
        MySqlCommand(sqlAzurirajSkladiste, veza))
                    {
                        azurirajSkladiste.Parameters.AddWithValue("@novoStanje", novoStanje);
                        azurirajSkladiste.Parameters.AddWithValue("@proizvodId", proizvodId);
                        azurirajSkladiste.ExecuteNonQuery();
                    }
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Greška prilikom automatskog naručivanja
        proizvoda: {ex.Message}");
        }
    }
}

```

*Primjer 2. Logika koja automatski naruči proizvod*

### 2.2.3.4. Administracija računa

Zaposlenik može odabrati račun te pregledati, i urediti stanje računa (čekanje, plaćeno, stornirano, odbačeno). Za svaki odabrani račun generira se pdf datoteka sa nazivom Račun\_ID. ID je ID odabranog računa. Računi na čekanju nisu izdani i mogu se mijenjati. Svaki račun zaposlenik može stornirati ali stornirani račun ne može se mijenjati. Kada zaposlenik stornira račun, stanje količine proizvoda vrati se za iznos koji je bio oduzeti.



BLAGAJNA ADMINISTRACIJA

### Administracija računa

Odaberite id računa:

Odaberite željeno stanje računa:

POTVDRI

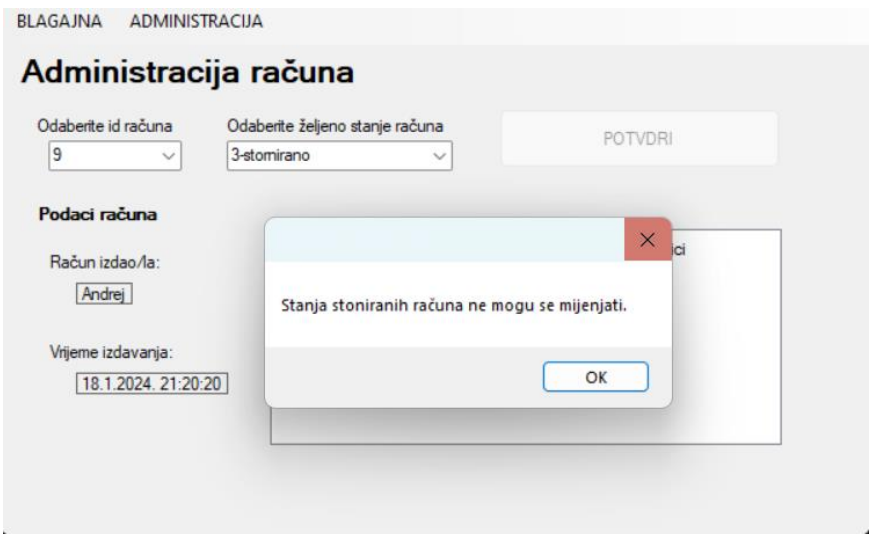
**Podaci računa**

Račun izdao/la:

Vrijeme izdavanja:

Proizvod ID	Količina	Cijena po jedinici
1	1	2,99

Slika 15. Sučelje za administraciju računa



BLAGAJNA ADMINISTRACIJA

### Administracija računa

Odaberite id računa:

Odaberite željeno stanje računa:

POTVDRI

**Podaci računa**

Račun izdao/la:

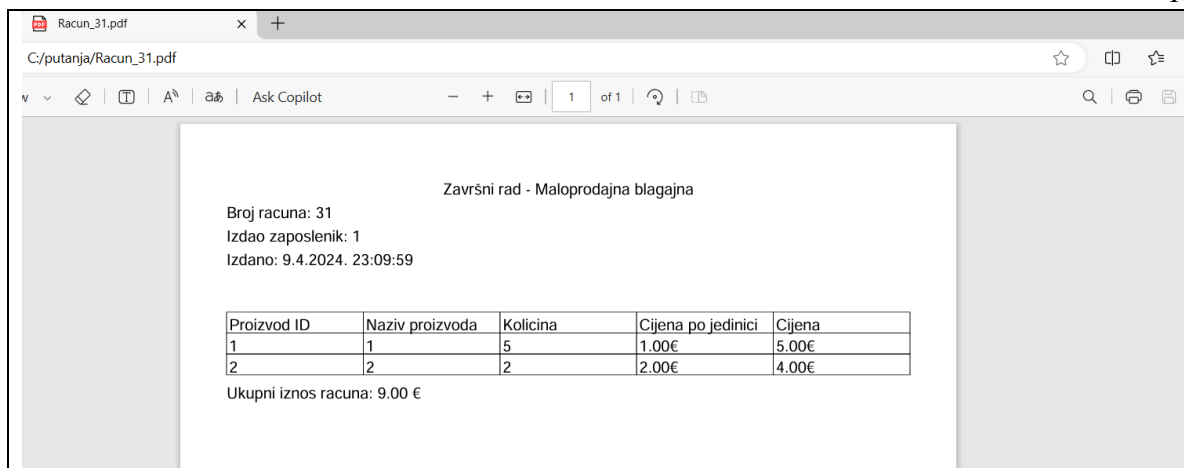
Vrijeme izdavanja:

×

Stanja stoniranih računa ne mogu se mijenjati.

OK

Slika 16. Poruka kod storniranog računa



*Slika 17. Prikaz odabranog računa u pdf datoteci*

```
private void OtvoriPDF(int racunId)
{
    string putanja = "C:\\\\putanja\\";
    string nazivDatoteke = "Racun_" + racunId + ".pdf";
    if (File.Exists(putanja + nazivDatoteke))
    {
        Process.Start(putanja + nazivDatoteke);
    }
    else
    {
        GenerirajPDF(racunId);
        Process.Start(putanja + nazivDatoteke);
    }
}
```

*Primjer 3. Logika koja otvara generirani pdf računa*

```
private void GenerirajPDF(int racunId)
{
    string putanja = "C:\\\\putanja\\";
    string nazivDatoteke = "Racun_" + racunId + ".pdf";
    Document doc = new Document();
    PdfWriter.GetInstance(doc, new FileStream(putanja + nazivDatoteke,
    FileMode.Create));
    doc.Open();

    Paragraph naslov = new Paragraph("Završni rad - Maloprodajna blagajna");
    naslov.Alignment = Element.ALIGN_CENTER;
    doc.Add(naslov);

    Paragraph infoRacun = new Paragraph($"Broj racuna: {racunId}");
    doc.Add(infoRacun);

    decimal ukupniIznos = 0;

    try
    {
        using (MySqlConnection veza = new
        MySqlConnection(upraviteljBazom.VezaNaBazu))
        {
            veza.Open();

            // Dohvati id zaposlenika koji je izdao račun iz tablice racun
            string sqlDohvatiZaposlenika = "SELECT zaposlenik_id,
            datum_izdavanja FROM Racun WHERE id = @racun_id";
            using (MySqlCommand dohvatiZaposlenika = new
            MySqlCommand(sqlDohvatiZaposlenika, veza))
            {
```

```

        dohvatiZaposlenika.Parameters.AddWithValue("@racun_id",
racunId);
        using (MySqlDataReader reader =
dohvatiZaposlenika.ExecuteReader())
        {
            if (reader.Read())
            {
                int zaposlenik_id = reader.GetInt32("zaposlenik_id");
                DateTime datum_izdavanja =
reader.GetDateTime("datum_izdavanja");
                Paragraph infoRacun2 = new Paragraph($"Izdao
zaposlenik: {zaposlenik_id}");
                doc.Add(infoRacun2);
                Paragraph infoRacun3 = new Paragraph($"Izdano:
{datum_izdavanja}");
                doc.Add(infoRacun3);
            }
        }

        // Dodavanje zaglavlje tablice
        PdfPTable tablica = new PdfPTable(5);
        tablica.WidthPercentage = 100;
        tablica.AddCell("Proizvod ID");
        tablica.AddCell("Naziv proizvoda");
        tablica.AddCell("Kolicina");
        tablica.AddCell("Cijena po jedinici");
        tablica.AddCell("Cijena");

        // Dohvati stavke računa iz baze podataka i izračunaj ukupni
iznos
        string sqlUpit = "SELECT sr.proizvod_id, p.naziv, sr.kolicina,
sr.cijena_po_jedinici " +
                        "FROM StavkaRacuna sr " +
                        "INNER JOIN Proizvodi p ON sr.proizvod_id = p.id
" +
                        "WHERE sr.racun_id = @racunId";

        using (MySqlCommand naredba = new MySqlCommand(sqlUpit, veza))
        {
            naredba.Parameters.AddWithValue("@racunId", racunId);

            using (MySqlDataReader reader = naredba.ExecuteReader())
            {
                while (reader.Read())
                {
                    tablica.AddCell(reader["proizvod_id"].ToString());
                    tablica.AddCell(reader["naziv"].ToString());
                    tablica.AddCell(reader["kolicina"].ToString());
                    tablica.AddCell(reader["cijena_po_jedinici"].ToString()+"€");

                    // Izračuna ukupni iznos za svaku stavku i dodaj ga
na ukupni iznos računa
                    decimal cijenaPoJedinici =
reader.GetDecimal("cijena_po_jedinici");
                    int kolicina = reader.GetInt32("kolicina");
                    decimal iznosStavke = cijenaPoJedinici * kolicina;
                    tablica.AddCell(iznosStavke.ToString() + "€");
                    ukupniIznos += iznosStavke;
                }
            }

            doc.Add(new Chunk("\n"));
            doc.Add(tablica);

            Paragraph ukupniIznosParagraf = new Paragraph($"Ukupni iznos
racuna: {ukupniIznos} €");
            doc.Add(ukupniIznosParagraf);

```

```
    }  
  }  
  catch (Exception ex)  
  {  
    MessageBox.Show($"Greška: {ex.Message}");  
  }  
  
  doc.Close();  
  MessageBox.Show("PDF generiran.");  
}
```

*Primjer 4. Logika koja generira pdf datoteku računa*

### **3. ZAKLJUČAK**

Ovaj projekt predstavlja simulaciju maloprodajne blagajne koja omogućuje upravljanje zaposlenicima, dobavljačima, proizvodima te računima. Razvojem ovog projekta, stekao sam dragocjeno iskustvo i znanje o programiranju desktop aplikacija.

Sustavom komunikacije između različitih formi aplikacije, povezivanjem s bazom podataka te implementacijom klasa postignuta je željena funkcionalnost.

## 4. LITERATURA

Fajković, Saša Programski jezik C# & Windows Forms 2014

<https://carpediem.hr/wp-content/uploads/2021/10/C-programiranje.pdf>

(pristupljeno 15.5.2024.)

<https://stackoverflow.com/questions/5420491/c-sharp-can-i-make-an-sqlconnection-public-and-refer-to-it-from-other-forms>

(pristupljeno 15.5.2024.)

<https://www.w3schools.com/cs/index.php>

(pristupljeno 15.5.2024.)

<https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/>

(pristupljeno 15.5.2024.)

<https://www.w3schools.com/sql/default.asp>

(pristupljeno 15.5.2024.)

<https://www.w3schools.com/mysql/default.asp>

(pristupljeno 15.5.2024.)

## 5. POPIS SLIKA

Slika 1. Alati korišteni u izradi aplikacije .....	7
Slika 2. Baza podataka napravljena u MySQL .....	9
Slika 3. Prikaz prijave u sustav .....	11
Slika 4. Promjena lozinke ako je prva prijava .....	11
Primjer 1. Postavke za spajanje na bazu koje se trebaju konfigurirati u App.config .....	11
Slika 5. Sučelje za izdavanje računa .....	12
Slika 6. Izbornik administracijskih opcija .....	12
Slika 7. Sučelje za administraciju zaposlenika .....	13
Slika 8. Potvrda brisanja zaposlenika .....	13
Slika 9. Poruka o odbijenom brisanju .....	13
Slika 10. Poruka o uspješnom kreiranju zaposlenika .....	13
Slika 11. Sučelje za administraciju dobavljača .....	14
Slika 12. Poruka o uspješno dodanom dobavljaču .....	14
Slika 13. Sučelje za administraciju proizvoda .....	15
Slika 14. Poruka o automatsko naručenim proizvodima .....	15
Primjer 2. Logika koja automatski naruči proizvod .....	16
Slika 15. Sučelje za administraciju računa .....	17
Slika 16. Poruka kod storniranog računa .....	17
Slika 17. Prikaz odabranog računa u pdf datoteci .....	18
Primjer 3. Logika koja otvara generirani pdf računa .....	18
Primjer 4. Logika koja generira pdf datoteku računa .....	20



**KONZULTACIJSKI LIST IZRADE ZAVRŠNOG RADA**Ime i prezime učenika: Noa TurkRazred: 4.Program - zanimanje: Tehničar za računalstvoMentor: Mirko Jambrošić, mag. ing. inf. et comm. techn.

<b>R. br.</b>	<b>DATUM KONZULTACIJE</b>	<b>SADRŽAJ RADA</b>	<b>POTPIS MENTORA</b>
<b>1.</b>	<b>5.11.2023.</b>	Rasprava o strukturi baze podataka.	
<b>2.</b>	<b>15.12.2023.</b>	Pregled logike funkcioniranja aplikacije.	
<b>3.</b>	<b>14.2.2024.</b>	Prikaz prve verzije aplikacije i rasprava o poboljšanjima.	
<b>4.</b>	<b>16.2.2024.</b>	Rasprava i pregled elaborata.	
<b>5.</b>	<b>17.4.2024.</b>	Predaja završne verzije aplikacije. Savjeti oko izrade elaborata završnog rada.	
<b>6.</b>	<b>16.5.2024.</b>	Pregled završne verzije elaborata.	

Završni rad ocijenjen pozitivno  
DA / NE

---

(potpis mentora)