

PRIJAVA

Klikom na dugme "Prijava se" uspostavlja se veza sa bazom podataka, vade se pozicija i jmbg korisnika sa unetim imenom, prezimenom i sifrom. Nakon toga se menja status iz false u true kod korisnika sa tacnim jmbg. Trazi se pozicija korisnika i na osnovu toga se odredjuje koji ce sledeci prozor da se pojavi. U slucaju da nesto nije bilo u redu pojavljuje se poruka "Prijava nije uspeła", a u suprotnom se prikazuje poruka sa obavestenjem da je korisnik prijavljen i prikazuje mu njegovu poziciju.

```
private void prijavi_se_Click(object sender, EventArgs e) {
    try {
        using (SqlConnection konekcija = new
SqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            MySqlCommand cmd;

            cmd = new MySqlCommand("SELECT pozicija, jmbg FROM radnik WHERE sifra=@sifra AND
ime=@ime AND prezime=@prezime;", konekcija);
            cmd.Parameters.AddWithValue("@sifra", this.tbSifra.Text);
            cmd.Parameters.AddWithValue("@ime", this.tbIme.Text);
            cmd.Parameters.AddWithValue("@prezime", this.tbPrezime.Text);

            MySqlDataReader reader = cmd.ExecuteReader();

            if (reader.HasRows) {
                while (reader.Read()) {
                    pozicija = reader["pozicija"].ToString();
                    jmbg = reader["jmbg"].ToString();
                }
            }

            reader.Close();

            cmd = new MySqlCommand("UPDATE radnik SET status=true WHERE jmbg=@jmbg;", konekcija);
            cmd.Parameters.AddWithValue("@jmbg", jmbg);
            cmd.ExecuteNonQuery();

            konekcija.Close();
        }

        if (pozicija == "administrator") {
            this.Hide();
            Administracija administracija = new Administracija();
            administracija.Show();
        } else if (pozicija == "magacioner") {
            this.Hide();
            Magacin magacin = new Magacin();
            magacin.Show();
        } else if (pozicija == "radnik") {
            this.Hide();
            Sastavljanje sastavljanje = new Sastavljanje();
            sastavljanje.Show();
        }

        MessageBox.Show("Prijavljeni ste kao " + pozicija.ToUpper());
    }
    catch (Exception) {
        MessageBox.Show("Prijava nije uspeła");
    }
}
```

ADMINISTRACIJA

Klikom na dugme "Magacin" otvara se prozor u kome je omogucen pregled magacina.

```
private void magacin_Click(object sender, EventArgs e) {
    Magacin magacin = new Magacin();
    magacin.ShowDialog();
}
```

Klikom na dugme "Radnici" otvara se prozor u kome je omogucen pregled radnik.

```
private void radnici_Click(object sender, EventArgs e) {
    Radnici radnici = new Radnici();
}
```

```
radnici.ShowDialog();
}
```

Klikom na dugme “Kraj rada” uspostavlja se veza sa bazom podataka i menja se status iz true u false kod korisnika sa određenim jmbg koji je unet u formi “Prijava”.

```
private void kraj_rada_Click(object sender, EventArgs e) {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            MySqlCommand cmd = new MySqlCommand("UPDATE radnik SET status=false WHERE
            jmbg=@jmbg;", konekcija);
            cmd.Parameters.AddWithValue("@jmbg", Prijava.jmbg);
            cmd.ExecuteNonQuery();

            konekcija.Close();

            this.Close();
        }
    } catch (Exception) {
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
    }
}
```

RADNICI

Klikom na dugme “Potvrdi” vrednosti iz liste se kopiraju u polja sa desne strane.

```
private void potvrdi_Click(object sender, EventArgs e) {
    ListView.SelectedListViewItemCollection selektovano = this.spisak_radnika.SelectedItems;

    foreach (ListViewItem radnik in selektovano) {
        this.tbSifra.Text = radnik.SubItems[0].Text;
        this.tbIme.Text = radnik.SubItems[1].Text;
        this.tbPrezime.Text = radnik.SubItems[2].Text;
        this.tbJmbg.Text = radnik.SubItems[3].Text;
        this.tbPozicija.Text = radnik.SubItems[4].Text;
        this.tbStatus.Text = radnik.SubItems[5].Text;
    }
}
```

Klikom na dugme “Obrisi” uspostavlja se veza sa bazom podataka i brise se radnik sa određenim jmbg koji je unet u polje sa desne strane.

```
private void obrisi_Click(object sender, EventArgs e) {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            string jmbg = this.tbJmbg.Text;

            MySqlCommand cmd = new MySqlCommand("DELETE FROM radnik WHERE jmbg=@jmbg;",
            konekcija);
            cmd.Parameters.AddWithValue("@jmbg", jmbg);
            cmd.ExecuteNonQuery();

            konekcija.Close();

            osveziListu();

            MessageBox.Show("Uspesno obrisan radnik");
        }
    } catch (Exception) {
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
    }
}
```

Klikom na dugme "Izmeni" uspostavlja se veza sa bazom podataka koja se potom azurira izmenjenim vrednostima unetim u polja sa desne strane. Jmbg se ne menja, jer to znaci da je dodat novi radnik sa unikatnim brojem koji se u bazi podataka obelezava sa "PRIMARY KEY". "PRIMARY KEY" omogućava da se u odredjenom stolu u bazi podataka ne sme naci drugi red sa istom vrednoscu te kolone.

```
private void izmeni_Click(object sender, EventArgs e) {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            string sStatus = this.tbStatus.Text;
            int status;

            if (sStatus == "True") {
                status = 1;
            } else {
                status = 0;
            }

            MySqlCommand cmd = new MySqlCommand("UPDATE radnik SET sifra=@sifra, ime=@ime,
            prezime=@prezime, pozicija=@pozicija, status=@status WHERE jmbg=@jmbg;", konekcija);
            cmd.Parameters.AddWithValue("@sifra", this.tbSifra.Text);
            cmd.Parameters.AddWithValue("@ime", this.tbIme.Text);
            cmd.Parameters.AddWithValue("@prezime", this.tbPrezime.Text);
            cmd.Parameters.AddWithValue("@jmbg", this.tbJmbg.Text);
            cmd.Parameters.AddWithValue("@pozicija", this.tbPozicija.Text);
            cmd.Parameters.AddWithValue("@status", status);
            cmd.ExecuteNonQuery();

            konekcija.Close();

            osvezilistu();

            MessageBox.Show("Uspesna izmena podataka radnika");
        }
    } catch (Exception) {
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
    }
}
```

Klikom na dugme "Novi radnik" uspostavlja se veza sa bazom podataka i ubacuje se novi red(u ovom slucaju je to korisnik) sa vrednostima iz polja sa desne strane. <OVO SE NE PISE>status mora da bude 0 ili 1, a ne "True" ili "False" u polju odredjenom sa njegovo unosenje</OVO SE NE PISE>

```
private void novi_radnik_Click(object sender, EventArgs e) {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            MySqlCommand cmd = new MySqlCommand("INSERT INTO radnik(sifra, ime, prezime,
            jmbg, pozicija, status) VALUES(@sifra, @ime, @prezime, @jmbg, @pozicija, @status);", konekcija);

            cmd.Parameters.AddWithValue("@sifra", this.tbSifra.Text);
            cmd.Parameters.AddWithValue("@ime", this.tbIme.Text);
            cmd.Parameters.AddWithValue("@prezime", this.tbPrezime.Text);
            cmd.Parameters.AddWithValue("@jmbg", int.Parse(this.tbJmbg.Text));
            cmd.Parameters.AddWithValue("@pozicija", this.tbPozicija.Text);
            cmd.Parameters.AddWithValue("@status", this.tbStatus.Text);
            cmd.ExecuteNonQuery();

            konekcija.Close();

            osvezilistu();

            MessageBox.Show("Uspesno dodat novi radnik");
        }
    }
}
```

```

    }
    catch (Exception) {
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
    }
}

```

“osveziListu” je metoda koja omogućava aruriranje liste pred nasim ocima. U njoj se uspostavlja veza sa bazom podataka i izdvaja se sve iz stola/tejbla “radnik” i prikazuje se na listi, ali se pre toga stavke iz liste brisu. Ova metoda je pozvana i u samom otvaranju forme “Radnici”.

```

private void osveziListu() {
    try{
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            MySqlDataAdapter ada = new MySqlDataAdapter("SELECT * FROM radnik;", konekcija);
            DataTable dt = new DataTable();
            ada.Fill(dt);

            spisak_radnika.Items.Clear();

            for (int i = 0; i < dt.Rows.Count; i++) {
                DataRow dr = dt.Rows[i];
                ListViewItem radnik = new ListViewItem(dr["sifra"].ToString());
                radnik.SubItems.Add(dr["ime"].ToString());
                radnik.SubItems.Add(dr["prezime"].ToString());
                radnik.SubItems.Add(dr["jmbg"].ToString());
                radnik.SubItems.Add(dr["pozicija"].ToString());
                radnik.SubItems.Add(dr["status"].ToString());
                spisak_radnika.Items.Add(radnik);
            }

            konekcija.Close();
        }
    }
    catch (Exception) {
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
    }
}

```

MAGACIN

Klikom na dugme “Kraj rada” uspostavlja se veza sa bazom podataka i menja se status iz true u false kod korisnika sa odredenim jmbg koji je unet u formi “Prijava”.

```

private void kraj_rada_Click(object sender, EventArgs e) {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;"))
        {
            konekcija.Open();

            if (Prijava.pozicija != "administrator") {
                MySqlCommand cmd = new MySqlCommand("UPDATE radnik SET status=false WHERE
                jmbg=@jmbg;", konekcija);
                cmd.Parameters.AddWithValue("@jmbg", Prijava.jmbg);
                cmd.ExecuteNonQuery();
            }

            konekcija.Close();

            this.Close();
        }
    }
    catch (Exception) {
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
    }
}

```

```
}
```

Klikom na dugme "Potvrdi" vrednosti iz liste se kopiraju u polja sa desne strane.

```
private void potvrdi_Click(object sender, EventArgs e) {  
    ListView.SelectedListViewItemCollection selektovano = this.delovi.SelectedItems;  
  
    foreach (ListViewItem deo in selektovano) {  
        this.tbNaziv.Text = deo.SubItems[0].Text;  
        this.tbKolicina.Text = deo.SubItems[1].Text;  
        this.tbCena.Text = deo.SubItems[2].Text;  
        this.tbUkupno.Text = deo.SubItems[3].Text;  
    }  
}
```

Klikom na dugme "Izmeni" uspostavlja se veza sa bazom podataka koja se potom azurira izmenjenim vrednostima unetim u polja sa desne strane.

```
private void izmeni_Click(object sender, EventArgs e) {  
    try {  
        using (MySqlConnection konekcija = new  
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {  
            konekcija.Open();  
  
            int cena = Convert.ToInt16(this.tbCena.Text);  
            int kolicina = Convert.ToInt16(this.tbKolicina.Text);  
            int ukupno = cena * kolicina;  
  
            MySqlCommand cmd = new MySqlCommand("UPDATE delovi SET kolicina=@kolicina,  
            cena=@cena, ukupno=@ukupno WHERE naziv=@naziv;", konekcija);  
            cmd.Parameters.AddWithValue("@kolicina", this.tbKolicina.Text);  
            cmd.Parameters.AddWithValue("@cena", this.tbCena.Text);  
            cmd.Parameters.AddWithValue("@ukupno", ukupno);  
            cmd.Parameters.AddWithValue("@naziv", this.tbNaziv.Text);  
            cmd.ExecuteNonQuery();  
  
            konekcija.Close();  
  
            osveziDelove();  
  
            MessageBox.Show("Uspesna izmena podataka dela");  
        }  
    }  
    catch (Exception) {  
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");  
    }  
}
```

"osveziRacunare" je metoda koja omogućava aruriranje liste pred nasim ocima. U njoj se uspostavlja veza sa bazom podataka i izdvaja se sve iz stola/tejbla "racunari" i prikazuje se na listi, ali se pre toga stavke iz liste brisu. Ova metoda je pozvana i u samom otvaranju forme "Magacin". Racunari se ne mogu menjati za razliku od delova.

```
private void osveziRacunare() {  
    try {  
        using (MySqlConnection konekcija = new  
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {  
            konekcija.Open();  
  
            MySqlDataAdapter ada = new MySqlDataAdapter("SELECT * FROM racunari;",  
            konekcija);  
            DataTable dt = new DataTable();  
            ada.Fill(dt);  
  
            racunari.Items.Clear();  
  
            for (int i = 0; i < dt.Rows.Count; i++) {  
                DataRow dr = dt.Rows[i];
```

```

        ListViewItem racunar = new ListViewItem(dr["naziv"].ToString());
        racunar.SubItems.Add(dr["kolicina"].ToString());
        racunar.SubItems.Add(dr["cena"].ToString());
        racunar.SubItems.Add(dr["ukupno"].ToString());
        racunari.Items.Add(racunar);
    }
}

```

```

        konekcija.Close();
    }
}
catch (Exception) {
    MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
}
}

```

“osveziDelove” je metoda koja omogućava aruriranje liste pred nasim ocima. U njoj se uspostavlja veza sa bazom podataka i izdvaja se sve iz stola/tejbla “delovi” i prikazuje se na listi, ali se pre toga stavke iz liste brisu. Ova metoda je pozvana i u samom otvaranju forme “Magacin”.

```

private void osveziDelove() {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

```

```

            MySqlDataAdapter ada = new MySqlDataAdapter("SELECT * FROM delovi;", konekcija);
            DataTable dt = new DataTable();
            ada.Fill(dt);

```

```

            delovi.Items.Clear();

```

```

            for (int i = 0; i < dt.Rows.Count; i++) {
                DataRow dr = dt.Rows[i];
                ListViewItem deo = new ListViewItem(dr["naziv"].ToString());
                deo.SubItems.Add(dr["kolicina"].ToString());
                deo.SubItems.Add(dr["cena"].ToString());
                deo.SubItems.Add(dr["ukupno"].ToString());
                delovi.Items.Add(deo);
            }

```

```

        konekcija.Close();
    }
}
catch (Exception) {
    MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
}
}

```

SASTAVLJANJE

Klikom na dugme “Kraj rada” uspostavlja se veza sa bazom podataka i menja se status iz true u false kod korisnika sa odredenim jmbg koji je unet u formi “Prijava”.

```

private void kraj_rada_Click(object sender, EventArgs e) {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

```

```

            MySqlCommand cmd = new MySqlCommand("UPDATE radnik SET status=false WHERE
jmbg=@jmbg;", konekcija);
            cmd.Parameters.AddWithValue("@jmbg", Prijava.jmbg);
            cmd.ExecuteNonQuery();

```

```

            konekcija.Close();

```

```

            this.Close();
        }
    }
}

```

```

        catch (Exception) {
            MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
        }
    }
}

```

U metodi koja se poziva na klik na dugme "Sastavi" se prvo pozivaju metode "izdvojiDelove" i "izdvojiRacunare". Nakon toga se proverava da li je korisnik uneo naziv modela i broj sastavljanja u polja koja su predviđena za to. Ako korisnik nije uneo ono što se od njega očekivalo prikazace se poruka "Modela racunara nije unet" ili "Broja sastavljanja nije unet", a ako je ulaz dobar onda ce preci na glavni deo ove metode, a to je smanjivanje vrednosti kolicine delova i povecavanje vrednosti kolicine racunara. Svaki od 3 dostupna modela zahteva odredjenju kolicinu delova, tako da se cena razlikuje medju modelima. Nakon ovog dela sledi pozivanje medota "azurirajDelove" i "azurirajRacunare". Na kraju se pojavljuje poruka sa obavestenjem da je sastavljanje bilo uspesno. <<MOZE DA SE PISE, A I NE MORA>>ProgressBar je u formi ubacen cisto radi estetike.<</MOZE DA SE PISE, A I NE MORA>>

```

private void sastavi_Click(object sender, EventArgs e) {
    izdvojiDelove();
    izdvojiRacunare();
}

```

```

    if (String.IsNullOrEmpty(this.tbModel.Text)){
        MessageBox.Show("Model racunara nije unet");
    }else if (String.IsNullOrEmpty(this.tbBrojSastavljanja.Text)) {
        MessageBox.Show("Broj sastavljanja nije unet");
    }else {
        repeat = Convert.ToInt16(this.tbBrojSastavljanja.Text);
    }
}

```

```

    if (naziv == "pc_lq") {
        ram -= 2 * repeat;
        cpu -= repeat;
        hdd -= 250 * repeat;
        cooler -= repeat;
    } else if (naziv == "pc_mq") {
        ram -= 4 * repeat;
        cpu -= 2 * repeat;
        hdd -= 500 * repeat;
        cooler -= 2 * repeat;
    }
    else if (naziv == "pc_hq") {
        ram -= 8 * repeat;
        cpu -= 4 * repeat;
        hdd -= 1000 * repeat;
        cooler -= 3 * repeat;
    }
}

```

```

    uRam = cRam * ram;
    uCpu = cCpu * cpu;
    uHdd = cHdd * hdd;
    uCooler = cCooler * cooler;
}

```

```

    kolicina += repeat;
    ukupno = cena * kolicina;
}

```

```

    azurirajRacunare();
    azurirajDelove();
}

```

```

    MessageBox.Show("Uspesno sastavljanje");
}
}

```

U metodi "izdvojiRacunare" se uspostavlja veza sa bazom podataka. Nakon toga se izdavaju vrednosti kolicine i cene modela racunara koji je unet u polje koje je predviđeno za to.

```

private void izdvojiRacunare() {
    try {
        using (MySqlConnection konekcija = new
        MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {

```

```

        konekcija.Open();

        naziv = tbModel.Text;

        MySqlCommand cmd = new MySqlCommand("SELECT kolicina, cena FROM racunari WHERE naziv=@naziv;", konekcija);
        cmd.Parameters.AddWithValue("@naziv", naziv);

        MySqlDataReader reader = cmd.ExecuteReader();

        if (reader.HasRows) {
            while (reader.Read()) {
                kolicina = Convert.ToInt16(reader["kolicina"].ToString());
                cena = Convert.ToInt16(reader["cena"].ToString());
            }
        }

        reader.Close();

        konekcija.Close();
    }
}
catch (Exception) {
    MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
}
}

```

U metodi "izdvojiDelove" se uspostavlja veza sa bazom podataka. Nakon toga se izdavaju vrednosti kolicine i cene svakog dela pojednicano.

```

private void izdvojiDelove() {
    try {
        using (MySqlConnection konekcija = new MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            MySqlCommand cmd = new MySqlCommand("SELECT kolicina, cena FROM delovi WHERE naziv=@naziv;", konekcija);
            cmd.Parameters.AddWithValue("@naziv", "ram");

            MySqlDataReader reader = cmd.ExecuteReader();

            if (reader.HasRows) {
                while (reader.Read()) {
                    ram = Convert.ToInt16(reader["kolicina"].ToString());
                    cRam = Convert.ToInt16(reader["cena"].ToString());
                }
            }

            reader.Close();

            cmd = new MySqlCommand("SELECT kolicina, cena FROM delovi WHERE naziv=@naziv;", konekcija);
            cmd.Parameters.AddWithValue("@naziv", "cpu");
            reader = cmd.ExecuteReader();

            if (reader.HasRows) {
                while (reader.Read()) {
                    cpu = Convert.ToInt16(reader["kolicina"].ToString());
                    cCpu = Convert.ToInt16(reader["cena"].ToString());
                }
            }

            reader.Close();

            cmd = new MySqlCommand("SELECT kolicina, cena FROM delovi WHERE naziv=@naziv;", konekcija);
            cmd.Parameters.AddWithValue("@naziv", "hdd");
            reader = cmd.ExecuteReader();

```



```

        if (reader.HasRows) {
            while (reader.Read()) {
                hdd = Convert.ToInt16(reader["kolicina"].ToString());
                cHdd = Convert.ToInt16(reader["cena"].ToString());
            }
        }

        reader.Close();

        cmd = new MySqlCommand("SELECT kolicina, cena FROM delovi WHERE naziv=@naziv;",
konekcija);
        cmd.Parameters.AddWithValue("@naziv", "cooler");
        reader = cmd.ExecuteReader();

        if (reader.HasRows) {
            while (reader.Read()) {
                cooler = Convert.ToInt16(reader["kolicina"].ToString());
                cCooler = Convert.ToInt16(reader["cena"].ToString());
            }
        }

        reader.Close();

        konekcija.Close();
    }
}
catch (Exception) {
    MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
}
}

```

U metodi “azurirajRacunare” se uspostavlja veza sa bazom podataka i vrednosti iz stola/tejbila “racunari” se azuriraju.

```

private void azurirajRacunare() {
    try {
        using (MySqlConnection konekcija = new
MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {
            konekcija.Open();

            MySqlCommand cmd = new MySqlCommand("UPDATE racunari SET kolicina=@kolicina,
ukupno=@ukupno WHERE naziv=@naziv;", konekcija);
            cmd.Parameters.AddWithValue("@naziv", this.tbModel.Text);
            cmd.Parameters.AddWithValue("@kolicina", kolicina);
            cmd.Parameters.AddWithValue("@ukupno", ukupno);
            cmd.ExecuteNonQuery();

            konekcija.Close();
        }
    }
    catch (Exception) {
        MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
    }
}

```

U metodi “azurirajDelove” se uspostavlja veza sa bazom podataka i vrednosti iz stola/tejbila “delovi” se azuriraju.

```

private void azurirajDelove() {
    try {
        using (MySqlConnection konekcija = new
MySqlConnection("server=localhost;database=racunari;uid=root;pwd=root123;")) {

            MySqlCommand cmd;

            konekcija.Open();

```

```
WHERE naziv=@naziv;", konekcija);
cmd = new MySqlCommand("UPDATE delovi SET kolicina=@kolicina, ukupno=@ukupno
cmd.Parameters.AddWithValue("@naziv", "ram");
cmd.Parameters.AddWithValue("@kolicina", ram);
cmd.Parameters.AddWithValue("@ukupno", uRam);
cmd.ExecuteNonQuery();
```

```
WHERE naziv=@naziv;", konekcija);
cmd = new MySqlCommand("UPDATE delovi SET kolicina=@kolicina, ukupno=@ukupno
cmd.Parameters.AddWithValue("@naziv", "cpu");
cmd.Parameters.AddWithValue("@kolicina", cpu);
cmd.Parameters.AddWithValue("@ukupno", uCpu);
cmd.ExecuteNonQuery();
```

```
WHERE naziv=@naziv;", konekcija);
cmd = new MySqlCommand("UPDATE delovi SET kolicina=@kolicina, ukupno=@ukupno
cmd.Parameters.AddWithValue("@naziv", "hdd");
cmd.Parameters.AddWithValue("@kolicina", hdd);
cmd.Parameters.AddWithValue("@ukupno", uHdd);
cmd.ExecuteNonQuery();
```

```
WHERE naziv=@naziv;", konekcija);
cmd = new MySqlCommand("UPDATE delovi SET kolicina=@kolicina, ukupno=@ukupno
cmd.Parameters.AddWithValue("@naziv", "cooler");
cmd.Parameters.AddWithValue("@kolicina", cooler);
cmd.Parameters.AddWithValue("@ukupno", uCooler);
cmd.ExecuteNonQuery();
```

```
konekcija.Close();
}
}
catch (Exception)
{
    MessageBox.Show("Veza sa bazom podataka nije uspostavljena");
}
}
```