ВИСОКА ШКОЛА СТРУКОВНИХ СТУДИЈА ЗА
ИНФОРМАЦИОНЕ И *КОМУНИКАЦИОНЕ* ТЕХНОЛОГИЈЕ

АПЛИКАЦИЈА ЗА ВОЂЕЊЕ ПОСЛОВАЊА СЕРВИСНОГ ОДЕЉЕЊА

ЗАВРШНИ РАД

Ментор:

мр. Миланко Краговић

Студент:

Вукићевић Алоша 348/13

Београд, 2017.

ВИСОКА ШКОЛА СТРУКОВНИХ СТУДИЈА ЗА
ИНФОРМАЦИОНЕ И *КОМУНИКАЦИОНЕ* ТЕХНОЛОГИЈЕ

Смер: Интернет технологије

Предмет: ООП - "C# 2"

Тема: **Апликација за вођење пословања сервисног одељења**

Ментор:

мр. Миланко Краговић

Студент:

Вукићевић Алоша 348/13

Београд, 2017.

САДРЖАЈ

1. Увод

У времену у којем живимо рачунар је већ увелико зашао у скоро све сфере нашег живота, почевши од "desktop" рачунара до мобилног уређаја. Како се рачунари користе код куће тако се користе и у пословном окружењу, с тим што се његова корист најбоље може видети баш у пословном окружењу.

Свака фирма одређује свој начин вођења пословања.Током рада сваког предузећа свакодневно се обрађује велика количина података о којима је потребно водити евиденцију и одредити начин чувања тих података, као и документација која настаје унутар фирме као резултат пословања.

У оквиру овог рада биће представљено једно од могућих решења у облику "desktop" аппликације. Сама апликација је програмски пакет намењен бољој организацији пословања и вођењу посла у сервисном одељењу у предузећи "Техноклиник". Омогућава једноставније вођење евиденције о клијентима и њиховим уређајима који се сервисирају, преглед и ажурирање инвентара резервних делова, и рад са извештајима.

2. Циљеви

Циљеви пројектовања софтверског пакета "Сервис" је да се унапреди ефикасност рада у серисном одељењу предузећа "Техноклиник"у смислу боље организације рада и евиденције одређене документације која настаје као резултат пословања. Боља организација рада подразумева да корисник може са лакоћом да пронађе одређени податак који је везан за клијента или уређај који је примљен на сервис односно чији реверс има активан статус.

Једноставност у управљачком смислу, да корисник може са лакоћом да се креће кроз контроле корисничког интерфејса и да манипулише са њима.

Боља евиденција и поузданије чување неопходних података о клијентима и њиховим уређајима. Подаци требају у сваком тренутку да буду видљиви да би корисник могао да има увид у њих. Подацима је могуће манипулисати (додати, заменити, обрисати) независно за клијента, уређај или инвентарски артикал.

Боља евиденција инвентара резервих делова путем каталошког броја и праћење количинских промена делова.

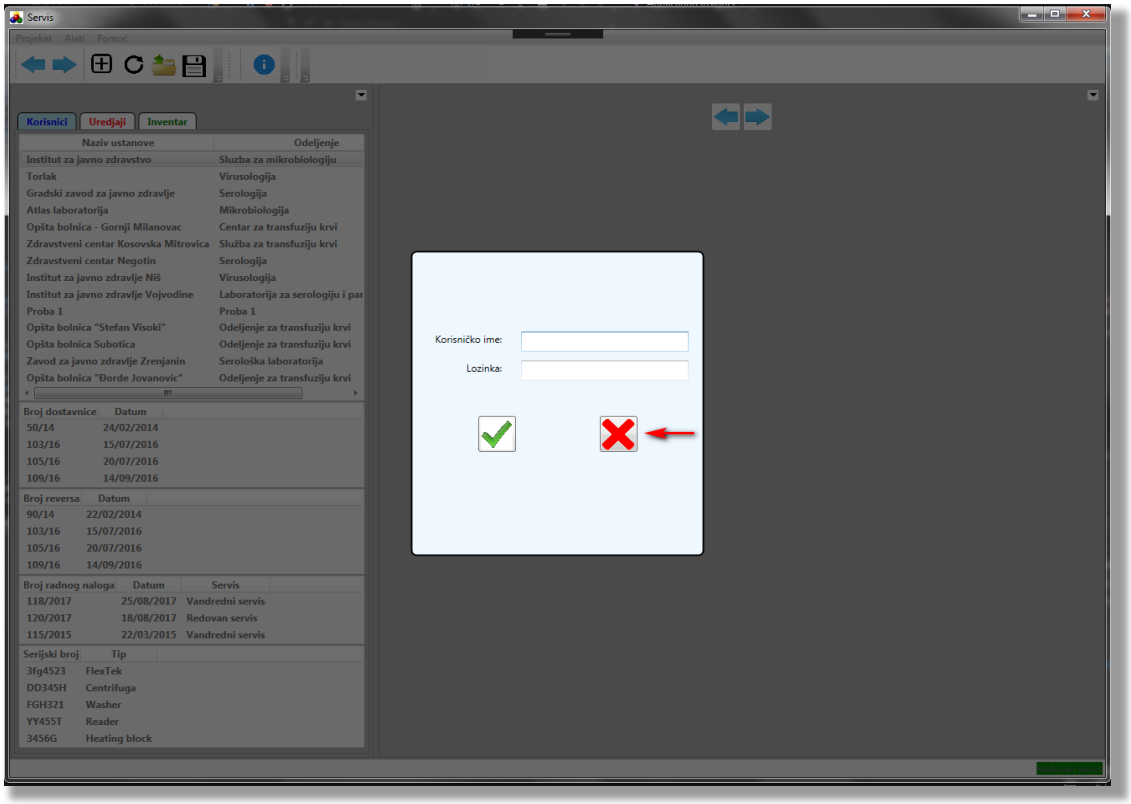
Документи одељења за сервис су: налози, доставнице и радни налози. Циљ је да сваки од наведених докумената који има свој регистарски број преко кога се врши архивирање и да буде приказано коме он припада. Преглед докумената и активни документи са њиховим детаљима могу да се виде у самом едитору односно креираним табелама. С обзиром да се ради о званичним документима које је потребно оверити и архивирати, било је потребно направити функционалност прослеђивања података из табеле апликације у "word" документ.

3. Функционалана спецификација

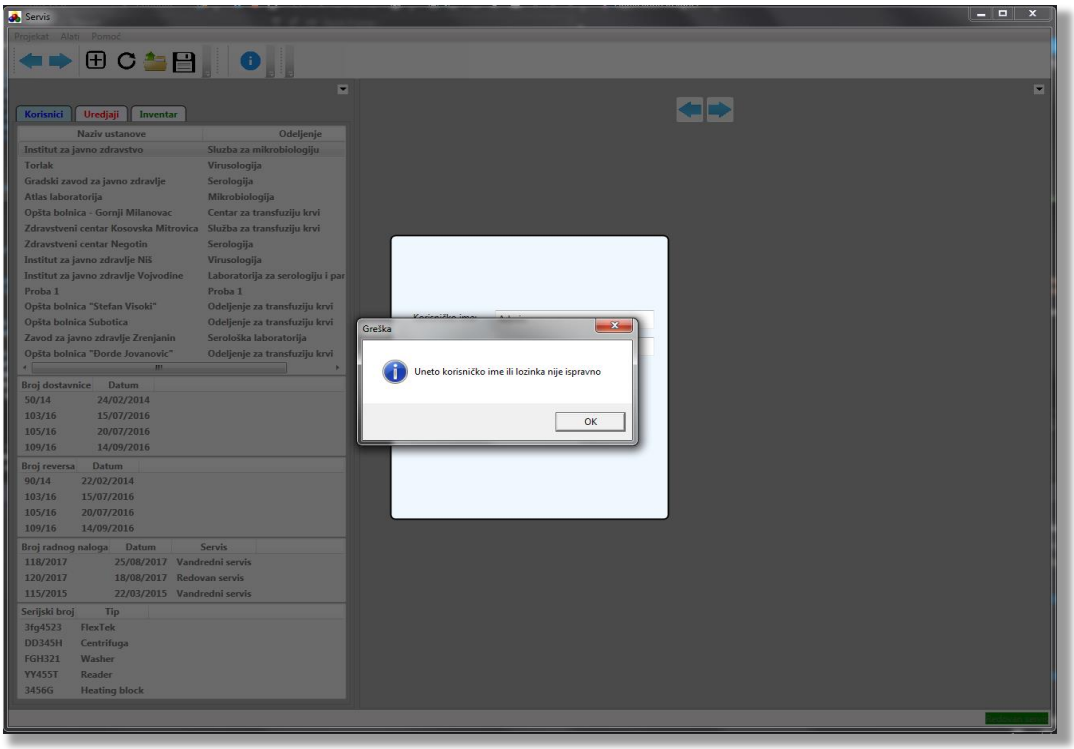
Програмски пакет се састоји из једне апликације "Сервис". Апликација поседује графички кориснички интерфејс који се састоји од једног главног прозора, неколико модалних дијалога који служе за пријављивање корисника пре уласка у апликацију, креирање новог корисничког налога, за унос и ажурирање новог клијента, уређаја и резервног дела у инвентар.

3.1 Дијалог за пријављивање

Стартовањем апликације прво се појављује модални дијалог за пријављивање корисника на кориснички налог (Слика 1), приликом уноса корисничког имена и лозинке врши се провера преко "SQL" упита да ли постоји такав корисник у бази података, у случају да је корисничко име или лозинка погрешно унето, дијалог обавештава корисника о томе (Слика 2). У случају да корисник не зна корисничко име или лозинку, кликом на дугме поништи које је означено са црвеним иксом(Слика1), позива Модални сервис који затвара дијалог и родитељски прозор. Ако је кориник исправно унео тражене корисничке параметре, кликом на дугме потврди, модални сервис затвара дијалог и главни прозор је омогућен за даљи рад корисника.



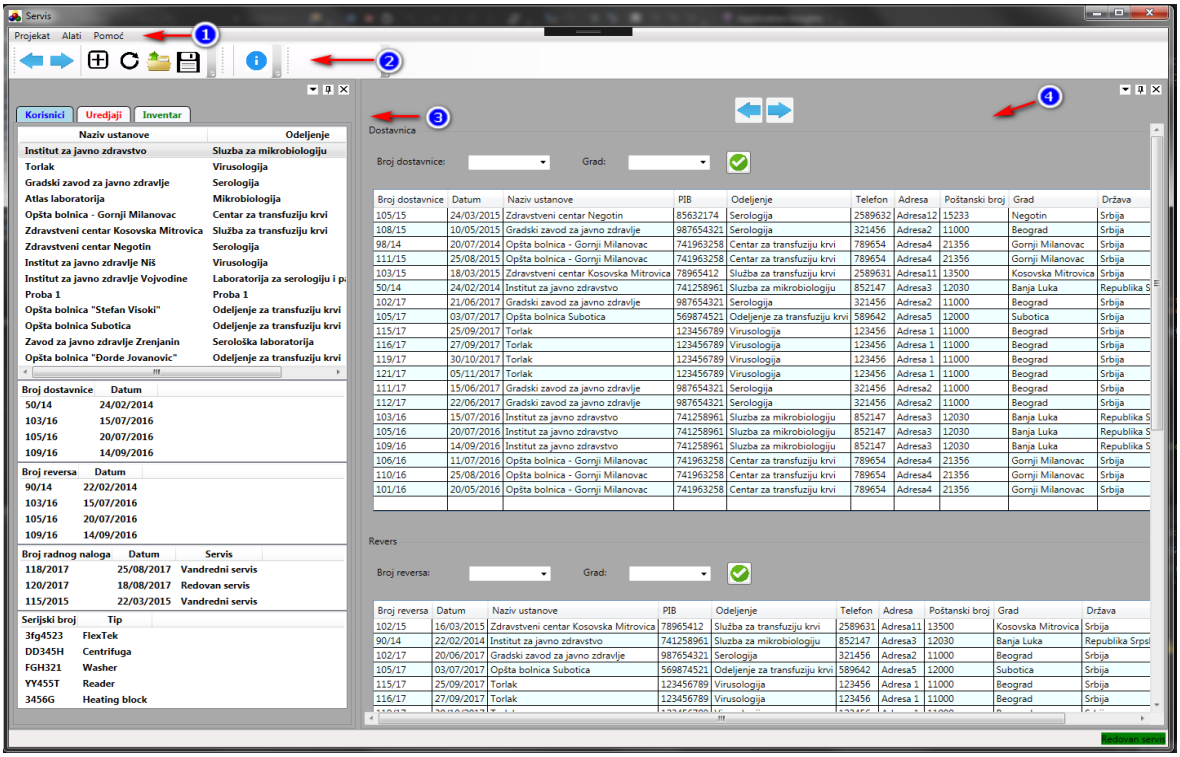
Слика 1. Приказ главног прозора апликације Сервис пре пријављивања на кориснички налог



Слика 2: Приказ дијалога који обавештава да је погрешно унето корисничко име и лозинка

3.2 Главни прозор

После успешног пријављивања на кориснички налог отвара се главни прозор апликације (Слика 3) који се састоји од наслова, [**1**]менија (menu bar), [**2**]палете са алаткама (toolBar) и [**3**][**4**]радне површине која је подељена на две секције.



3.2.1 Мени бар [1]

Мени бар апликације садржи подмените: **Projekat, Alat, Pomoc.**

Подмени **Projekat** има опције:

- **Novi unos** отвара главни дијалог на коме се налазе дугмићи преко којих се отварају дијалози за унос корисника, уређаја и инвентара и затвара текући дијалог (Слика 4).
- **Otvori** отвара одређени документ у **Word** едитору. Документ се односи на радни налог, реверс и доставницу.
- **Sačuvaj** покреће контекст **SaveChanges()** методу која чува све промене насатале у бази података.
- **Izađi** прекида све процесе и излази из апликације.

Подмени **Alat** има опције:

- **Podešavanja** опција отвара дијалог са две картице, на првој картици се налази checkBox опција за приказивање палете са алаткама (**toolBar**), на другој картици се налази палета са бојама **ColorPicker** за бирање позадинске боје и боје слова.
- **Kreiranje naloga** опција отвара дијалог за креирање новог корисничког налога, у дијалогу се налазе текстуална поља **TextBox** за унос имена и презимена корисника као и корисничког имена и лозинке. Обавезна текстуална поља имају основну валидацију за потврду уноса.

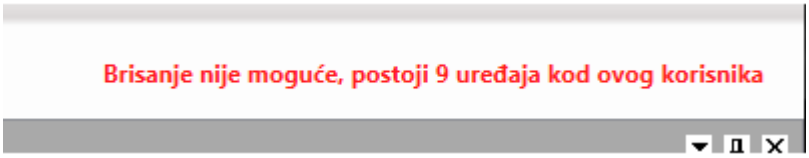
Подмени **Pomoc** има опције:

- **Dokumentacija** опција отвара документ о спецификацији апликације у подразумеваном **PDF** читачу који је инсталиран на рачунару са кога се покреће апликација.
- **O Servis aplikaciji** опција отвара дијалог са информацијама о самој апликацији

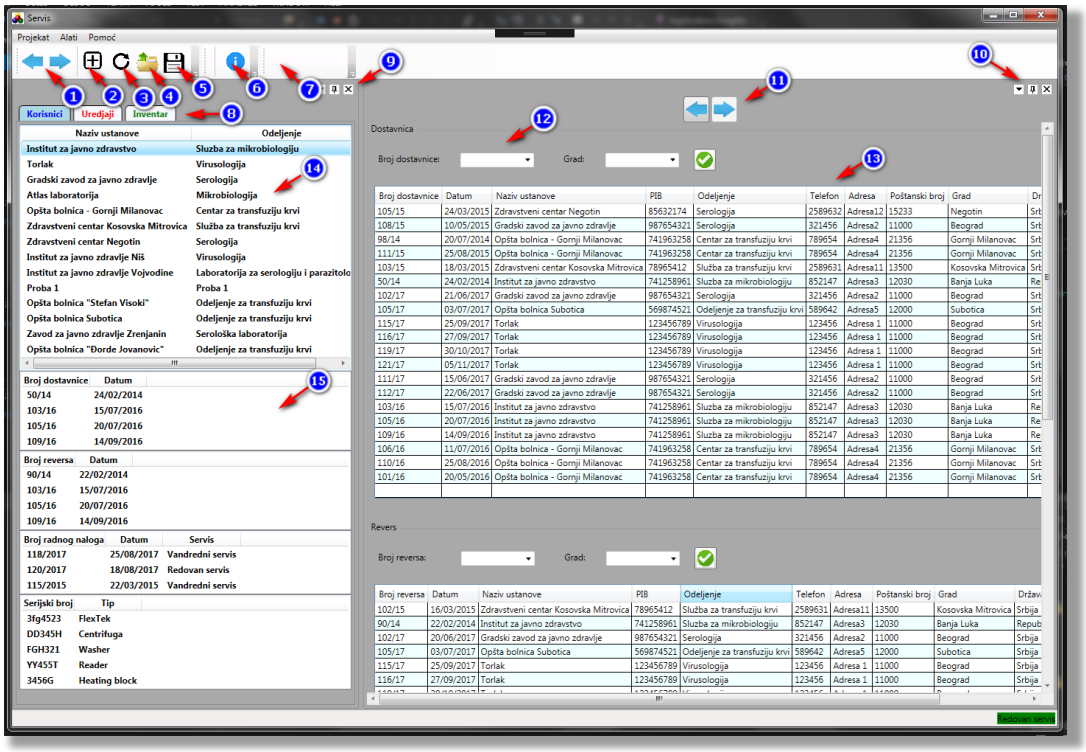
3.2.2. Палета са алаткама [2]

Палета са алаткама садржи опције **Nazad** и **Napred** које представљају **навигацију, Unesi** , **Osveži, Otvori, Sačuvaj, O Servis aplikaciji, Poruka.** Одређеним командама из менија је омогућен бржи приступ преко палете са алаткама.

- [**1**]**Навигација** са опцијама **Nazad i Napred** служи да се мењају [**8**]картице које се налазе у левом делу главног прозора Слика 5.
- [**2**]**Unesi** дугме има исту функцију као и **Novi unos** у мени бару, да отвара дијалог са дугмићима за унос корисника, уређаја и инвентара, Слика 5.
- [**3**]**Osveži** опција позива методу **OsveziPodatke()** која поново учитава податке који су приказани у листи са леве стране главног прозора и у табелама са десне стране прозора, Слика 5.
- [**4**]**Otvori** дугме има исту функцију као опција **Otvori** у мени бару, да отвори дукумент у **Word** editoru, Слика 5.
- [**5**]**Sačuvaj** дугме има исту функцију као опција **Sačuvaj** у мени бару да ажурира унету измену података у бази података, Слика 5.
- [**6**]**O Servis aplikaciji** дугме отвара дијалог са информацијама о самој апликацији, Слика 5.
- [**7**]**Poruka** је део палете који је резервисан за поруке које се приказују када апликација шаље у току интеракције корисника са контролама у корисничком интерфејсу, Слика 5.



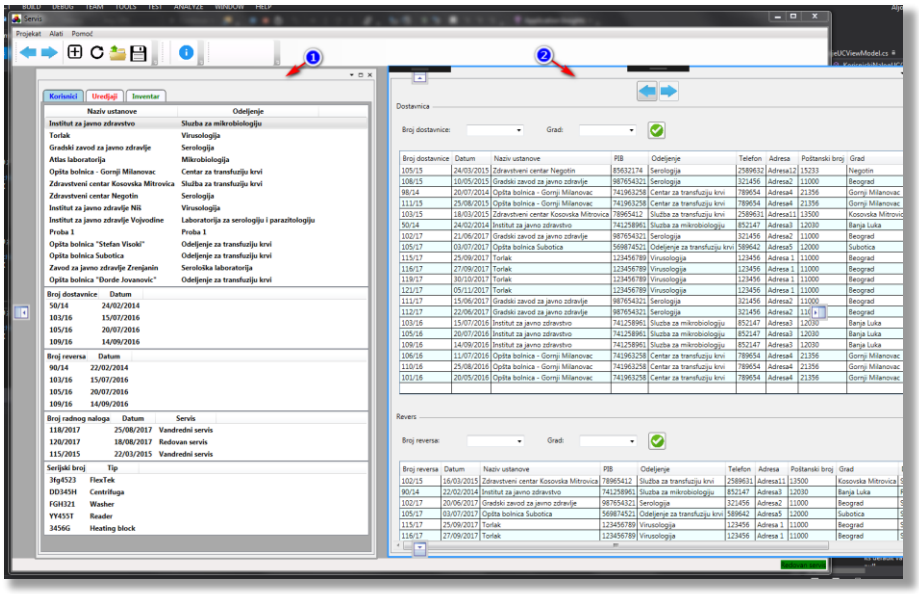
Слика 4: Пример поруке



Слика 5: Главни прозор апликације са означеним контролама

3.2.3. Главни прозор - радна површина

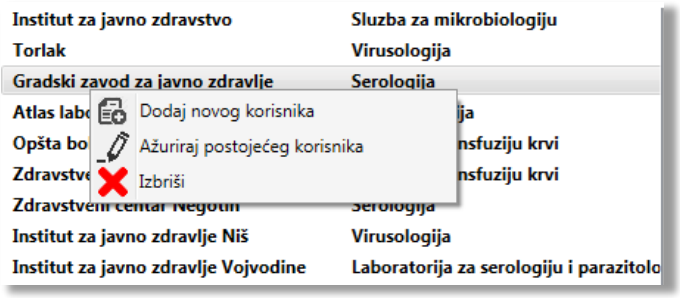
Радна површина главног прозора се дели на две секције, део са картицама (tab control) и дела са табелама и филтерима за претрагу. С обзиром да апликација поседује контролу за издвајање одређених делова прозора из контекста главног прозора **AvalonDock control**, она омогућава персонализацију самог прозора тако да је могуће померати [**1**]леву и [**2**]десну секцију и подешавати по жељи корисника, Слика 6.



Слика 6: Главни прозор апликације са издвојеним секцијама прозора уз помоћ AvalonDock контроле

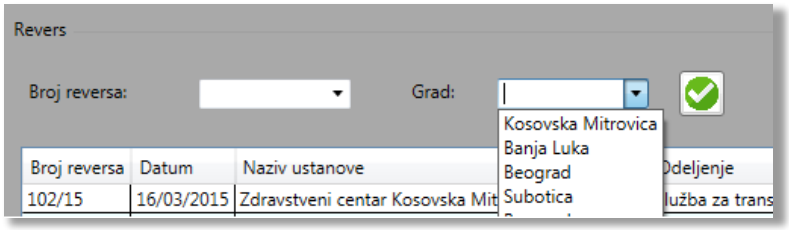
У секцији са картицама налазе се [**8**]три картице, **Korisnici, Уређаји** и **Inventar,** Слика 6. На картици **Korisnici** налази се [**14**]листа клијената са још неколико листа испод које служе за приказ [**15**]детаља, односно података који су везани за њега. Кликом на неког од клијената се приказује сва документација која је заведена под тим клијентом као и уређаји који су у његовом власништву. Ова **MasterDetail** шема је направљена са циљем лакшег проналажења заведене документације у архиви. Картица **Уређаји** приказује инфорамције о уређајима, битни детаљи везани за уређај су информације о редовним сервисима односно вандредним сервисима као и број радног налога под којим се воде. Приказују се и детаљи у смислу датума обављеног сервиса и датума предстојећег сервиса код редовних сервиса. Картица **Inventar** приказује резервне делове који су заведени под катлошким бројем, да ли су на стању и количину делова. Детаљи везани за резервне делове односно инвентар су типови уређаја у смилу који уређај може да користи одређен део, јер неки делови могу да се користе за више уређаја.

Десним кликом на неку од листа отвара се падајући мени **ContextMenu**, преко кога може да се унесе нови корисник, ажурира постојећи или да се обрише селектовани, исто важи и за уређај и резервни део. У случају да постоји клијент или уређај са детаљима или везаним подацима исписује се порука о забрани брисања и операција се прекида, Слика 6.



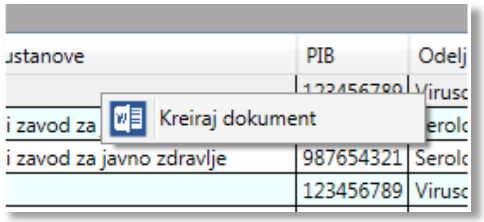
Слика 6: Падајући мени

Смењивањем картица у левој секцији, смењују се и **DataGrid** табеле у другој (десној) секцији радне површине главног прозора. За сваку картицу у десној страни се приказује посебан сет табела који је у вези са подацима из леве секције. Изнад сваке табеле се налазе **ComboBox** [**12**]филтери за претрагу табеле, свака табела има по два критеријума претраге.



Кликом на картицу **Korisnici** отварају се табеле доставнице, реверси и радни налози. Табеле представљају активне податке са детаљима о сваком документу и ком клијенту припада. Критеријуми за претрагу табеле доставница су број доставнице и град клијента на кога се односи доставница, за табелу реверс број реверс и град клијента и за радни налог број радног налога и врста сервиса.

Десним кликом на било коју од табела са документима отвара се падајући мени са командом креирај документ, ова комада прослеђује податке из табеле у **Word** текст едитор у већ припремљена документа у виду **Word template-a,** Слика 7.



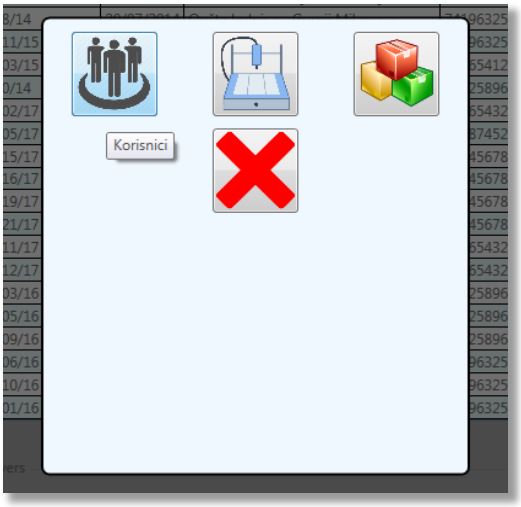
Слика 7: Команда креирај документ

Кликом на картицу **Уређаји** отварају се табеле у којима су приказани вандредни сервиси и редовни сервиси са детаљима о самом сервису. Датуми предстојећих редовних сервиса су обојени зеленом бојом. Критеријуми за претрагу табеле доставница су датум сервиса и број радног налога клијента на кога се односи доставница.

Кликом на картицу **Inventar** отварају се табела на којој су приказани детаљи о уређајима . Критеријуми претраге су Серијски број уређаја и тип уређаја.

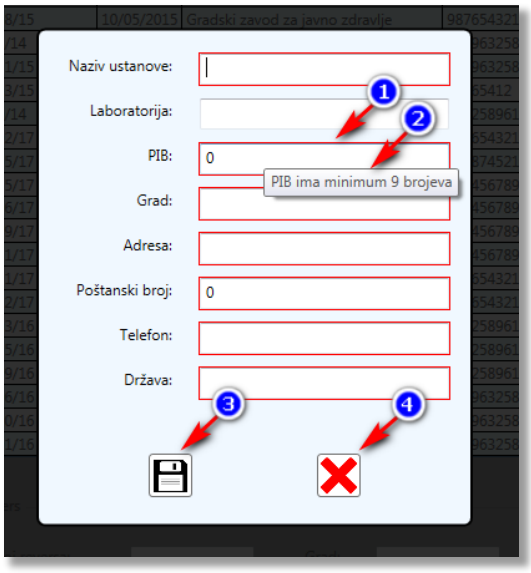
3.2.3. Дијалози за унос

Преко главног дијалога за унос, отварају се дијалози за унос корисника, уређаја и инвентара. **Novi unos** опција у мени бару и **Unesi** опција у палети са алаткама отвара главни дијалог на коме се налазе дугмићи преко којих се отварају дијалози за унос клијента, уређаја и инвентара,Слика 8.



Слика 8: Приказ дијалога са дугмићима за унос корисника, уређаја и инвентара

С обзиром да су дијалози идентични у смислу уноса и валидације приказан је само дијалог за унос клијента, Слика 9.



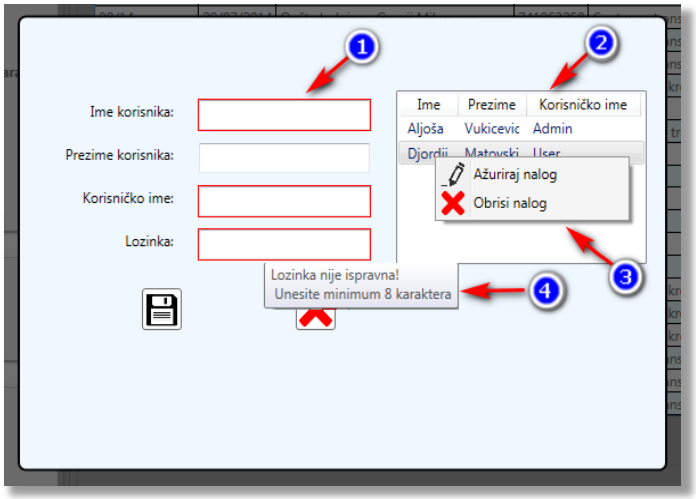
Слика 9: Приказ дијалога за унос корисника

Иницјално при отварању дијалога појављује се [**1**]црвени оквир - бордура око текстуалног поља и преласком курсора миша преко текстуалног поља појављује [**2**]порука која означава да у поља нису исправно унети подаци,Слика 9. Када корисник унесе

тражене податке унутар текстуалних поља, црвена бордура нестаје и кликом на дугме [**3**]**Unesi** уносе се подаци у базу података и дијалог се затвара, у случају да корисник жели да изађе из дијалога или да поништи унос, може да кликне на дугме [**4**]поништи.

Дијалог за креирање корисничког налога

Дијалог се отвара преко опције **Kreiranje naloga** која се налази у **Alat** опцији у менију. У дијалогу се налазе [**1**]текстуална поља **TextBox** за унос имена и презимена корисника као и корисничког имена и лозинке, обавезна текстуална поља имају основну валидацију за потврду уноса, у сличају да у поља нису исправно унети подаци [**1**]бордура текстуалног поља остаје црвене боје и преласком курсора миша се појављује **ToolTip** [**4**]порука о грешци. Поред текстуалних поља се налази [**2**]листа **ListView** на којој су приказани унети корисници, десним кликом на неког од корисника орвара се [**3**]падајући мени **ContextMenu** преко кога је могуће ажурирати параметре већ унетог корисника или избрисати корисника ,Слика 5.



Слика 10: Приказ дијалог за креирање нових корисничких налога

4. Коришћене технологије

Током развоја апликације за вођење пословања сервисног одељења коришћене су следеће технологије и развојни алати:

- .**NET Framwork v4.6.0** окружење за развој апликација намењених за рад на **Windows** оперативним системима
- Коришћена је развојна верзија **Visual Studio SDK 4.6**
- Као развојно окружење апликације је коришћен **Microsoft Visual Studio 2015 и 2017**
- **WPF(Windows Presentation Foundation)** део **NET Framworka,** за развијање графичког корисничког интерфејса апликаиције коришћен је **XAML (zammel) - Extensible Application Markup Language** који је део стандардног **WPF** пакета. **XAML** је декларативан језик заснован на **XML-y,** а користи се за иницијализацију структурираних вредности и објеката.
- Апликација је писана у програмском језику **C#** и **T SQL i Linq** језици за писање упита над базом података
- **Microsoft SQL Server v11.0.3000.0** пакет за управљање релационом базом података
- **Microsoft SQL Server 2012 Management Studio** - окружење за развој базе података у смислу креирања релационих табела и функција, у овом окружењу је развијена "**Tehnoklinik.db**" база података
- **Entity Framwork v6.0** релацијски мапер који решава проблем приступног кода за приступ подацима у бази података тако што креира модел са класама у апликацији по релацијском моделу базе података. Коришћена је метода **Code first on existing database.**

Коришћени су и специфични пакети који су преузети са **NuGet Package** менагера и нису у стандарднот издању **C# i WPF-a:**

- **Extended Wpf Toolkit v3.0** колекција **Wpf** компоненти и алата за креирање напредних **Windows** апликаиција, коришћена је за елементе у графичком корисничком интерферјсу **XAML-a.**
- **Mvvm Light Libs v5.3.0** колекција помоћних библиотека које олакшавају распоређивање компоненти у апликацији која се заснива на **Model-View-ViewModel** шаблону. У овом пројекту је коришћена за прослеђивање порука и приказ истих између слојева у апликацији конципираној по MVVM шаблону.
- **System.Windows.Interactivity.WPF v2.0.20525** укључује у апликацију нове могућности као што је поновно коришћење делова кода који могу да се региструју на разне објекте. Ова особина је коришћена за регистровање догађаја у **XAML** објектима, као што је нпр. иницијализација текстуалних поља у дијалогу за унос приликом самог отварања дијалога.

5. Пројекат софтвера

Апликација **"Servis"** се састоји из једног пројекта апликације и базе података са којом апликација ради.

5.1 База података

База података **"Tehnoklinik"** је креирана у **Microsoft SQL Server 2012 Management Studio** окружењу. Представља поуздан и скалабилан начин чувања података, састоји се од девет релацијских табела које садрже податке и седам везних табела које представљају M : M везу.

5.1.1 Шема релационе базе података

LogKorisnik (IDLogKorisnik, Ime, Prezime, Uloga, KorisnickoIme, Lozinka),

Uredjaj (IDUredjaj, [S/N], Tip, Detalji, IDKorisnik),

Deo (IDDeo,K_Broj, Naziv, Kolicina, NaStanju, Cena),

Korisnik (IDKorisnik, NazivUstanove, Laboratorija, PIB, Grad, Adresa, PostanskiBroj, Telefon, Drzava),

Dostavnica (IDDostavnica, Datum, BrDostavnica, IDKorisnik),

Revers (IDRevers, Datum, BrRevers, IDKorisnik),

Nalog (IDNalog, BrRadNalog, Komentar, Datum, IDKorisnik),

RedovanServis (IDRedovanServis, DatumServisa, DatumSledecegServisa, OpisPosla, IDNalog),

VandredanServis (IDVandredniServis, DatumServisa, OpisPosla, IDNalog),

UredjajDeo (IDUredjaj, IDDeo),

ReversUredjaj (IDRevers, IDUredjaj),

DostavnicaUredjaj (IDDostavnica, IDUredjaj),

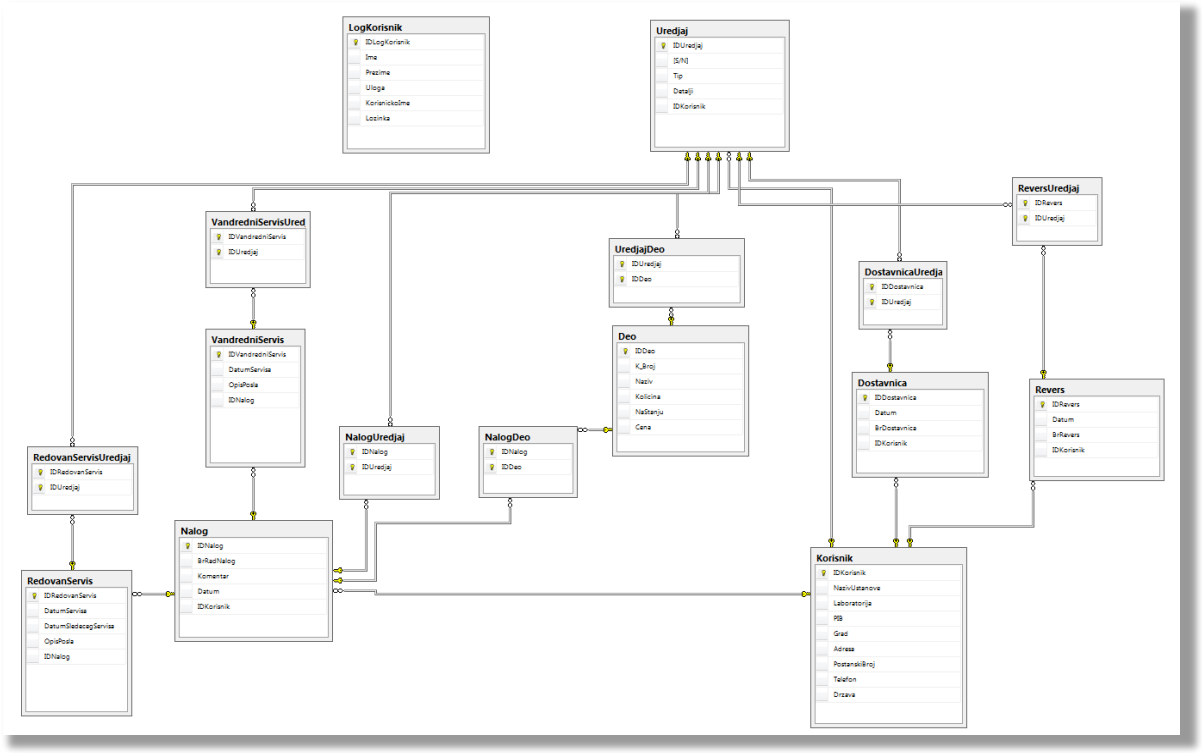
NalogUredjaj (IDNalog, IDUredjaj),

NalogDeo (IDNalog, IDUredjaj),

RedovanServisUredjaj (IDRedovanServis, IDUredjaj),

VandredniServisUredjaj (IDVandredniServis, IDUredjaj)

5.1.2 ЕР дијаграм



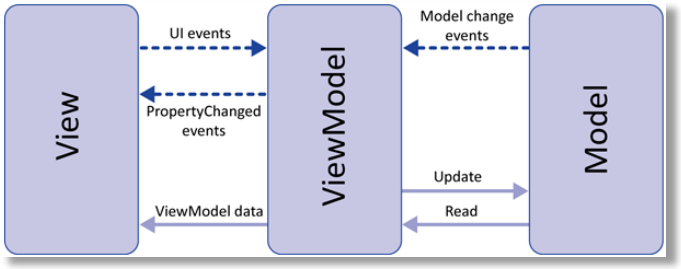
Слика 11: Релацијски дијаграм "Техноклиник" базе података

5.2 Пројекат апликаиције

Идеја је била да се направи апликација за вођење пословања сервисног одељења и да буде урађена у виду десктоп апликације, њена сврха је да се уведе боља организација пословања у сервисном одељењу.

5.2.1 Архитектура апликације

Архитектура која је примењена у развијању апликације је **Model - View-ViewModel** или скраћено **MVVM. Model - View-ViewModel** је софтверски шаблон, идеја је раслојавање. Он олакшава раздвајање **View** слоја корисничког интерфејса од бизнис логике или логике модела. **ViewModel** представља конвертер вредности, одговоран је за обраду објеката из модела тако да објекти могу лако да се презентују. Још једна идеја **MVVM** шаблона је модуларност, у смислу да одређени делови (нпр. класе) пројекта могу да се примењују и у другим пројектима.



Слика 12: Модел MVVM шаблона

5.2.2 Структура апликације

Модел чине класе које су генерисане преко **Entity Framwork** технологије, коришћена је метода **Code first on existing database.** Модел је генерисан тако што се узима релацијски модел базе података као модел по коме се креирају класе у апликацији, Свака класа представља један ентитет базе односно табелу, а свака колона у табели представња објекат који је одређеног типа податка. Генеришу се **POCO (Plain Old CLR Object)** класе**,** што значи да су исте као и било које друге класе и да су едитабилне јер нпр. у случају да се примени метода **Existing Database** класе се генеришу из T4 темплејта и те класе су део .edmx фајла и оне су **partal** што значи да могу да се налазе само унутар .edmx и нису едитабилне, односно кориник не може да мења њихов садржај.

Model представља слој у коме се налазе ентитет класе: **Korisnik, Uredjaj, Deo, Dostavnica, Revers, Nalog, RedovanServis, VandredniServis, LogKorisnik i TehnoklinikModel.** Везне класе се не генеришу јер се унутар генерисаних класа везе дефинису као својства и колекције.

Контекст класа **TehnoklinikModel** представља мапу између модела и базе података, наслеђује **DbContext** klasu која представља примарну класу која интереагује са подацима као са објектима. У класи се налази виртуална колекција **DbSet** која представља колекцију ентитета у контексту и преко ње се уз помоћ упита, може приступати подацима. У **TehnoklinikModel** класи се налазе **DbSet** колекције за сваку класу из модела.

Поред **DbSet** колекција налази се и метода **OnModelCreating** која дефинише конвенције за својства и врши мапирање класа по моделу базе података. У овој методи су дефинисани и атрибути за конфигурисање нпр. **.IsUnicode** метода.

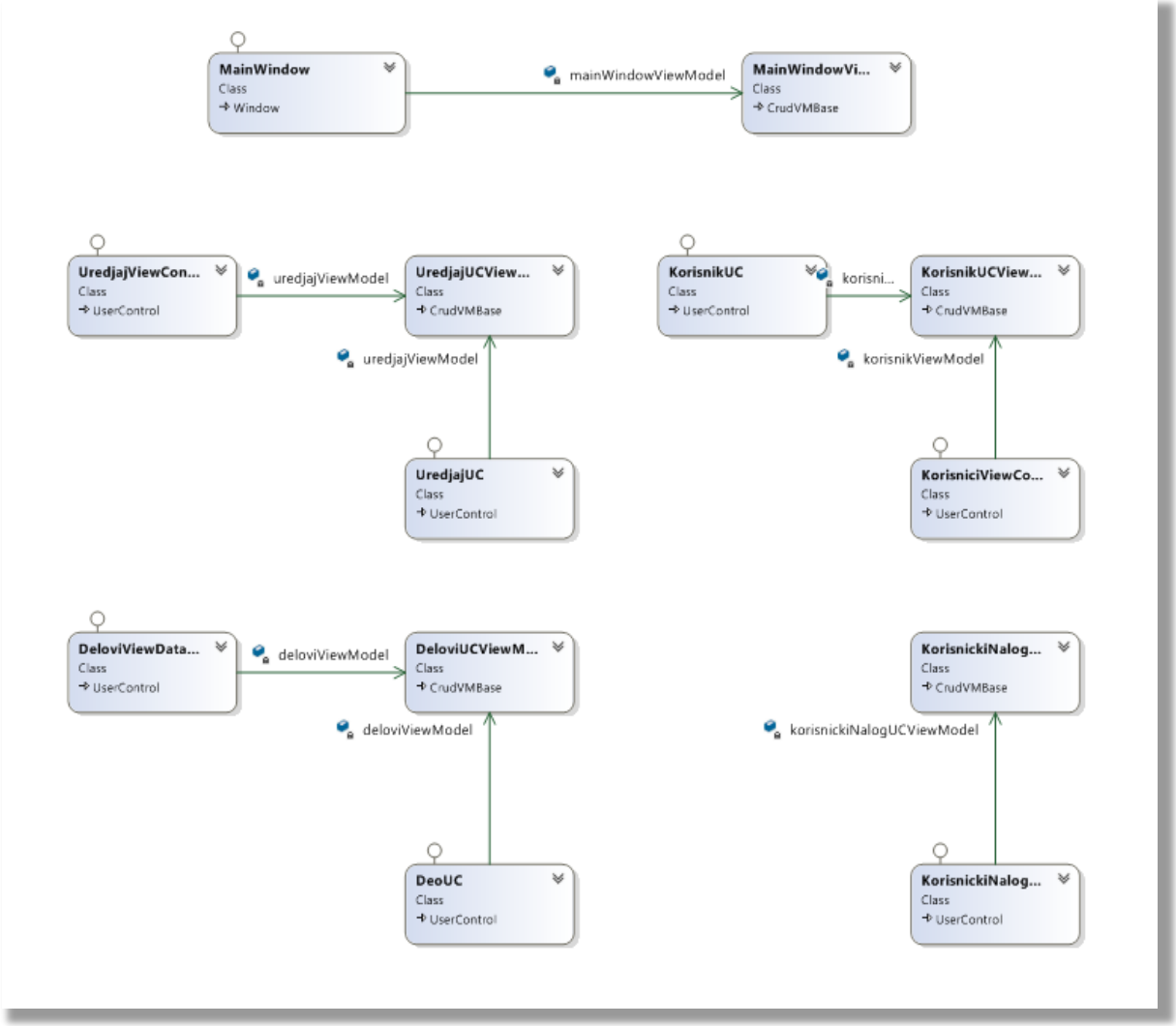
View слој представља презентациони слој у њему се налазе сви прозори и дијалози. Сваки прозор односно **Window** или **UserControl** који ради са подацима у **View** слоју као **DataContext** има **ViewModel,** Слика**:**

Главни прозор и контроле са табелама за приказ података:

1. **MainWindow - DataContext -> MainWindowViewModel**
2. **KorisnikViewControl - DataContext -> KorisnikUCViewModel**
3. **UredjajViewControl - DataContext -> UredjajUCViewModel**
4. **DeoViewControl - DataContext -> DeoUCViewModel**

Дијалози за унос:

1. **KorisnickiNalogUC - DataContext -> KorisnickiNalogUCViewModel**
2. **KorisnikUC - DataContext -> KorisnikUCViewModel**
3. **UredjajUC - DataContext -> UredjajUCViewModel**
4. **DeoUC - DataContext -> DeoUCViewModel**

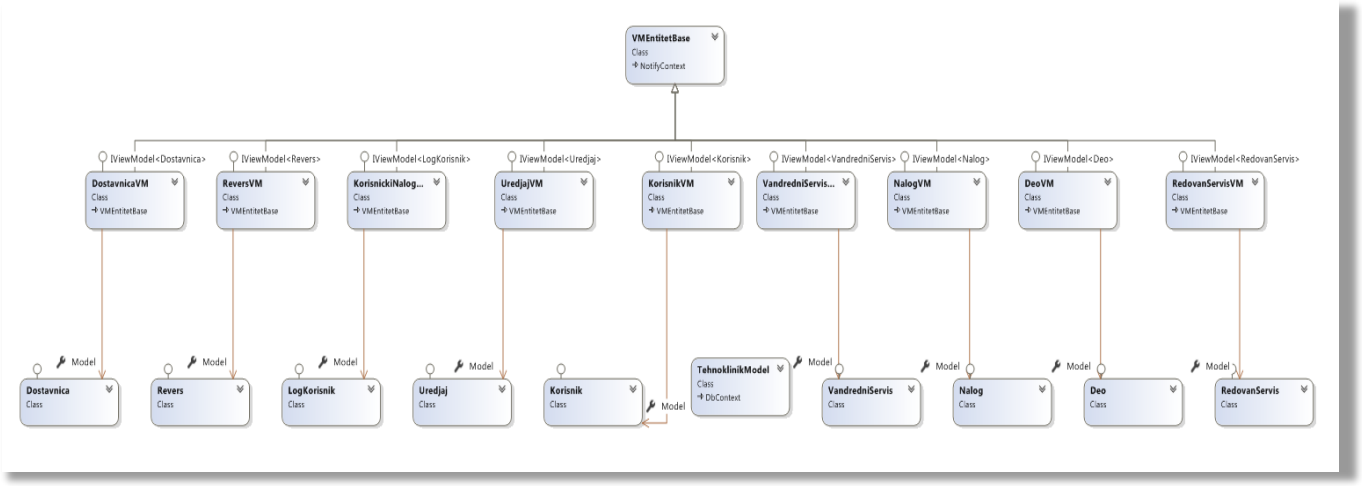


Слика 13: Представља везу између класа презентационог **View** слоја и **ViewModel** класа које представљају њихове **DataContext-e**

ViewModel представља омотач Модела и самим тим још један слој који је суштина "чистоће" **MVVM** шаблона. У класама **ViewModel-a** које представљају **DataContext** се дефинишу функционалности апликације.

Свака класа у Моделу има свој **ViewModel** омотач, Слика :

1. **Korisnik -> KorisnikVM**
2. **Uredjaj -> UredjajVM**
3. **Deo -> DeoVM**
4. **Dostavnica -> DostavnicaVM**
5. **Revers -> ReversVM**
6. **Nalog -> NalogVM**
7. **RedovanServis -> RedovanServisVM**
8. **VandredniServis -> VandredniServisVM**
9. **LogKorisnik -> LogKorisnikVM**



Слика 14: Представља класни дијаграм између ентитет класа у Моделу и њихових "омотач" класа у **ViewModel-u**

Пошто су **DataContext** класе идентичне по питању дефинисане функционалности, као пример је описана само **KorisnikUCViewModel**

Табела 1: Опис **DataContext** класе **KorisnikUCViewModel**

Инстанце објеката унутар	Скоро све функционалности која апликација поседује су дефинисане у **ViewModel DataContext** класи. **KorisnikViewControl view** класа је подешена на инстанцу **KorisnikUCViewModel** класе која имплементира колекције типа **ObservableCollection** које су типа **KorisnikVM, DostavniceVM, ReversVM, NaloziVM** и имплементира инстанцу , а сваки од ових типова инстанцира ентитет класу из модела.
CRUD (Create, Read, Update, Delete) функционалности	**CRUD (Create, Read, Update, Delete)** функционалности везане за унос, ажурирање брисање и читање података из базе података. Унос и ажурирање се врши преко модалног дијалога за унос док се брисање реализује преко **ContextMeni-a** десним кликом на неког од корисника. У случају да постоји клијент или уређај са детаљима или везаним подацима исписује се порука о забрани брисања и операција се прекида, ова рестрикција је регулисана методама за проверу зависности између ентитета, унутар методе се налазе Linq izrazi koji preko instance konteksta baze podataka proveravaju zavisnost. Читање података се реализује исто преко Linq упита који преко колекције чита податке из базе података. Колекција је **ItemsSource ListView** елемента унутар **XAML** објекта у коме се приказују

	подаци.
Филтери за претрагу	Филтери имају своје инстанце које су дефинисане као текући објекти унутар **ListCollectionView** класе преко које се реализује филтер над колекцијом објеката. Преко листе која имплементира делегат **Prediacate** преко кога се дефинишу критеријуми за филтрирање података унутар **DataGrid-a.**
Office.Interop.Word функционалност за рад са **Word** документима	Унутар **KorisnikUCViewModel** су дефинисане и методе за прослеђивање података у **Word** документ за **Dostavnice, Reverse i Radne Naloge.** Прослеђивање се врши у већ припремљена документа у виду **Word** темплејта, методом **MailMerge** где се преко **Microsoft.Office.Interop** библиотеке приступа **Microsoft.Office** апликацији. Преко стринга локације се проналази тражени документ, отвара се **Word** документ и проналазе се тагови поља унутат документа где се смештају подаци одређени за то поље.

6. Закључак

У овом раду је презентиран развој апликације за вођење пословања сервисног одељења "Сервис" и базе података "Техноклиник" коју користи аплкикација. Апликација није завршена и постоји неколико проблема које треба решити:

- Команда за ажурирање селектованих објеката не ради, проблем је у делу имплементиране архитектуре
- Дијалог за креирање корисничког налога уместо **PasswordBox** контроле има **TextBox** контролу која не може да криптује лозинку па је због тога приказана приликом уноса и ажурирања. Проблем је у томе што **PasswordBox** нема контролу за везивање (**Bind**) него је потребна имплементација конвертора (**Converter**) и **DependencyProperty-a**

7. Литература

[1] др John Allwor , "*C# програмирање за Windows i Android*"
[2] Chris Sells, Ian Griffiths, "*Programming WPF*"
[3] MVVM Tutorials point, https://www.tutorialspoint.com/mvvm/mvvm_tutorial.pdf
[4] *Fabrice Marguerie, Steve Eichert, Jim Wooley*, "LINQ in Action"
[5] Microsoft Developer Network, https://code.msdn.microsoft.com/
[6] Code Project, https://www.codeproject.com/
[7] Entity Framwork Code-Based Configuration (EF6), https://msdn.microsoft.com/en-us/library/jj680699(v=vs.113).aspx
[8] Entity Framwork, https://msdn.microsoft.com/en-us/library/gg696172(v=vs.103).aspx