Izrada stolne aplikacije za izračun režija u jeziku C#

Vukosavić, Danijel

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:954519

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-24**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

Danijel Vukosavić

Izrada stolne aplikacije za izračun režija u jeziku C#

Završni rad

Pula, rujan, 2019.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

Danijel Vukosavić

Izrada stolne aplikacije za izračun režija u jeziku C#

Završni rad

JMBAG: 0303069617, redoviti student
Studijski smjer: Sveučilišni preddiplomski studij Informatike

Predmet: Programsko Inženjerstvo
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija
Mentor: doc. dr. sc. Tihomir Orehovački

Pula, rujan, 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Danijel Vukosavič, kandidat za prvostupnika informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, rujan 2019. godine

IZJAVA
o korištenju autorskog djela

Ja, Danijel Vukosavić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom „Izrada stolne aplikacije za izračun režija u jeziku C#“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan, 2019.godine

Potpis

_____

SAŽETAK

Aplikacija MojeRežije služi za izračun režija. Režije koje se mogu izračunati u aplikaciji su struja, voda i plin za određene gradove. Korisnik unosi vrijednosti, te mu se ispisuje koliki iznos bi trebao platiti za određene režije, te iste može spremiti u bazu podataka i kasnije vidjeti u aplikaciji ili izvesti na svoje računalo. Aplikacija nudi računanje režija po godini. U aplikaciji postoji još i ocjenjivanje same aplikacije gdje korisnik odabere ocjenu i napiše svoju recenziju. Za izradu same aplikacije korištena je platforma Visual Studio i jezik C#.

Ključne Riječi : MojeRežije , Režije, Računanje Režija, Visual Studio , C#

ABSTRACT

The MojeRežije application is used for computing overheads. Utilites that can be computed are Electricity, Water, Gas, for specific cities. The user enters the values, gets what the bill should be for certain overhead, which he can save to the database and later see in the application or export to his computer. The app offers computation of overhead per year. The app offers computation of overhead per year. To build the application myself, I used the Visual Studio platform and the C # language.

Keywords: MojeRežije, Utilites, Computing Overheads , Visual Studio, C #

SADRŽAJ

1.Uvod 1
2.Dijagrami 2
2.1 Dijagram slučaja 2
2.2 Dijagram slijeda 3
2.3 Klasni dijagram 4
3.Forme 5
3.1 Forma za registraciju 6
3.2 Forma izbornik 9
3.3 Forma za računanje vode 10
3.4 Forma za računanje struje 14
3.5 Forma za računanje plina 16
4. Forme za ostale režije 18
4.1 Forma za internet usluge 18
4.2 Forma za mobilne pretplate 19
4.3 Forma za unos odvoza smeća 20
4.4 Forma za ispis potrošnji 20
4.4.1 Forma za ispis potrošnje vode 21
4.4.2 Forma za ispis potrošnje struje 23
4.4.3 Forma za ispis potrošnje plina 24
4.4.4 Forma za ispis Internet računa, mobilne pretplate i odvoz smeća 25
4.5 Forma za recenziju aplikacije 27
4.6 Forma za administratora 29
Zaključak 32
Literatura 33
Popis slika 34

1. Uvod

MojeRežije je aplikacija za računanje režija kao što su voda, plin i struja, ali postoji i mogućnost spremanja ostalih režija kao što su : Internet, mobilne pretplate i komunalne naknade. Osim što se režije mogu računati, iste je moguće i spremiti u bazu podataka ili izvesti na osobno računalo u obliku xlsx formata.

Prednost ove aplikacije u odnosu na druge aplikacije je ta što se sve režije mogu izračunati i spremiti na jednom mjestu bez da se ide na službene stranice poduzeća koje nude usluge. Spremljene režije jedino može vidjeti onaj korisnik koji ih je spremio. Prilikom provjere spremljenih režija postoji i mogućnost računanja režija po godini kao i samo brisanje režija.

Korisnik je obvezan napraviti korisnički račun kako bih mogao koristiti aplikaciju. Korisnik također ima mogućnost ocjenjivanja same aplikacije ukoliko on to poželi. Potrebno je dati ocjenu od 1 do 5 i napisati recenziju koja se sprema u bazu podataka.

Administrator jedini ima mogućnost provjere svih ocjena i recenzija. Administrator također ima i ispis prosječne ocjene aplikacije kao i graf koji mu prikazuje koliko je puta unesena neka ocjena. Razlog zbog kojeg korisnik ima mogućnost ocjenjivanja aplikacije je ta da bi se aplikacija u budućnosti mogla poboljšati i tako pridonijeti najbolje korisničko iskustvo. Osim ispisa režija postoji i mogućnost brisanja tih istih režija tako da klikne na onu spremljenu režiju koju želi obrisati.

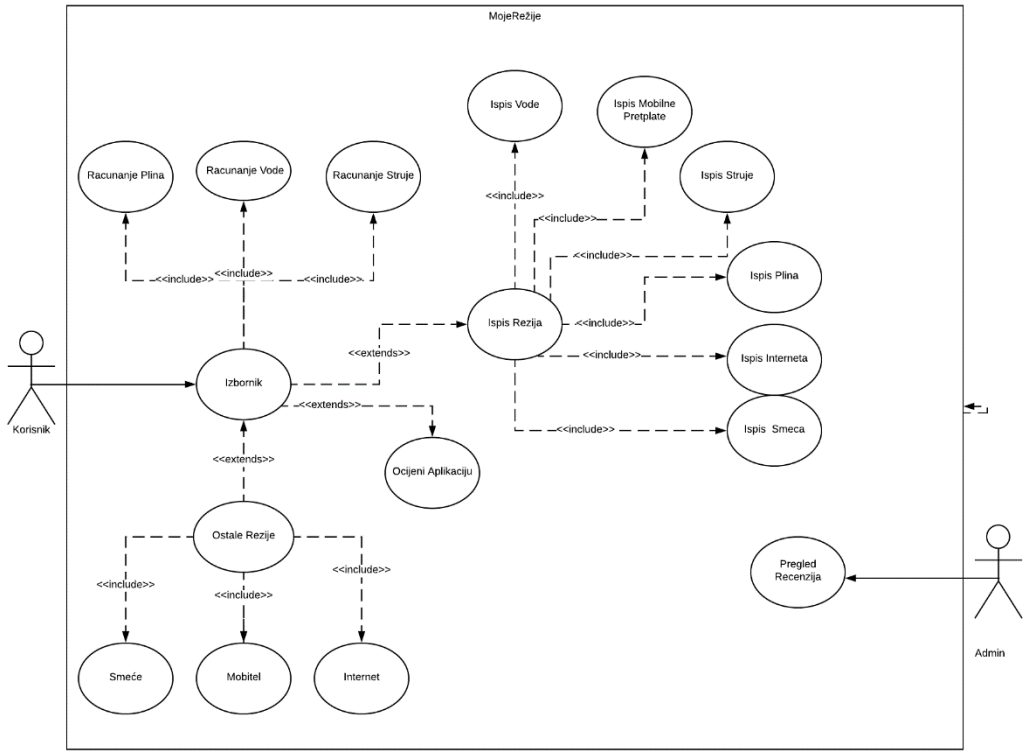
Aplikacija još ima i restrikcije koje onemogućuju korisniku unos ne pravilnih email formata. Postoji još i restrikcija koja korisnika upozorava na ispunjavanje svih polja koje su potrebne da bih se određena režija mogla spremiti i izračunati. Sve te restrikcije su ispisane na ekran da upozore korisnika ukoliko dođe do pogreške.

Režije poput interneta, mobilne pretplate i komunalne naknade jedino je moguće spremiti i izvesti zato što su one pretežito fiksne pa se ne moraju računat.

2. Dijagrami

2.1 Dijagram slučaja

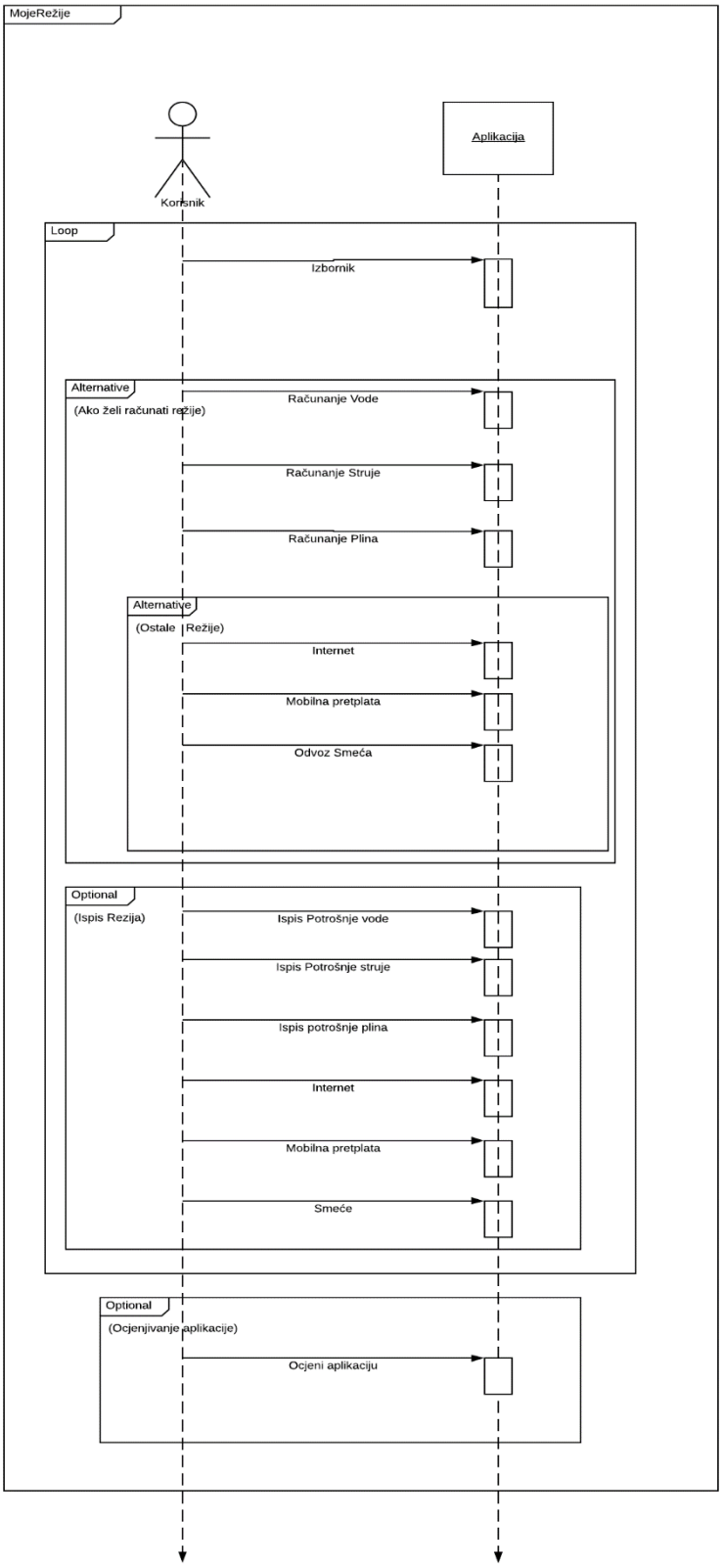
Dijagram slučaja sadrži dva aktera. Korisnika i Administratora. Administrator ima samo jednu mogućnost, a to je provjera pregled recenzija. Korisnik ima puno više mogućnosti kao što su računanje ili unos samih režija, te pregled režija i ocjenjivanje aplikacije. Ako korisnik odabere ispis režija on vidi sve režije koje je spremio u bazu podataka, te ih može izbrisati ili pogledati potrošnju za godinu koju odabere. Ukoliko se korisnik odluči za računanje režija iste može pohraniti u bazu podataka ili spremiti na osobno računalo. Slika 1 prikazuje interakciju korisnika i administratora s aplikacijom.



Slika 1. Dijagram slučaja

2.2 Dijagram slijeda

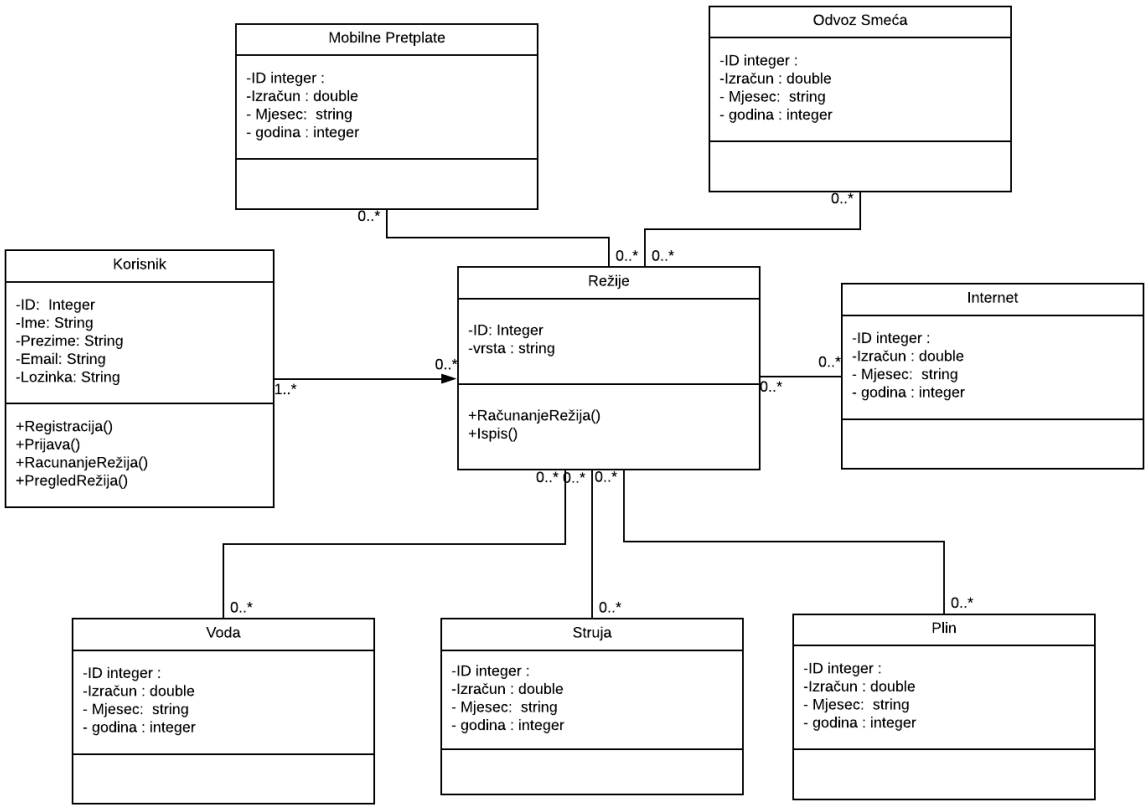
Ovaj dijagram prikazuje samo korisnika. Prilikom same prijave korisnik ulazi na izbornik , te onda bira što želi sljedeće. Ukoliko korisnik odabere ostale režije tu može spremiti režije kao što su Internet, mobitel i odvoz smeća. Ukoliko se odluči na ispis onda može provjeriti režije koje je pohranio, te iste izbrisati i za kraj može ocijeniti aplikaciju. Slika 2 prikazuje ilustraciju dijagrama slijeda.



Slika 2. Dijagram slijeda

2.3 Klasni dijagram

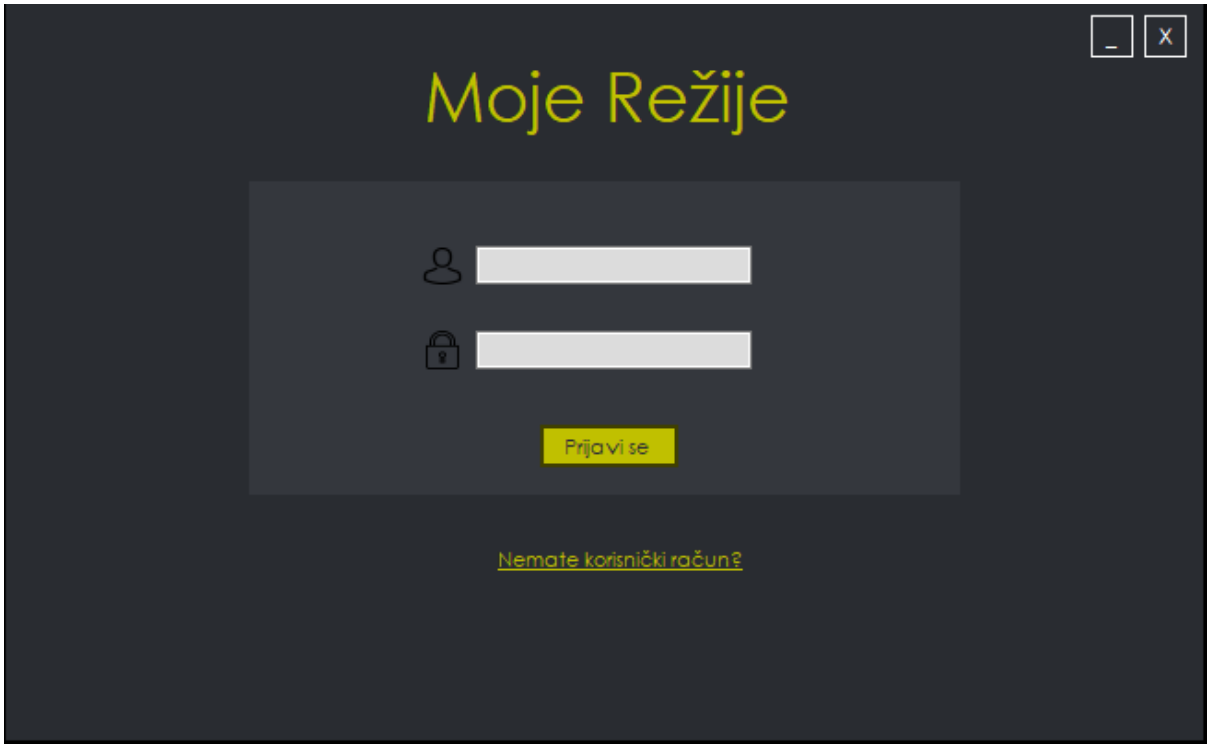
Klasni dijagram prikazuje samu strukturu sustava. Klasni dijagram objašnjava klase unutar sustava , njihove atribute i funkcije. Također postoje i odnosi između klasa. Npr. Ukoliko izbrišemo klasu režije brišu se i sve ostale klase osim klase korisnik. Klase imaju i svoje kardinalnosti. Primjer kardinalnosti je ta da korisnik može unijeti jednu ili više režija , dok režije mogu unijeti nula ili više korisnika. Slika 3 prikazuje klase i odnose između klasa.



Slika 3. Klasni dijagram

3. FORME

Početna forma u aplikaciji je forma prijave (Slika 4) gdje se korisnik treba prijaviti da bi mogao nastaviti dalje raditi u aplikaciji. Na formi imamo dva tekstualna okvira u kojoj korisnik mora unijeti svoj email i lozinku, te onda kliknuti na tipku „Prijavi se“ da bi se prijavio. Kada korisnik klikne na tipku prijava aplikacija prvo provjerava da li se nalazi korisnik s unesenim emailom i lozinkom u bazi, ukoliko se ne nalazi pojavljuje se poruka u kojoj piše da email ili lozinka nisu točni. Ukoliko se korisnik nalazi u bazi ispisat će se poruka da se uspješno prijavio, te aplikacija prelazi na formu izbornik. Treba još napomenuti da se na formi nalazi i tekst link na kojeg korisnik treba kliknut ukoliko nema račun da ga se odvede na formu registracije, te se također u gornjem desnom kutu nalaze tipke za gašenje i minimiziranje same aplikacije.

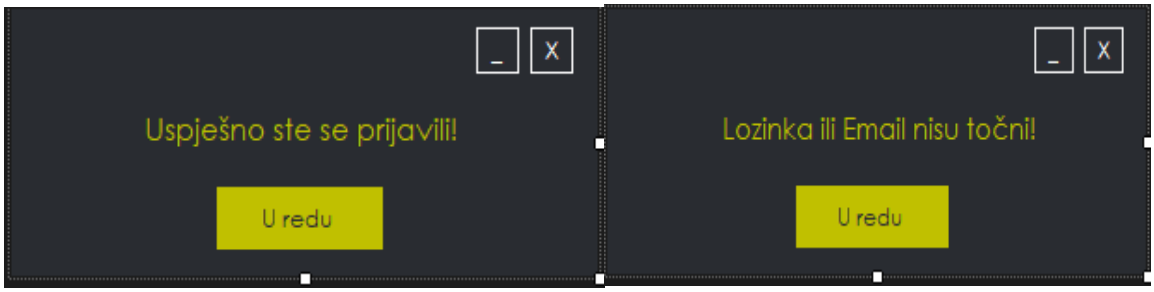


Slika 4. Forma za prijavu

Ovaj dio koda provjerava da li se korisnik s unesenim emailom i lozinkom nalazi u bazi, te ukoliko se nalazi poruka s uspješnom prijavom se pojavljuje na ekranu i tada aplikacija prelazi na formu izbornik, ukoliko se podaci ne nalaze u bazi podataka tada se pojavljuje poruka s ne uspješnom prijavom. Sam prikaz koda nalazi se na Slici 5. dok poruke za uspješnu i ne uspješnu prijavu nalaze se na slici 6.

```
else
{
    string provjera = "SELECT Email , Lozinka FROM Registracija  WHERE Email = '" + tb_Email.Text + "' AND Lozinka = '" + tb_Lozinka.Text + "';";

    SqlCommand nadji = new SqlCommand(provjera, con);
    SqlDataReader pronadji = nadji.ExecuteReader();
    if (pronadji.Read())
    {

        PrijavaKorisnika.ImePrijavljenog = (string)pronadji["Email"];
        this.Hide();
        Form Izbornik = new Izbornik();
        Izbornik.Show();
        Form UspjesnoPrijavljen = new UspjesnoPrijavljen();
        UspjesnoPrijavljen.Show();


    }
    else
    {
        Form LozEmailNetocni = new LozEmailNetocni();
        LozEmailNetocni.Show();
    }
}
con.Close();
```

Slika 5. Dio koda koji provjerava autentifikaciju korisnika



Slika 6. Poruka za uspješnu i ne uspješnu prijavu

3.1 Forma za registraciju

Na formi registracije nalaze se vise tekstualnih okvira koje korisnik mora točno popunit da bi se mogao uspješno registrirati (Slika 7). Ukoliko korisnikov email nije u točnom formatu pojavit će mu se poruka da taj email nije ispravan, te da ga ispravi, a

ukoliko se lozinke ne podudaraju pojavit će se poruka da se lozinke ne podudaraju. Kada korisnik unese sve točne podatke treba stisnuti na tipku „Registriraj se“ , te se onda pojavljuje poruka da se uspješno registrirao. Nakon uspješne registracije korisnikovi podatci se spremaju u bazu, te ga vodi na formu prijave. Na formi se još nalazi i tekstualni link koji vodi na prijavu ukoliko korisnik već ima korisnički račun. Na slici 8. su prikazane poruke za uspješnu registraciju, ali i poruku ukoliko korisnik ne upiše točne lozinke, dok slika 9 prikazuje poruku za postojeći email i poruku koja govori da se moraju popuniti sva polja.

Slika 7. Forma za registraciju

Slika 8. Poruka za provjeru lozinki i poruka za uspješnu registraciju

Slika 9. Poruka za postojeći email i prazna polja

U prvom dijelu koda aplikacija uzima sve email-ove, te ih provjerava s onim email-om koje je upisan u polje. Ukoliko korisnik s istim emailom postoji već unutar baze podataka izbacit će se poruka koja javlja da email već postoji. Aplikacija provjerava da li je korisnik popunio sva polja , te da li se lozinke podudaraju. Tek onda kada je korisnik upisao sve validne podatke obavlja se spremanje u bazu podataka. Na slici 10 nalazi se sam kod za registraciju.

```
SqlCommand Uzmi = new SqlCommand("SELECT COUNT(*) FROM [Registracija] WHERE Email=@email", con);
Uzmi.Parameters.AddWithValue("@email", tb_EmailReg.Text);
object R = Uzmi.ExecuteScalar();
int Registracija = Convert.ToInt32(R);

if (tb_ImeReg.Text == "" || tb_PrezimeReg.Text == "" || tb_EmailReg.Text == "" || tb_LozinkaReg.Text == "" || tb_PotvrdaLoz.Text == "")
{
    Form PopunitePolja = new PopunitePolja();
    PopunitePolja.Show();

}


else if(tb_LozinkaReg.Text != tb_PotvrdaLoz.Text)
 {

       Form LozinkePodudarati = new LozinkePodudarati();
       LozinkePodudarati.Show();


 }

else if (Registracija > 0)
{
   Form EmailPostoji = new EmailPostoji();
   EmailPostoji.Show();
}
else
     {
          string query = "INSERT INTO Registracija(Ime, Prezime, Email, Lozinka) VALUES " +
                         "('" + tb_ImeReg.Text + "', '" + tb_PrezimeReg.Text + "', '" + tb_EmailReg.Text + "', '" +
                         tb_LozinkaReg.Text + "')";
          SqlCommand sqlcomm = new SqlCommand(query, con);
          sqlcomm.ExecuteNonQuery();
          this.Hide();
    Form Prijava = new Prijava();
    Prijava.Show();
    Form UspjesnoRegistriran = new UspjesnoRegistriran();
    UspjesnoRegistriran.Show();
```

Slika 10. Kod koji radi provjeru za uspješnu registraciju

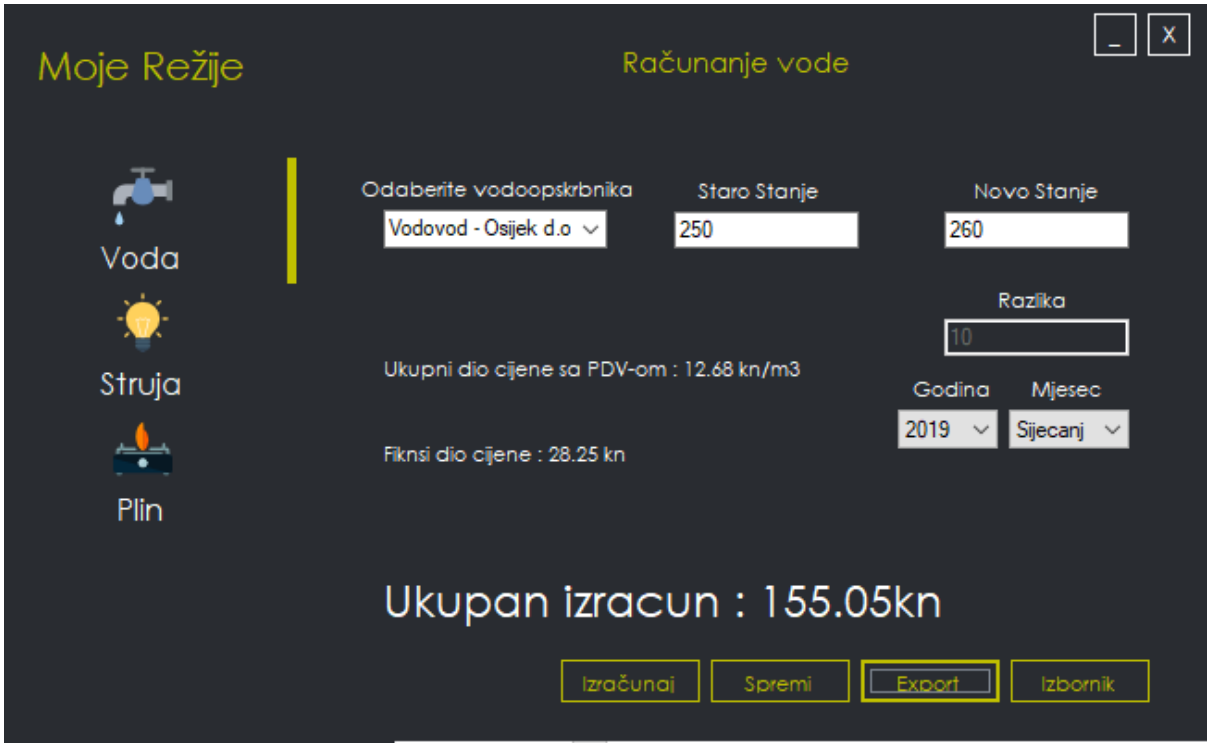
3.2 Forma izbornik

Na sljedećoj formi se nalaze pet opcija koje korisnik može odabrat, te opcije su : voda, struja, plin, ispis, ocjenjivanje aplikacije što je prikazano na slici 11 Svaka od ovih opcija vodi na sljedeću formu. Na ovoj formi treba još napomenut da se nalazi opcija za odjavu koja korisnika odjavi, te ga odvede na formu prijave, te još postoji i tekst u kojem piše koji korisnik je prijavljen.

Slika 11. Forma za izbornik

3.3 Forma za računanje vode

Forma za računanje vode sadrži polja koje se moraju ispuniti da bi se dobio rezultat. Ukoliko korisnik proba izračunati podatak, a da nije ispunio sva polja pojavit će se poruka da mora popuniti sva polja ili ako se unese da je staro stanje veće od novog onda će se također pojavit poruka da to nije moguće. Korisnik također mora odabrati jednu od tri ponuđene vodoopskrbe. Kada se popune sva polja onda je potrebno pritisnuti tipku „Izračunaj“ . Kada se klikne na tipku izračunaj ispišu se podatci kao što su : fiksni dio cijene i ukupan udio cijene, te pri samom dnu i sama ukupna cijena potrošnje. Korisnik još može i birati da unese za koju godinu i mjesec je taj račun , te ima mogućnost i izvesti svoj račun na svoje računalo ili spremiti u bazu podataka aplikacije. Treba napomenuti da korisnik može spremiti samo jedan račun u mjesecu i godini, ako pokuša spremiti dva puta račun u istom mjesecu i godini onda će se pojaviti poruka da to ne smije, te će ga se odvesti na formu za ispis gdje prvo treba obrisati postojeći račun. Na formi se još nalaze i opcije da se može otići na formu za računanje plina i struje ili da ode nazad na izbornik. Slika 12 prikazuje izgled forme sa ispunjenim vrijednostima.



Slika 12. Forma za računanje potrošnje vode.

Ovdje se odvija provjera da li je novo stanje manje od starog. Nakon što su se upisali valjajući podaci onda se provjerava koji vodovod je odabran. Svaki vodovod ima drugačiji sustav računanja potrošnje. Na slici 13 je prikazan kod za računanje i provjeru stanja. Vrijednosti potrošnje su preuzete sa službenih stranica vodovoda [4][5][6].

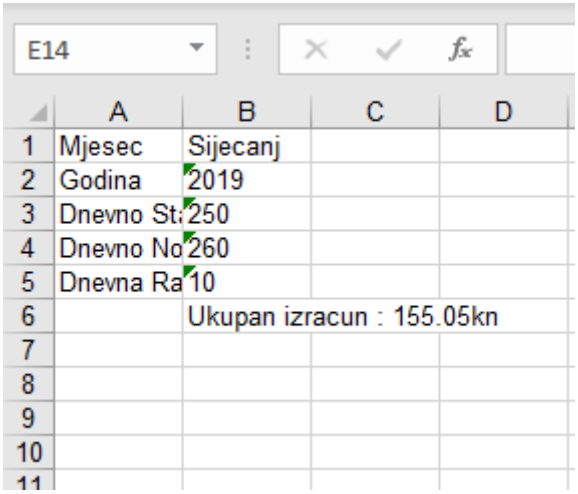
```
if (Convert.ToInt32(tb_novo.Text) < Convert.ToInt32(tb_Staro.Text))
{
    Form NovoStanje = new NovoStanje();
    NovoStanje.Show();
}
else
{
    switch (cb_Voda.SelectedItem.ToString())
    {
        case "Vodovod - Osijek d.o.o":
            tb_RazlikaVoda.Text = (Convert.ToInt32(tb_novo.Text) - Convert.ToInt32(tb_Staro.Text)).ToString();
            lbl_VodaIzracun.Text = "Ukupan izracun : " + Math.Round((Convert.ToDouble(tb_RazlikaVoda.Text) * 12.68 + 28.25), 2).ToString() + "kn";
            Podatak = Math.Round((Convert.ToDouble(tb_RazlikaVoda.Text) * 12.68 + 28.25), 2);
            lbl_VodaIzracun.Visible = Enabled;
            lbl_CijenaPDV.Text = "Ukupni dio cijene sa PDV-om : 12.68 kn/m3";
            lbl_CijenaPDV.Visible = Enabled;
            lbl_FiksniDio.Text = "Fiknsi dio cijene : 28.25 kn";
            lbl_FiksniDio.Visible = Enabled;
            break;

        case "Vodovod i Kanalizacija d.o.o Rijeka":
            tb_RazlikaVoda.Text = (Convert.ToInt32(tb_novo.Text) - Convert.ToInt32(tb_Staro.Text)).ToString();
            lbl_VodaIzracun.Text = "Ukupan izracun : " + Math.Round((Convert.ToDouble(tb_RazlikaVoda.Text) * 12.7157 + 19.20), 2).ToString() + "kn";
            Podatak = Math.Round((Convert.ToDouble(tb_RazlikaVoda.Text) * 12.7157 + 19.20), 2);
            lbl_VodaIzracun.Visible = Enabled;
            lbl_CijenaPDV.Text = "Ukupni dio cijene sa PDV-om : 12.7157 kn/m3";
            lbl_CijenaPDV.Visible = Enabled;
            lbl_FiksniDio.Text = "Fiknsi dio cijene : 19.20 kn";
            lbl_FiksniDio.Visible = Enabled;
            break;

        case "Vodoopskrba i odvodnja d.o.o Zagreb":
            tb_RazlikaVoda.Text = (Convert.ToInt32(tb_novo.Text) - Convert.ToInt32(tb_Staro.Text)).ToString();
            lbl_VodaIzracun.Text = "Ukupan izracun : " + Math.Round((Convert.ToDouble(tb_RazlikaVoda.Text) * 15.2887 + 18.92), 2).ToString() + "kn";
            Podatak = Math.Round((Convert.ToDouble(tb_RazlikaVoda.Text) * 15.2887 + 18.92), 2);
            lbl_VodaIzracun.Visible = Enabled;
            lbl_CijenaPDV.Text = "Ukupni dio cijene sa PDV-om : 15.2887 kn/m3";
            lbl_CijenaPDV.Visible = Enabled;
            lbl_FiksniDio.Text = "Fiknsi dio cijene : 18.92 kn";
            lbl_FiksniDio.Visible = Enabled;
            break;

    }


}
```

Slika 13. Kod za računanje vode i provjeru stanja

Dio koda koji prikazuje spremanje podatka na radnu površinu računala u obliku xlsx formata koji nalazi se na slici 14 Kod sprema podatke u određene ćelije. Na slici 15 je prikazan ispis unutar programa Microsoft Excel [7].

```
private void Btn_Export_Click(object sender, EventArgs e)
{
    try
    {


        WorkBook ExcelPodatak = WorkBook.Create(ExcelFileFormat.XLS);
        ExcelPodatak.Metadata.Author = "IronXL";
        WorkSheet SpremiPodatak = ExcelPodatak.CreateWorkSheet("new_shee
        SpremiPodatak["A1"].Value = "Mjesec";
        SpremiPodatak["B1"].Value = cb_VMjesec.SelectedItem.ToString();
        SpremiPodatak["A2"].Value = "Godina";
        SpremiPodatak["B2"].Value = cb_Godina.SelectedItem.ToString();
        SpremiPodatak["A3"].Value = "Dnevno Staro Stanje";
        SpremiPodatak["B3"].Value = tb_Staro.Text;
        SpremiPodatak["A4"].Value = "Dnevno Novo Stanje";
        SpremiPodatak["B4"].Value = tb_novo.Text;
        SpremiPodatak["A5"].Value = "Dnevna Razlika";
        SpremiPodatak["B5"].Value = tb_RazlikaVoda.Text;
        SpremiPodatak["B6"].Value = lbl_VodaIzracun.Text;
        ExcelPodatak.SaveAs(@"C:\Users\Dado\Desktop\Voda.xls");
        Form DatotekaSpremljena = new DatotekaSpremljena();
        DatotekaSpremljena.Show();
    }
    catch
    {
        MessageBox.Show("Zatvorite postojecu datoteku prvo");
    }
```

Slika 14. Kod za izvoz xlsx formata na radnu površinu

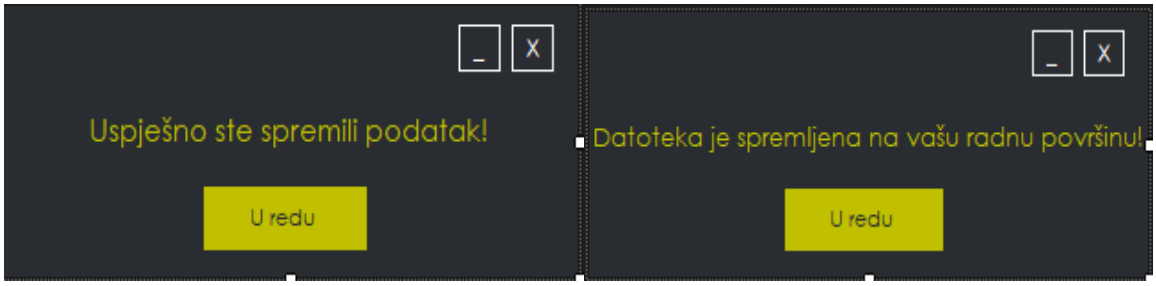


Slika 15. Izvoz potrošnje vode u xlsx formatu na radnu površinu

Ovaj dio koda provjerava da li se u bazi podataka nalazi izračun s istom godinom i mjesecom, te ukoliko se nalazi pojavljuje se poruka da nije moguće unijeti podatak. Kod isto tako provjerava da li je korisnik izračunao podataka, te ako nije da ga izračuna iz razloga zato što nije moguće spremiti praznu vrijednost u bazu podataka. Na samom kraju koda se odvija spremanje podatka u bazu. Kod je prikazan na slici 16, dok su na slici 17. prikazane poruke za uspješno spremanje podatka i spremanje podatka na radnu površinu.

```
bool flag = true;
var con = DataBase.GetSqlkonekcija();
con.Open();


for (int i = 0; i < DT.Rows.Count; i++)
{

    if (PrijavaKorisnika.ImePrijavljenog == DT.Rows[i][0].ToString() && cb_VMjesec.SelectedItem.ToString() == DT.Rows[i][1].ToString() && cb_Godina.SelectedItem.ToString() == DT.Rows[i][2].ToString())
    {
        Form IstiIzracun = new IstiIzracun();
        IstiIzracun.Show();
        flag = false;
    }
}
    if (tb_RazlikaVoda.Text == "")
    {
    MessageBox.Show("Prvo izracunajte");

    }
    else if (flag == true)
{
    string Unos = "INSERT INTO Voda(Email, Izracun, Mjesec, Godina) VALUES " +
                  "('" + PrijavaKorisnika.ImePrijavljenog + "', '" + Podatak + "', '" + cb_VMjesec.Text + "','" +
                    cb_Godina.Text + "')";

    SqlCommand Spremi = new SqlCommand(Unos, con);
    Spremi.ExecuteNonQuery();
    Form UspjesnoSpremljeno = new UspjesnoSpremljeno();
    UspjesnoSpremljeno.Show();
    RefreshDataTable();
    Spremi.Dispose();
}
```

Slika 16. Kod za spremanje izračuna u bazu podataka



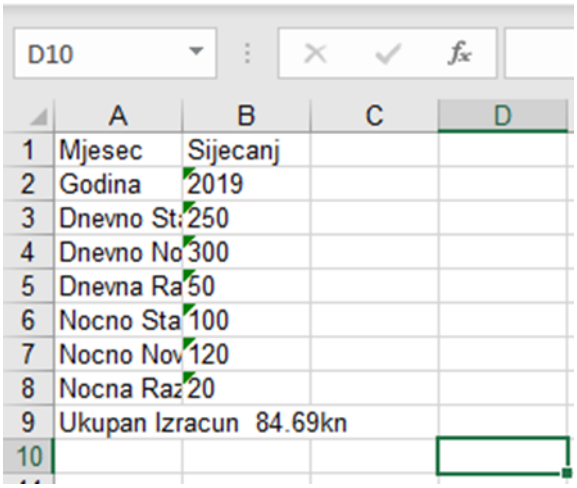
Slika 17. Poruke za uspješno spremanje podatka

.

3.4 Forma za računanje struje

Na ovoj formi potrebno je kao i u prethodnoj unijeti potrošnju prošlog i sadašnjeg mjeseca da bi se mogao dobiti izračun (Slika 18) . Ukoliko se ne ispune podatci, ne može se izračunati unesenu vrijednost. Kao i na prethodnoj formi moguće je spremiti podatak na osobno računalo (Slika 19) ili u bazu podataka od same aplikacije. Da bi se podatci uspješno izračunali i spremili potrebno je unijeti podatke za dnevnu i noćnu potrošnju struje. Kod za provjeru i računanje je sličan onome za računanje vode samo s različitim vrijednostima Kao i na prethodnoj formi ukoliko ne unese neki podatak pojavit će se poruka da to napravi. Podaci za računanje su preuzeti sa službenih stranica [1].

Slika 18. Forma za računanje potrošnje struje

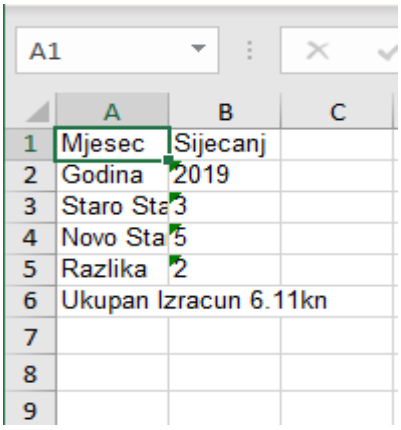


Slika 19. Izvoz potrošnje struje u xlsx formatu na radnu površinu

3.5 Forma za računanje plina

Zadnja forma za računanje je forma za računanje plina (Slika 20). Kao i dvije prethodne forme ima sve iste karakteristike kao što su da ne može izračunati vrijednosti ukoliko nisu popunjena sva polja ili da se ne može spremiti isti podatak u istoj godini i istom mjesecu. Potrebno je odabrati plinaru i tarifni model, kod za izračun je također sličan prethodnim samo s drugačijim formulama i vrijednostima. Na slici 21. je prikazan ispis izračuna u programu microsoft excel. Sve vrijednosti za izračun su preuzete sa službenih stranica plinara [2.][3.].

Slika 20. Forma za računanje potrošnje plina



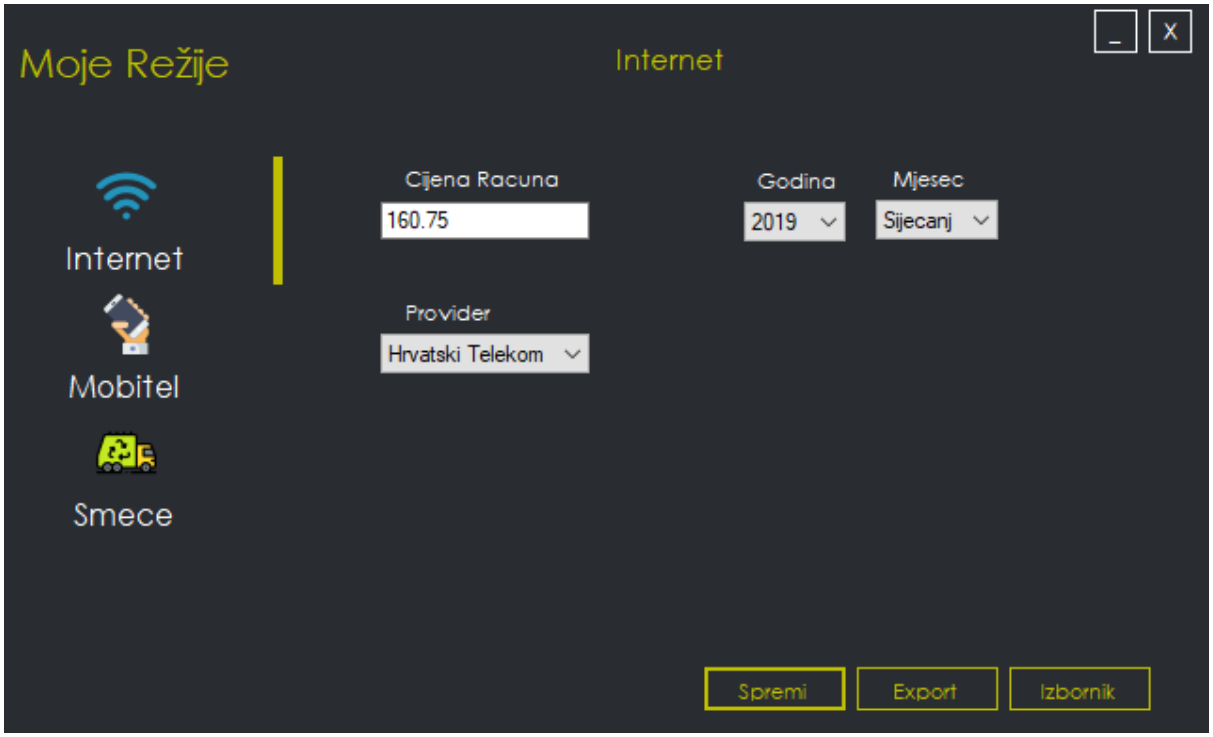
	A	B	C
1	Mjesec	Sijecanj	
2	Godina	2019	
3	Staro Sta	3	
4	Novo Sta	5	
5	Razlika	2	
6	Ukupan Izracun 6.11kn		
7			
8			
9			

Slika 21. Izvoz potrošnje struje u xlsx formatu na radnu površinu

4. Forme za ostale režije

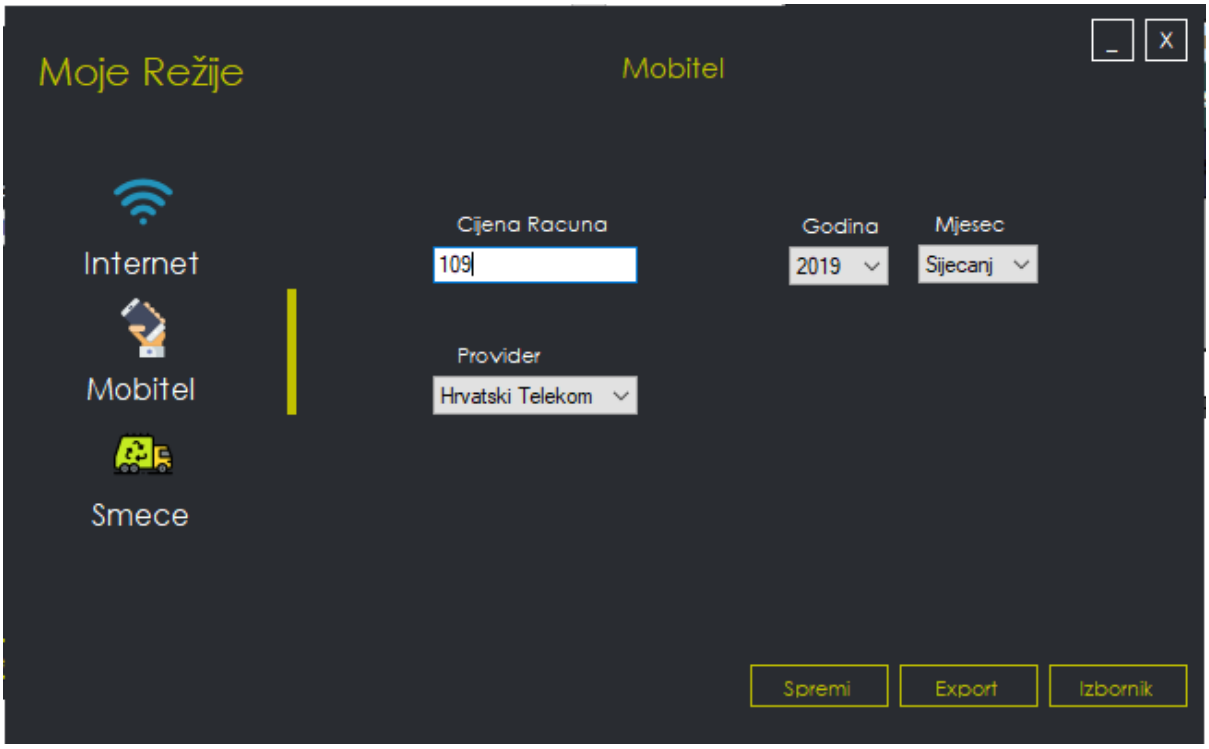
Forma za ostale režije sadrži potrošnju interneta, mobilne pretplate i odvoza smeća. Radi tako da se upisuje izračun, mjesec, godina i ime poduzeća, te se to onda sprema u bazu podataka. Forma sadrži sve restrikcije kao i prethodne forme, podaci se također upisuje na isti način. Isto tako postoji i mogućnost spremanja podatka na radnu površinu. Slika 22 prikazuje unos podataka za spremanje ili izvoz Internet računa, slika 23 prikazuje unos podataka za spremanje ili izvoz mobilnih pretplata i slika 24 prikazuje unos podataka za spremanje ili izvoz usluge za odvoz smeća.

- Forma za internet usluge



Slika 22. Forma za upis internet računa

- Forma za mobilne pretplate



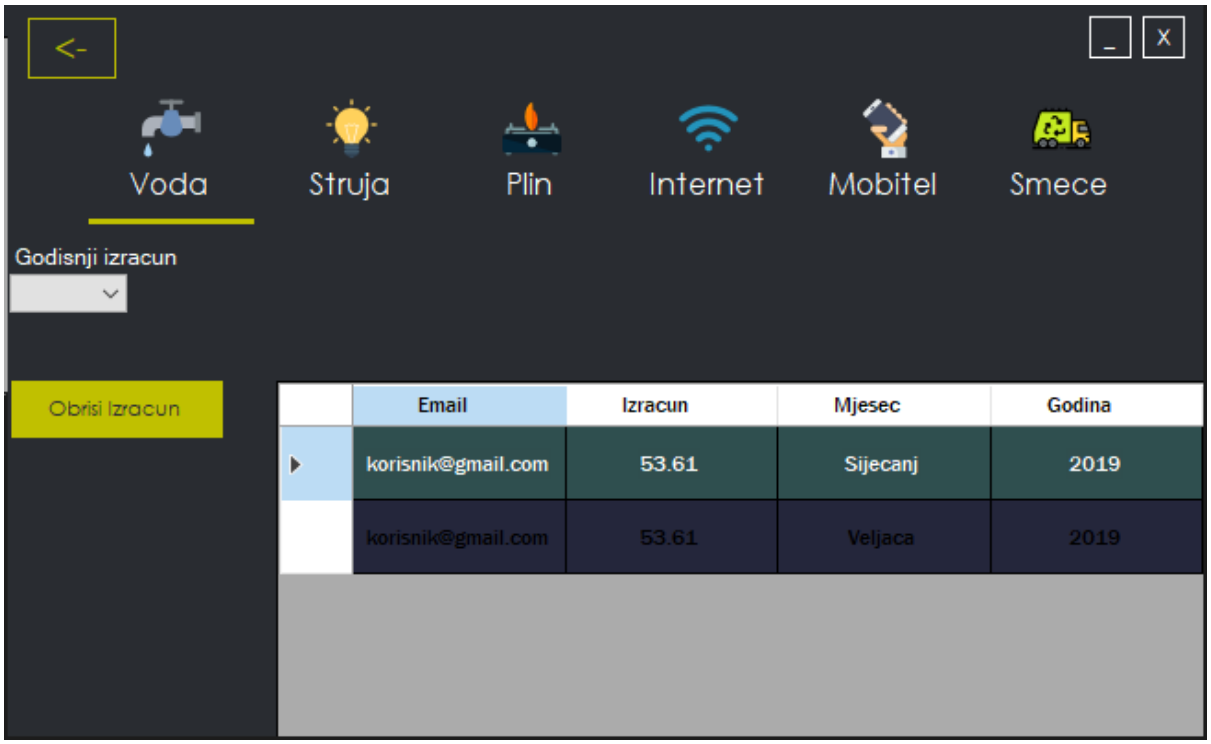
Slika 23. Forma za unos mobilne pretplate

- Forma za unos odvoza smeća

Slika 24. Forma za unos odvoza smeća

4.4 Forma za ispis potrošnji

Ove forma služe da ispišu sve potrošnje od prijavljenog korisnika, te također da mu izračuna potrošnju po godini ukoliko to korisnik poželi. Na ovoj formi moguće je birati koja potrošnja je poželjna za pogledati. Klikom na prikazane ikone moguće je vidjeti sve potrošnje, te je moguće izbrisati potrošnje. Slika 25 prikazuje podatke spremljene unutar same baze podataka od korisnika koji je trenutno prijavljen.



Slika 25. Forma za ispis potrošnje vode

Odmah kada se učita forma funkcija puni se polje podataka, te se tako ispisuju podaci koji se nalaze od onog korisnika koji je trenutno prijavljen. Druga funkcija radi na način da briše onaj podataka koji je kliknut tako što se uzima vrijednost ID-a. Na slici 26 je prikazan kod za sam ispis i brisanje.

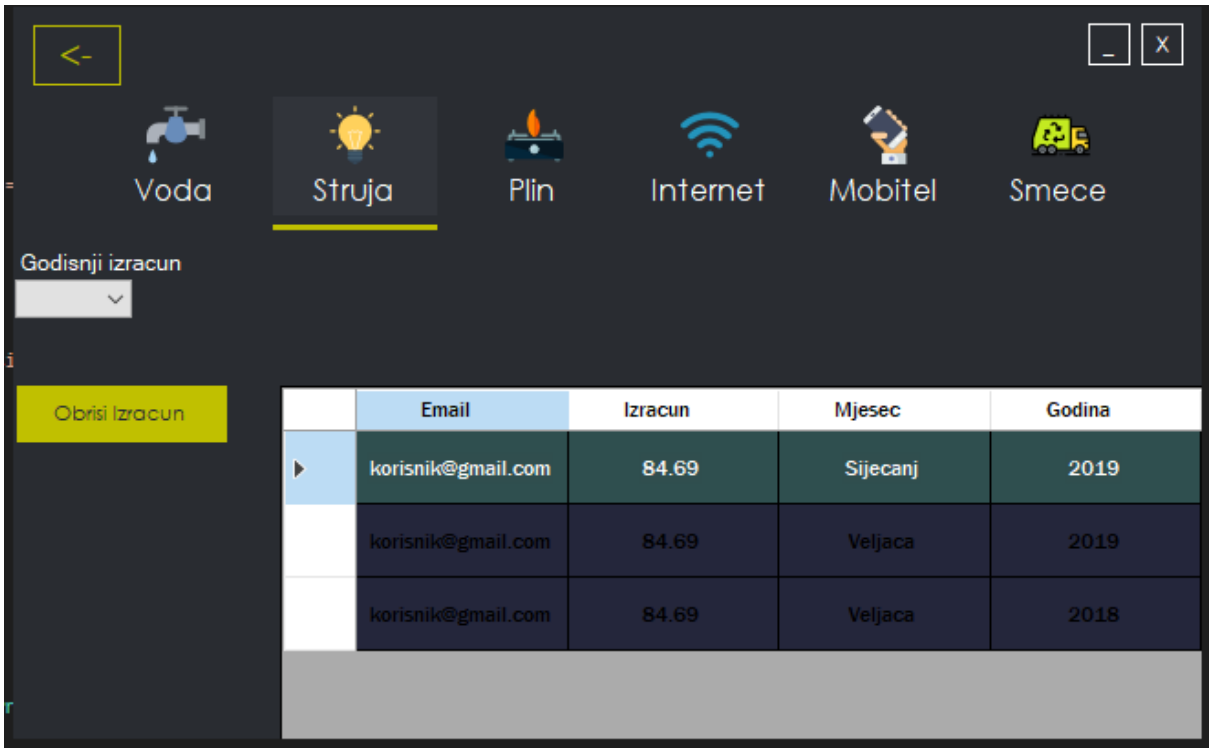
```
1 reference
private void IspisVode_Load(object sender, EventArgs e)
{
    var con = DataBase.GetSqlkonekcija();
    con.Open();
    RefreshDataTable();
    GridIspis.RowTemplate.Height = 40;
    GridIspis.AllowUserToAddRows = false;
    GridIspis.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
    GridIspis.ForeColor = Color.Black;
    GridIspis.DataSource = DT;
    GridIspis.Columns[0].Visible = false;
    RefreshDataTable();

    con.Close();
}

1 reference
private void Btn_Obrisi_Click(object sender, EventArgs e)
{
    var con = DataBase.GetSqlkonekcija();
    con.Open();
    GridIspis.DataSource = DT;
    string str = GridIspis.Rows[GridIspis.SelectedRows[0].Index].Cells[0].Value.ToString();

    SqlCommand Brisi;
    SqlDataAdapter Adapter = new SqlDataAdapter();
    string Obrisi = "Delete from Voda where ID='" + str + "'";
    Brisi = new SqlCommand(Obrisi, con);
    Adapter.DeleteCommand = new SqlCommand(Obrisi, con);
    Adapter.DeleteCommand.ExecuteNonQuery();
    Brisi.Dispose();
    Form UspjesnoObrisano = new UspjesnoObrisano();
    UspjesnoObrisano.Show();
    RefreshDataTable();
    con.Close();

}
```

Slika 26. Kod za ispis i brisanje

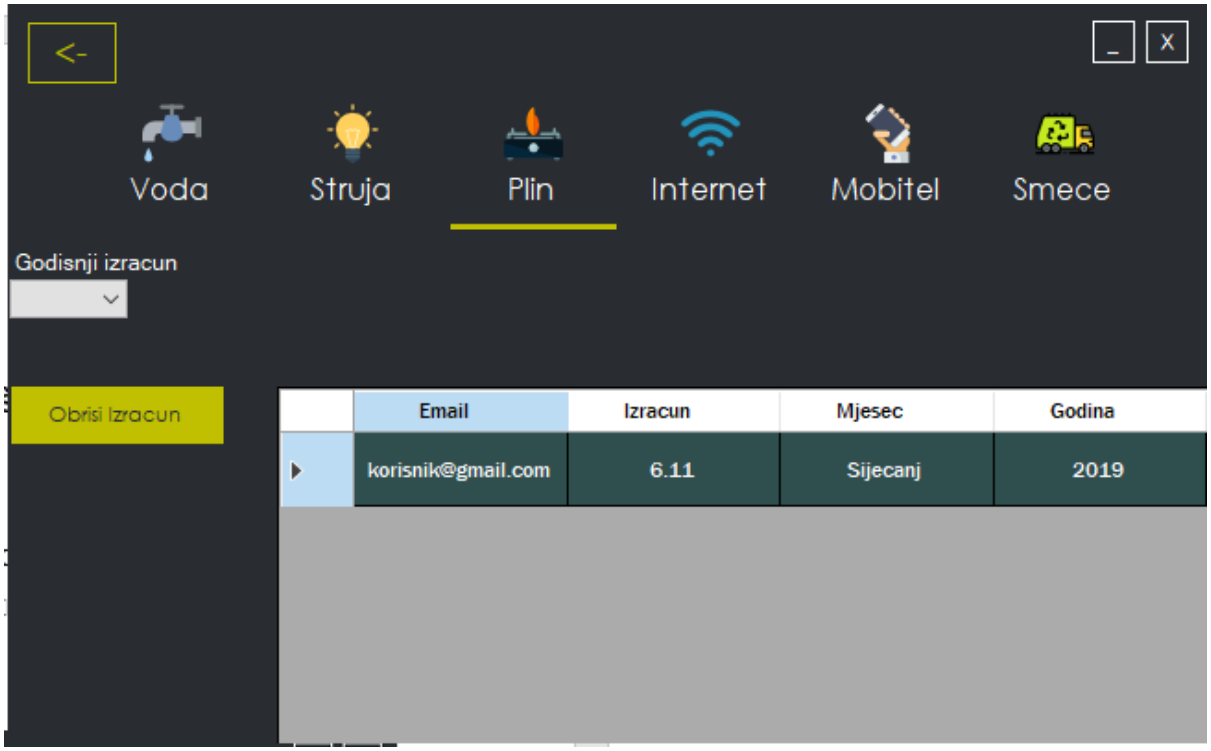
Slika 27 prikazuje sam kod za izračun potrošnje po godini i punjenje tablice podataka. Prva funkcija uzima sve vrijednosti iz baze podataka za onu godinu koja je odabrana te ih zbraja i ispisuje na formu, dok druga funkcija uzima podatke iz baze te ih popunjava u data table koji kasnije popunjava polje podataka za ispis. Slika 28 prikazuje formu za ispis potrošnje struje, dok slika 29 prikazuje formu za ispis potrošnje plina.

```
private void Cb_GodinaVoda_SelectedIndexChanged(object sender, EventArgs e)
{

    double ukupno = 0;
    var con = DataBase.GetSqlkonekcija();
    con.Open();

        SqlCommand Uzmi = new SqlCommand("SELECT Izracun, Godina FROM Voda Where Email = '" + PrijavaKorisnika.ImePrijavljenog + "'AND Godina = '" + cb_GodinaVoda.Text + "'", con);
        SqlDataReader Citac;
        Citac = Uzmi.ExecuteReader();
        while (Citac.Read())
        {
            ukupno = ukupno + Convert.ToDouble(Citac.GetValue(0));
        }
        lbl_Ispis.Text = "Vasa Ukupna cijena za godinu " + cb_GodinaVoda.Text + " iznosi " + ukupno + "kn";
        lbl_Ispis.Visible = true;
        Citac.Close();
        Uzmi.Dispose();
        lbl_Ispis.Refresh();
        con.Close();

    }
3 references
public void RefreshDataTable()
{
    DT.Rows.Clear();
    var con = DataBase.GetSqlkonekcija();
    con.Open();
    string sql = "SELECT ID, Email, Izracun,Mjesec,Godina From Voda  WHERE Email='" + PrijavaKorisnika.ImePrijavljenog + "'";
    SqlDataAdapter Adapter = new SqlDataAdapter(sql, con);
    Adapter.Fill(DT);
    con.Close();

}
```

Slika 27. Kod za računanje potrošnje po godini i punjenje tablice podataka.



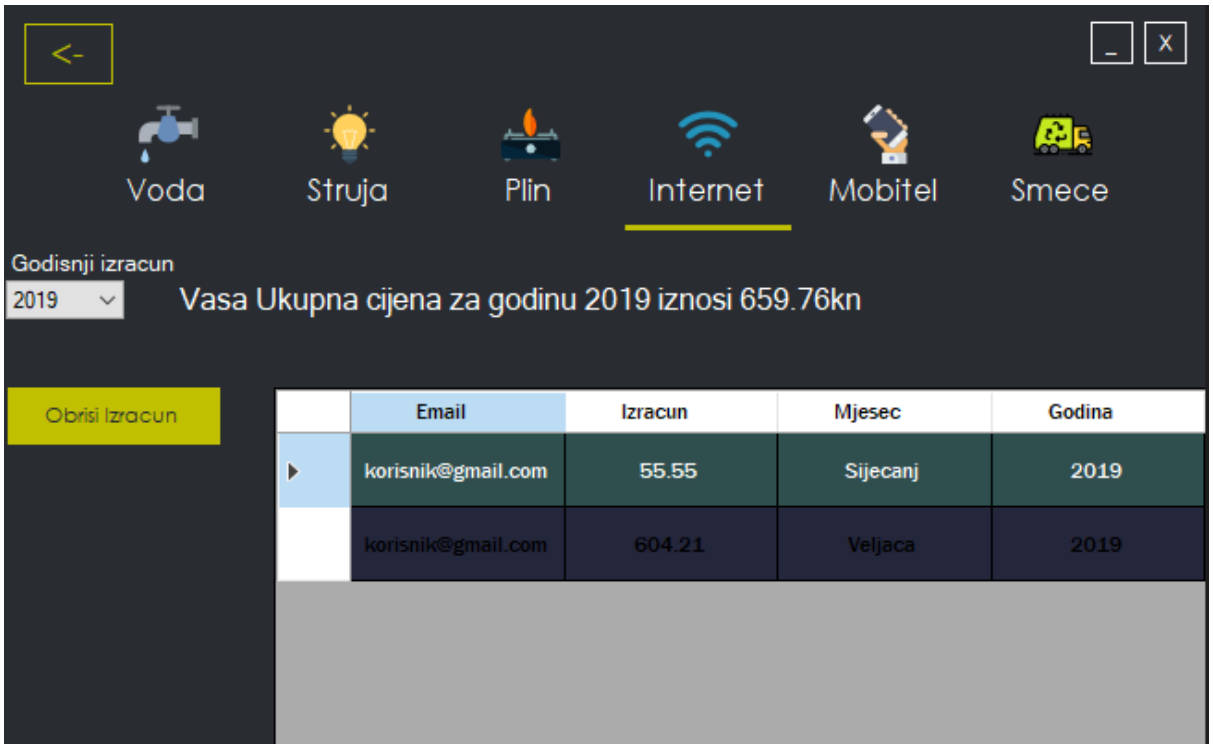
Slika 28. Forma za ispis potrošnje struje



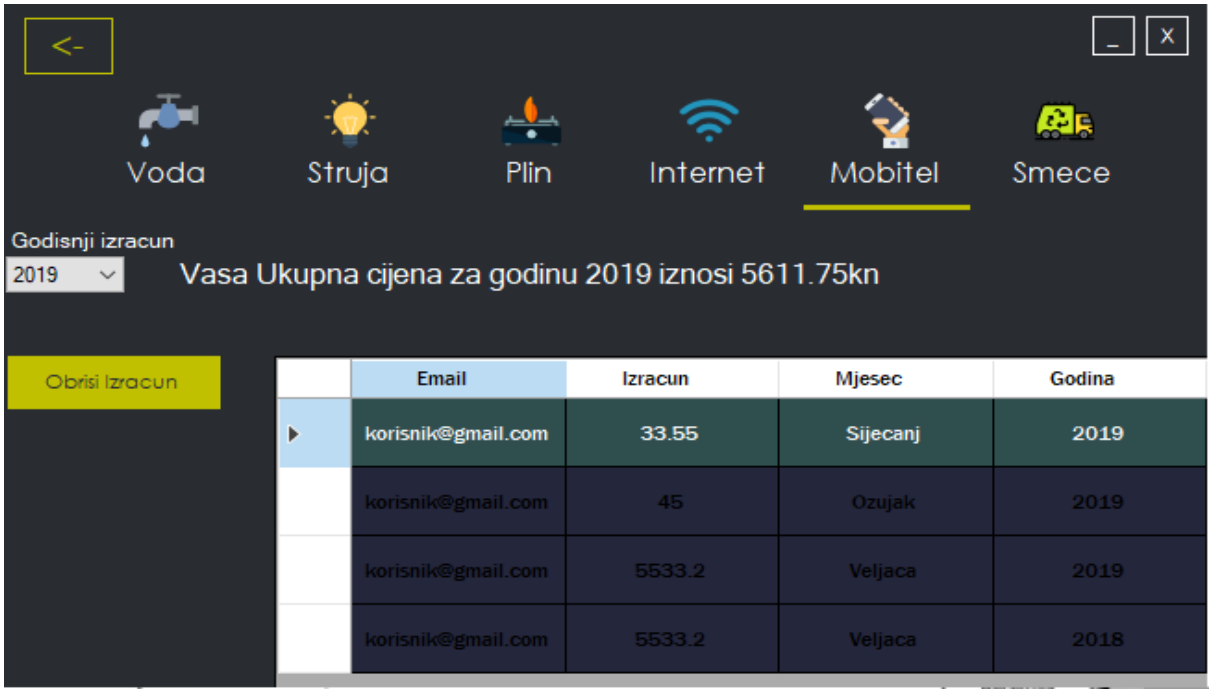
Slika 29. Forma za ispis potrošnje plina

4.4.4. Forme za ispis internet računa, mobilne pretplate i odvoza smeća

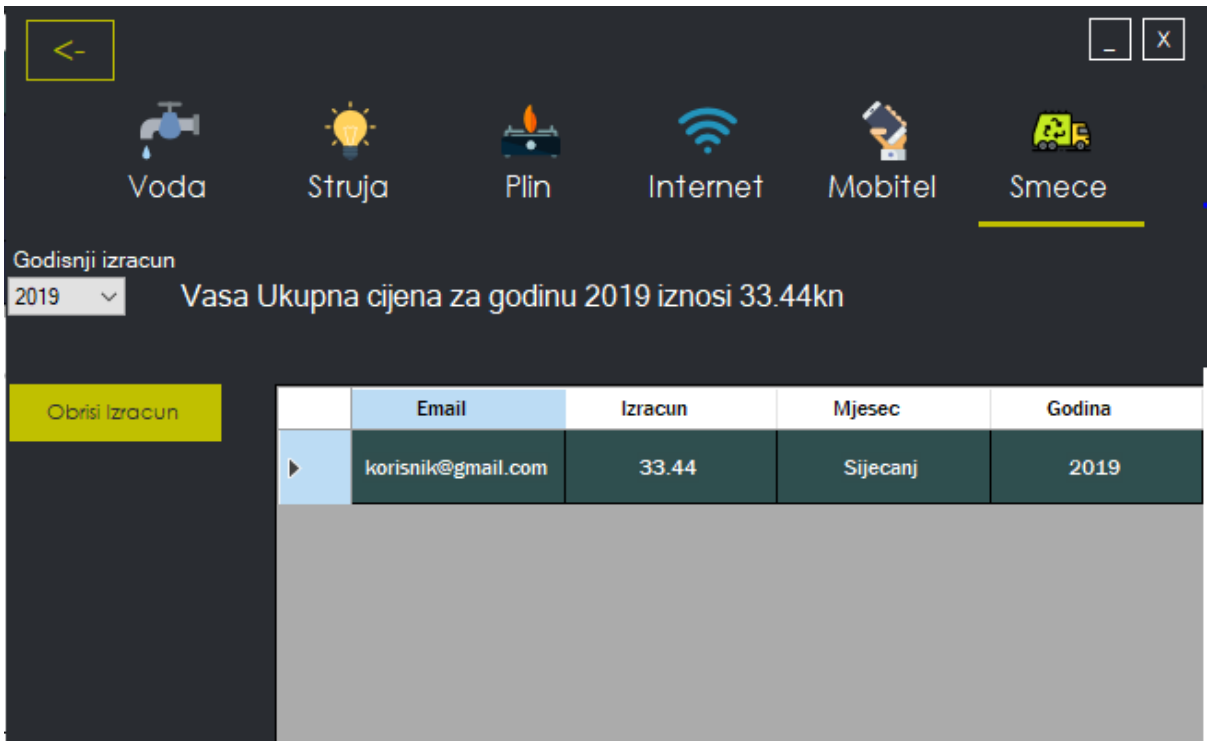
Sljedeće forme rade na isti način kao i forme za ispis ostalih potrošnji. Slika 30 prikazuje formu za ispis internet računa, slika 31 prikazuje formu za ispis mobilnih pretplata i slika 32 prikazuje formu za ispis usluga odvoza smeća.



Slika 30. Forma za ispis Internet računa



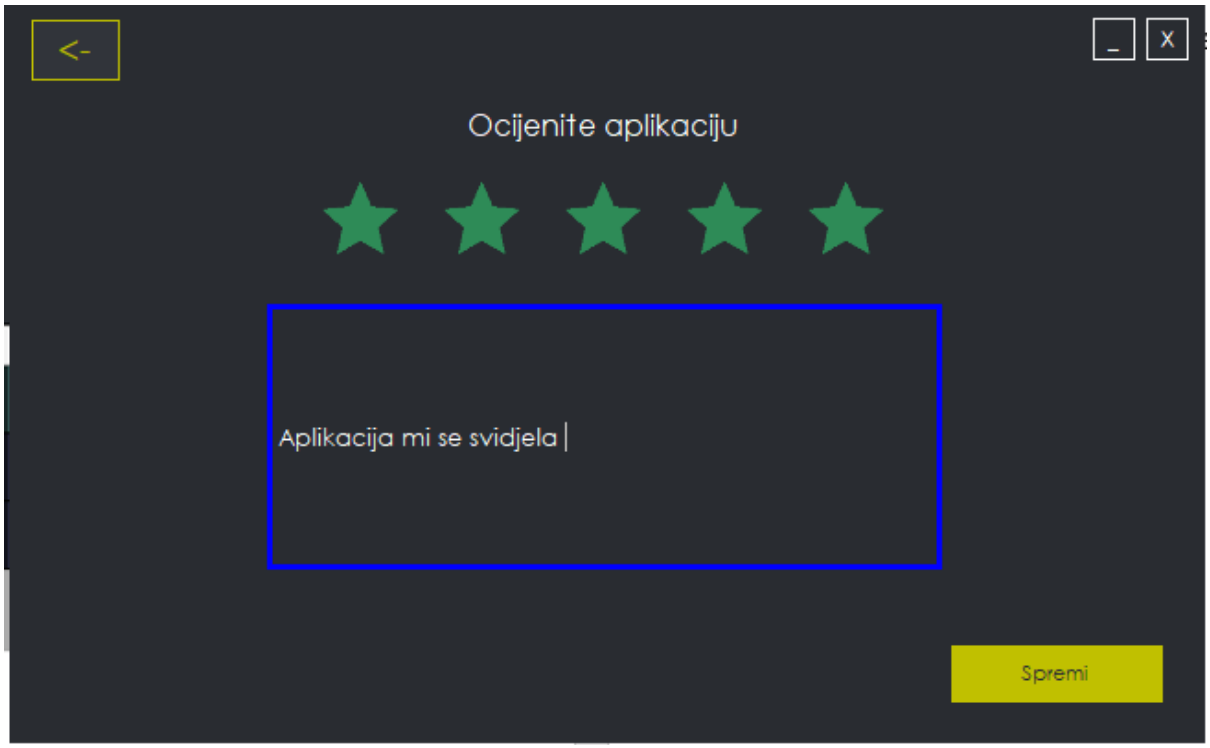
Slika 31. Forma za ispis mobilnih pretplata



Slika 32. Forma za ispis odvoza smeća

4.5 Forma za recenziju aplikacije

Forma za recenziju aplikacije omogućava korisniku za ocjenjivanje i recenziranje same aplikacije. Korisnik treba unijeti ocjenu od 1 do 5 i napisati recenziju. Slika 33 prikazuje formu za ocjenu aplikacije. Forma služi za poboljšavanje same aplikacije u budućnosti.



Slika 33. Forma za recenziju aplikacije

Slika 34 prikazuje kod za spremanje recenzije i ocjene aplikacije. Aplikacija ima zabranu unosa samo ocijene , što znači da korisnik mora s ocjenom ostaviti i samu recenziju.

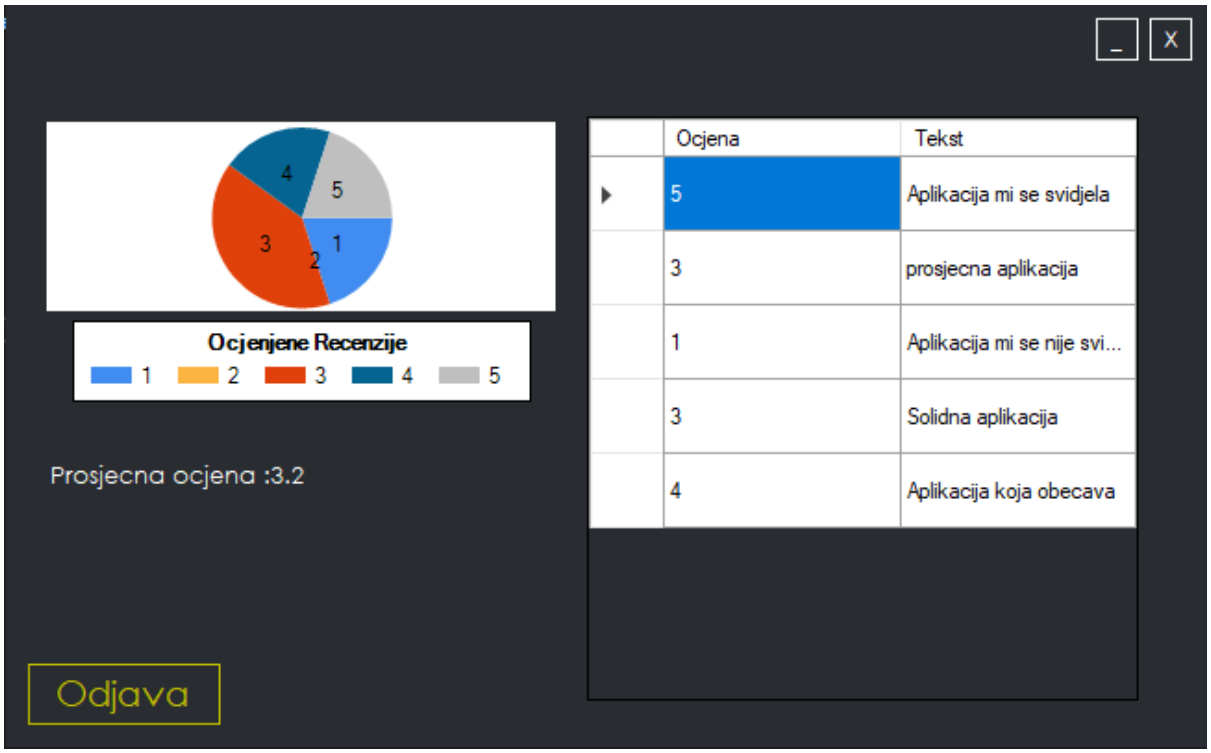
```
private void Btn_Spremi_Click(object sender, EventArgs e)
{
    var con = DataBase.GetSqlkonekcija();
    con.Open();
    if (tb_Recenzija.Text == "")
    {
        Form PopunitePolja = new PopunitePolja();
        PopunitePolja.Show();
    }
    else
    {


        if (bunifu.Value == 1)
        {
            if (tb_Recenzija.Text == "")
            {
                Form PopunitePolja = new PopunitePolja();
                PopunitePolja.Show();
            }
            else
            {


                string query = "INSERT INTO Recenzija(Ocjena,Tekst) VALUES " +
                               "('" + bunifu.Value + "', '" + tb_Recenzija.Text + "')";
                SqlCommand sqlcomm = new SqlCommand(query, con);

                sqlcomm.ExecuteNonQuery();
                this.Hide();
                Form Izbornik = new Izbornik();
                Izbornik.Show();
                Form HvalaOcjena = new HvalaOcjena();
                HvalaOcjena.Show();
            }

        }
```

O

Slika 34. Kod za unos ocjene i recenzije

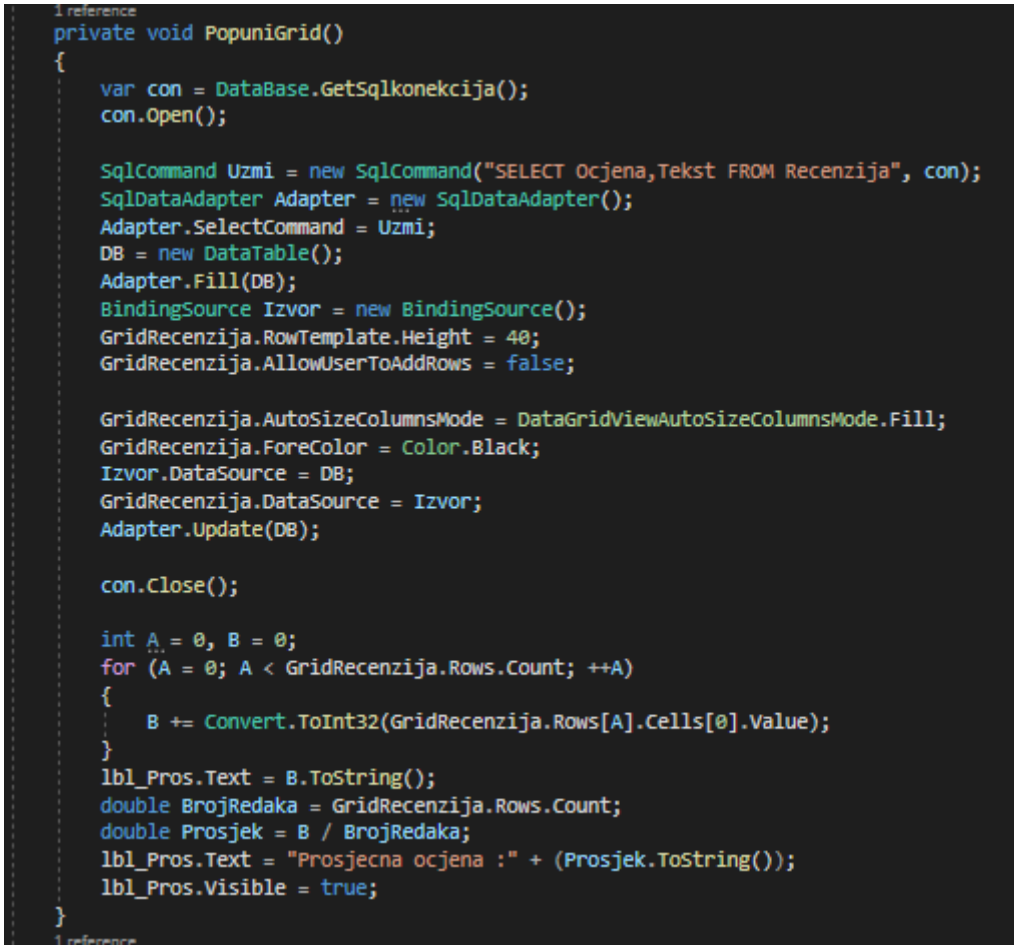
4.6 Forma za administratora

Slika 35 Prikazuje formu za administratora koju može vidjeti samo administrator. Administrator vidi prosječnu ocjenu ali i sve recenzije koje su korisnici unijeli. Na formi se vidi graf koji prikazuje koliki je broj ocjena. Administrator ovo koristi kako bih unaprijedio svoju aplikaciju u budućnosti.



Slika 35. Forma za administratora

Slika 36. prikazuje kod koji radi na način da se prvo uzimaju sve ocjene i recenzije iz baze podataka koje se onda spremaju u tablicu gdje se prije ispisa na polje podataka zbroje i podijele u prosječnu ocjenu.



```
1 reference
private void PopuniGrid()
{
    var con = DataBase.GetSqlkonekcija();
    con.Open();

    SqlCommand Uzmi = new SqlCommand("SELECT Ocjena,Tekst FROM Recenzija", con);
    SqlDataAdapter Adapter = new SqlDataAdapter();
    Adapter.SelectCommand = Uzmi;
    DB = new DataTable();
    Adapter.Fill(DB);
    BindingSource Izvor = new BindingSource();
    GridRecenzija.RowTemplate.Height = 40;
    GridRecenzija.AllowUserToAddRows = false;

    GridRecenzija.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
    GridRecenzija.ForeColor = Color.Black;
    Izvor.DataSource = DB;
    GridRecenzija.DataSource = Izvor;
    Adapter.Update(DB);

    con.Close();

    int A = 0, B = 0;
    for (A = 0; A < GridRecenzija.Rows.Count; ++A)
    {
        B += Convert.ToInt32(GridRecenzija.Rows[A].Cells[0].Value);
    }
    lbl_Pros.Text = B.ToString();
    double BrojRedaka = GridRecenzija.Rows.Count;
    double Prosjek = B / BrojRedaka;
    lbl_Pros.Text = "Prosjecna ocjena :" + (Prosjek.ToString());
    lbl_Pros.Visible = true;
}
1 reference
```

Slika 36. Kod za popunjavanje polja podataka i računanje prosječne ocjene

Ovaj dio koda sprema sve podatke u određene dijelove PieChart modela , te ih onda ispisuje na formu kako bi ih administrator mogao vidjeti. Slika 37 prikazuje kod koji popunjava PieChart.

```
private void DrawPieChart(int Ocjena1, int Ocjena2, int Ocjena3, int Ocjena4, int Ocjena5)
{
    Graf.Series.Clear();
    Graf.Legends.Clear();

    Graf.Legends.Add("Legenda");
    Graf.Legends[0].LegendStyle = LegendStyle.Table;
    Graf.Legends[0].Docking = Docking.Bottom;
    Graf.Legends[0].Alignment = StringAlignment.Center;
    Graf.Legends[0].Title = "Ocjenjene Recenzije";
    Graf.Legends[0].BorderColor = Color.Black;

    string seriesname = "Ocjene";
    Graf.Series.Add(seriesname);

    Graf.Series[seriesname].ChartType = SeriesChartType.Pie;

    Graf.Series[seriesname].Points.AddXY("1", Ocjena1);
    Graf.Series[seriesname].Points.AddXY("2", Ocjena2);
    Graf.Series[seriesname].Points.AddXY("3", Ocjena3);
    Graf.Series[seriesname].Points.AddXY("4", Ocjena4);
    Graf.Series[seriesname].Points.AddXY("5", Ocjena5);
}
```

Slika 37. Izrada PieChart modela

Zaključak

Aplikacija korisniku može donijeti određenu organiziranost tako što bih korisnik uvijek mogao provjeriti potrošnju svojih režija , te unaprijed odvojiti novac za plaćanje istih.

Cilj rada ove aplikacije je da se zadovolji korisnike svih uzrasta, ali isto tako i savladavanje programiranja C# jezika unutar visual studio programa.

U dokumentaciji su objašnjene sve funkcije programa i njegovog koda koji su priloženi.

Velika prednost ove aplikacije je u tom što je jako jednostavna za koristit. Sve režije su na jednom mjestu pa nema potrebe da se ide na stranice od ovlaštenih poduzeća. Korisnik ima mogućnost spremanja režija, te onda kasnije to koristit da provjeri koliko je izašao račun za koji mjesec i tako ostat organiziran. Još jedna od velikih prednosti su te da pokazuju korisniku ukoliko on pogriješi tijekom rada s aplikacijom. Ukoliko unese slovo u okvir za računanje pojavit će mu se greška da to ispravi , ili ukoliko već ima unesen isti račun.

Spremanje izračuna na računalo je isto jedna prednost aplikacije koje druge aplikacije nemaju. Korisnik može spremiti izračun u xlsx formatu , te tako isto držati tok plaćanja svojih režija. Da bih korisnik otvorio taj format potreban mu je alat microsoft excel.

Korisnik može ocijeniti aplikaciju i napisati recenziju u svrhu poboljšavanja same aplikacije. Također jedino administrator može vidjeti same ocijene i recenzije aplikacije.

Korisničko sučelje aplikacije je vrlo jednostavno i pristupačno korisniku.

Literatura

[1]. https://www.hep.hr

[2]. www.pis.com.hr

[3]. www.ppd.hr

[4]. http://www.kdvik-rijeka.hr/

[5]. https://vodovod.com/

[6]. https://www.vio.hr

[7.]. https://ironsoftware.com/csharp/excel/

Popis slika

Slika 1. Dijagram slučaja 2
Slika 2. Dijagram slijeda 3
Slika 3. Klasni dijagram 4
Slika 4. Forma za prijavu 5
Slika 5. Dio koda koji provjerava autentifikaciju korisnika 6
Slika 6. Poruka za uspješnu i ne uspješnu prijavu 6
Slika 7. Forma za registraciju 7
Slika 8. Poruka za provjeru lozinki i poruka za uspješnu registraciju 7
Slika 9. Poruka za postojeći email i prazna polja 8
Slika 10. Kod koji radi provjeru za uspješnu registraciju 8
Slika 11. Forma za izbornik 9
Slika 12. Forma za računanje potrošnje vode 10
Slika 13. Kod za računanje vode i provjeru stanja 11
Slika 14. Kod za izvoz xlsx formata na radnu površinu 12
Slika 15. Izvoz potrošnje vode u xlsx formatu na radnu površinu 12
Slika 16. Kod za spremanje izračuna u bazu podataka 13
Slika 17. Poruka za uspješno spremanje podatka 13
Slika 18. Forma za računanje potrošnje struje 14
Slika 19. Izvoz potrošnje struje u xlsx formatu na radnu površinu 15
Slika 20. Forma za računanje potrošnje plina 16
Slika 21. Izvoz potrošnje struje u xlsx formatu na radnu površinu 17
Slika 22. Forma za upis internet računa 18
Slika 23 Forma za unos mobilne pretplate 19
Slika 24. Forma za unos odvoza smeća 20
Slika 25. Forma za ispis potrošnje vode 21
Slika 26. Kod za ispis i brisanje 22
Slika 27. Kod za računanje potrošnje po godini i punjenje data table. 23

Slika 28. Forma za ispis potrošnje struje 23
Slika 29. Forma za ispis potrošnje plina 24
Slika 30. Forma za ispis Internet računa 25
Slika 31. Forma za ispis mobilnih pretplata 26
Slika 32. Forma za ispis odvoza smeća 26
Slika 33. Forma za recenziju aplikacije 27
Slika 34. Kod za unos ocjene i recenzije 28
Slika 35. Forma za administratora 29
Slika 36. Kod za popunjavanje data grida i računanje prosječne ocijene 30
Slika 37. Izrada PieChart modela 31