

**Univerzitet u Novom Sadu**

**Tehnički fakultet „Mihajlo Pupin“**

**Zrenjanin**

**WPF aplikacija - Zubna Ordinacija**

SEMINARSKI RAD

Razvoj softvera otvorenog koda

**Predmetni profesor** **Studenti**

Prof. dr Zoltan Kazi Alen Mesaroš SI 43/21

**Predmetni asistent** Mihailo Tanackov SI 7/21  
Msc Velibor Premčevski Kristijan Kečkeš SI 36/21

Dušan Bilbija SI 2/21

**Zrenjanin, 2025.**

**Sadržaj**

[Cilj seminarskog rada 1](#_Toc207801597)

[1. Uvod 2](#_Toc207801598)

[2. Korišćene tehnologije 3](#_Toc207801599)

[2.1 Microsoft Visual Studio 3](#_Toc207801600)

[2.2 C# programski jezik 3](#_Toc207801601)

[2.3 Windows Presentation Foundation 4](#_Toc207801602)

[2.4 Microsoft SQL Server 4](#_Toc207801603)

[2.5 Git 5](#_Toc207801604)

[2.6 GitHub 5](#_Toc207801605)

[3. Postupak kreiranja aplikacije 6](#_Toc207801606)

[3.1 Kreiranje baze podataka 6](#_Toc207801607)

[3.2 Prozor MainWindow – Login 8](#_Toc207801608)

[3.3 Prozor Meni 8](#_Toc207801609)

[3.4 Prozor Pacijenti 9](#_Toc207801610)

[3.5 Prozor Termini 10](#_Toc207801611)

[3.6 Prozor Pregledi 11](#_Toc207801612)

[3.7 Prozor Lekovi 12](#_Toc207801613)

[3.8 Prozor Zubari 13](#_Toc207801614)

[4. Poslovna pravila 14](#_Toc207801615)

[4.1 Pravila za lekove 14](#_Toc207801616)

[4.2 Pravila za pacijente 14](#_Toc207801617)

[4.3 Pravila za termine 15](#_Toc207801618)

[4.4 Pravila za preglede 15](#_Toc207801619)

[5. Ključni delovi koda 16](#_Toc207801620)

[6. Zaključak 19](#_Toc207801621)

[Literatura 20](#_Toc207801622)

# Cilj seminarskog rada

Cilj ovog seminarskog rada je izrada aplikacije koja će olakšati svakodnevno poslovanje jedne zubne ordinacije. Aplikacija je zamišljena tako da omogući jednostavno čuvanje i upravljanje podacima o pacijentima, zubarima, terminima, pregledima i lekovima.

Umesto vođenja evidencije na papiru ili kroz nepregledne tabele, aplikacija pruža moderan i praktičan način rada: dodavanje novih podataka, njihova izmena ili brisanje obavlja se brzo i lako, a sve informacije ostaju sigurno sačuvane u bazi. Pored toga, sistem omogućava i preglednu bazu svih lekova i zaposlenih zubara, što dodatno doprinosi boljoj organizaciji.

Glavna ideja je da se poslovanje ordinacije učini bržim, preglednijim i sigurnijim za sve korisnike. Time se smanjuju greške, štedi vreme i omogućava kvalitetnija komunikacija između pacijenta i lekara. Na kraju, aplikacija predstavlja alat koji modernizuje rad ordinacije i doprinosi pružanju boljih usluga pacijentima.

# Uvod

U okviru ovog seminarskog rada predstavljena je aplikacija za zubnu ordinaciju, razvijena korišćenjem troslojne arhitekture. Ovakav pristup omogućava jasnu podelu sistema na tri nezavisna dela, što olakšava održavanje, proširivanje i ponovnu upotrebu koda.

Prvi sloj je sloj prezentacije, koji čini korisnički interfejs i obezbeđuje komunikaciju između korisnika i aplikacije. Drugi sloj je sloj servisa (poslovna logika), gde se nalaze pravila i procesi vezani za tabele, kao i validacija i regulisanje ispravnog korišćenja aplikacije. Treći sloj je sloj podataka, koji komunicira sa SQL bazom preko SQL queries-a i vraća željene podatke korisniku.

Ovakva podela doprinosi boljoj organizaciji aplikacije, omogućava jednostavnije testiranje i kasniji razvoj novih funkcionalnosti.

Korisnici aplikacije zubne ordinacije su njeni zaposleni.

# Korišćene tehnologije

Korišćene tehnologije predstavljaju sve tehnologije upotrebljene prilikom izrade

Windows Presentation Foundation aplikacije. Postoji veliki broj tehnologija koji je

potreban za izradu ovakve aplikacije.

## 2.1 Microsoft Visual Studio

Microsoft Visual Studio je integrisano razvojno okruženje Microsoft-a. Koristi se za

razvoj kompjuterskih programa za Microsoft Windows, kao i veb sajtove, veb aplikacije

i veb servise. Visual Studio koristi Microsoft platforme za razvoj softvera kao što su

Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i

Microsoft Silverlight. Može da se piše i upravlja kodom.

Visual Studio uključuje editor koda koji podržava IntelliSense. Dolazi sa ugrađenim C#

i C++ , VB NET (Visual Basic .NET) i F# programskim jezicima. Takođe se može

programirati i u drugim programskim jezicima kao što su JavaScript, CSS, XML,

HTML, itd. instaliranjem jezičkih servisa za ovo razvojno okruženje. Microsoft Visual

Studio je jedan od najpopularnijih razvojnih okruženja koji pruža velike mogućnosti u

razvoju programa. Njegove mane su donekle ispravljene pomoću raznih ekstenzija

razvijenih kako od „Microsofta“ tako i od strane drugih programera. Konkretno kod

programiranja Windows form aplikacija, on sam generiše skelet aplikacije i sve podesi

automatski kako bi korisnik mogao da se bavi problemom koji treba da reši, a ne

pravljenjem prozora aplikacije. Pored desktop aplikacija takođe nudi mogućnost

kreiranja i veb aplikacija. Microsoft je jako dobro razvio pronalaženje i uklanjanje

grešaka pri programiranju. Zbog toga predstavlja jako popularnu i korisnu alatku

programerima.

## 2.2 C# programski jezik

C# (C Sharp) programski jezik je jedan od mlađih programskih jezika nastao 2002.

godine kao sastavni deo MS .NET Framework-a 1.0. C# je objektno orijentisani

programski jezik namenjen razvoju aplikacija na .NET Framework platformi. Pošto je

objekno orijentisan programski jezik, prvenstveno je sagrađen od klasa koje čine

skupove objekata koji imaju zajedničku strukturu i ponašanje. Takođe kao i mnogi

objektno orijentisani programski jezici, sadrži sposobnost ekapsulacije, pisanje metoda,

nasleđivanje između klasa i visoke performanse tokom kompajliranja. Iz razloga

enkapsulacije postoji obrazac kojim atributima različitih klasa možemo spolјa pristupiti

indirektno i to metodama get i set. C# je case sensitive programski jezik, što znači da je

od velike važnosti da li kucamo malim ili velikim slovima.

## 2.3 Windows Presentation Foundation

WPF, tj. Windows Presentation Foundation, je najnoviji Microsoftov pristup GUI

framework-u, koji se koristi sa .NET framework-om. GUI framework je skraćenica za

Graphical User Interface (grafički korisnički interfejs). Windows ima GUI za rad sa

računarom, a veb pretraživač ima GUI koji omogućava surfovanje internetom.

GUI framework omogućava da se kreira aplikacija sa širokim spektrom GUI elemenata,

kao što su labele, tekstboks-ovi i drugi dobro poznati elementi. Bez GUI framework-a

treba ručno da se nacrtaju ovi elementi i da obradi sve scenarije i interakcije korisnika

kao što su unos teksta i pomeranje miša. Sve ovo predstavlja mnogo posla, pa

zbog toga većina programera koristi GUI framework koji će obaviti osnovne radnje i

omogućiti programerima da se fokusiraju na razvijanju aplikacija.

Postoji dosta GUI framework-a, ali za .NET programere, najzanimljivje su Windows

Forme i WPF. WPF je noviji, ali Microsoft i dalje održava i podržava Windows Forme.

Postoji nekoliko razlika između ova dva framework-a, ali njihova svrha je ista, da se

kreira na najlakši našin sa odličnim GUI.

## 2.4 Microsoft SQL Server

Microsoft SQL Server je relacijska baza podataka kojoj je primarni jezik za upite

Transact SQL (T-SQL), što znači da osim osnovnih i klasičnih (SELECT tipa) SQL upita

dozvoljava i složenije stvari poput menjanja programskog toka (IF naredba) i slično.

Transact SQL nastao je kao plod saradnje između Microsofta i Sybasea. SQL server je

baza podataka koja se smestila na prag između manjih i srednjih baza. Prvi SQL Server

za Windows NT izašao je isto kada i sami Windows-i.

SQL Server je prva baza podataka na svetu koja je posedovala korisnički interfejs.

Takođe, SQL Server je prva komercijalna baza podataka koja je podržala Intelovu 64-

bitnu arhitekturu procesora. Od SQL Servera 2000 do SQL Servera 2005 napravljeni su

golemi pomaci u poboljšanju same baze podataka. Najbitniji od njih su: Klijentski deo

IDE alata, SQL Server Integration Services, Reporting Server, OLAP server, podrška za

data mining, Notification Services itd.

Microsoft SQL podržava „Open Database Connectivity“, odnosno skraćeno ODBC

tehnologiju. SQL Server 2005 poseduje i podršku za Web servise, tj. za Simple Object

Access Protocol, Service Oriented Architecture Protocol odnosno skraćeno SOAP W3C

standard. To je izvanredna stvar zato što omogućuje ne-Windows klijentima da

komuniciraju preko te platforme sa SQL Serverom. U verziji 2005 takođe je izdan i

sertifikovan JDBC API za komunikaciju sa Java aplikacijama. SQL Server takođe

poseduje i mogućnosti poput mirroringa, clusteringa i snapshots-ova. Microsoft se

stvarno potrudio, te je omogućio brojne aplikacije koje spremaju podatke na SQL

Server. SQL Server 2005 uključuje i Common Language Runtime odnosno CLR.

## 2.5 Git

Git je sistem za verzionisanje koda, napravljen 2005. godine od strane Linusa Torvaldsa, tvorca Linux-a. Njegov glavni zadatak je da beleži sve promene u kodu i da omogući da više programera radi istovremeno na istom projektu, a da pri tom ne dolazi do gubljenja podataka ili neželjenih konflikata. Za razliku od starijih sistema, Git čuva celu istoriju projekta i na lokalnom računaru, pa programer može da radi i kada nema internet konekciju.

Velika prednost Gita je rad sa granama (branches). Svaka nova funkcionalnost može da se razvija odvojeno od glavnog koda, a kada se završi, jednostavno se spaja (merge) sa glavnom verzijom projekta. Na taj način timovi mogu paralelno da rade na različitim delovima aplikacije bez ometanja i uz manji rizik od grešaka.

Git se koristi kroz jednostavne komande poput *git init*, *git add*, *git commit*, *git push* i *git pull*. Pored toga, nudi i napredne mogućnosti, na primer vraćanje na starije verzije koda, kreiranje različitih tokova rada (workflow), kao i potpunu kontrolu nad istorijom promena. Zbog svega ovoga, Git je postao neizostavan alat u modernom razvoju softvera i danas predstavlja industrijski standard.

## 2.6 GitHub

GitHub je internet platforma zasnovana na Git sistemu za verzionisanje, ali sa mnogo dodatnih opcija koje olakšavaju rad pojedincima i timovima. Pokrenut je 2008. godine i vrlo brzo je postao najpoznatije mesto za čuvanje i deljenje koda, kao i za zajednički rad na softverskim projektima.

Glavna prednost GitHub-a je što omogućava da se Git repozitorijumi hostuju na internetu, pa kod više nije vezan samo za jedan računar, već mu se može pristupiti bilo gde i bilo kada. Osim običnog čuvanja koda, GitHub nudi i praktične funkcionalnosti: pull requests (zahtevi za spajanje koda), issues (praćenje grešaka i zadataka), actions (automatizacija testiranja i deploy-a), kao i wiki i dokumentaciju.

Zbog ovoga je saradnja između više programera znatno jednostavnija – svako može da radi u svojoj grani, predlaže izmene i učestvuje u diskusiji pre nego što promene postanu deo glavnog projekta. Osim toga, GitHub ima i elemente društvene mreže: korisnici mogu da prate jedni druge, ocenjuju projekte i istražuju ogroman broj javno dostupnih open-source projekata.

Danas se GitHub koristi širom sveta, kako u industriji tako i na fakultetima, i predstavlja jedan od ključnih alata u modernom razvoju softvera.

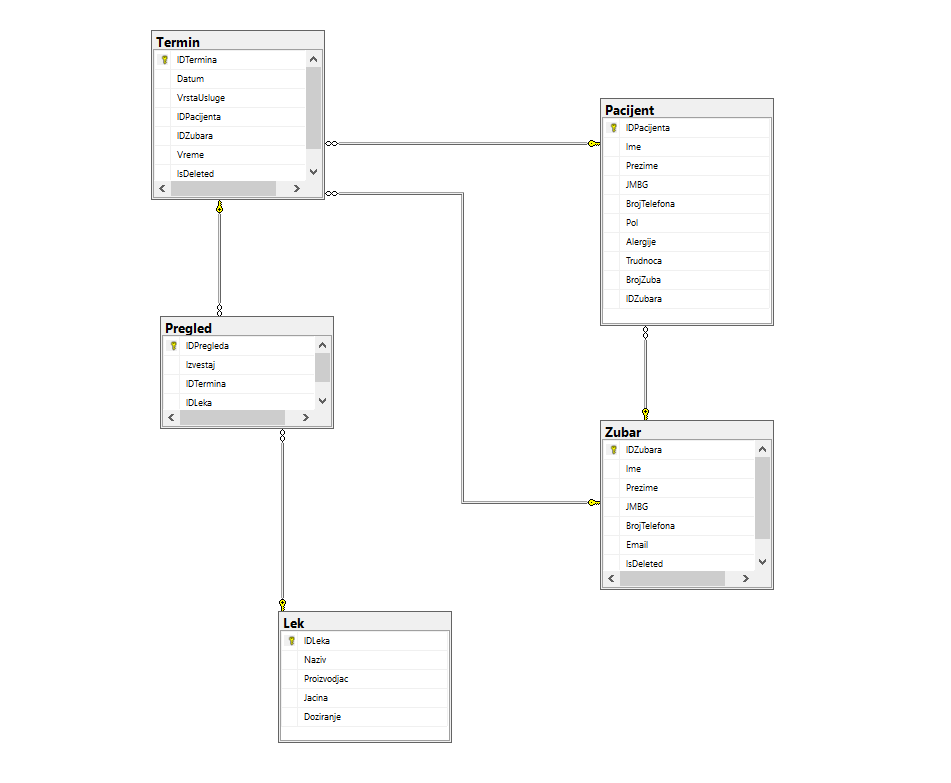
# Postupak kreiranja aplikacije

Kreiranjem aplikacije kreira se prozor MainWindow (Login). Kreira se još i glavni meni, kao i 5 prozor: Pacijenti, Termini, Pregledi, Lekovi i Zubari. Pored toga, odnosno pre, kreira se i

baza podataka Zubne Ordinacije.

## 3.1 Kreiranje baze podataka

Kreira se baza podataka sa nazivom ZubnaOrdinacija. Nakon toga kreiraju se

tabele sa njihovim atributima. U svakoj tabeli treba da se nalazi jedan od atributa koji je primarni ključ te tabele. Tabele se povezuju pomoću stranih ključeva.

*Slika 1.0 – Šema (dijagram) baze podataka ZubnaOrdinacija*

Baza podataka se sastoji od skupa tabela koje su međusobno povezane. Svaka tabela ima svoj naziv, i podatke koji su određenog tipa. Tabele koje se nalaze u bazi podataka su: ***Pacijent***, ***Zubar***, ***Termin***, ***Pregled*** i ***Lek***.

Tabela ***Pacijent*** sadrži sledeće podatke: *IDPacijenta*, *Ime*, *Prezime*, *JMBG*, *BrojTelefona*, *Pol*, *Alergije*, *Trudnoca*, *BrojZuba* i *IDZubara*. *IDPacijenta* je primarni ključ tabele i ima vrednost identity(1,1). Ovo zapravo služi kao automatski brojač jer povećava za jedan ID svakog unetog pacijenta. Unos podatala *Ime*, *Prezime*, *JMBG*, *BrojTelefona*, *Alergije*, *Trudnoca* i *BrojZuba* se vrši preko polja namenjenih tome. Unos za *Pol* se vrši preko padajućeg menija u kom se bira “M“ za muški i “Ž“ za ženski pol. Unos za *IDZubara*, što zapravo dalje predstavlja izabranog zubara, se vrši takođe preko padajućeg menija gde se može videti ID i ime svakog zubara, unetog u bazu podataka, radi lakše preglednosti.

Tabela ***Zubar*** sadrži sledeće podatke: *IDZubara*, *Ime*, *Prezime*, *JMBG*, *BrojTelefona*, *i Email*. *IDZubara* je primarni ključ tabele i ima vrednost identity(1,1). Ovo zapravo služi kao automatski brojač jer povećava za jedan ID svakog unetog zubara. Unos podataka *Ime*, *Prezime*, *JMBG*, *BrojTelefona* i *Email* se vrši preko preko polja namenjenih tome.

Tabela ***Termin*** sadrži sledeće podatke: *IDTermina*, *Datum*, *Vreme*, *VrstaUsluge*, *IDPacijenta* i *IDZubara*. *IDTermina* je primarni ključ tabele i ima vrednost identity(1,1). Ovo zapravo služi kao automatski brojač jer povećava za jedan ID za svaki uneti termin. Unos podatka *Datum* se vrši preko birača datuma, *Vreme* preko polja namenjenog tome, *VrstaUsluge* takođe preko polja namenjenog tome, a unos *IDPacijenta* i *IDZubara* se vrši preko padajućih menija gde se može videti ID i ime pacijenta, kao i ID i ime Zubara, unetih u bazu podataka, radi lakše preglednosti. ID i ime pacijenta predstavljaju pacijenta koji je zakazao termin, a ID i ime zubara predstavljaju zubara koji je za taj termin zadužen.

Tabela ***Pregled*** sadrži sledeće podatke: *IDPregleda*, *Izvestaj*, *IDTermina i IDLeka*. *IDPregleda* je primarni ključ tabele i ima vrednost identity(1,1). Ovo zapravo služi kao automatski brojač jer povećava za jedan ID za svaki uneti termin. Unos podatka *Izvestaj* se vrši preko polja namenjenog tome, unos IDTermina preko polja namenjenog za to, i mora se pogledati u tabeli termina kako bi se znalo o kom terminu se radi.Unos podatka *IDLeka* se vrši preko padajućeg menija gde se može videti ID i naziv leka, unetih u bazu podataka, radi lakše preglednosti. ID termina predstavlja termin iz kojeg je pregled nastao (termini se moraju zakazivati). ID i naziv leka predstavljaju lek koji bi trebao da se uzima posle pregleda (ukoliko za to ima potrebe, ukoliko ne, selektuje se polje sa kosim crtama).

Tabela ***Lek*** sadrži sledeće podatke: *IDLeka*, *Naziv*, *Proizvodjac*, *Jacina* i *Doziranje*. *IDLeka* je primarni ključ tabele i ima vrednost identity(1,1). Ovo zapravo služi kao automatski brojač jer povećava za jedan ID za svaki uneti lek. Unos podataka *Naziv*, *Proizvodjac*, *Jacina* i *Doziranje* se vrši preko preko polja namenjenih tome.

## 3.2 Prozor MainWindow – Login

Login prozor služi za prijavljivanje pre pristupa samoj aplikaciji. S obzirom da je aplikacija za Zubnu ordinaciju, neovlašćenim ljudima pristup nije dozvoljen.

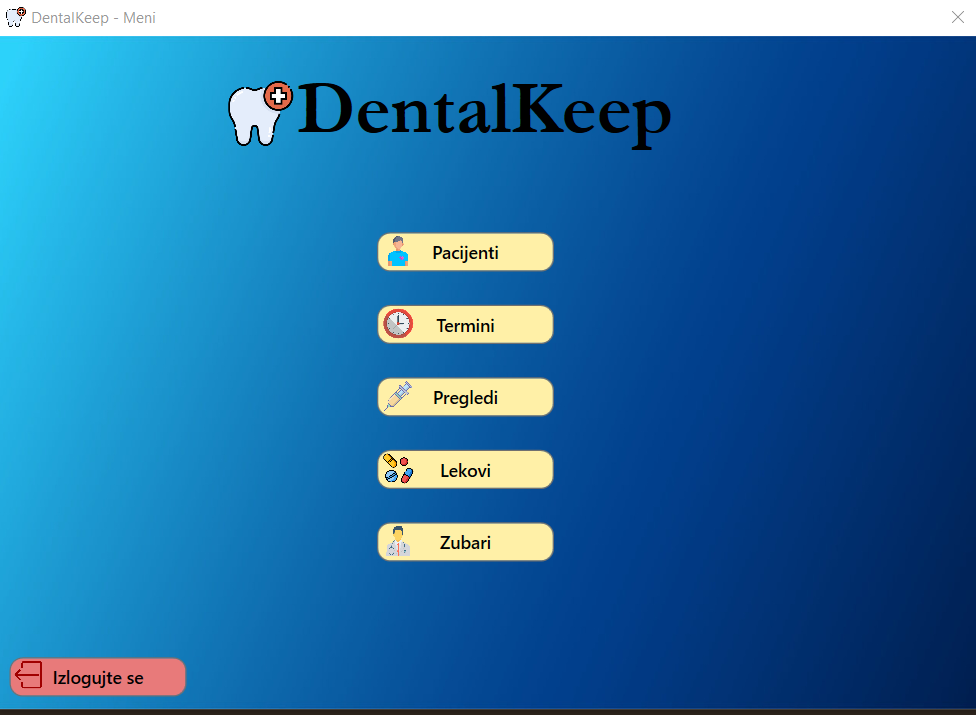
U TextBox se unosi korisničko ime, a u PasswordField lozinka.

Prilikom pritiska na dugme „Ulogujte se” proverava se korisničko ime i lozinka. Provera se vrši jednostavnom if petljom koja prikuplja tekst iz TextBox-a i PasswordField-a i pokušava da ih izjedači sa vrednostima prethodno zadatog stringa. Ukoliko to ne uspe izbacuje se greška, a ukoliko su u redu prebacuje se na sledeći prozor – Meni.



*Slika 2.0 – Izgled Login prozora*

## 3.3 Prozor Meni

Prozor Meni sadrži 5 Buttona (dugmadi) koji vode do 5 menija – Pacijenti, Termini, Pregledi, Lekovi i Zubari. Takođe, sadrži jedan Button za izlogovanje.

*Slika 3.0 – Izgled prozora Meni*

## 3.4 Prozor Pacijenti

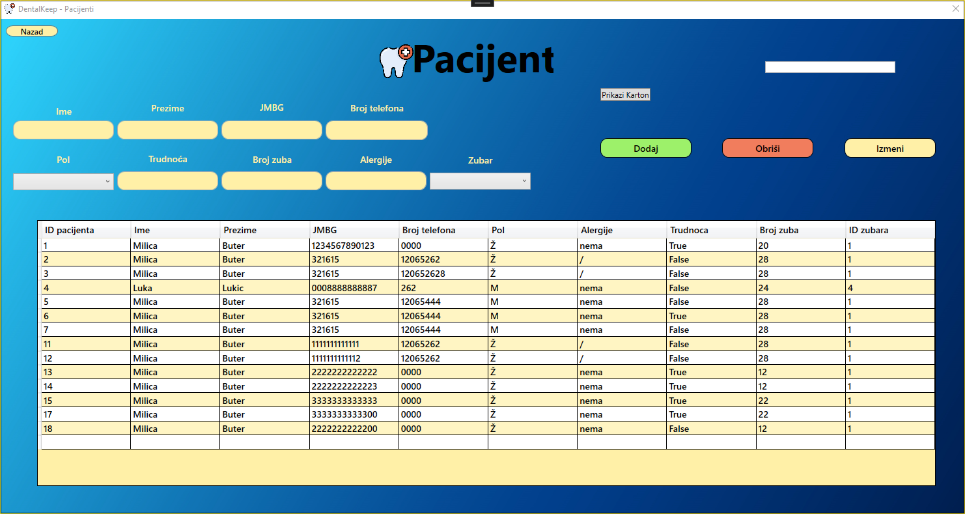
Prozor Pacijenti sadrži 8 TextBox-eva za Ime, Prezime, JMBG, Broj Telefona, Trudnoću, Broj Zuba i Alergije. Sadrži ComboBox Pol kojim se bira pol pacijenta, ponuđeni su M-muški i Ž-ženski. ComboBox Zubar prikazuje ID i Ime svih Zubara iz Baze podataka, te nudi izbor jednog.

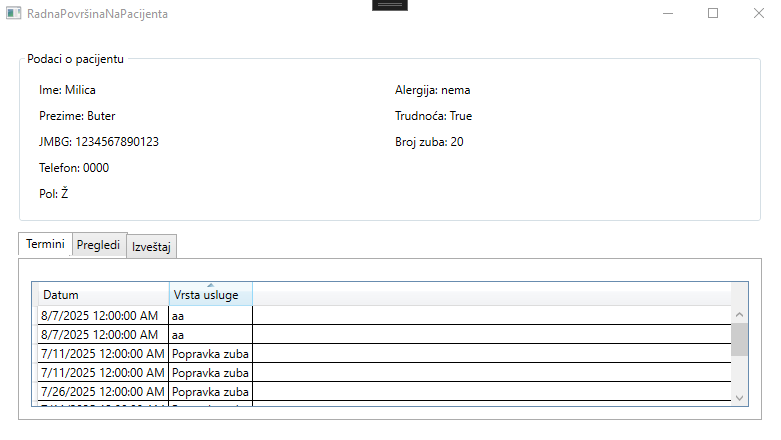
Buttoni (Dugmadi) za dodavanje (zeleno), brisanje (crveno) i izmenu (žuto) koje omogućavaju unos, brisanje i izmenu iz baze podataka preko prethodno navedenih alatki.

Button Nazad koji se nalazi u gornjem levom uglu, vraća na glavni meni.

DataGrid omogućava tabelarni prikaz svih polja tabele Pacijent iz baze podataka, te omogućava veoma pregledan prikaz svih informacija o Pacijentima.

Na desnoj strani nalazi se search bar pomoću kog je moguće pretraživati pacijente po imenu i prezimenu, kao i po JMBG-u. Takođe, implementirano je i dugme za prikaz kartona pacijenta, kojim se otvara detaljan uvid u istoriju termina i pregleda određenog pacijenta.



*Slika 4.0 - Izgled prozora Pacijenti*

*Slika 4.1 - Izgled prozora prikaza kartona*

## 3.5 Prozor Termini

Prozor Termini sadrži DatePicker za Datum i 2 TextBox-a za Vreme i Vrstu Usluge.

Combobox Pacijent prikazuje ID i Ime svih Pacijenata iz Baze podataka, te nudi izbor jednog.

Ispod njega nalazi se dodatni Combobox sa četiri opcije za sortiranje termina: za danas, za 7 dana i za 30 dana, takođe je moguć prikaz svih termina, čime se omogućava brza filtracija prikaza.

Combobox Zubar prikazuje ID i Ime svih Zubara iz Baze podataka, te nudi izbor jednog.

Buttoni (Dugmadi) za dodavanje (zeleno), brisanje (crveno) i izmenu (žuto) koje omogućavaju unos, brisanje i izmenu iz baze podataka preko prethodno navedenih alatki.

Button Nazad koji se nalazi u gornjem levom uglu, vraća na glavni meni.

DataGrid omogućava tabelarni prikaz svih polja tabele Termin iz baze podataka, te omogućava veoma pregledan prikaz svih informacija o Terminima.

*Slika 5.0 – Izgled prozora Termini*

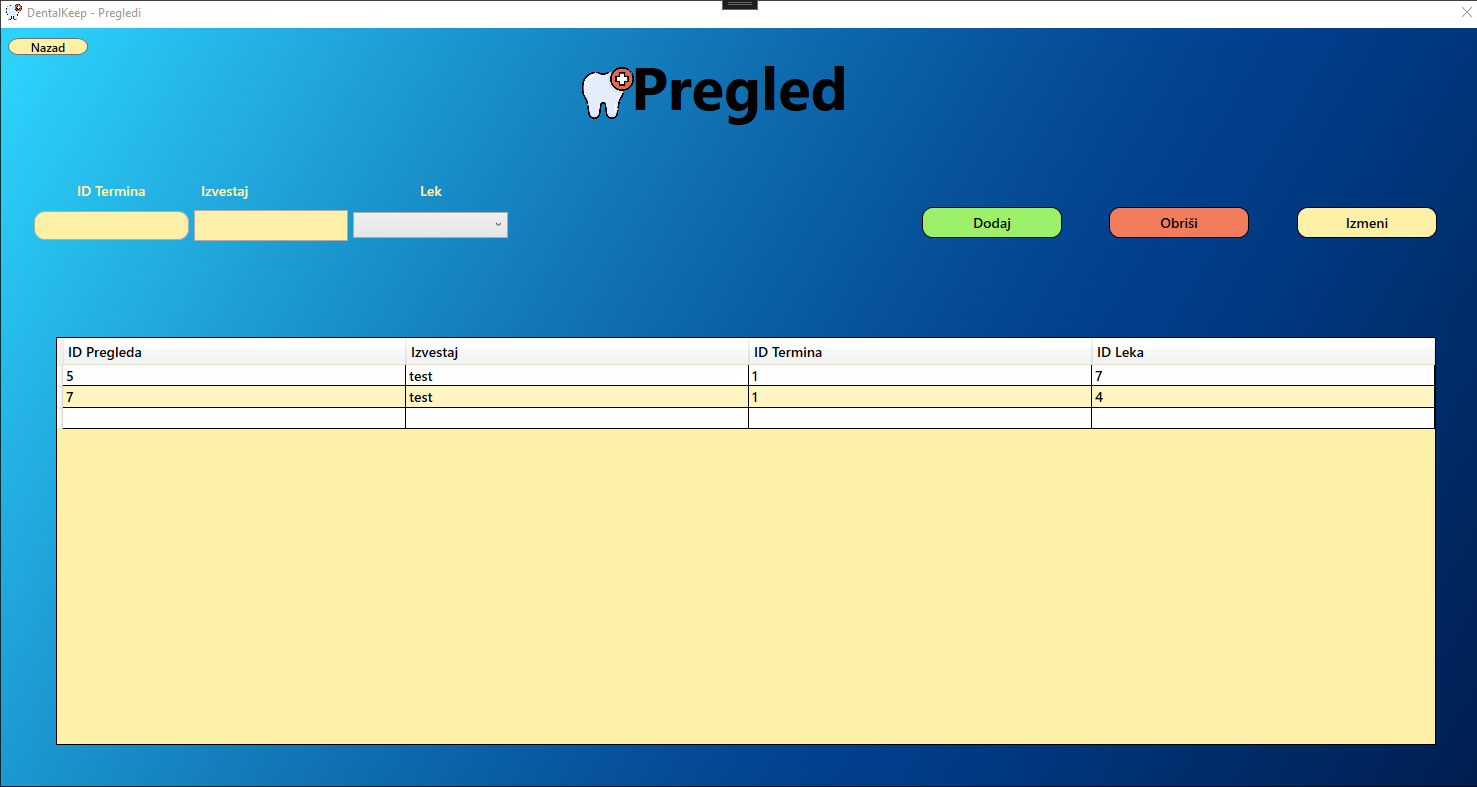
## 3.6 Prozor Pregledi

Prozor Pregledi sadrži DatePicker za Datum Sledeće Posete, TextBox za IDTermina, koji se mora znati (iz prozora Termini), kako bi se Pregled povezao sa Terminom. Combobox Lek prikazuje ID i Naziv svih Lekova iz Baze podataka, te nudi izbor jednog.

Buttoni (Dugmadi) za dodavanje (zeleno), brisanje (crveno) i izmenu (žuto) koje omogućavaju unos, brisanje i izmenu iz baze podataka preko prethodno navedenih alatki.

Button Nazad koji se nalazi u gornjem levom uglu, vraća na glavni meni.

DataGrid omogućava tabelarni prikaz svih polja tabele Pregled iz baze podataka, te omogućava veoma pregledan prikaz svih informacija o Pregledima.



*Slika 6.0 – Izgled prozora Pregledi*

## 3.7 Prozor Lekovi

Prozor Lekovi sadrži 4 TextBox-a za Naziv, Proizvođača, Jačinu i Doziranje.

Buttoni (Dugmadi) za dodavanje (zeleno), brisanje (crveno) i izmenu (žuto) koje omogućavaju unos, brisanje i izmenu iz baze podataka preko prethodno navedenih alatki.

Button Nazad koji se nalazi u gornjem levom uglu, vraća na glavni meni.

DataGrid omogućava tabelarni prikaz svih polja tabele Lek iz baze podataka, te omogućava veoma pregledan prikaz svih informacija o Lekovima.



*Slika 7.0 – Izgled prozora Lekovi*

## 3.8 Prozor Zubari

Prozor Lekovi sadrži 5 TextBox-eva za Ime, Prezime, JMBG, Broj Telefona i Email.

Buttoni (Dugmadi) za dodavanje (zeleno), brisanje (crveno) i izmenu (žuto) koje omogućavaju unos, brisanje i izmenu iz baze podataka preko prethodno navedenih alatki.

Button Nazad koji se nalazi u gornjem levom uglu, vraća na glavni meni.

DataGrid omogućava tabelarni prikaz svih polja tabele Zubar iz baze podataka, te omogućava veoma pregledan prikaz svih informacija o Zubarima.



*Slika 8.0 - Izgled prozora Zubari*

# Poslovna pravila

## 4.1 Pravila za lekove

1) Naziv leka:

* Obavezno polje – ne sme biti prazno;
* Može sadržati do 100 karaktera;
* Dozvoljeni su samo slova (A–Z, č, ć, š, đ, ž), cifre, razmak i znak -;
* Zabranjeni su specijalni znakovi.

2) Proizvođač leka:

* Obavezno polje – ne sme biti prazno;
* Može sadržati do 100 karaktera;
* Dozvoljeni su samo slova (A–Z, č, ć, š, đ, ž), cifre, razmak i znak -;
* Zabranjeni su specijalni znakovi.

3) Jačina leka:

* Obavezno polje – ne sme biti prazno;
* Mora biti broj praćen opcionalnom jedinicom: mg, g, ml, mcg;
* Primer validnog formata: *500mg*

4) Doziranje leka:

* Obavezno polje – ne sme biti prazno;
* Može sadržati do 100 karaktera;
* Dozvoljeni su slova (A–Z, č, ć, š, đ, ž), cifre, razmak, znak - i znak /;
* Zabranjeni su specijalni znakovi.

## 4.2 Pravila za pacijente

1) Ime pacijenta:

* Obavezno polje – ne sme biti prazno;
* Maksimalno 50 karaktera;
* Dozvoljena su samo slova (A–Z, č, ć, š, đ, ž) i razmak;
* Brojevi i specijalni znakovi nisu dozvoljeni.

2) Prezime pacijenta:

* Obavezno polje – ne sme biti prazno;
* Maksimalno 50 karaktera;
* Dozvoljena su samo slova (A–Z, č, ć, š, đ, ž) i razmak;
* Brojevi i specijalni znakovi nisu dozvoljeni.

3) JMBG:

* Mora sadržati tačno 13 cifara;
* Ne sme biti prazno;
* JMBG mora biti jedinstven – sistem proverava da li već postoji u bazi.

4) Broj telefona:

* Obavezno polje – ne sme biti prazno;
* Maksimalno 20 karaktera;
* Dozvoljeni karakteri: cifre, +, -, (, ), razmak;
* Primeri validnih formata:

*+381641234567*

*064-123-4567*

*0631234567890123456*

5) Pol:

* Obavezno polje – ne sme biti prazno;
* Ako je pol = "M" → polje Trudnoća ne može biti true (validacija greške: "Kako?!").

6) Trudnoća:

* Polje ne sme biti null;
* Dozvoljeni odgovori (pozitivni): "da", "trudna", "jeste", "yes", "true", "1";
* Dozvoljeni odgovori (negativni): "ne", "nije", "no", "false", "0", "/".

7) Broj zuba:

* Ne sme biti manji od 0;
* Sistem podržava i vrednosti preko 50 (mogući genetski poremećaji).

8) ID Zubara:

* Mora biti veći od 0;
* Obavezan je izbor zubara.

## 4.3 Pravila za termine

1) Datum termina:

* Datum ne može biti u prošlosti;
* Datum ne može biti zakazan više od godinu dana unapred.

2) Vreme termina:

* Vreme termina mora biti između 08:00 i 20:59.

3. Vrsta usluge:

* Obavezno polje – ne sme biti prazno.

4) Konflikt termina:

* Termin ne može biti zakazan na isti dan u razmaku manjem od 45 minuta.

## 4.4 Pravila za preglede

1) Izveštaj:

* Polje ne sme biti prazno;
* Ako je prazno → greška: "Datum sledećeg pregleda ne može biti u prošlosti."

2) ID Termina:

* Mora biti veći od 0;
* Termin mora postojati u tabeli termina.

3) ID Leka:

* Mora biti veći od 0.

## 4.5 Pravila za zubare

1) Ime zubara:

* Obavezno polje – ne sme biti prazno;
* Maksimalno 50 karaktera;
* Dozvoljena su samo slova (A–Z, č, ć, š, đ, ž) i razmak;
* Brojevi i specijalni znakovi nisu dozvoljeni.

2) Prezime zubara:

* Obavezno polje – ne sme biti prazno;
* Maksimalno 50 karaktera;
* Dozvoljena su samo slova (A–Z, č, ć, š, đ, ž) i razmak;
* Brojevi i specijalni znakovi nisu dozvoljeni.

3) JMBG:

* Mora sadržati tačno 13 cifara;
* Ne sme biti prazno;
* JMBG mora biti jedinstven – sistem proverava da li već postoji u bazi.

4) Broj telefona:

* Obavezno polje – ne sme biti prazno;
* Maksimalno 20 karaktera;
* Dozvoljeni karakteri: cifre, +, -, (, ), razmak;
* Primeri validnih formata:

*+381641234567*

*064-123-4567*

*0631234567890123456*

5) Email:

* Obavezno polje – ne sme biti prazno;
* Mora biti u validnom formatu;
* Primer validnog formata: *korisnik@domen.com*.

# Ključni delovi koda

Prilikom prijave, proverava se podudaranje korisničkog imena i lozinke. Dodatno, ukoliko se podudaraju otvara se novi Window – Meni, a ukoliko se ne podudaraju dobijamo ispis greške na ekranu (*Listing 1.0*).

|  |
| --- |
| private void Button\_Click(object sender, RoutedEventArgs e)  {  string Username, Password;  Username = TextboxKorisnickoIme.Text;  Password = PasswordfieldLozinka.Password;  if (Username == "Doktor" && Password == "1234")  {  Meni menipage = new Meni();  menipage.Show();  this.Close();  }  else  {  MessageBox.Show("Pogrešno korisničko ime ili lozinka!");  }  } |

*Listing 1.0 – Provera podudaranja korisničkog imena i lozinke, otvaranje novog Windowa ili poruka o grešci*

Prilikom inicijalizacije svakog Window-a potrebno je da se doda deo koda koji stavlja taj Window na centar ekrana (*Listing 2.0*).

|  |
| --- |
| public MainWindow()  {  InitializeComponent();  WindowStartupLocation = WindowStartupLocation.CenterScreen;  } |

*Listing 2.0 Inicijalizacija MainWindow-a i centriranje na ekranu*

Fajl ZubarPravila predstavlja ključnu vezu između prezentacionog sloja i sloja podataka. Njegova uloga je da validira korisnički unos i osigura da svi podaci budu u skladu sa poslovnim pravilima definisanim u softverskoj arhitekturi. Na taj način se obezbeđuje konzistentnost, integritet i predvidivo ponašanje sistema (*Listing 3.0*).

|  |
| --- |
| public class ZubarPravila  *// Klasa za poslovna pravila vezana za entitet Zubar*  {  *// Privatno polje (repozitorijum za rad sa bazom)*  private readonly ZubarDB \_repo;  *// Konstruktor*  public ZubarPravila()  {  \_repo = new ZubarDB();  }  *// Metoda: vraća sve zubare*  public List<Zubar> VratiSveZubare() => \_repo.GetAll();  *// Metoda: dodavanje zubara*  public Obavestenje DodajZubara(Zubar zubar)  {  var ispravan = ProveraPodatakaZaZubara(zubar);  if (ispravan.Uspeh)  \_repo.Insert(zubar);  return ispravan;  }  *// Metoda: izmena zubara*  public Obavestenje IzmeniZubara(Zubar zubar)  {  var ispravan = ProveraPodatakaZaZubara(zubar);  if (ispravan.Uspeh)  \_repo.Update(zubar);  return ispravan;  }  *// Metoda: brisanje zubara*  public void ObrisiZubara(int id) => \_repo.SoftDelete(id);  *// Interna metoda: pravila i validacija*  internal Obavestenje ProveraPodatakaZaZubara(Zubar zubar)  {  *//\*\*\*\*PRAVILA ZA INSERT/UPDATE ZA PACIJENTA\*\*\*\**  *//Lista provera (pravila) da se vidi da li je sve ispravno*  *//IF(uslov==true) -> pada*  *//Na kraju ako nigde nije pao onda se vraća Obavestenje sa uspehom (true)*  *//Samo treba da se prilagodi na WPF-u kada da proverava input-e*  *//i da prikaze poruku kada nađe grešku*  return new Obavestenje { Uspeh = true, Poruka = "Uspešno je dodat entitet" };  }  } |

*Listing 3.0 – ZubarPravila – pravila poslovne logike za entitet Zubar*

Klasa obaveštenje je format koji koristimo da obavestimo da je željena radnja izvršena ili ne. (*Listing 4.0*).

|  |
| --- |
| *// Klasa za vraćanje poruke o uspehu ili grešci*  public class Obavestenje  {  *// Svojstvo: označava da li je operacija uspešna*  public bool Uspeh { get; set; }  *// Svojstvo: tekstualna poruka (uspeh ili greška)*  public string Poruka { get; set; }  } |

*Listing 4.0 Klasa obaveštenje – poruka o uspešnosti radnje*

Ako neki od unosa krši poslovna pravila onda će vratiti uspeh false (*Listing 5.0*).

|  |
| --- |
| if (pacijent.BrojZuba < 0)  return new Obavestenje  {  Uspeh = false,  Poruka = "-Zuba?!"  };  *// Ako je sve u redu, vraća se obaveštenje o uspehu*  return new Obavestenje  {  Uspeh = true,  Poruka = "Uspešno je dodat pacijent"  };  *// nema poruke* |

*Listing 5.0 Provera validacije broja zuba pacijenta*

U WPF se return Obavestenje koje potvrđuje da li je pao u nekom od pravila. WPF prikazuje ovaj tip poruke samo se prekrši neko od pravila. (*Listing 6.0*).

|  |
| --- |
| *// Dodavanje pacijenta kroz repozitorijum*  var poruka = \_pacijentRepo.DodajPacijenta(pacijent);  // Provera da li je operacija uspešna  if (!poruka.Uspeh)  {  MessageBox.Show(poruka.Poruka);  return;  }  *// Ako je uspešno, osvežava se DataGrid i poništava unos u poljima*  binDataGrid();  ponistiUnosTxt(); |

*Listing 6.0 Obaveštenje na UI o uspešnosti radnje*

U nastavku je prikazan deo koda kojim se implementira poslovno pravilo za zakazivanje termina u zubnoj ordinaciji. Pravilo podrazumeva proveru da li već postoji zakazani termin istog dana, kod istog zubara, u vremenskom razmaku manjem od 45 minuta. Na ovaj način se sprečava dupliranje termina i omogućava pravilno funkcionisanje sistema zakazivanja.. (*Listing 7.0*).

|  |
| --- |
| *// Preuzimanje vrednosti novog termina*  var vremeNovogTermina = termin.Vreme;  var datumNovogTermina = termin.Datum;  var IDZubara = termin.IDZubara;  *// Preuzimanje svih zakazanih termina iz baze*  var zakazaniTermini = \_repo.GetAll();  *// Provera da li već postoji termin kod istog zubara u razmaku manjem od 45 minuta*  bool postojiKonflikt = zakazaniTermini.Any(t =>  t.Datum == datumNovogTermina && t.IDZubara == IDZubara &&  Math.Abs((t.Vreme - vremeNovogTermina).TotalMinutes) < 45);  if (postojiKonflikt)  {  return new Obavestenje  {  Uspeh = false,  Poruka = "Već postoji termin na isti dan u razmaku manjem od 45 minuta."  };  } |

*Listing 7.0 Provera konflikta prilikom zakazivanja termina*

# Zaključak

Aplikacija za Zubne Ordinacije nazvana “**DentalKeep**” služi za čuvanje, i tabelarni (čist) prikaz najbitnijih informacija za jednu Zubnu Ordinaciju. Čistim i lakim interfejsom postiže se lak i brz rad, a podaci se čuvaju u Bazi podataka što je u današnjici veoma bitno.

Dalji razvoj aplikacije će teći u smeru kompletnog pretvaranje termina u pregled. To jest automatizovano brisanje termina iz baze podataka ukoliko je njegov ID stavljen u pregled. Termin treba da postane pregled i da se posle nekog vremena briše iz baze podataka, on je samo tu kao organizator vremena za doktore. Pregled je tu za evidenciju.

# Literatura

1. <https://sh.wikipedia.org/wiki/Microsoft_SQL_Server>

2. <http://en.wikipedia.org/wiki/Microsoft_Visual_Studio>

3. <https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29>

4. <https://wpf-tutorial.com/about-wpf/what-is-wpf/>

6. <http://www.tfzr.uns.ac.rs/Predmet/baze-podataka-1>

7. <https://www.youtube.com/watch?v=oU-nRo40YYU&t>

8. <https://www.youtube.com/watch?v=oSeYvMEH7jc&t>