

Razvoj sustava za predaju i obradu zahtjev za izdavanje beskontaktnih kartica

Radin-Mačukat, Mikaela

Master's thesis / Diplomski rad

2019

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:459074>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-01**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

MIKAELA RADIN – MAČUKAT

**RAZVOJ SUSTAVA ZA PREDAJU I OBRADU ZAHTJEVA ZA IZDAVANJE
BESKONTAKTNIH KARTICA**

Diplomski rad

Pula, lipanj, 2019. godine

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

MIKAELA RADIN – MAČUKAT

**RAZVOJ SUSTAVA ZA PREDAJU I OBRADU ZAHTJEVA ZA IZDAVANJE
BESKONTAKTNIH KARTICA**

Diplomski rad

JMBAG: 0303054703, redoviti student
Studijski smjer: Informatika

Kolegij: Napredni algoritmi i strukture podataka

Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Tihomir Orehovački

Pula, lipanj, 2019. godine



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Mikaela Radin-Mačukat, kandidat za magistra informatike, ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega 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 diplomskog 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.

Studentica

U Puli, _____, 2019. godine



IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, Mikaela Radin-Mačukat dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Razvoj sustava za predaju i obradu zahtjeva za izdavanje beskontaktnih kartica“ 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.

Studentica

U Puli, _____, 2019. godine

SADRŽAJ

| | |
|--|----|
| UVOD | 8 |
| 1. KORIŠTENE TEHNOLOGIJE | 10 |
| 1.1. C# | 10 |
| 1.2. .NET Framework..... | 10 |
| 1.3. MVC | 11 |
| 1.4. WPF | 12 |
| 1.5. MVVM | 12 |
| 2. Razrada funkcionalnosti | 14 |
| 2.1. Dijagram obrasca upotrebe | 14 |
| 2.2. Klasni dijagram | 15 |
| 2.3. Prototip | 16 |
| 3. Model podataka | 20 |
| 3.1. Struktura baze podataka | 20 |
| 3.2. Relacijska baza podataka | 40 |
| 3.3. SQL Server baza podataka | 40 |
| 3.4. Entity Framework | 41 |
| 4. Web usluge | 44 |
| 4.1. Podjela i karakteristike | 44 |
| 4.1.1. SOAP Web usluge | 44 |
| 4.1.2. REST Web usluge | 46 |
| 4.2. Web API | 47 |
| 4.3. Primjer dodavanja javno dostupnih krajnjih točaka | 47 |
| 5. Implementacija web aplikacije | 53 |
| 5.1. Prikaz programskih komponenti | 53 |
| 5.2. Primjeri koda za pojedine dijelove aplikacije | 55 |
| 6. Implementacija desktop aplikacije | 58 |
| 6.1. Prikaz programskih komponenti | 58 |
| 6.2. Primjeri koda za pojedine dijelove aplikacije | 60 |

| | |
|--|----|
| 7. Korisničke upute | 62 |
| 7.1. Ispunjavanje web zahtjeva u internetskom pregledniku | 62 |
| 7.2. Korištenje desktop aplikacije | 64 |
| 7.2.1. Prijava u aplikaciju | 65 |
| 7.2.2. Izbornik aplikacije | 65 |
| 7.2.3. Prozor fizička osoba | 66 |
| 7.2.4. Prozor zahtjev za fizičku osobu | 67 |
| 7.2.5. Prozor ažuriranje podataka fizičke osobe pomoću predanog zahtjeva preko mreže | 68 |
| 7.2.6. Dodatne funkcionalnosti | 70 |
| ZAKLJUČAK | 70 |
| LITERATURA | 71 |
| POPIS SLIKA | 73 |
| POPIS TABLICA | 75 |

Zahvala

Zahvaljujem se svom mentoru doc.dr.sc. Tihomiru Orehovačkom na strpljenju i danim savjetima pri izradi diplomskog rada. Želim se zahvaliti svim djelatnicima tvrtke Penta d.o.o., a posebno mentorima koji su mi svojim stručnim savjetima pomogli pri izradi diplomskog rada. Najveću zahvalu želim iskazati svojim roditeljima koji su mi bili podrška i oslonac kroz cijelo moje školovanje. Na kraju se želim zahvaliti svojoj obitelji i prijateljima koji su mi vrijeme provedeno na fakultetu uljepšali svojim prisustvom i pomogli da to vrijeme smatram najljepšim dijelom svog života.

M.R.M.

UVOD

Tema ovog diplomskog rada je izrada sustava za predaju i obradu zahtjeva za izdavanje beskontaktnih kartica. Diplomski rad je izrađen unutar tvrtke Penta d.o.o iz Pule [16] koja je dala suglasnost da se koriste potrebni resursi u svrhu izrade diplomskog rada. Dio baze podataka koji je korišten pri izradi diplomskog rada je zatečeno stanje tvrtke Penta. Unutar njihovog projekta BusCARD izrađena je nova funkcionalnost – obrada zahtjeva predanog putem internetskog preglednika za izradu beskontaktnih kartica. BusCARD je integralni informacijski sustav koji pokriva kompletno poslovanje jednog prometnog poduzeća do nivoa integracije s ERP sustavima.

Diplomski rad će se baviti konceptualnim i stvarnim rješenjem predviđenog sustava. Sustav je namijenjen za poduzeća koja vrše prijevoz putnika kako bi im olakšao izradu mjesečnih beskontaktnih pretplatnih karti. Ovaj sustav će riješiti dugotrajno čekanje u redovima te smanjiti gužve na prodajnim mjestima koja vrše izradu kartica. Svaki korisnik će putem web zahtjeva predati svoje podatke, sliku i potrebne dokumente te nakon toga pomoću ostavljenog kontakt telefona ili e-maila dobiti povratnu informaciju.

Obrada zahtjeva i ažuriranje podataka se vrši u desktop aplikaciji. Radnik na zaslonu ima pregled ispunjenog zahtjeva od strane korisnika, te mu je jedino omogućen prikaz podataka i opcije za mijenjanje statusa zahtjeva iz novog zahtjeva u obrađeni ili odbijeni. Postoji mogućnost da je korisnik krivo unio neki dokument ili sliku, i stoga se taj zahtjev odbija. Zatim se kontaktira korisnika zbog čega je došlo do problema te ukoliko je potrebno da ponovi zahtjev s točnim podacima. Također ako je zahtjev odobren ide u izradu kartice te nakon toga korisnik dobiva povratnu informaciju o preuzimanju kartice.

Cilj ovog sustava je korisnicima olakšati postupak izrade beskontaktna kartice te smanjiti redove čekanja. Navedeni cilj postići će se koristeći aplikacijski okvir ASP.NET MVC, MVVM, Entity Framework kao Object Relational Mapper te Microsoft SQL Server.

U diplomskom radu nakon uvoda slijedi opis korištenih programskih tehnologija, među kojima su ASP.NET MVC uz njegove tri komponente – model- pogled- upravljač, te model – pogled – model pogleda (MVVM). Nakon toga slijedi razrada funkcionalnosti. Zatim slijedi opis Entity Framework

i modela podataka. Slijedeća cjelina se odnosi na programsko rješenje web aplikacije koje se sastoji od prikaza programskih komponenti i primjera koda za pojedine dijelove aplikacije. Nakon toga slijedi programsko rješenje desktop aplikacije koje se sastoji od prikaza programskih komponenti i primjera koda za pojedine dijelove aplikacije. Zadnja cjelina prikazuje korisničke upute za web i desktop aplikaciju.

1. KORIŠTENE TEHNOLOGIJE

1.1. C#

Microsoft Visual C# je jednostavan jezik namijenjen prije svega za razvoj aplikacija koje koriste radno okruženje Microsoft.NET. C# je razvijen od strane tima stručnjaka Anders Hejlsberg-a i Scott Wiltamuth-a koji je nastao u tvrtki Microsoft. Prvi puta se pojavio na tržištu 2000. godine zajedno sa .NET platformom. C# je nastao na temelju objektnih jezika Java, C++ i Visual Basic te je vrlo sličan Javi i C++ jeziku. Sintaksa i semantika je dobrim dijelom preuzeta iz Jave, koja je kao i C# potpuno objektno orijentirani jezik. C# je kreiran za izradu stolnih (desktop) i Internet aplikacija u Microsoft .NET okruženju. U sklopu .NET platforme omogućava kreiranje vizualnih aplikacija tako omogućavajući lakše snalaženje onim korisnicima koji nemaju programerskog iskustva. .NET platforma je razvojni okvir koji omogućava novo sučelje za programiranje aplikacije (eng. application programming interface, tj. API) i klasično sučelje Windows operativnog sustava [15].

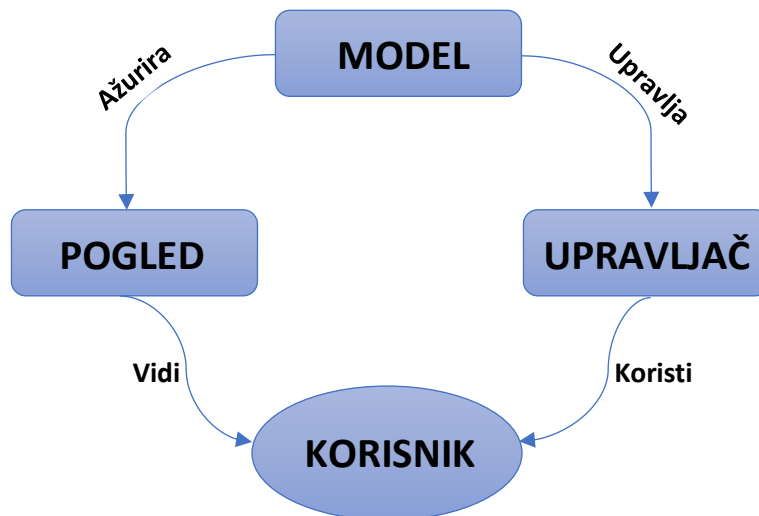
1.2. .NET Framework

Microsoft .NET okvir je infrastruktura pomoću koje se programi na Windows operacijskom sustavu jednostavnije izrađuju i pokreću. .NET okvir se sastoji od CLR (eng. Common Language Runtime) komponente koja služi za izvršavanje .NET programa. Pri pokretanju aplikacije pisane u .NET platformi, CLR ju izvršava kako bi osigurao stabilnost i funkcionalnost. Aplikacije za .NET platformu se mogu pisati u raznim programskim jezicima, međutim CLR ne poznaje niti jedan jezik, on isključivo dobiva naredbe u jeziku MSIL (eng. Microsoft Intermediate Language), temeljen na pravilima koja se nazivaju CLS (eng. Common Language Specifications). Stoga postoji kompajler koji programski jezik prevede u MSIL kako bi ga CLR razumio. Kompajleri se nazivaju IL-kompajleri i dostupni su samo za pet jezika: C#, J#, C++, Visual Basic i JScript [14].

1.3. MVC

Model-Pogled-Upravljač (eng. Model-View-Controller) je software za implementaciju aplikativnog sučelja na računalima. Koristi se za elegantno razdvajanje problema unutar aplikacije što se najčešće primjenjuje kod web aplikacija. MVC razdvaja korisničko sučelje (eng. User Interface) aplikacije i omogućava detaljnu razradu prezentacijskog sloja u 3 komponente (model, pogled i upravljač) koje su prikazane na slici 1.

1. Model – skup klasa koje opisuju podatke s kojima radimo, kao i aplikacijska logika kako vršiti izmjenu i manipulaciju podacima.
2. Pogled – definira kako će korisničko sučelje aplikacije biti prikazano.
3. Upravitelj – skup klasa koji se brine o komunikaciji s korisnikom, sveukupnom toku aplikacije i specifičnoj logici aplikacije [11].



Slika 1. Odnos između modela, pogleda i upravljača (izrada: autor)

Prednosti MVC aplikacija su:

- Olakšano upravljanje razvojem složenih aplikacija zbog dijeljenja aplikacije na model, pogled i upravljač
- Olakšano testiranje aplikacije (eng. TDD - Test-Driven Development), jer je moguće odvojeno testirati pojedine komponente aplikacije i nije uvijek za testiranje potrebno koristiti web server, neke komponente se mogu testirati lokalno
- Koriste se kod velikog tima programera i web dizajnera kojima je potreban veliki stupanj kontrole nad ponašanjem aplikacije [11].

1.4. WPF

Windows Presentation Foundation je okvir korisničkog sučelja koji stvara desktop aplikacije. Omogućava programerima da putem grafičkog sučelja lakše stavljaju grafičke elemente. WPF podržava model aplikacije, resurse, kontrole, grafiku, raspored, povezivanje podataka, dokumente i sigurnost. WPF koristi Extensible Application Markup Language (XAML) koji služi za označavanje podataka, te je poseban po tome što skraćuje kod. Koristi se kod razvijanja WPF aplikacija [12].

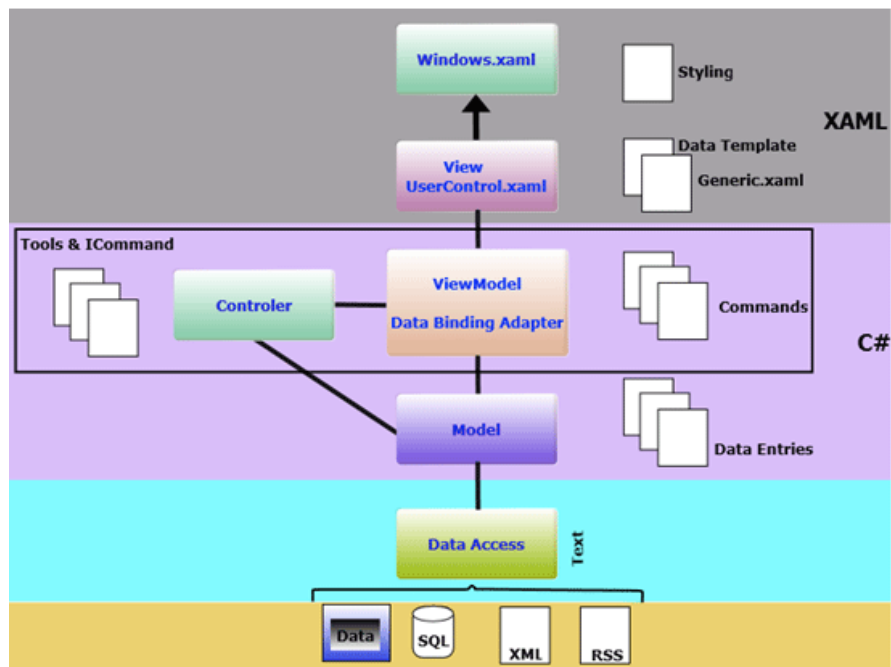
1.5. MVVM

Model-View-ViewModel arhitektura se dijeli na tri dijela. Na slici 2. prikazana je arhitektura modela, pogleda i modela pogleda.

1. Model – predstavlja model domene aplikacije koji obično uključuje model podatka zajedno s poslovnom logikom i logikom provjere valjanosti. Primjeri modela objekta uključuju objekte za prijenos podataka (eng. DTO – Data Transfer Object), obične stare CLR objekte (eng. POJO – Plain Old CLR Object) i generirane objekte entiteta i proxy-a.
2. Pogled – je odgovoran za definiranje strukture i izgleda što korisnik vidi na zaslonu. Svaki pogled je definiran u XAML-u.
3. Model pogleda – implementira svojstva i naredbe na koje se pogled može vezati i obavještavati o promjenama stanja. Svojstva i naredbe koje model pogleda definira su

funkcionalnosti koje nudi UI, ali pogled određuje kako će se ta funkcija prikazivati. Može sadržavati i komande (eng. Commands) preko kojih korisnik vrši interakciju s aplikacijom [13].

Svaki model i model pogleda trebaju se implementirati INotifyPropertyChanged interface iz System.ComponentModel namespace-a. Implementacija ovog korisničkog sučelja se sastoji od samo jednog pristupa tipa PropertyChangedEventHandler (iz istog namespace-a) koji nosi naziv PropertyChanged. Prilikom promijene vrijednosti nekog svojstva potrebno je podići ovaj event da bi se obavijestio pogled da je došlo do promjene, da bi View (putem Data Binding-a) ažurirao kontrole sa novim podacima. Svaka aplikacija koja prati MVVM obrazac bi trebala da ima tri direktorija, jedan za Views, jedan za ViewModels i jedan za Models [6].



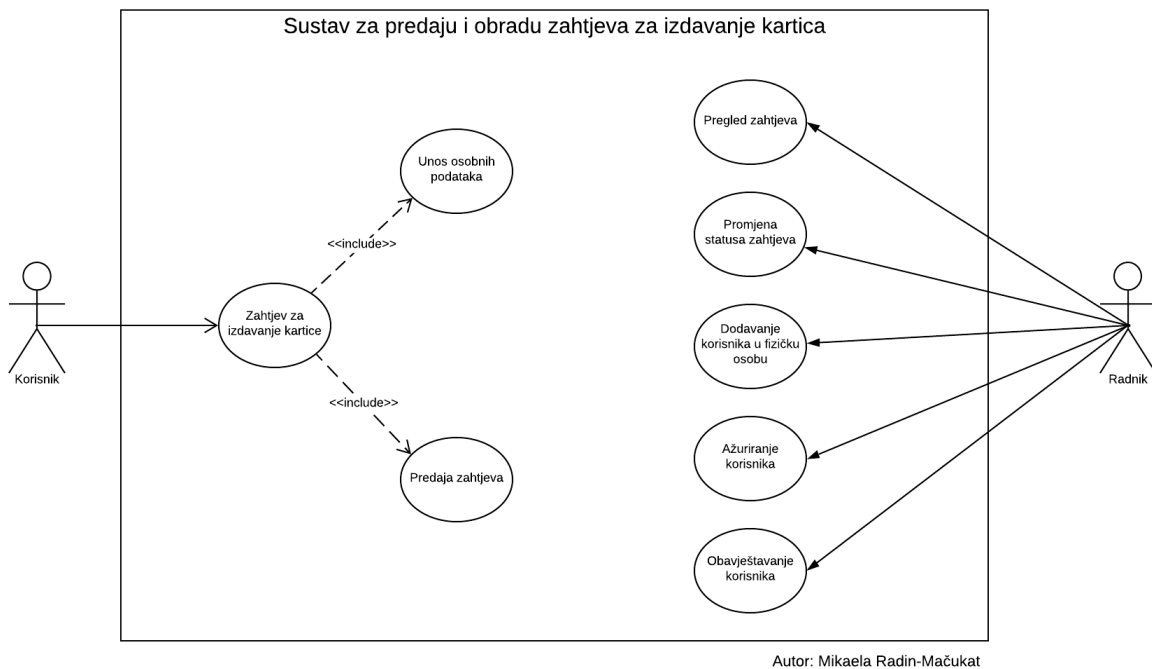
Slika 2. MVVM arhitektura [2]

2. RAZRADA FUNKCIONALNOSTI

Funkcionalnosti od strane korisnika su ispunjavanje zahtjeva za izradu beskontaktno kartice, a od strane radnika su pregledavanje zahtjeva u desktop aplikaciji, promjena statusa zahtjeva, dodavanje korisnika u fizičke osobe, ažuriranje korisnika te obavještanje korisnika.

2.1. Dijagram obrasca upotrebe

Dijagram obrasca upotrebe služi za definiranje komunikacije između aktera i sustava. To je skup scenarija koji su povezani putem jednog cilja korisnika. Scenarij je slijed koraka koji opisuju interakciju između korisnika i sustava.



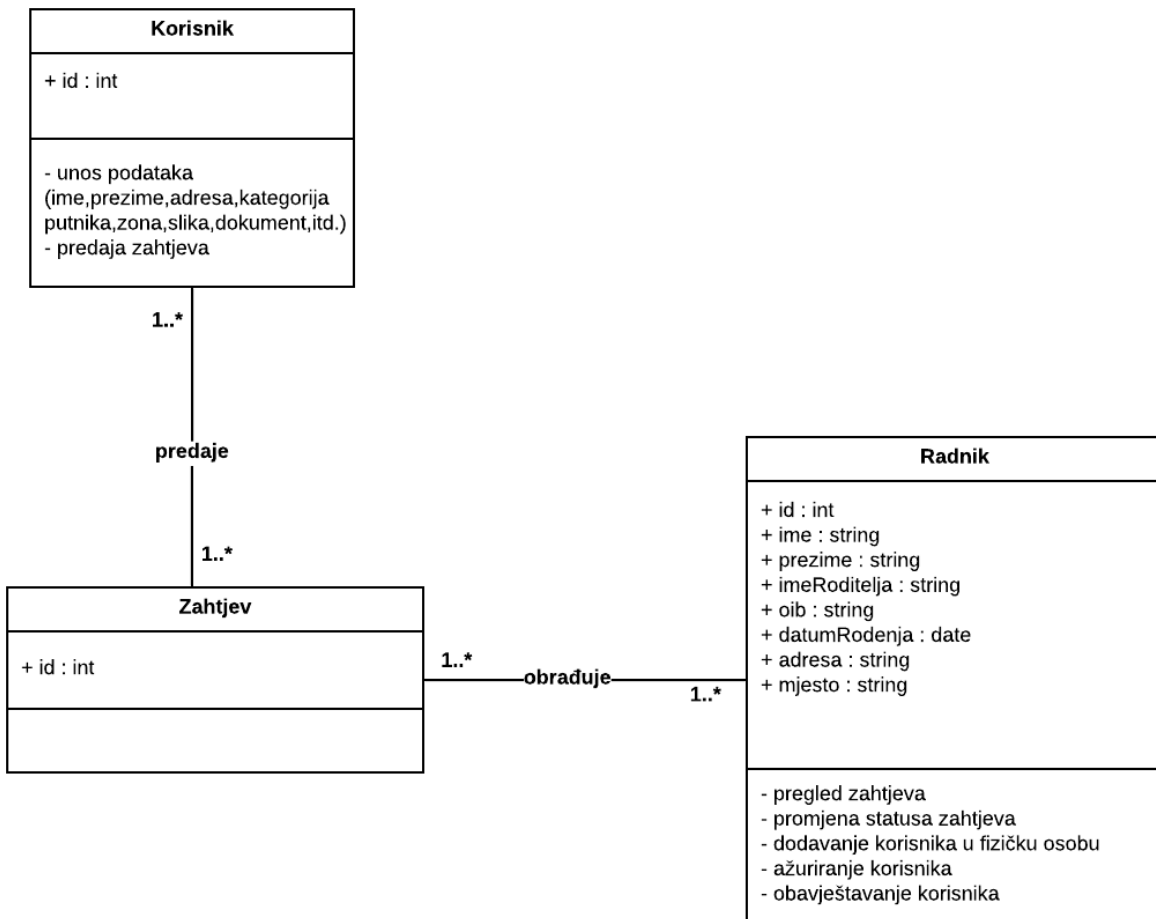
Slika 3. Dijagram obrasca upotrebe (Izvor: autor)

Na slici 3. je prikazana granica sustava i 2 aktera, a to su korisnik i radnik. Akter korisnik vrši radnju ispunjavanja zahtjeva za izradom beskontaktno kartice koji uključuje unos osobnih podataka te predaju zahtjeva (prikazano s vezom <<include>>). Akter radnik vrši radnje pregleda zahtjeva, promjene statusa zahtjeva (novi zahtjev, odbijen, unesena fizička osoba, izrada kartice, izrađena

kartica), dodavanje korisnika u fizičku osobu, ažuriranje korisnika te obavještanje korisnika o krivim podacima i dokumentima tijekom predaje web zahtjeva.

2.2. Klasni dijagram

Na slici 4. se nalazi klasni dijagram koji prikazuje odnose između klasa koji će predstavljati objekte u sustavu.



Slika 4. Klasni dijagram (Izvor: autor)

2.3. Prototip

Prototip je najčešće skica finalnog izgleda aplikacija koja se dogovara sa sudionicima i krajnjim korisnicima projekta. Stvaranje prototipa može uštedjeti vrijeme i novac u kasnijem razvoju projekta. Za izradu skica i prototipa aplikacija korišten je alat Figma. Figma je jednostavni alat za dizajniranje prototipa i povezivanje komponenti aplikacije. Prototip web aplikacije je prikazan na slici 5.

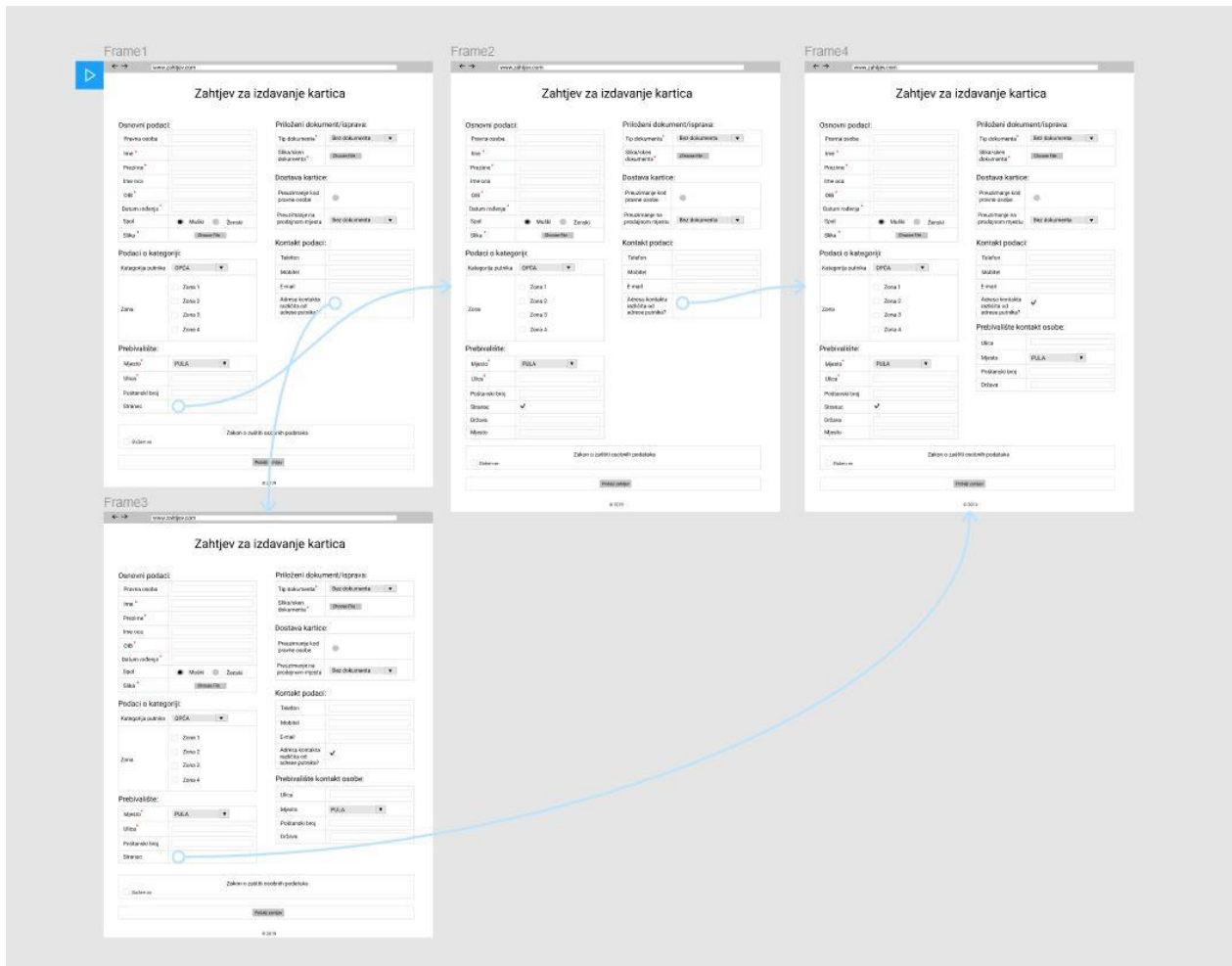
The image shows a web browser window with the URL 'www.zahtjev.com'. The page title is 'Zahtjev za izdavanje kartica'. The form is divided into several sections:

- Osnovni podaci:** Includes fields for 'Pravna osoba', 'Ime', 'Prezime', 'Ime oca', 'OIB', 'Datum rođenja', 'Spot' (with radio buttons for 'Muški' and 'Ženski'), and 'Slika' (with a 'Choose File' button).
- Priloženi dokument/isprava:** Includes 'Tip dokumenta' (dropdown menu set to 'Bez dokumenta') and 'Slika/sken dokumenta' (with a 'Choose File' button).
- Dostava kartice:** Includes 'Preuzimanje kod pravne osobe' (radio button) and 'Preuzimanje na prodajnom mjestu' (dropdown menu set to 'Bez dokumenta').
- Podaci o kategoriji:** Includes 'Kategorija putnika' (dropdown menu set to 'OPĆA') and 'Zona' (checkboxes for 'Zona 1', 'Zona 2', 'Zona 3', and 'Zona 4').
- Kontakt podaci:** Includes 'Telefon', 'Mobilitel', 'E-mail', and 'Adresa kontakta različita od adrese putnika?' (checkbox checked).
- Prebivalište kontakt osobe:** Includes 'Ulica', 'Mjesto' (dropdown menu set to 'PULA'), 'Poštanski broj', and 'Država'.

At the bottom, there is a checkbox for 'Slazem se' next to the text 'Zakon o zaštiti osobnih podataka'. Below this is a 'Pošalji zahtjev' button and a copyright notice '© 2019'.

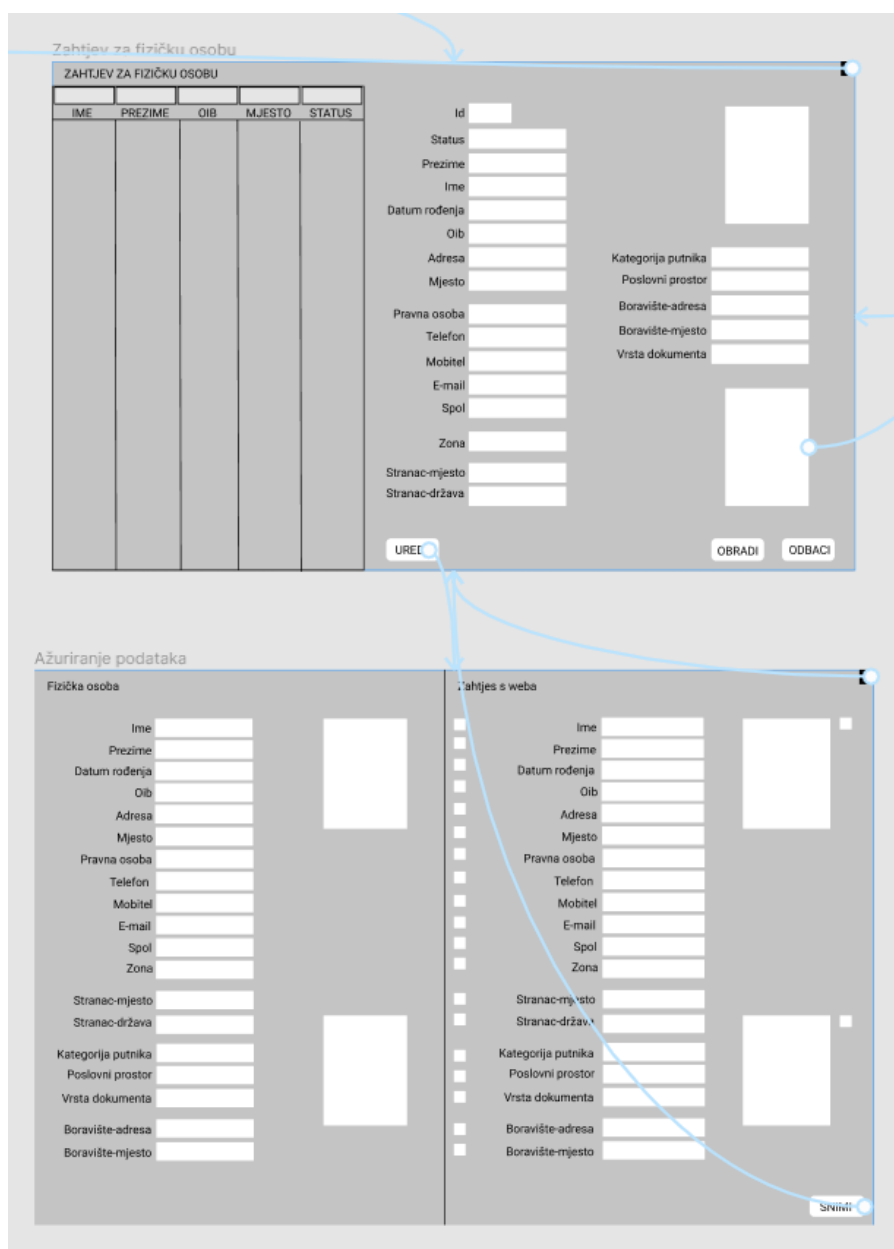
Slika 5. Prototip web aplikacije

Izgled prototipa web aplikacije je jednostavan i lako se koristi, što je najbitnije za krajnje korisnike. Na vrhu stranice se nalazi ime aplikacije, zatim ispod su prikazana prazna polja sa informacija koje će krajnji korisnici ispuniti sa svojim osobnim podacima. Postoje combobox komponente, to su komponente u obliku liste koje sadrže informacije iz baze podataka te omogućava izbor samo jedne stavke sa liste. Zatim postoje komponente za učitavanje slike ili dokumenta sa računala. Na kraju web aplikacije postoji komponenta gumb (eng. button), pomoću koje se predaje zahtjev za izdavanje beskontaktno kartice. Sve funkcionalnosti web aplikacije prikazane su na slici 6.

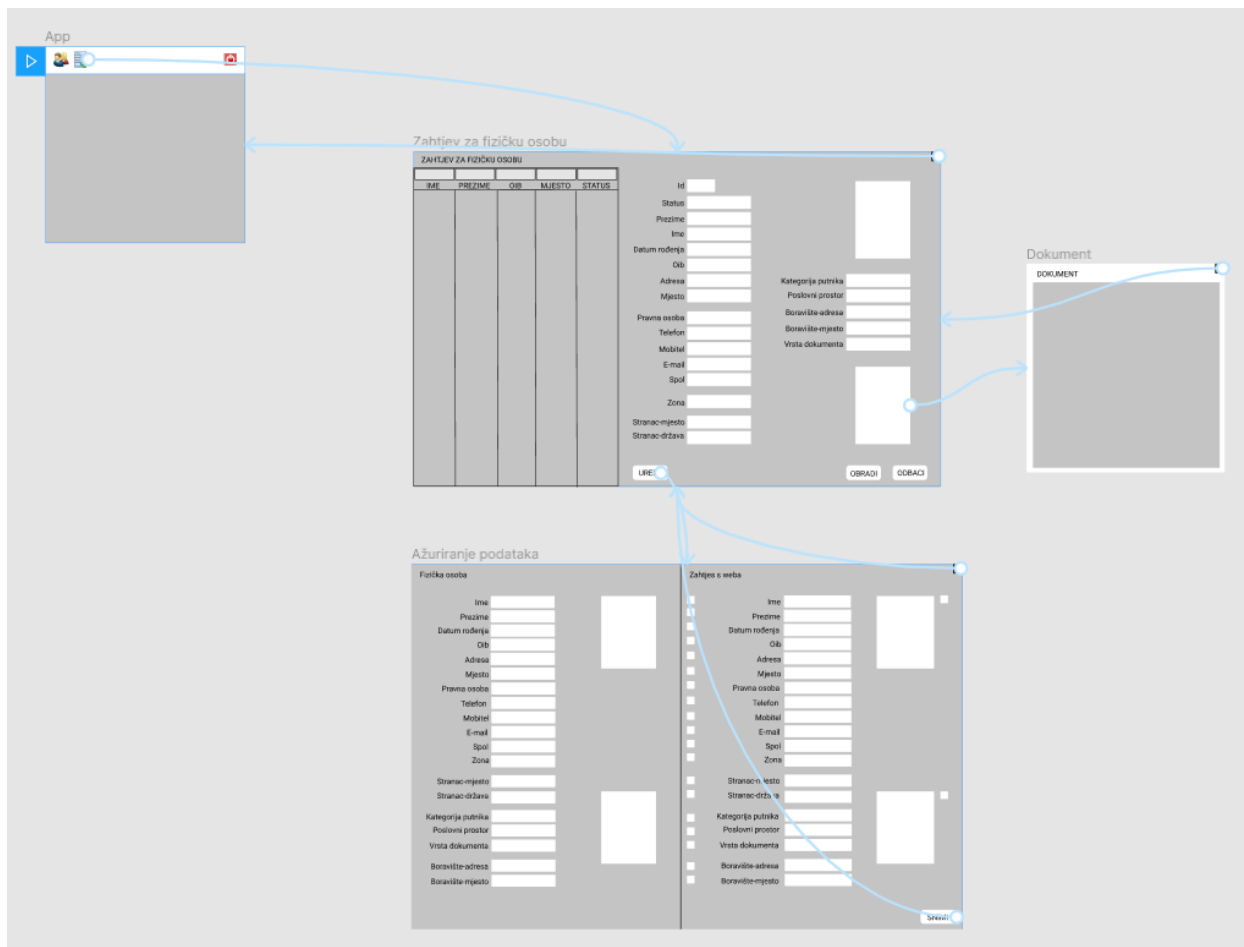


Slika 6. Prototip web aplikacije

Prototip desktop aplikacije prikazan je na slici 7. i sastoji se od 4 okvira. Prvi okvir je početni dio aplikacije, sadrži ulaz u fizičku osobu te ulaz u zahtjev za fizičku osobu. Unutar okvira zahtjev se nalazi komanda 'Uredi' u kojoj je drugi okvir. On sadrži podatke o fizičkoj osobi s lijeve strane i podatke o korisniku koji je zahtjev ispunio preko web aplikacije, s desne strane. Također unutar okvira zahtjev se nalazi u donjem desnom kutu slika dokumenta na koju se s dvoklikom otvara novi okvir koji prikazuje dokument s komandama za povećanje i smanjenje. Sve funkcionalnosti desktop aplikacije prikazane su na slici 8.



Slika 7. Prototip desktop aplikacije



Slika 8. Prototip desktop aplikacije

3. MODEL PODATAKA

3.1. Struktura baze podataka

Za izradu tablica korišten je Microsoft SQL. U njemu je definirano ukupno 14 tablica koje su potrebne za razvoj proizvoda. Glavni podaci se nalaze u :

- Tablica dbo.Zahtjev – sadrži podatke o korisnicima koji su predali zahtjev putem web aplikacije
- Tablica dbo.FizickaOsoba – sadrži podatke o fizičkim osobama
- Tablica dbo.ZahtjevStatus – sadrži podatke za pet statusa zahtjeva
- Tablica dbo.Zona – sadrži podatke o zonama grada
- Tablica dbo.Mjesto – sadrži podatke o nazivima mjesta u Hrvatskoj
- Tablica dbo.DokumentTip – sadrži podatke o vrsti dokumenta
- Tablica dbo.KategorijaPutnika – sadrži podatke o kategoriji putnika
- Tablica dbo.PoslovniProstor – sadrži podatke o poslovnim prostorima

3.1.1. Tablica za pohranu podataka koji su predani putem web aplikacije

Tablica za pohranu podataka koji su predani putem web aplikacije se sastoji od 32 atributa. Popis atributa je prikazan u tablici 1. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „KATEGORIJAPUTNIKA_Id“ je tipa integer ili cjelobrojni tip podatka. Atribut „PravnaOsoba“ je tipa nvarchar i duljina znakova je ograničena na 150. Atribut „FIZICKAOSOBA_Id“ je cjelobrojni tip podataka(integer) te on povezuje zahtjev i unesenu fizičku osobu. Atributi „OIB“ , „Ime“, „Prezime“, „Adresa“ su zapisi znakovnog niza varijabilne duljine. Atribut „MJESTO_Id“ je cjelobrojni tip podatka. Atribut „Stranac“ je tipa bit te 0 označava ako su korisnici iz Hrvatske, 1 ako je stranac. Atributi „StranacMjesto“ i „StranacDrzava“ su zapisi varijabilne duljine. Ako prethodni atribut „Stranac“ ima vrijednost 1, onda se mjesto i država unose tekstualno u posebna polja. Atributi „Telefon“, „Mobitel“, „E-mail“ su zapisi varijabilne duljine te služe za kontakt korisnika. Atribut „Spol“ je tipa nchar, odnosno duljina samo jednog

znaka, u ovom slučaju M ili Ž. Atribut „DatumRodjenja“ je datumski tip podatka koji sprema datum u formatu dd.MM.yyyy zajedno s vremenom dana, koji se inače ne prikazuje. Atribut „DOKUMENTTIP_Id“ je cjelobrojni tip podatka te je obavezan unos, a sadrži nazive vrste dokumenata. Atribut „KontaktUlica“ je zapis znakovnog niza varijabilne duljine, a odnosi se na adresu koja se razlikuje od prebivališta. Atribut „MJESTO_Id_Kontakt“ je cjelobrojni tip podatka, unos mu nije obavezan te se odnosi na mjesto boravišta. Atribut „Slika“ je tipa varbinary, što bi značilo da je taj tip podatka namijenjen za skladištenje binarnih podataka. Atribut „Napomena“ je tipa nchar, što bi značilo da je namijenjen za skladištenje kraćih nizova karaktera. Atribut „STATUS_Id“ je tipa tinyint, sadrži 1 bajt te atribut ima oznake 0 za novi zahtjev, 1 za obrađen zahtjev, 2 za odbijen zahtjev, 3 za unesenu fizičku osobu, 4 za izrađenu karticu. Atribut „Roditelj“ je zapis znakovnog niza varijabilne duljine te omogućava unos jednog roditelja. Atribut „SkenDokumenta“ ima također isti tip podatka kao atribut „Slika“, varbinary, te je obavezno polje. Atribut „POSLOVNIPROSTOR_Id“ je cjelobrojni tip podatka i omogućuje odabir jednog poslovnog prostora gdje će korisnik moći preuzeti svoju karticu. Atribut „PRAVNAOSOBA_Id“ je cjelobrojni tip podatka i dodjeljuje se pravna osoba kod obrade zahtjeva. Atribut „PlacaPO“ je tipa bit i označava da li karticu plaća pravna osoba ili sam korisnik. Atribut „KARTICA_Id“ je cjelobrojni tip podatka koji označava da li je kartica izrađena. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „ZONA_Id“ je cjelobrojni tip podatka koji sadrži vrste zona.

Tablica 1. Popis atributa koji se nalaze u tablici dbo.Zahtjev

| Naziv | Tip podataka | Ograničenja | Opis |
|-----------------------------|--------------|---|--|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| KATEGORIJAPUTNIKA_Id | int | Nije obavezan unos | Vrste kategorije putnika |
| PravnaOsoba | nvarchar | Duljine do 150 znakova; nije obavezan unos | Unos naziva pravne osobe |
| FIZICKAOSOBA_Id | int | Nije obavezan unos | Povezuje se Zahtjev i unesena fizička osoba |
| OIB | nvarchar | Duljine do 50 znakova; obavezan unos | Osobni identifikacijski broj svakog hrvatskog državljana i pravne osobe |
| Ime | nvarchar | Duljine do 50 znakova; obavezan unos | Ime korisnika |
| Prezime | nvarchar | Duljine do 50 znakova; obavezan unos | Prezime korisnika |
| Adresa | nvarchar | Duljine do 50 znakova; obavezan unos | Adresa korisnika |
| MJESTO_Id | int | Nije obavezan unos | Mjesto korisnika |
| Stranac | bit | Nije obavezno | 0 - iz Hrvatske, 1 – stranac; ako je stranac mjesto i država se unose tekstualno u posebna polja |
| StranacMjesto | nvarchar | Duljine do 50 znakova; nije obavezan unos | Mjesto izvan Hrvatske |
| StranacDrzava | nvarchar | Duljine do 50 znakova; nije obavezan unos | Strana država |
| Telefon | nvarchar | Duljine do 50 znakova; nije obavezan unos | Broj telefona korisnika za kontakt |
| Mobitel | nvarchar | Duljine do 50 znakova; nije obavezan unos | Broj mobitela korisnika za kontakt |

| | | | |
|---------------------------|-----------|--|--|
| Email | nvarchar | Duljine do 50 znakova; nije obavezan unos | Elektronička adresa korisnika |
| Spol | nchar | Duljina jednog znaka; obavezan unos | M ili Ž |
| DatumRodnja | date | Obavezan unos | Datum rođenja korisnika |
| DOKUMENTTIP_Id | int | Obavezan unos | Vrste dokumenta |
| KontaktUlica | nvarchar | Duljine do 150 znakova; nije obavezan unos | Kontakt ulica korisnika; odnosi se na boravište |
| MJESTO_Id_Kontakt | int | Nije obavezan unos | Kontakt mjesto korisnika; odnosi se na boravište; ako je stranac odabрати IZVAN HRVATSKE |
| Slika | varbinary | Obavezno polje | Slika korisnika |
| Napomena | nchar | Duljine do 300 znakova; nije obavezan unos | Napomena korisniku |
| STATUS_Id | tinyint | Obavezno polje | 0 - Novi zahtjev; 1 – Obrađen; 2 – Odbijen; 3 - Unesena fizička; osoba; 4 – Izrađena kartica |
| Roditelj | nvarchar | Duljine do 50 znakova; nije obavezan unos | Ime roditelja kontakta |
| SkenDokumenta | varbinary | Obavezno polje | Sken dokumenta korisnika |
| POSLOVNIPROSTOR_Id | int | Nije obavezno polje | Odabir poslovnog prostora gdje se preuzima kartica |
| PRAVNAOSOBA_Id | int | Nije obavezno polje | Dodijeljena pravna osoba kod obrade zahtjeva |
| PlacaPO | bit | Nije obavezno polje | Karticu plaća PO(Pravna osoba) |
| KARTICA_Id | int | Nije obavezno polje | Izrađena kartica |
| LastUpdate | datetime | Nije obavezno polje | Zadnje vrijeme izmjene |
| LastAction | nchar | Duljina jednog znaka; Nije obavezno polje | D(Delete), U(Update), I(Insert) |
| LASTUSER_Id | int | Nije obavezno polje | Radnik |
| ZONA_Id | int | Nije obavezno polje | Vrste zona |

3.1.2. Tablica za pohranu podataka fizičke osobe

Tablica za pohranu podataka fizičke osobe sastoji se od 22 atributa. Popis atributa je prikazan u tablici 2. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „MJESTO_Id“ je cjelobrojni tip podatka. Atributi „Ime“, „Prezime“, „OIB“ , „Adresa“ su zapisi znakovnog niza varijabilne duljine. Atributi „Telefon“, „Mobitel“, „E-mail“ su zapisi varijabilne duljine te služe za kontakt korisnika. Atribut „Spol“ je tipa nchar, odnosno duljina samo jednog znaka, u ovom slučaju M ili Ž. Atribut „DatumRodjenja“ je datumski tip podatka koji sprema datum u formatu dd.MM.yyyy zajedno s vremenom dana, koji se inače ne prikazuje. Atribut „Roditelj“ je zapis znakovnog niza varijabilne duljine te omogućava unos jednog roditelja. Atributi „PravoNaPratioca“, „Stranac“, „Slijepac“ su tipa bit i nisu obavezna polja. Atributi „Razred“ i „Semestar“ su tipa tinyint. Atribut „KmDoSkole“ je tipa decimal i nije obavezno polje. Atribut „Drzava“ je zapis znakovnog niza varijabilne duljine i nije obavezno polje. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „SIF_Osoba“ je cjelobrojni tip podatka.

Tablica 2. Popis atributa koji se nalaze u tablici dbo.FizickaOsoba

| Naziv | Tip podataka | Ograničenja | Opis |
|------------------|--------------|---|---|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| MJESTO_Id | int | Obavezan unos | Mjesto fizičke osobe |
| Ime | nvarchar | Duljine do 50 znakova; nije obavezan unos | Ime fizičke osobe |
| Prezime | nvarchar | Duljine do 50 znakova; obavezan unos | Prezime fizičke osobe |
| OIB | nvarchar | Duljine do 50 znakova; obavezan unos | Osobni identifikacijski broj svakog hrvatskog državljana i pravne osobe |

| | | | |
|------------------------|----------|--|--------------------------------------|
| Adresa | nvarchar | Duljine do 50 znakova; obavezan unos | Adresa fizičke osobe |
| Telefon | nvarchar | Duljine do 50 znakova; nije obavezan unos | Telefon fizičke osobe |
| Mobitel | nvarchar | Duljine do 50 znakova; nije obavezan unos | Mobitel fizičke osobe |
| Email | nvarchar | Duljine do 50 znakova; nije obavezan unos | Elektronička adresa fizičke osobe |
| Spol | nchar | Duljina jednog znaka; obavezan unos | M ili Ž |
| DatumRodenja | date | Obavezan unos | Datum rođenja fizičke osobe |
| Roditelj | nvarchar | Duljine do 100 znakova; nije obavezan unos | Ime roditelja fizičke osobe |
| PravoNaPratioca | bit | Nije obavezno polje | |
| Stranac | bit | Nije obavezno polje | |
| Slijepac | bit | Nije obavezno polje | |
| Razred | tinyint | Nije obavezno polje | |
| Semestar | tinyint | Nije obavezno polje | |
| KmDoSkole | decimal | Nije obavezno polje | |
| Drzava | nvarchar | Nije obavezno polje | |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; Obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
| SIF_Osoba | int | Nije obavezno polje | |

3.1.3. Tablica sadrži podatke o statusu zahtjeva

Tablica koja sadrži podatke o statusu zahtjeva sastoji se od 6 atributa. Popis atributa je prikazan u tablici 3. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine. Atribut „Aktivan“ je tipa bit i on označava da li status zahtjeva aktivan. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika.

Tablica 3. Popis atributa koji se nalaze u tablici dbo.ZahtjevStatus

| Naziv | Tip podataka | Ograničenja | Opis |
|--------------------|--------------|---|---------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| Naziv | nvarchar | Duljine do 50 znakova; obavezan unos | Naziv statusa |
| Aktivan | bit | Obavezan unos | Aktivnost statusa |
| LastUpdate | datetime | Nije obavezno polje | Zadnje vrijeme izmjene |
| LastAction | nchar | Duljina jednog znaka; nije obavezno polje | D(Delete), U(Update), I(Insert) |
| LASTUSER_Id | int | Nije obavezno polje | Radnik |

3.1.4. Tablica sadrži podatke o zonama grada

Tablica koja sadrži podatke o zonama grada sastoji se od 6 atributa. Popis atributa je prikazan u tablici 4. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „Kratki“ je zapis znakovnog niza varijabilne duljine, maksimalno 10 znakova. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene.

Tablica 4. Popis atributa koji se nalaze u tablici dbo.Zona

| Naziv | Tip podataka | Ograničenja | Opis |
|---------------|--------------|---|--------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| Naziv | nvarchar | Duljine do 50 znakova; obavezan unos | Naziv statusa |
| Kratki | nvarchar | Duljine do 10 znakova; obavezan unos | Kratki naziv statusa |

| | | | |
|--------------------|----------|---|------------------------------------|
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |

3.1.5. Tablica sadrži nazive mjesta u Hrvatskoj

Tablica koja sadrži nazive mjesta u Hrvatskoj sastoji se od 10 atributa. Popis atributa je prikazan u tablici 5. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „Sifra“ je cjelobrojni tip podatka koji sadrži šifru mjesta. Atribut „MaticniBroj“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „OPCINA_Id“ je cjelobrojni tip podatka i sadrži nazive općina. Atribut „PTT“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 200 znakova i sadrži naziv mjesta. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „Zona“ je cjelobrojni tip podatka koji sadrži zone grada.

Tablica 5. Popis atributa koji se nalaze u tablici dbo.Mjesto

| Naziv | Tip podataka | Ograničenja | Opis |
|--------------------|--------------|--|-----------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| Sifra | int | Nije obavezan unos | Šifra mjesta |
| MaticniBroj | nvarchar | Duljine do 50 znakova; nije obavezan unos | Matični broj mjesta |
| OPCINA_Id | int | Obavezan unos | Općina |
| PTT | nvarchar | Duljine do 50 znakova; nije obavezan unos | Poštanski broj |
| Naziv | nvarchar | Duljine do 200 znakova; obavezan unos | Naziv mjesta |

| | | | |
|--------------------|----------|---|------------------------------------|
| LASTUSER_Id | int | Nije obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
| Zona | int | Nije obavezno polje | Naziv zone |

3.1.6. Tablica sadrži podatke o vrsti dokumenta

Tablica koja sadrži podatke o vrsti dokumenta sastoji se od 5 atributa. Popis atributa je prikazan u tablici 6. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova i sadrži naziv dokumenta. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene.

Tablica 6. Popis atributa koji se nalaze u tablici dbo.DokumentTip

| Naziv | Tip podataka | Ograničenja | Opis |
|--------------------|--------------|--|------------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| Naziv | nvarchar | Duljine do 50 znakova; obavezan unos | Naziv dokumenta |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |

3.1.7. Tablica sadrži podatke o kategoriji putnika

Tablica koja sadrži podatke o kategoriji putnika sastoji se od 8 atributa. Popis atributa je prikazan u tablici 7. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos

dvije iste vrijednosti u tablicu. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova i sadrži naziv kategorije putnika. Atribut „Aktivan“ je tipa bit i on označava da li kategorija putnika aktivna. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „StariTipKartice“ je cjelobrojni tip podatka. Atribut „IspisNaKartici“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova.

Tablica 7. Popis atributa koji se nalaze u tablici dbo.KategorijaPutnika

| Naziv | Tip podataka | Ograničenja | Opis |
|------------------------|--------------|---|---------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| Naziv | nvarchar | Duljine do 50 znakova; obavezan unos | Naziv kategorije putnika |
| Aktivan | bit | Obavezan unos | Aktivnost kategorije putnika |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
| StariTipKartice | int | Nije obavezno polje | |
| IspisNaKartici | nvarchar | Duljine do 50 znakova; nije obavezan unos | |

3.1.8. Tablica sadrži podatke o poslovnim prostorima

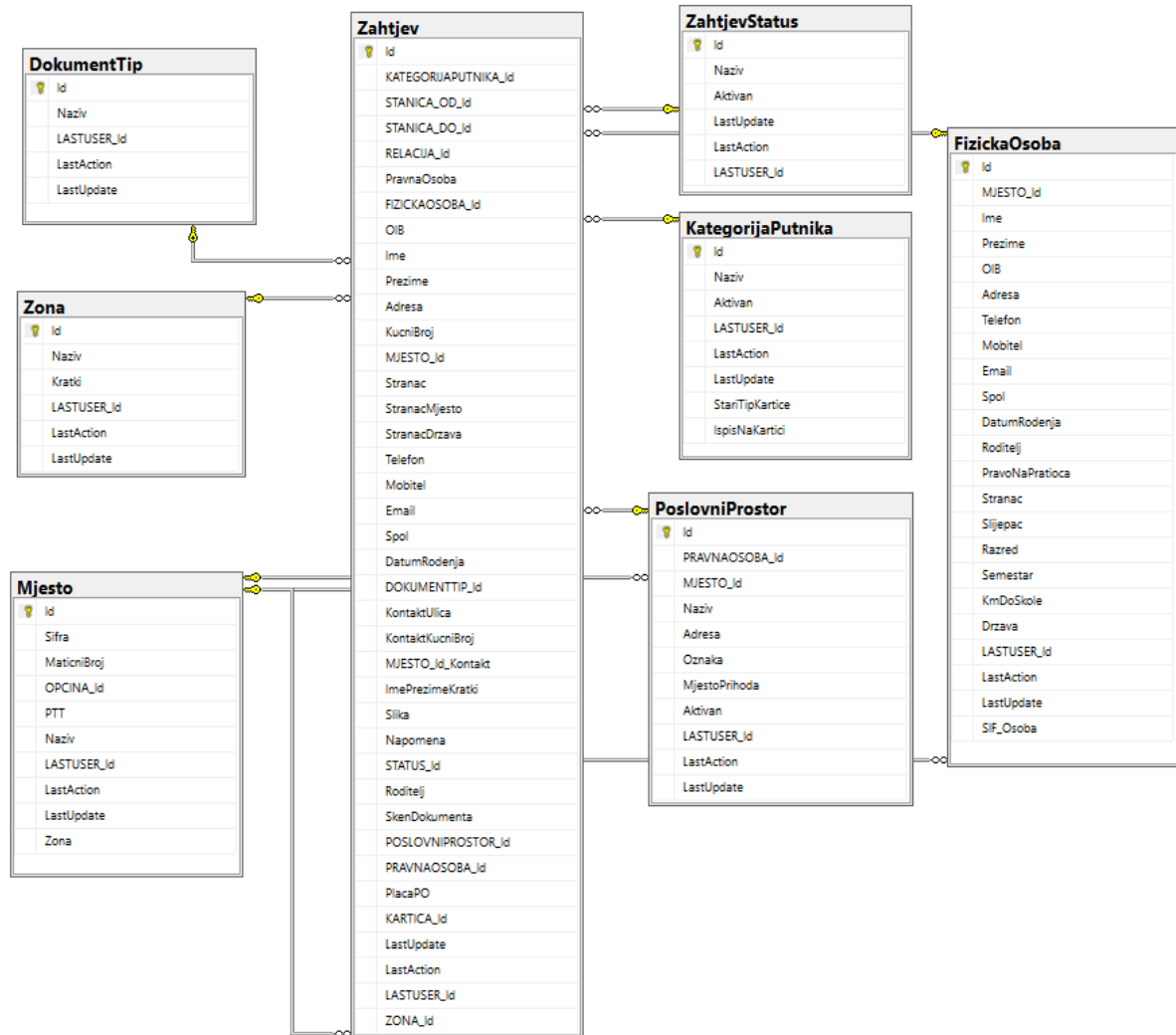
Tablica koja sadrži podatke o poslovnim prostorima sastoji se od 11 atributa. Popis atributa je prikazan u tablici 8. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „PRAVNAOSOBA_Id“ je cjelobrojni tip podatka i dodjeljuje se pravna osoba kod obrade zahtjeva. Atribut „MJESTO_Id“ je cjelobrojni tip podatka. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova i sadrži naziv kategorije putnika. Atributi „Adresa“ i „Oznaka“ su zapisi znakovnog niza varijabilne

duljine i obavezan im je unos. Atribut „MjestoPrihoda“ je zapis znakovnog niza varijabilne duljine i nema obavezan unos. Atribut „Aktivan“ je tipa bit i on označava da li kategorija putnika aktivna. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene.

Tablica 8. Popis atributa koji se nalaze u tablici dbo.PoslovniProstor

| Naziv | Tip podataka | Ograničenja | Opis |
|-----------------------|--------------|---|--|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| PRAVNAOSOBA_Id | int | Nije obavezno polje | Dodijeljena pravna osoba kod obrade zahtjeva |
| MJESTO_Id | int | Obavezan unos | Mjesto korisnika |
| Naziv | nvarchar | Duljine do 200 znakova; obavezan unos | Naziv poslovnog prostora |
| Adresa | nvarchar | Duljine do 200 znakova; obavezan unos | Adresa poslovnog prostora |
| Oznaka | nvarchar | Duljine do 50 znakova; obavezan unos | Oznaka poslovnog prostora |
| MjestoPrihoda | nvarchar | Duljine do 50 znakova; nije obavezan unos | Mjesto prihoda |
| Aktivan | bit | Obavezan unos | Aktivnost poslovnog prostora |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |

Na slici 9. je prikazan dijagram entiteta i veza glavnih tablica: dbo.Zahtjev, dbo.DokumentTip, dbo.Zona, dbo.Mjesto, dbo.ZahtjevStatus, dbo.KategorijaPutnika, dbo.PoslovniProstor, dbo.FizickaOsoba.



Slika 9. Dijagram entiteta i veza glavnih tablica

Sporedni podaci se nalaze u tablicama:

- Tablica dbo.Drzava – sadrži podatke o državama
- Tablica dbo.Kartica – sadrži podatke o kartici fizičke osobe
- Tablica dbo.Oblast – sadrži podatke o županijama u Hrvatskoj
- Tablica dbo.Opcina – sadrži podatke o općinama u Hrvatskoj
- Tablica dbo.Parametar – sadrži podatke o parametrima
- Tablica dbo.Radnik – sadrži podatke o radniku

3.1.9. Tablica sadrži podatke o državama

Tablica koja sadrži podatke o državama sastoji se od 7 atributa. Popis atributa je prikazan u tablici 9. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „VALUTA_Id“ je cjelobrojni tip podatka. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova i sadrži naziv države. Atribut „Oznaka“ je zapis znakovnog niza varijabilne duljine, maksimalno 2 znaka. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene.

Tablica 9. Popis atributa koji se nalaze u tablici dbo.Drzava

| Naziv | Tip podataka | Ograničenja | Opis |
|--------------------|--------------|---|---------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| VALUTA_Id | int | Obavezno polje | Valuta države |
| Naziv | nvarchar | Duljine do 50 znakova; obavezan unos | Naziv države |
| Oznaka | nvarchar | Duljine do 2 znaka; obavezno polje | Oznaka države |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |

| | | | |
|-------------------|----------|----------------|------------------------|
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
|-------------------|----------|----------------|------------------------|

3.1.10. Tablica sadrži podatke o kartici fizičke osobe

Tablica koja sadrži podatke o kartici fizičke osobe sastoji se od 16 atributa. Popis atributa je prikazan u tablici 10. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „FIZICKAOSOBA_Id“ je cjelobrojni tip podataka(integer) te on povezuje zahtjev i unesenu fizičku osobu. Atributi „KARTICAVRSTA_Id“ i „KARTICA_Id_Reizdavanje“ imaju cjelobrojni tip podatka. Atribut „UID“ je zapis znakovnog niza varijabilne duljine, maksimalno 14 znakova. Atributi „DatumIzrade“, „DatumAktivacije“ i „DatumStorna“ su datumski i vremenski tipovi podataka. Atributi „VrijediOd“ i „VrijediDo“ su tipa nchar i imaju ograničenje na duljinu do 5 znaka. Atribut „Status“ je tipa nchar i ima ograničenje na jedan znak. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „RADNIK_Id“ je cjelobrojni tip podatka i označava kojoj kategoriji pripada radnik. Atribut „SIF_Kartica“ je cjelobrojni tip podatka i predstavlja šifru kartice.

Tablica 10. Popis atributa koji se nalaze u tablici dbo.Kartica

| Naziv | Tip podataka | Ograničenja | Opis |
|-------------------------------|--------------|---|--------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| FIZICKAOSOBA_Id | int | Nije obavezno polje | Fizička osoba |
| KARTICAVRSTA_Id | int | Obavezan unos | Vrsta kartice |
| KARTICA_Id_Reizdavanje | int | Nije obavezno polje | Reizdavanje kartice |
| UID | nvarchar | Duljine do 50 znakova; obavezan unos | |
| DatumIzrade | datetime | Obavezno polje | Datum izrade kartice |
| DatumAktivacije | datetime | Nije obavezno polje | Datum aktivacije kartice |
| DatumStorna | datetime | Nije obavezno polje | Datum storniranja kartice |

| | | | |
|--------------------|----------|--------------------------------------|--|
| VrijediOd | nchar | Duljina do 5 znaka, obavezan unos | Datum od kad vrijedi kartica |
| VrijediDo | nchar | Duljina do 5 znaka, obavezan unos | Datum do kad vrijedi kartica |
| Status | nchar | Duljina jednog znaka; obavezan unos | Status kartice |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
| RADNIK_Id | int | Nije obavezno polje | Radnik, vozač, prodavač ili ostali radnici |
| SIF_Kartica | int | Nije obavezno polje | Šifra kartice |

3.1.11. Tablica sadrži podatke o županijama u Hrvatskoj

Tablica koja sadrži podatke o županijama u Hrvatskoj sastoji se od 6 atributa. Popis atributa je prikazan u tablici 11. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „DRZAVA_Id“ je cjelobrojni tip podatka. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova i sadrži naziv države. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene.

Tablica 11. Popis atributa koji se nalaze u tablici dbo.Oblast

| Naziv | Tip podataka | Ograničenja | Opis |
|--------------------|--------------|---|---------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| DRZAVA_Id | int | Obavezno polje | Država |
| Naziv | nvarchar | Duljine do 50 znakova; obavezan unos | Naziv županije u Hrvatskoj |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |

| | | | |
|-------------------|----------|----------------|------------------------|
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
|-------------------|----------|----------------|------------------------|

3.1.12. Tablica sadrži podatke o općinama u Hrvatskoj

Tablica koja sadrži podatke o općinama u Hrvatskoj sastoji se od 8 atributa. Popis atributa je prikazan u tablici 12. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „OBLAST_Id“ je cjelobrojni tip podatka. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 200 znakova i sadrži naziv države. Atribut „MaticniBroj“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „Subvencija“ je tipa bit i označava da li općina daje subvenciju. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene.

Tablica 12. Popis atributa koji se nalaze u tablici dbo.Opcina

| Naziv | Tip podataka | Ograničenja | Opis |
|--------------------|--------------|---|---------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| OBLAST_Id | int | Obavezno polje | Županija |
| Naziv | nvarchar | Duljine do 200 znakova; obavezan unos | Naziv općine |
| MaticniBroj | nvarchar | Duljine do 50 znakova; nije obavezan unos | Matični broj općine |
| Subvencija | bit | Obavezno polje | Subvencija općine |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |

3.1.13. Tablica sadrži podatke o parametrima

Tablica koja sadrži podatke o parametrima za zahtjev sastoji se od 10 atributa. Popis atributa je prikazan u tablici 13. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „Sifra“ je zapis znakovnog niza varijabilne duljine, maksimalno 20 znakova. Atribut „Naziv“ je zapis znakovnog niza varijabilne duljine, maksimalno 500 znakova i sadrži naziv parametra. Atributi „VrijediOd“ i „VrijediDo“ su datumski i vremenski tip podataka. Atribut „Vrijednost“ je zapis znakovnog niza varijabilne duljine, maksimalno do 6000 znakova. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „Nevidljivo“ je tipa bit i nije obavezno polje.

Tablica 13. Popis atributa koji se nalaze u tablici dbo.Parametar

| Naziv | Tip podataka | Ograničenja | Opis |
|--------------------|--------------|---|---------------------------------|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| Sifra | int | Nije obavezan unos | Šifra mjesta |
| Naziv | nvarchar | Duljine do 20 znakova; nije obavezan unos | Naziv parametra |
| VrijediOd | datetime | Nije obavezno polje | Datum od kad vrijedi parametar |
| VrijediDo | datetime | Nije obavezno polje | Datum do kad vrijedi parametar |
| Vrijednost | varchar | Duljine do 6000 znakova; nije obavezno polje | Vrijednost |
| LASTUSER_Id | int | Nije obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
| Nevidljivo | bit | Nije obavezno polje | |

3.1.14. Tablica sadrži podatke o radniku

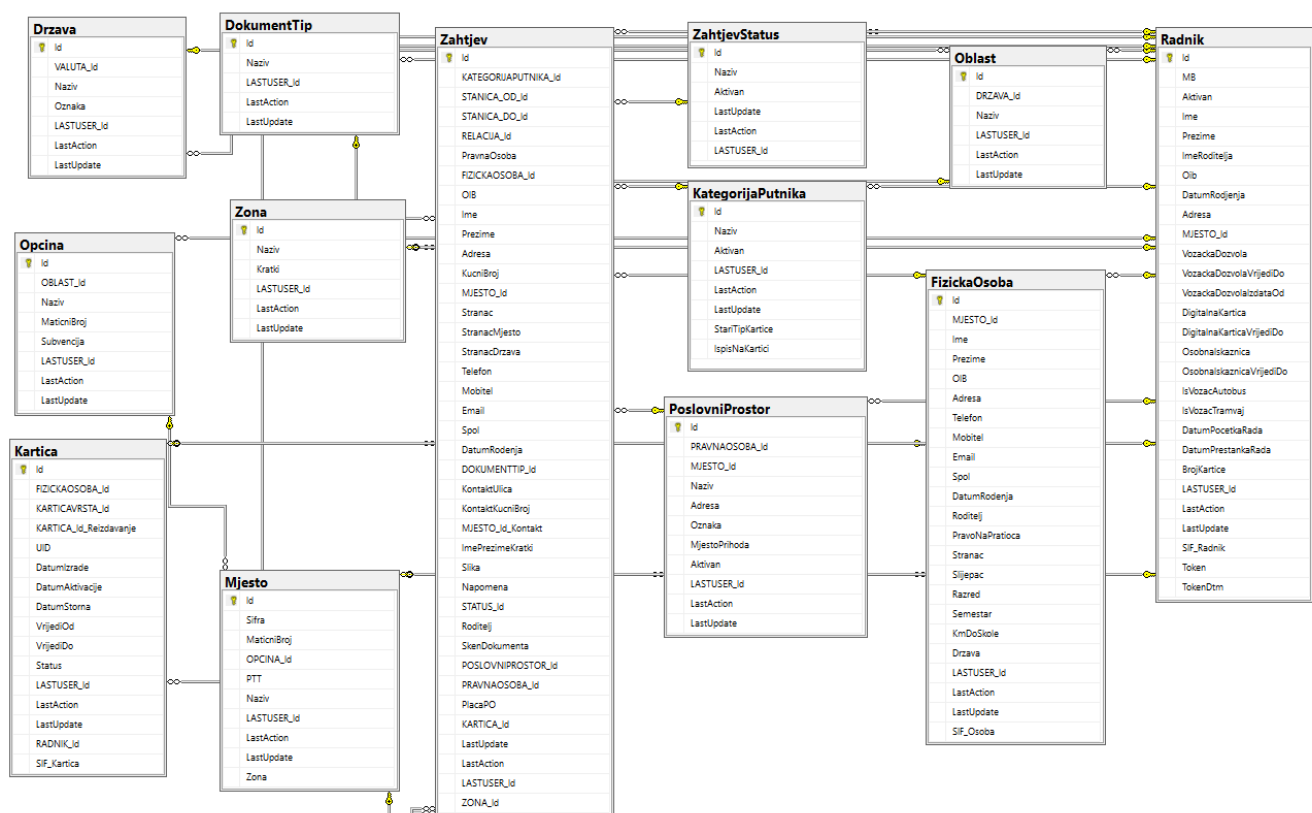
Tablica koja sadrži podatke o radniku sastoji se od 28 atributa. Popis atributa je prikazan u tablici 14. Atribut „Id“ je primarni ključ tablice i polje bez dupliciranja, odnosno ne dopušta unos dvije iste vrijednosti u tablicu. Atribut „MB“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „Aktivan“ je tipa bit i označava da li je radnik aktivan. Atributi „Ime“, „Prezime“, „ImeRoditelja“, „OIB“, su zapisi znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „DatumRodjenja“ je datumski tip podatka koji sprema datum u formatu dd.MM.yyyy zajedno s vremenom dana, koji se inače ne prikazuje. Atribut „Adresa“ je zapis znakovnog niza varijabilne duljine, maksimalno 100 znakova. Atribut „MJESTO_Id“ je cjelobrojni tip podatka. Atribut „VozackaDozvola“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „VozackaDozvolaVrijediDo“ je datumski tip podatka. Atribut „VozackaDozvolaIzdanaOd“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „DigitalnaKartica“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „DigitalnaKarticaVrijediDo“ je datumski tip podatka. Atribut „Osobnalskaznica“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „OsobnalskaznicaVrijediDo“ je datumski tip podatka. Atribut „IsVozacAutobus“ je tipa bit i označava da li je radnik vozač autobusa. Atribut „IsVozacTramvaj“ je tipa bit i označava da li je radnik vozač tramvaja. Atribut „DatumPocetkaRada“ je datumski i vremenski tip podatka i označava početak rada. Atribut „DatumPrestanakRada“ je datumski i vremenski tip podatka i označava prestanak rada. Atribut „BrojKartice“ je zapis znakovnog niza varijabilne duljine, maksimalno 50 znakova. Atribut „LASTUSER_Id“ je cjelobrojni tip podatka koji označava „Id“ radnika. Atribut „LastAction“ je tipa nchar i ima ograničenje na jedan znak koji može biti D za „DELETE“ – obrisani redak, U za „UPDATE“ – ažurirani redak te I za „INSERT“ – dodani novi redak. Atribut „LastUpdate“ je datumski i vremenski tip podatka i označava zadnje vrijeme izmjene. Atribut „SIF_Radnik“ je cjelobrojni tip podatka i sadrži šifru radnika. Atribut „Token“ je uniqueidentifier što označava jedinstveni identifikacijski broj koji se koristi prilikom poziva udaljenih procedura. Atribut „TokenDtm“ je datumski i vremenski tip podatka.

Tablica 14. Popis atributa koji se nalaze u tablici dbo.Radnik

| Naziv | Tip podataka | Ograničenja | Opis |
|----------------------------------|--------------|---|---|
| Id | int | Primarni ključ tablice; indeksirano polje bez dupliciranja; obavezan unos | Unikatni identifikacijski broj |
| MB | nvarchar | Duljine do 50 znakova; nije obavezan unos | |
| Aktivan | bit | Obavezan unos | Aktivnost radnika |
| Ime | nvarchar | Duljine do 50 znakova; obavezan unos | Ime radnika |
| Prezime | nvarchar | Duljine do 50 znakova; obavezan unos | Prezime radnika |
| ImeRoditelja | nvarchar | Duljine do 50 znakova; obavezan unos | Ime roditelja |
| OIB | nvarchar | Duljine do 50 znakova; obavezan unos | Osobni identifikacijski broj svakog hrvatskog državljana i pravne osobe |
| DatumRodenja | date | Obavezan unos | Datum rođenja radnika |
| Adresa | nvarchar | Duljine do 50 znakova; obavezan unos | Adresa korisnika |
| MJESTO_Id | int | Nije obavezan unos | Mjesto radnika |
| VozackaDozvola | nvarchar | Duljine do 50 znakova; nije obavezan unos | Vozačka dozvola |
| VozackaDozvolaVrijediDo | date | Nije obavezan unos | Datum do kad vrijedi vozačka dozvola |
| VozackaDozvolaIzdanaOd | nvarchar | Duljine do 50 znakova; nije obavezan unos | Podataka od kud je izdata vozačka dozvola |
| DigitalnaKartica | nvarchar | Duljine do 50 znakova; nije obavezan unos | Digitalna kartica |
| DigitalnaKarticaVrijediDo | date | Nije obavezan unos | Datum do kad vrijedi digitalna kartica |
| Osobnalskaznica | nvarchar | Duljine do 50 znakova; obavezan unos | Osobna iskaznica |

| | | | |
|---------------------------------|------------------|---|--|
| OsobnalskaznicaVrijediDo | date | Obavezan unos | Datum do kad vrijedi osobna iskaznica |
| IsVozacAutobus | bit | Obavezan unos | Podatak da li je radnik vozač autobusa |
| IsVozacTramvaj | bit | Obavezan unos | Podatak da li je radnik vozač tramvaja |
| DatumPocetkaRada | datetime | Nije obavezan unos | Datum početka rada |
| DatumPrestankaRada | datetime | Obavezan unos | Datum prestanka rada |
| BrojKartice | nvarchar | Duljine do 50 znakova; nije obavezan unos | Broj kartice |
| LASTUSER_Id | int | Obavezno polje | Radnik |
| LastAction | nchar | Duljina jednog znaka; Obavezno polje | D(Delete), U(Update), I(Insert) |
| LastUpdate | datetime | Obavezno polje | Zadnje vrijeme izmjene |
| SIF_Radnik | int | Nije obavezno polje | Šifra radnika |
| Token | uniqueidentifier | Nije obavezno polje | Token |
| TokenDtm | datetime | Nije obavezno polje | Datum |

Na slici 10. je prikazan kompletan dijagram entiteta i veza.



Slika 10. Kompletan dijagram entiteta i veza

3.2. Relacijska baza podataka

U relacijskom modelu podaci su organizirani u više tablica, koje su u međusobnoj vezi. Najčešće je u upotrebi tzv. ER (eng. entity-relationship) model. U relacijskom modelu postoje polja (stupci) i slogovi (redci), pri čemu se izdvajaju polja koja su ključevi (po kojima se tablice međusobno povezuju). Primarni ključ je jedinstveni identifikator entiteta u jednoj tablici. Karakteristike relacijskog modela podataka su: neredundantnost podataka, manje su podložne greškama, moguć lakši unos gotovo neograničenog broja podataka u retke tablice. Dizajn tablice u relacijskoj bazi čine stupci (polja) u kojima se određuje što će se pohraniti u pojedino polje, npr. Ime, Prezime, Adresa. itd., zatim tipovi podataka u poljima, ostala svojstva kao što su duljina, način zapisa, indeksi, i dr. svojstva prikazana u prozoru Properties, primarni ključ tablice (identifikator), npr. ID. Kod programskih aplikacija se dio aplikacije koji kontrolira unos i ispis podataka (odnosno komunikaciju s korisnikom) naziva FRONT END, dok se sam izvor podataka naziva BACK END. ASP .NET razvojna okolina sadrži alate za rad sa SQL Serverom [20].

3.3. SQL Server baza podataka

SQL Server baza podataka je relacijska baza podataka, odnosno sustav za upravljanje bazom podataka temeljen na relacijskom modelu, čiji je proizvođač Microsoft i namijenjena je pohrani i upravljanju podacima. Prednosti korištenja su široko korišten alat za upravljanje bazom, robustan skup korisničkih alata za upravljanje podacima (alati Enterprise Manager i Query Analyzer za analize podataka), velike mogućnosti manipulacije podacima i prijenos u skladište podataka (DTS alati za transformaciju podataka). Prednost je i rukovanje velikom količinom podataka i mogućnost rada na Internetu. Nedostaci su rukovanje samom bazom jer još uvijek nije jednostavno, Visual Studio 2017 sadrži alate koji olakšavaju pristup i obradu podataka iz baze SQL Server, te klase i kontrole za rad s podacima [20].

Glavni alat za rukovanje bazom podataka iz Visual Studia 2017 je Server Explorer. Server Explorer alat omogućuje povezivanje s bazom podataka (konekciju) koja se nalazi na lokalnom računalu, kreiranje baze podataka i dodavanje objekata u bazi (tablice, pogledi i dr.), održavanje tablica u bazi (dizajniranje strukture tablica), održavanje pogleda (eng. views), održavanje uskladištenih

procedura (u Transact SQL-u koji se koristi za modifikaciju podataka). U prozoru Solution Explorer, u mapi Tables se nalaze tablice iz baze podataka koja je povezana s aplikacijom, te se otvaranjem te mape može dobiti uvid u podatke u tablici.

ADO.NET (eng. ActiveX Data Objects .NET) je skup klasa koje .NET okolina sadrži za rad s podacima u aplikacijama. Razvijen je od strane Microsofta kako bi programerima olakšao pristup podacima iz .NET aplikacija [20].

3.4. Entity Framework

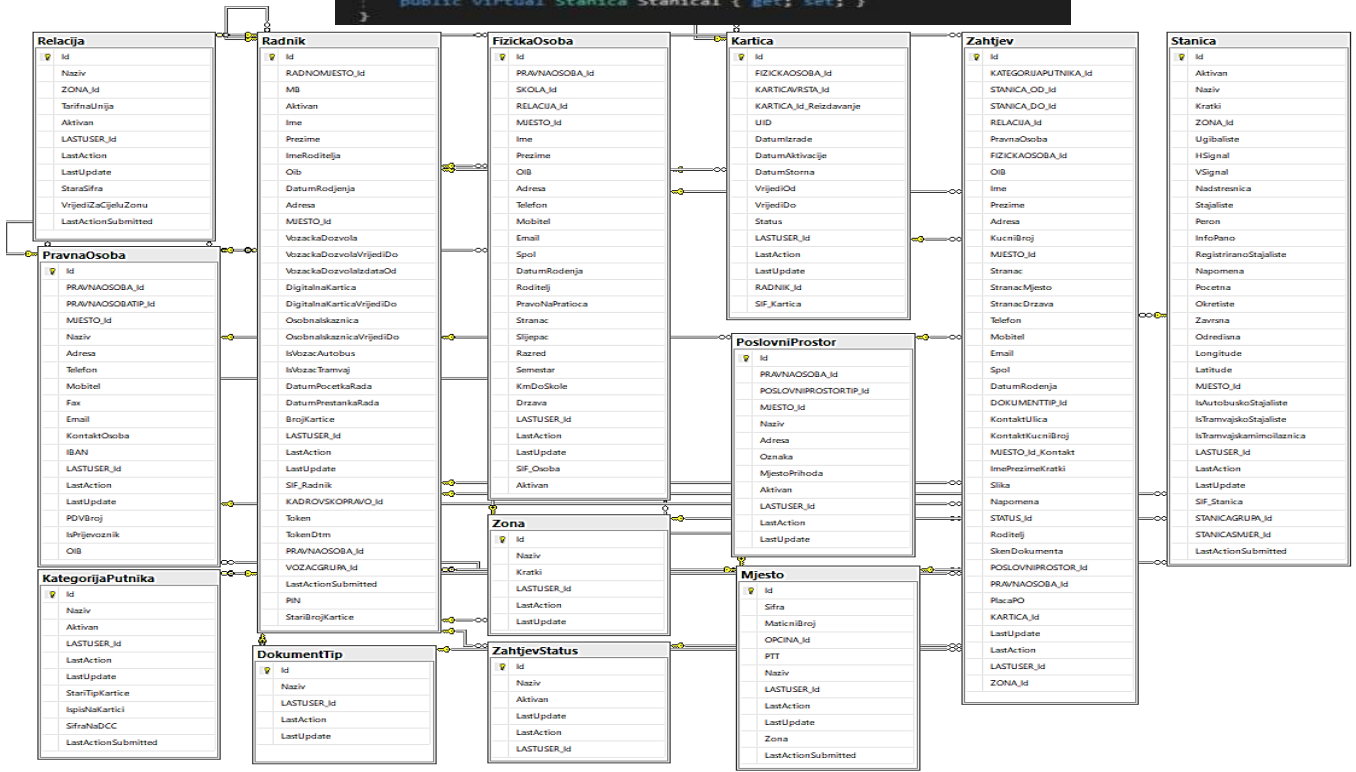
Glavni zadatak EF-a (eng. Entity Framework-a) je olakšati C# kodu rad s podacima u bazi podataka. C# objekti su sasvim različiti od tablica pohranjenih u relacijskoj bazi podataka, a upravljanje tim razlikama i prijenos podataka između C# koda i SQL server baze podataka se zove mapiranje (eng. mapping). Zbog toga se Entity framework naziva Object relational mapper-om (ORM) jer mapira relacijske tablice u objekte razumljive C# jeziku [20].

```

public partial class Zahtjev
{
    public int Id { get; set; }
    public int KATEGORIJAPUTNIKA_Id { get; set; }
    public Nullable<int> STANICA_OD_Id { get; set; }
    public Nullable<int> STANICA_DO_Id { get; set; }
    public Nullable<int> RELACIJA_Id { get; set; }
    public string PravnaOsoba { get; set; }
    public Nullable<int> FIZICKAOSOBA_Id { get; set; }
    public string OIB { get; set; }
    public string Ime { get; set; }
    public string Prezime { get; set; }
    public string Adresa { get; set; }
    public string KucniBroj { get; set; }
    public Nullable<int> MJESTO_Id { get; set; }
    public bool Stranac { get; set; }
    public string StranacMjesto { get; set; }
    public string StranacDrzava { get; set; }
    public string Telefon { get; set; }
    public string Mobitel { get; set; }
    public string Email { get; set; }
    public string Spol { get; set; }
    public System.DateTime DatumRodjenja { get; set; }
    public int DOKUMENTTIP_Id { get; set; }
    public string KontaktUlica { get; set; }
    public string KontaktKucniBroj { get; set; }
    public Nullable<int> MJESTO_Id_Kontakt { get; set; }
    public string ImePrezimeKratki { get; set; }
    public byte[] Slika { get; set; }
    public string Napomena { get; set; }
    public string Roditelj { get; set; }
    public byte[] SkenDokumenta { get; set; }
    public Nullable<int> POSLOVNIPROSTOR_Id { get; set; }
    public Nullable<int> PRAVNAOSOBA_Id { get; set; }
    public Nullable<bool> PlacaPO { get; set; }
    public Nullable<int> KARTICA_Id { get; set; }
    public Nullable<System.DateTime> LastUpdate { get; set; }
    public string LastAction { get; set; }
    public Nullable<int> LASTUSER_Id { get; set; }
    public byte STATUS_Id { get; set; }
    public Nullable<int> ZONA_Id { get; set; }

    public virtual DokumentTip DokumentTip { get; set; }
    public virtual FizickaOsoba FizickaOsoba { get; set; }
    public virtual Kartica Kartica { get; set; }
    public virtual Mjesto Mjesto { get; set; }
    public virtual PoslovniProstor PoslovniProstor { get; set; }
    public virtual PravnaOsoba PravnaOsoba { get; set; }
    public virtual Relacija Relacija { get; set; }
    public virtual Radnik Radnik { get; set; }
    public virtual ZahtjevStatus ZahtjevStatus { get; set; }
    public virtual Zona Zona { get; set; }
    public virtual Mjesto Mjesto { get; set; }
    public virtual KategorijaPutnika KategorijaPutnika { get; set; }
    public virtual Stanica Stanica { get; set; }
    public virtual Stanica Stanica { get; set; }
}

```



Slika 11. Mapiranje u Entity Framework-u

Slika 11. prikazuje mapiranje koje se odvija u oba smjera. Podaci se preuzimaju iz baze podataka te se učitavaju u C# objekte. Ako se koji slučajem promijene objekti ili izrade novi, baza podataka ažurira tablice u skladu s tim. Za svaku tablicu/entitet u bazi podataka Entity Framework koristi odgovarajuću C# klasu s uobičajenim svojstvima za svaki atribut/kolonu u SQL tablici. Pored klasa koje predstavljaju tablice u bazi, mora biti definirana i jedna posebna klasa koja je prikazana na slici 12. i predstavlja kontekst baze podataka (DbContext). Koristi se za dobivanje objekata entiteta koji predstavljaju podatke koji se već nalaze u bazi podataka, te se također koristi da kreirane objekte entiteta iz aplikacije pohrane u bazu [20].

```
namespace Penta.BusCARD.DAL.Model
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;
    using System.Data.Entity.Core.Objects;
    using System.Linq;

    public partial class BusCARDEntities : DbContext
    {
        public BusCARDEntities()
            : base("name=BusCARDEntities")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Zahtjev> Zahtjev { get; set; }
        public virtual DbSet<DokumentTip> DokumentTip { get; set; }
    }
}
```

Slika 12. Kontekst klasa

Klasa sadrži konstruktora koji se izvršava prilikom instanciranja klase. On predaje baznoj klasi DbContext konekciju na bazu, prikazanu na slici 13. te je on traži u konfiguracijskoj datoteci aplikacije unutar tagova <connectionStrings> </connectionStrings>. Na slici 13. zbog autorske sigurnosti skriveni su naziv i lozinka baze podataka.

```
<connectionStrings>
  <add name="BusCARDEntities" connectionString="metadata=res://*/
Model.BusCARDModel.csdl|res://*/Model.BusCARDModel.ssd
l|res://*/Model.BusCARDModel.msl;provider=System.Data.SqlClient;
provider connection string="data source=LAPTOP-12-772111\SQLEXPRESS1;
initial catalog=PulaDev;persist security info=True;
user id=;password=;MultipleActiveResultSets=True;
App=EntityFramework"; providerName="System.Data.EntityClient" />
</connectionStrings>
```

Slika 13. Konekcija u App.Config datoteci

4. WEB USLUGE

Web usluge su kao mehanizmi za prikazivanje korisnih podataka koji se mogu preuzeti putem HTTP-a. Web se počeo masovno koristiti tijekom 1990-ih godina te su milijune ljudi širom svijeta pristupali novim web-aplikacijama. No, postojalo je i mnogo zastarjelih aplikacija koje su se mogle koristiti samo u jednoj tvrtki. U takvom okruženju, postojala je potreba za izlaganjem podataka široj publici, što je rezultiralo stvaranjem standardnog sučelja za pristup podacima preko weba. Razlika između tradicionalnih web-aplikacija i web-usluga je u tome što web-usluge pružaju samo informacije i nisu zainteresirane za to kako izgleda korisničko sučelje. Na primjer, banka može ponuditi informacije o svojim proizvodima putem web-usluge koja vraća XML dokumente. Klijenti ne moraju znati da se aplikacija banke izvodi na IBM računalu. Tisuće javno dostupnih web-usluga nude različite API-je za prikupljanje podataka. Velike tvrtke koriste i privatne web-usluge za vlastite potrebe [19].

4.1. Podjela i karakteristike

Postoje dva tipa web usluga:

1. SOAP Web usluge
2. REST Web usluge

4.1.1. SOAP Web usluge

SOAP je protokol utemeljen na XML-u. Najveća prednost korištenja SOAP Web usluge je vlastita sigurnost. SOAP je kratica za Simple Object Access Protocol. Na slici 14. prikazana je SOAP omotnica koja pruža omotnicu za slanje poruka Web usluga putem Interneta, koristeći HTTP protokol. Poruke su uglavnom u XML formatu [18].

SOAP omotnica



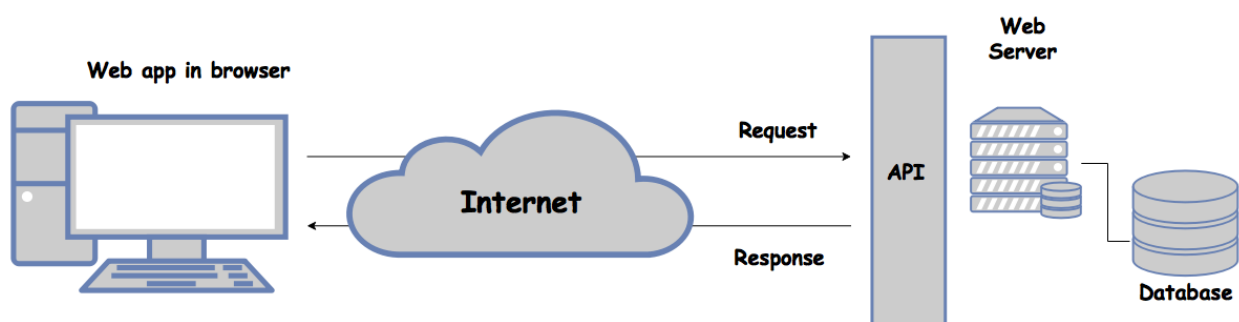
Slika 14. SOAP omotnica (izrada: autor)

Jednostavnim riječima, SOAP je tehnika za slanje XML zahtjeva preko Interneta pomoću HTTP protokola (udarajući URL), a zauzvrat dobivanje XML odgovora.

Svaka aplikacija koja služi SOAP zahtjevima ima WSDL datoteku. WSDL je XML. WSDL je kratica za Web Service Description Language, odnosno web jezik za opis usluge. WSDL opisuje sve metode dostupne u web usluzi, zajedno s vrstama zahtjeva i odgovora. Opisuje ugovor između usluge i klijenta. SOAP je trebao biti način za daljnje pozive procesa udaljenim objektima slanjem XML-a preko HTTP-a. Ako pogledamo trenutnu industriju softvera, SOAP se koristi u poslovnim aplikacijama, općenito u starom kodu. Danas se svijet brzo kreće prema RESTful web uslugama [18].

4.1.2. REST Web usluge

REST je kratica za Representational State Transfer. REST nije skup standarda ili pravila, već stil softverske arhitekture. Aplikacije koje slijede ovu arhitekturu, prikazanu na slici 15. nazivaju se RESTful. Za razliku od SOAP-a koji cilja akcije, REST se više tiče resursa. REST locira resurse pomoću URL-a i ovisi o vrsti transportnog protokola (s HTTP - GET, POST, PUT, DELETE, ...) za akcije koje se izvode na resursima. Usluga REST locira resurs na temelju URL-a i izvršava akciju temeljenu na glagolu transportne akcije. To je više arhitektonski stil i temeljen je na konvenciji [18].



Slika 15. RESTful WEB API [11]

U tablica 15. prikazane su osnovne razlike između REST i SOAP Web usluge.

Tablica 15. Prikaz osnovne razlike između dvije vrste Web usluga [18]:

| REST | SOAP |
|---|--|
| REST je stil softverske arhitekture. | SOAP je protokol ili skup standarda. |
| REST je kratica za Representational State Transfer | SOAP je kratica za Simple Object Access Protocol |
| REST može koristiti SOAP jer je to koncept i može koristiti bilo koji protokol kao što je HTTP, SOAP itd. | SOAP ne može koristiti REST jer je sam po sebi protokol. |
| REST koristi URI za izlaganje poslovne logike. No, kao REST radi na temelju vrste HTTP | SOAP koristi sučelje usluge kako bi razotkrio poslovnu logiku. |

| | |
|---|--|
| zahtjeva, stoga isti URI može raditi za više od jedne vrste operacije. | |
| REST ne definira previše standarda. | SOAP definira standarde koje treba strogo slijediti. |
| REST nasljeđuje sigurnosne mjere od temeljnih transportnih protokola. | SOAP definira vlastiti sigurnosni sloj. |
| REST prihvaća različite formate podataka, kao što su običan tekst, HTML, JSON, XML itd. | SOAP radi samo u XML formatu. |

4.2. Web API

Web API označava aplikacijsko programsko sučelje na strani poslužitelja ili klijenta u mrežnom sustavu. Na poslužiteljskoj strani je programsko sučelje koje čini skup javno dostupnih krajnjih točaka (eng. Endpoints) kojima klijenti mogu pristupiti. Krajnje točke se koriste u definiranom sustavu razmjene poruka između klijenta i poslužitelja putem web-a, odnosno najčešće HTTP protokola. Najpoznatiji tip web API-ja na poslužiteljskoj strani je REST API [1].

4.3. Primjer dodavanja javno dostupne krajnje točke Zona

Primjer se odnosi na dodavanje endpoint-a zona. U klasu sučelje (eng. Interface) Iver1 se dodaje metoda „POST“ prikazana na slici 16. Metoda POST služi za stvaranje zapisa.

```

269
270     [OperationContract]
271     [WebInvoke(Method = "POST",
272               ResponseFormat = WebMessageFormat.Json,
273               RequestFormat = WebMessageFormat.Json,
274               BodyStyle = WebMessageBodyStyle.Bare)]
275     GetZonesOut GetZones(LozIn inobj);
276

```

Slika 16. Dodavanje metode POST

Zatim se kreira klasa ZonaDTO, prikazana na slici 17., koja vraća Id i naziv zone.

```
ZonaDTO.cs  + X
Penta.BusCard.WebService  Penta.BusCard.WebService.Dto.ZonaDTO
using Penta.BusCard.WebService.DAL.Repositories.Osijek;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Web;

namespace Penta.BusCard.WebService.Dto
{
    [DataContract]
    public class ZonaDTO
    {
        int id;
        string naziv;

        [DataMember]
        public int Id { get => id; set => id = value; }

        [DataMember]
        public string Naziv { get => naziv; set => naziv = value; }

        public static ZonaDTO FromZone(Zona row)
        {
            if (row == null)
            {
                return null;
            }
            return new ZonaDTO
            {
                Id = row.Id,
                Naziv = row.Naziv,
            };
        }
    }
}
```

Slika 17. Kreiranje klase ZonaDTO

Klasa GetZonesOut prikazana na slici 18. sadrži listu Zona koja se puni iz baze podataka.

```
GetZonesOut.cs  Penta.BusCard.WebService.Composit

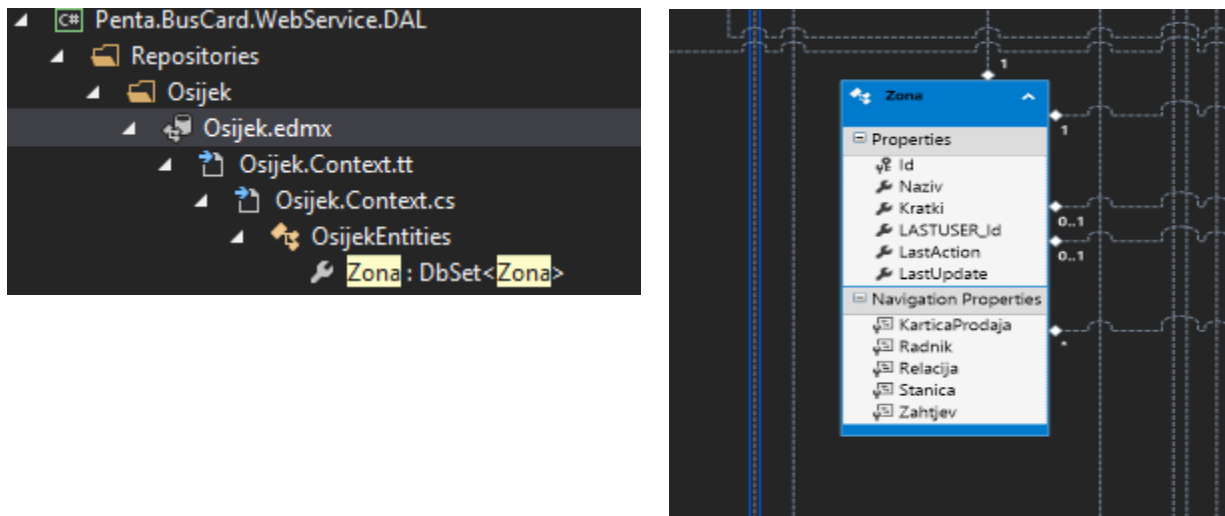
using Penta.BusCard.WebService.Dto;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Web;

namespace Penta.BusCard.WebService.CompositeTypes
{
    [DataContract]
    [KnownType(typeof(ZonaDTO))]
    public class GetZonesOut : BaseOut
    {
        List<ZonaDTO> zons = new List<ZonaDTO>();

        [DataMember]
        public List<ZonaDTO> Zone { get => zons; set => zons = value; }
    }
}
```

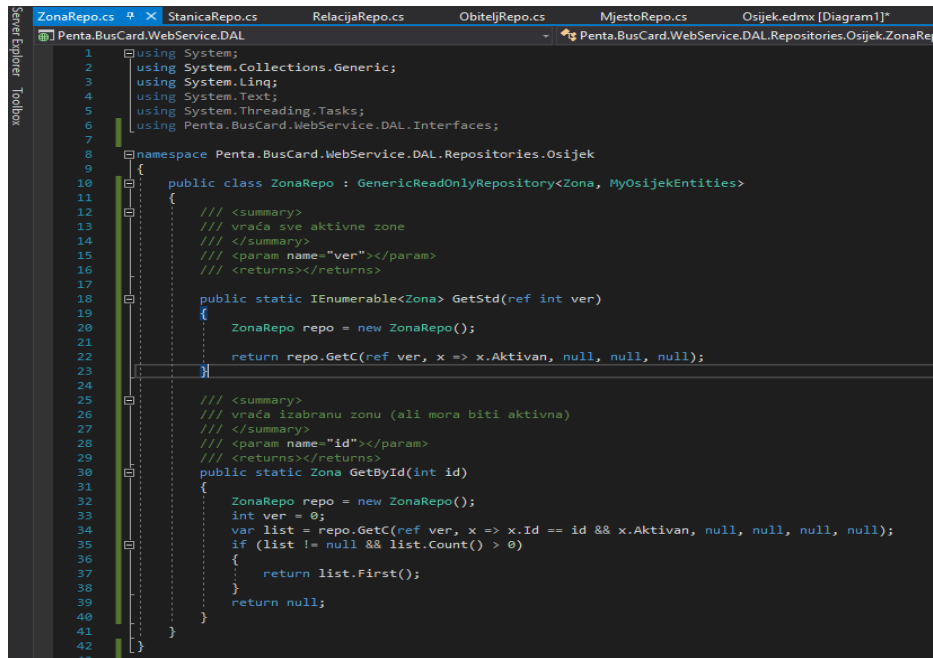
Slika 18. Klasa GetZonesOut.cs

Na slici 19. je prikazano dodavanje Zone u model.



Slika 19. Dodavanje Zone u model

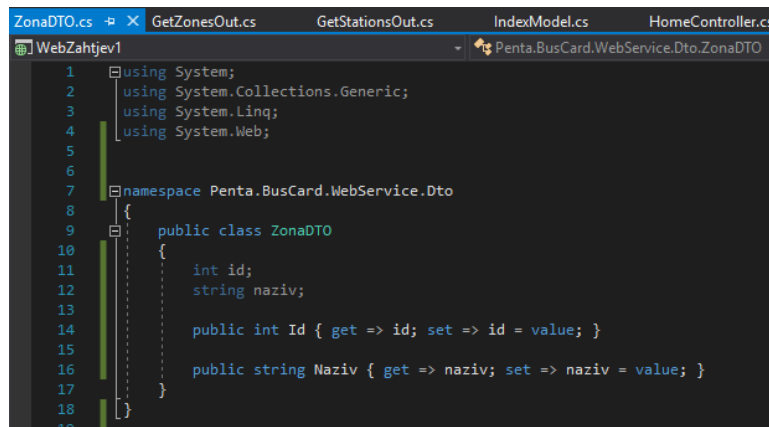
Klasa ZonaRepo prikazana na slici 20. vraća sve aktivne i izabrane zone koje moraju biti aktivne.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Penta.BusCard.WebService.DAL.Interfaces;
7
8 namespace Penta.BusCard.WebService.DAL.Repositories.Osijek
9 {
10     public class ZonaRepo : GenericReadOnlyRepository<Zona, MyOsijekEntities>
11     {
12         /// <summary>
13         /// vraća sve aktivne zone
14         /// </summary>
15         /// <param name="ver"></param>
16         /// <returns></returns>
17
18         public static IEnumerable<Zona> GetStd(ref int ver)
19         {
20             ZonaRepo repo = new ZonaRepo();
21             return repo.GetC(ref ver, x => x.Aktivan, null, null, null);
22         }
23
24         /// <summary>
25         /// vraća izabranu zonu (ali mora biti aktivna)
26         /// </summary>
27         /// <param name="id"></param>
28         /// <returns></returns>
29         public static Zona GetById(int id)
30         {
31             ZonaRepo repo = new ZonaRepo();
32             int ver = 0;
33             var list = repo.GetC(ref ver, x => x.Id == id && x.Aktivan, null, null, null);
34             if (list != null && list.Count() > 0)
35             {
36                 return list.First();
37             }
38             return null;
39         }
40     }
41 }
42
43
```

Slika 20. Klasa ZonaRepo.cs

Zatim se u Web aplikaciju dodaje endpoint Zona. Prvo se u mapu Communication dodaje klasa ZonaDTO koja je prikazana na slici 21. i sadrži Id i Naziv, te klasa GetZoneOut prikazana na slici 22. koja sadrži listu.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5
6
7 namespace Penta.BusCard.WebService.Dto
8 {
9     public class ZonaDTO
10     {
11         int id;
12         string naziv;
13
14         public int Id { get => id; set => id = value; }
15
16         public string Naziv { get => naziv; set => naziv = value; }
17     }
18 }
19
```

Slika 21. Klasa ZonaDTO.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using Penta.BusCard.WebService.Dto;
6
7  namespace Penta.BusCard.WebService.CompositeTypes
8  {
9      public class GetZonesOut : BaseOut
10     {
11         List<ZonaDTO> zones = new List<ZonaDTO>();
12
13
14         public List<ZonaDTO> Zones
15         {
16             get => zones;
17             set => zones = value;
18         }
19     }
20 }
21

```

Slika 22. Klasa GetZoneOut.cs

Slika 23. prikazuje klasu Zahtjev u koju se dodaje atribut Zona_Id.

```

namespace WebZahtjev1.Models
{
    public class Zahtjev
    {
        private int Kategorijaputnikaid;
        private string pravnaosoba;
        public int? MJESTO_Id { get; set; }
        public int? MJESTO_Id_Kontakt { get; set; }
        private string ime;
        private string prezime;
        private string adresa;
        private string kucnibroj;
        private string telefon;
        private string mobitel;
        private string email;
        private string spol;
        private string datumrodenja;
        private bool stranac;
        private int Zonaid;
    }
}

```

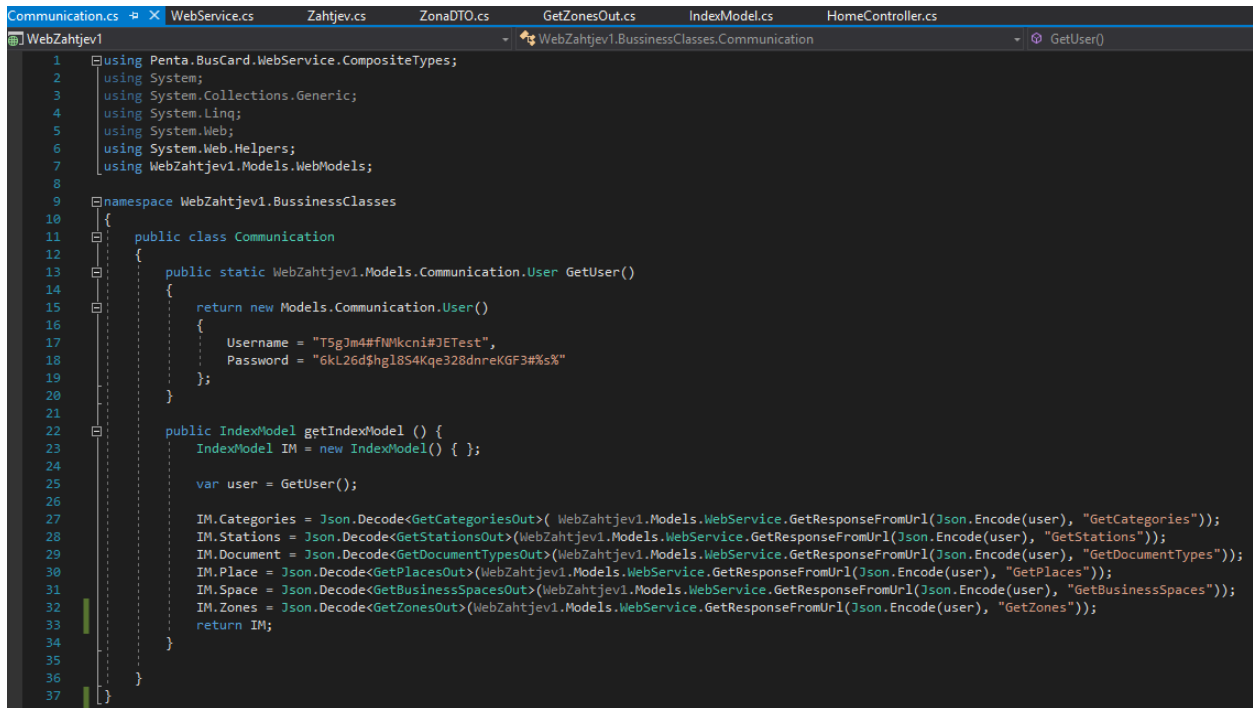
```

public int ZONA_Id
{
    get { return Zonaid; }
    set { Zonaid = value; }
}

```

Slika 23. Dodavanje u klasu Zahtjev atribut ZONA_Id

Zatim u klasu Communication, prikazanu na slici 24. instanciramo Zonu.



```
1 using Penta.BusCard.WebService.CompositeTypes;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Web;
6 using System.Web.Helpers;
7 using WebZajtjev1.Models.WebModels;
8
9 namespace WebZajtjev1.BusinessClasses
10 {
11     public class Communication
12     {
13         public static WebZajtjev1.Models.Communication.User GetUser()
14         {
15             return new Models.Communication.User()
16             {
17                 Username = "T5g2m4#fMkcnI#jETest",
18                 Password = "6kL26d$hg1854Kqe328dnreKGF3#%s%";
19             };
20         }
21
22         public IndexModel getIndexModel () {
23             IndexModel IM = new IndexModel() { };
24
25             var user = GetUser();
26
27             IM.Categories = Json.Decode<GetCategoriesOut>( WebZajtjev1.Models.WebService.GetResponseFromUrl(Json.Encode(user), "GetCategories"));
28             IM.Stations = Json.Decode<GetStationsOut>(WebZajtjev1.Models.WebService.GetResponseFromUrl(Json.Encode(user), "GetStations"));
29             IM.Document = Json.Decode<GetDocumentTypesOut>(WebZajtjev1.Models.WebService.GetResponseFromUrl(Json.Encode(user), "GetDocumentTypes"));
30             IM.Place = Json.Decode<GetPlacesOut>(WebZajtjev1.Models.WebService.GetResponseFromUrl(Json.Encode(user), "GetPlaces"));
31             IM.Space = Json.Decode<GetBusinessSpacesOut>(WebZajtjev1.Models.WebService.GetResponseFromUrl(Json.Encode(user), "GetBusinessSpaces"));
32             IM.Zones = Json.Decode<GetZonesOut>(WebZajtjev1.Models.WebService.GetResponseFromUrl(Json.Encode(user), "GetZones"));
33
34             return IM;
35         }
36     }
37 }
```

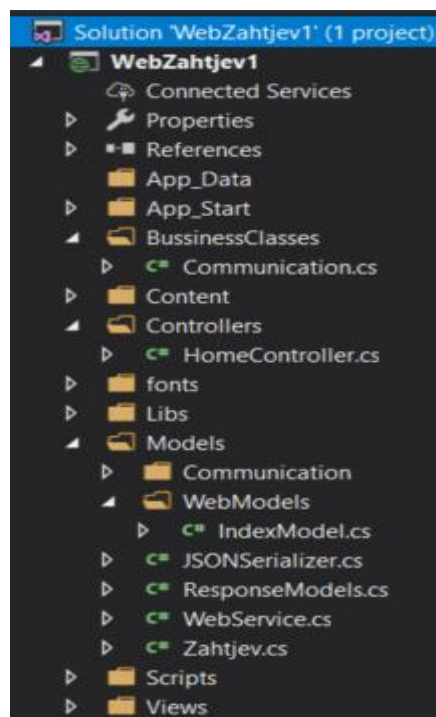
Slika 24. Instanciranje zone

5. IMPLEMENTACIJA WEB APLIKACIJE

Web aplikacija za predaju zahtjeva za izradom beskontaktne kartice realizirana je kao programsko rješenje – ASP.NET aplikacije, kroz Model-Pogled-Upravljač(eng. MVC – Model – View - Controller) arhitekturu uz Web uslugu preko koje se šalju podaci iz baze podataka.

5.1. Prikaz programskih komponenti

Sama aplikacija sastoji se od više datoteka koje su prikazane na slici 25. Mapa Controllers koja sadrži korišten kontroler (HomeController). Mapa Models koja sadrži podmapu WebModels i Communication, klase JsonSerializer, ResponseModels, Webservice i Zahtjev. Mapa View sadrži korištene poglede organizirane kao pogledi početne stranice, automatski generirani od strane razvojnog okruženja – Home, dijeljeni pogledi (eng. Shared) za Layout koji je zajednički većini pogleda. U Web.config datoteku se spremaju informacije o Entity Framework-u i Web usluzi. Mapa Libs sadrži klasu ImageOperations koja sadrži različite funkcije za slike. Mape Content i Scripts sadrže potrebne dodatne datoteke koje definiraju izgled i ponašanje određenih elemenata, tj. dijelova aplikacije.



Slika 25. Komponente web aplikacije

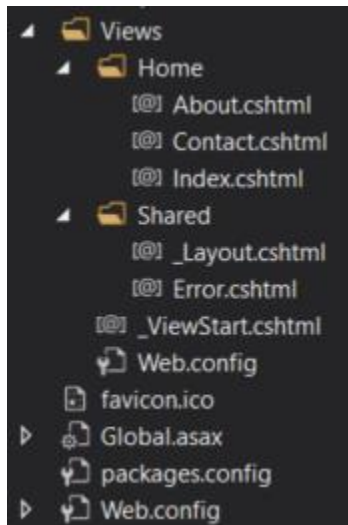
Unutar mape WebModels se nalazi klasa IndexModel koja služi kao kontejner cijele stranice. Mapa Communication sadrži klase za komunikaciju sa Web uslugom. Klasa JsonSerializer serijalizira podatke u JSON (služi kako bi Web usluga i aplikacija lakše ustanovili o kojim se podacima radi i što moraju s njima raditi) i deserijalizira ih u odgovarajuće podatkovne objekte koji se mogu koristiti kroz aplikaciju. Klasa ResponseModels sadrži funkciju getZahtjev, prikazanu na slici 26. u kojoj dohvaća sve podatke iz klase Zahtjev.

```
public Zahtjev getZahtjev()
{
    System.Globalization.CultureInfo cultureinfo = new System.Globalization.CultureInfo("hr-HR");
    return new Zahtjev()
    {
        Adresa = (Adresa == null) ? "" : Adresa,
        DatumRodenja = DateTime.Parse(DatumRodenja, cultureinfo).ToString("dd.MM.yyyy."),
        DOKUMENTTIP_Id = DOKUMENTTIP_Id,
        SkenDokumenta = toBase64Stream(SkenDokumenta),
        Slika = toBase64StreamResized(Slika),
        Email = (Email == null) ? "" : Email,
        Ime = Ime,
        KATEGORIJAPUTNIKA_Id = KATEGORIJAPUTNIKA_Id,
        KontaktDrzava = (KontaktDrzava == null) ? "" : KontaktDrzava,
        MJESTO_Id = MJESTO_Id,
        PTT = (PTT == null) ? "" : PTT,
        KontaktUlica = (KontaktUlica == null) ? "" : KontaktUlica,
        KucniBroj = (KucniBroj == null) ? "" : KucniBroj,
        KontaktMjesto = (KontaktMjesto == null) ? "" : KontaktMjesto,
        MJESTO_Id_Kontakt = MJESTO_Id_Kontakt,
        Mobitel = (Mobitel == null) ? "" : Mobitel,
        PravnaOsoba = (PravnaOsoba == null) ? "" : PravnaOsoba,
        Prezime = Prezime,
        Spol = Spol.ToString(),
        Stranac = Stranac,
        Telefon = (Telefon == null) ? "" : Telefon,
        OIB = OIB,
        Roditelj = (Roditelj == null) ? "" : Roditelj,
        StranacDrzava = (StranacDrzava == null) ? "" : StranacDrzava,
        StranacMjesto = (StranacMjesto == null) ? "" : StranacMjesto,
        POSLOVNIPROSTOR_Id = (Dostava == 1) ? POSLOVNIPROSTOR_Id : 0,
        ZONA1_Id = ZONA1_Id,
        ZONA2_Id = ZONA2_Id,
        ZONA3_Id = ZONA3_Id,
        ZONA4_Id = ZONA4_Id
    };
}
```

Slika 26. Klasa Zahtjev - funkcija getZahtjev

Klasa WebService sadrži URL localhost-a i upite za Web uslugu. U klasi Zahtjev su definirane metode get i set. Takve metode se još zovu Getteri i Setteri. Gettere koristimo za čitanje odnosno dohvaćanje vrijednosti nekog elementa, dok Settere za postavljanje ili izmjenu vrijednosti nego elementa [4].

Na osnovu HomeControllera stvoreni su pogledi u Home mapi prikazani na slici 27. About za detalje o aplikaciji, Contact za prikaz informacija o kontaktu, Indeks za početnu stranicu. Svi navedeni pogledi koriste zajednički Layout iz mape Shared kojim je definiran osnovni izgled pogleda (pozadina, izbornik, naslov).



Slika 27. Datoteke pogleda

U Content-u se nalaze biblioteke za CSS dizajn HTML elemenata u web aplikaciji i bootstrap biblioteke. U Scripts se nalaze JavaScript funkcije koje se mogu pozvati.

5.2. Primjeri koda za pojedine dijelove aplikacije

Sadržaj funkcije na slici 28. pisan je u programskom jeziku C#. Funkcija za provjeru OIB-a preuzeta je sa bloga od Domagoja Pavlešića [3]. OIB mora imati 11 znamenki, odnosno 10 znamenki koje se dodjeljuju slučajnim odabirom i jednu kontrolnu znamenku. OIB se provjerava na temelju zadnje kontrolne znamenke.


```

function CheckOIB(oib) {
    oib = oib.toString();

    if (oib.length != 11) return false;
    var b = parseInt(oib, 10);
    if (isNaN(b)) return false;

    var a = 10;
    for (var i = 0; i < 10; i++) {
        a = a + parseInt(oib.substr(i, 1), 10);
        a = a % 10;
        if (a == 0) a = 10;
        a *= 2;
        a = a % 11;
    }
    var kontrolni = 11 - a;
    if (kontrolni == 10) kontrolni = 0;

    return kontrolni == parseInt(oib.substr(10, 1));
}

```

Slika 28. Funkcija za provjeru OIB-a

Funkcija koja je prikazana na slici 29. pisana je u JavaScript-u. Ona provjerava da li je OIB ispravan te ako nije ispiše poruku „Neispravan OIB!“ i okvir od komponente TextBox pocrveni.

```

function ProvjeraOIBA(element){
    element.value=element.value.trim();
    var odg = "";
    element.style.borderColor = "";

    if (!CheckOIB(element.value.trim())) {

        odg="Neispravan OIB!\n";
        element.style.borderColor = "red";
        $(element).change(function () { setErrorLength(element, "") })
        return odg;
    }

    return odg;
}

```

Slika 29. Funkcija za neispravan OIB

Funkcija za zone prikazana na slici 30. je također pisana u JavaScript-u. U ovoj funkciji postoji jedan uvjet, a to je da se ne smije označiti prva i treća zona, bez da se označi i druga zona. U protivnom se pojavi upozorenje „Morate označiti i zonu 2“.

```
function validate(obje) {  
    if (document.getElementById('ZONA1_Id').checked &&  
        document.getElementById('ZONA3_Id').checked &&  
        !document.getElementById('ZONA2_Id').checked) {  
        alert("Morate označiti i zonu 2" );  
        obje.checked = obje.id == 'ZONA2_Id';  
    }  
}
```

Slika 30. Funkcija za označavanje zona

Postoje funkcije za stranca i kontakt podatke koje su prikazane na slici 31. One se aktiviraju nakon označenog potvrdnog okvira, nakon čega slijedu dodatna 2 ili 4 polja za popunjavanje podataka.

```
$("#Stranac").change(function () {  
    if ($(this).is(':checked')) {  
        $("#StranacDrzava").css("visibility", "visible");  
    } else {  
        $("#StranacDrzava").css("visibility", "hidden");  
    }  
});  
  
$("#KontaktRazlika").change(function () {  
    if ($(this).is(':checked')) {  
        $("#KontaktPodaci").css("display", "block");  
    } else {  
        $("#KontaktPodaci").css("display", "none");  
    }  
});
```

Slika 31. Funkcije za stranca i kontakt podatke

Funkcija za pravnu osobu, prikazana na slici 32. funkcionira na način da korisnik upiše na odgovarajuće mjesto naziv pravne osobe te se taj naziv kopira u odgovarajuće polje za dostavu kartice.

```
$("#PravnaOsoba").keyup(function () {  
    $("#copyPravnaOsoba").text(this.value);  
});
```

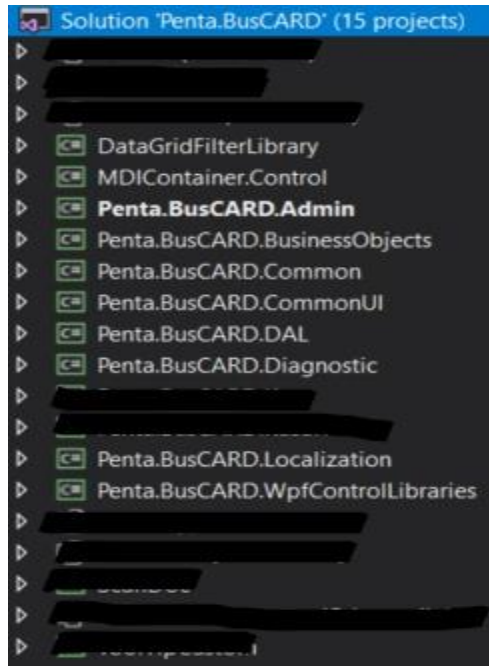
Slika 32. Funkcija za pravnu osobu

6. IMPLEMENTACIJA DESKTOP APLIKACIJE

Desktop aplikacija za obradu zahtjeva za izradom beskontaktna kartice realizirana je kao programsko rješenje – ASP.NET aplikacije, kroz Model-Pogled-Model pogleda (eng. MVVM – Model- View - ViewModel) arhitekturu uz Entity Framework, odnosno ADO.NET (eng. ActiveX Data Object .NET). ADO.NET je skup klasa koje .NET okruženje sadrži za rad s podacima u aplikaciji. Razvijen je od strane Microsofta kako bi programerima olakšao pristup podacima iz .NET aplikacije. Osnovni prostor za nazive za rad s podacima je: `Using System.Data;`

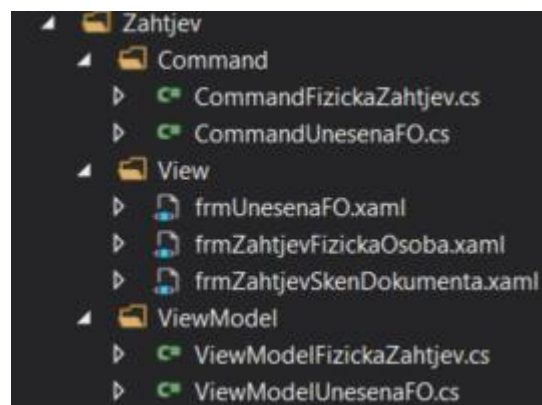
6.1. Prikaz programskih komponenti

Solucija (eng. Solution) aplikacije sastoji se od nekoliko važnijih projekata prikazanih na slici 33. Pojedini dijelovi slike 33. su zatamnjeni zbog sigurnosti tvrtke Penta d.o.o. u kojoj je ovaj diplomski rad izrađen. U projekt `Penta.BusCARD.Admin` se nalazi mapa `Forms`, a unutar mape `frmFizickaOsoba` i mapa `Zahtjev` sa podmapama `Command`, `View`, `ViewModel`. `MainWindow` je glavni zaslon aplikacije u kojem se nalazi izbornik i ostale komponente. U projektu `Penta.BusCARD.BusinessObjects`, u klasi `ZahtjevObject`, se nalaze definirani `business` i `basic` objekti. U `business` objektu dohvaćamo sve atribute, a u `basic` objektu dohvaćamo samo određene atribute, odnosno one koje prikazujemo na lijevom dijelu zaslona. Projekt `Penta.BusCARD.CommonUI` sadrži mapu `Resources` unutar koje se nalaze ikone komponenti gumba. Projekt `Penta.BusCARD.DAL` sadrži mapu `Mappers`, `Model` i `Repositories`. U mapi `Mappers` se nalazi klasa `ZahtjevMapper` i `ZahtjevStatusMapper` u kojoj se povezuju objekti sa entitetom iz baze. Entiteti su objekti, događaji koji nas zanimaju, a atributi svojstva ili obilježja entiteta.



Slika 33. Komponente desktop aplikacije

U Penta.BusCARD.Admin se nalazi mapa Forms unutar koje se nalazi mapa Zahtjev. Na slici 34. je prikazana mapa Zahtjev unutar koje se nalaze još tri mape: Command, View i ViewModel. U mapi Command se nalazi klasa u kojoj su definirane komande za gumbove (eng. buttons). U mapi View se nalazi frmZahtjevFizickaOsoba i frmUnesenaFO u kojima je definirana struktura i izgled zaslona. U mapi ViewModel-u se nalaze dvije klase u kojima su definirana svojstva i naredbe na koje se pogled može vezati i obavještavati o promjenama, poput CRUD operacija.



Slika 34. MVVM model Zahtjeva

6.2. Primjeri koda za pojedine dijelove aplikacije

U projektu Penta.BusCARD.DAL se nalazi mapa Repositories unutar koje se nalazi klasa FizickaOsobaRepository koja sadrži dvije glavne funkcije. Prva funkcija je InsertByZahtjev koja se referencira na ZahtjevObject i njena svrha je unos osobe, koja je predala zahtjev putem internetskog preglednika, u fizičku osobu. Druga funkcija je UpdateFromZahtjev koja se referencira na FizickaOsobaFromZahtjevObject i njena svrha je ažurirane postojeće fizičke osobe sa podacima predanim putem web zahtjeva. Funkcije sadrže kombinaciju nekoliko operacija u jednoj transakciji unutar istog konteksta. Na slici 35. je prikazana varijabla dbContextTransaction unutar koje je definirana metoda BeginTransaction() u kojoj počinje transakcija. Database.BeginTransaction() se koristi za jednostavniji način pokretanja i dovršavanja transakcije unutar postojećeg konteksta, uz kombinaciju nekoliko operacija unutar iste transakcije.

```
using (var dbContextTransaction = context.Database.BeginTransaction())
```

Slika 35. Metoda BeginTransaction()

Zatim se nakon transakcije definira stanje entiteta. Korištena su dva stanja entiteta, dodano (eng. Added) i modificirano (eng. Modified). Na slici 36. je prikazano stanje dodano koje se odnosi na dodavanje entiteta u kontekst, odnosno da će biti dodano u bazu podataka nakon što se pozove metoda spremanja SaveChange().

```
context.Set(typeof(FizickaOsobaDokument)).Add(entityFOD);  
context.Entry(entityFOD).State = EntityState.Added;  
context.SaveChanges();
```

Slika 36. Stanje entiteta dodano i metoda spremanja

Metoda SaveChange() se koristi za spremanje, svih promjena u kontekstu, u bazu podataka. Na slici 37. je prikazano stanje modificirano koje se odnosi na entitet koji već postoji u bazi podataka, a na kojem su moguće promjene. Nakon pozivanja metode, sva svojstva entiteta će biti označena

kao izmijenjena i uz poziv metode spremanja SaveChange() sve vrijednosti svojstva će biti poslane u bazu podataka.

```
context.Entry(entityFO).State = EntityState.Modified;
context.SaveChanges();
```

Slika 37. Stanje entiteta modificirano i metoda spremanja

Nakon metode spremanja, slijede metode Commit() i Rollback() koje su prikazane na slici 38. Metoda Commit() se koristi za potvrđivanje transakcije koja se izvršava, a metoda Rollback() se koristi za poništavanje transakcije koja se izvršava.

```
catch (Exception ex)
{
    dbContextTransaction.Rollback();
    throw new BusCARDCustomException(ex.Message);
}

dbContextTransaction.Commit();
```

Slika 38. Metode Commit() i Rollback()

Na slici 39. prikazana je klasa ZahtjevRepository u kojoj se nalazi funkcija Update, koja služi za ažuriranje statusa zahtjeva.

```
entity.STATUS_Id = item.ZahtjevStatus.Id;

entity.LASTUSER_Id = item.User.Id;
entity.LastAction = EnumeratorHelper.GetEnumValue(item.LastAction);
entity.LastUpdate = sad;

context.Entry(entity).State = EntityState.Modified;

context.SaveChanges();
```

Slika 39. Dio funkcije za ažuriranje statusa zahtjeva

7. KORISNIČKE UPUTE

7.1. Ispunjavanje web zahtjeva u internetskom pregledniku

Zahtjev za izdavanje kartica

| | |
|--|--|
| Osnovni podaci: | Priloženi dokument/isprava: |
| Pravna osoba <input type="text"/> | Tip dokumenta * <input type="text" value="Bez dokumenta"/> |
| Ime * <input type="text"/> | Slika/sken dokumenta* <input type="text" value="Odaberi datoteku"/> Nije odabrana niš jedr |
| Prezime * <input type="text"/> | |
| Ime oca <input type="text"/> | |
| OIB * <input type="text"/> | |
| Datum rođenja * <input type="text"/> | |
| Spol <input checked="" type="radio"/> Muški <input type="radio"/> Ženski | |
| Slika * <input type="text" value="Odaberi datoteku"/> Nije odabrana niš jedr | |
| Podaci o kategoriji: | Dostava kartice: |
| Kategorija putnika <input type="text" value="OPĆA KARTA"/> | Zahtjev za izdavanje kartice predaje pravna osoba ili institucija (fakultet, škola) za svoje radnike (studente, učenike). Pravna osoba plaća račun za izradne kartice i kupljene relacijske mjesečne karte svojih radnika. Kartice se dostavljaju pravnoj osobi gdje radnici preuzimaju svoju karticu. |
| Zona <input type="checkbox"/> Zona 1 | Preuzimanje kod pravne osobe <input type="radio"/> |
| <input type="checkbox"/> Zona 2 | Putnik sam predaje zahtjev za izradom kartice. Izabire prodajno mjesto gdje želi preuzeti karticu. Prilikom preuzimanja kartice, na odabranom prodajnom mjestu, plaća račun za izradnu karticu i kupljenu relacijsku mjesečnu kartu. |
| <input type="checkbox"/> Zona 3 | Preuzimanje na prodajnom mjestu <input checked="" type="radio"/> A1Smart |
| <input type="checkbox"/> Zona 4 | |
| Prebivalište: | Kontakt podaci: |
| Mjesto <input type="text" value="OSTALI GRADOVI V."/> | Morate upisati barem jedan od sljedeća tri podatka telefon/mobitel/e-mail * |
| Ulica * <input type="text"/> | Telefon <input type="text"/> |
| Pošanski broj * <input type="text"/> | Mobitel <input type="text"/> |
| Stranac <input type="checkbox"/> | E-mail <input type="text"/> |
| | Adresa kontakta različita od adrese putnika? <input type="checkbox"/> |
| <small>Tvrtka s Vašim će osobnim podacima postupati sukladno Općoj uredbi (EU) 2016/679 o zaštiti pojedinaca u vezi s obradom osobnih podataka i slobodnom kretanju takvih podataka, nacionalnim propisima kojima se regulira zaštita osobnih podataka uz primjenu odgovarajućih tehničkih i sigurnosnih mjera zaštite osobnih podataka od neovlaštenog pristupa, zlouporabe, otkrivanja, gubitka ili uništenja. Više pogledajte na: tvrtka.com.</small> | |
| <input type="checkbox"/> Slažem se s uvjetima Tvrtka d.o.o. | |
| <input type="button" value="Pošalji zahtjev"/> | |

© 2019

Slika 40. Zahtjev za izdavanje kartica

Scenarij korištenja je sljedeći: korisnik putem internetskog preglednika pristupa određenom web mjestu na kojem se nalazi zahtjev za izradom beskontaktnih kartica, prikazan na slici 40. Korisnik mora unijeti osobne podatke: ime, prezime, ime oca, OIB, datum rođenja, spol te učitati svoju fotografiju, zatim odabire kategoriju putnika te zonu.

Nakon toga ispunjava rubriku prebivališta: mjesto, ulicu, poštanski broj. Ako je korisnik strani državljanin, označava potvrdni okvir te mu se pojavljuju dva polja, država i mjesto, kao što je prikazano na slici 41.

Podaci o kategoriji:

| | |
|--------------------|--|
| Kategorija putnika | OPĆA |
| Zona | <input type="checkbox"/> Zona 1 <input type="checkbox"/> Zona 2 <input type="checkbox"/> Zona 3 <input type="checkbox"/> Zona 4 |

Prebivalište:

| | |
|------------------|-------------------------------------|
| Mjesto | OSTALI GRADOVI V/ |
| Ulica * | <input type="text"/> |
| Poštanski broj * | <input type="text"/> |
| Stranac | <input checked="" type="checkbox"/> |
| Država | <input type="text"/> |
| Mjesto | <input type="text"/> |

Slika 41. Prikaz potvrdnog okvira i dodatnog polja za unos stranca

Nakon toga je rubrika priloženi dokument/isprava u kojoj korisnik odabire tip dokumenta i učitava sliku/sken dokumenta. U rubrici dostava kartice korisnik odabire da li će preuzeti karticu kod pravne osobe ili na prodajnom mjestu. Ako odabere prodajno mjesto, tada mora izabrati na koje ponuđeno prodajno mjesto želi da mu se dostavi kartica.

Slika 42. predstavlja posljednju rubriku koju korisnik ispunjava. U kontakt podacima korisnik ispunjava barem jedan od tri podatka: telefon, mobitel, email. Ako se kojim slučajem adresa boravišta kontakta razlikuje od adrese prebivališta, korisnik označava potvrdni okvir te mu se pojavljuju četiri polja: ulica, mjesto, poštanski broj i država. Naposljetku korisnik prihvaća putem potvrdnog okvira Zakon o zaštiti podataka odgovarajućeg prijevoznika te pomoću gumba pošalje zahtjev za izdavanje kartice.

| Kontakt podaci: | |
|--|-------------------------------------|
| Morate upisati barem jedan od slijedeća tri podatka telefon/mobitel/e-mail * | |
| Telefon | <input type="text"/> |
| Mobitel | <input type="text"/> |
| E-mail | <input type="text"/> |
| Adresa kontakta različita od adrese putnika? | <input checked="" type="checkbox"/> |

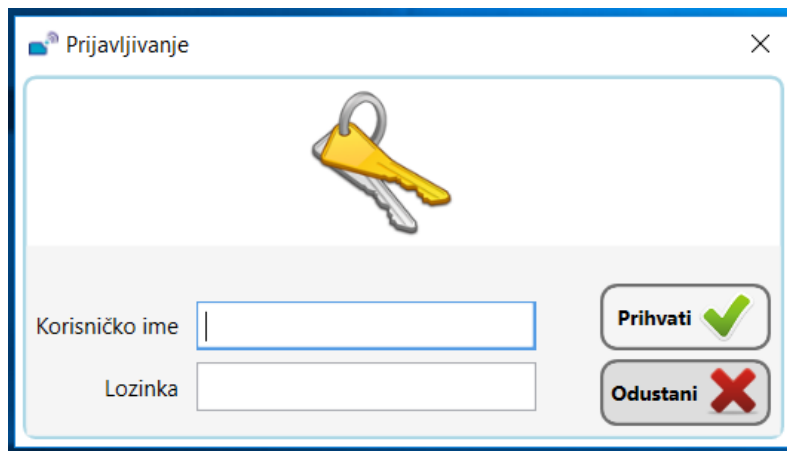
| Boravište kontakt osobe: | |
|---------------------------------|--|
| Ulica | <input type="text"/> |
| Mjesto | OSTALI GRADOVI V/ <input type="button" value="v"/> |
| Poštanski broj | <input type="text"/> |
| Država | <input type="text"/> |

Slika 42. Prikaz potvrdnog okvira i dodatnog polja za unos boravišta

7.2. Korištenje desktop aplikacije

7.2.1. Prijava u aplikaciju

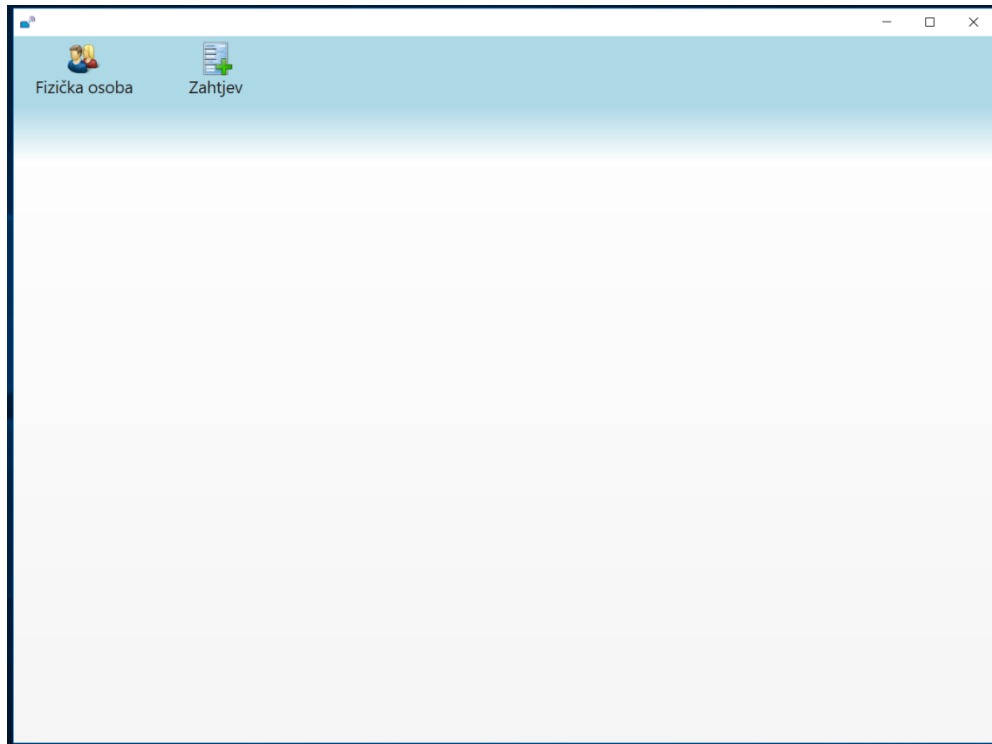
Scenarij korištenje je sljedeći: kod prijave, korisnik mora unijeti korisničko ime i lozinku unutar prozora za prijavljivanje, kako je prikazano na slici 43. Ako unese točne podatke, prijavljuje se u desktop aplikaciju. Nakon uspješne prijave u aplikaciju, otvara se početni zaslon aplikacije sa izbornikom.



Slika 43. Prijava u aplikaciju

7.2.2. Izbornik aplikacije

Na slici 44. je prikazan glavni izbornik koji sadrži stavku fizička osoba i zahtjev. Korisnik bira određenu stavku i pomoću nje mu se otvara odgovarajući prozor.



Slika 44. Glavni zaslon aplikacije

7.2.3. Prozor fizička osoba

Na slici 45. je prikazan prozor fizičke osobe gdje radnik može dodati novu fizičku osobu, izmjenjivati podatke fizičke osobe, pregledati fizičku osobu te izbrisati fizičku osobu.

The screenshot shows a web application window titled "Fizička osoba". On the left, there is a table with columns: A, Id, Prezime, Ime, and Pravna osoba. The second row is highlighted in yellow with values: 2, Test1, Test1. The main form contains the following fields:

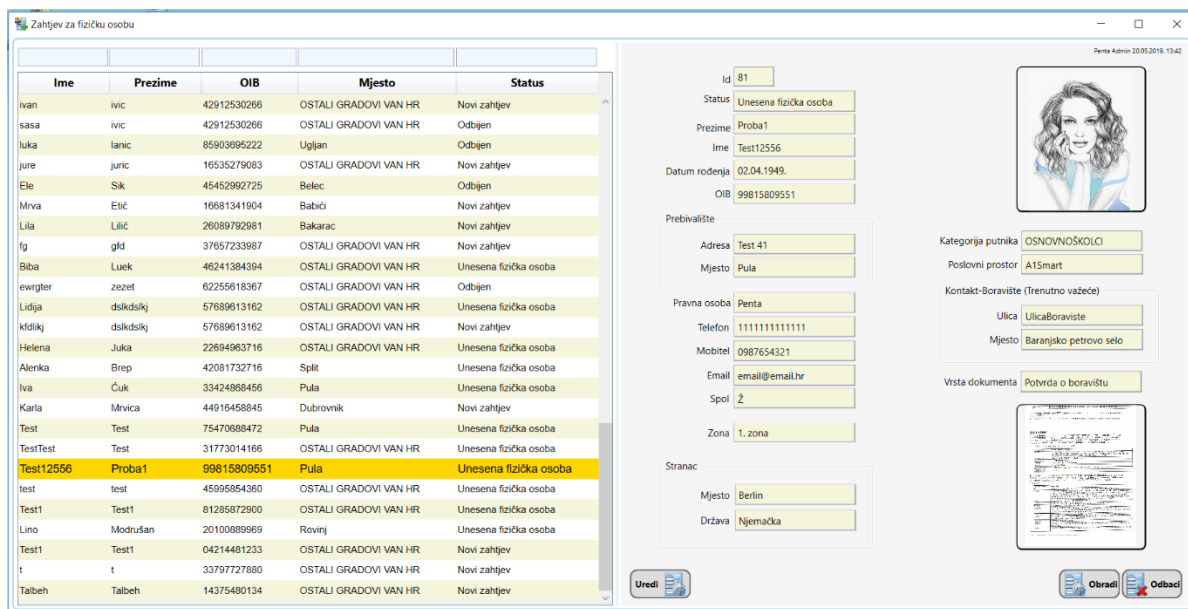
- Id:** 2
- Šifra:** (empty)
- Prezime:** Test1
- Ime:** Test1
- Datum rođenja:** 5.5.2005.
- OIB:** 81285872900
- Ime roditelja:** (empty)
- Prebivalište:** Adresa: Pulska 32; Mjesto: OSTALI GRADOVI VAN HR (Grad Zagreb)
- Pravna osoba:** (dropdown menu)
- Telefon:** 0321654987
- Mobitel:** (empty)
- Email:** (empty)
- Spol:** Muško, Žensko
- Relacija:** 1. ZONA
- Škola/Fakultet:** (dropdown menu)
- Razred:** (empty)
- Semestar:** (empty)
- Km do škole:** (empty)
- Stranac:** **Slijepac:** **Pravo na pratioca:**
- Država:** (empty)

On the right side, there is a photo upload area with a placeholder image of a woman. Below it, there are sections for "Kategorija putnika - trenutno važeća" (Vrsta: OPĆA KARTA, Vrijedi od: 9.6.2019., Vrijedi do: 9.7.2019.), "Boravište - trenutno važeće" (Mjesto: (empty), Vrijedi od: Odaberi datum, Vrijedi do: Odaberi datum), and "Dokument/Napomena" (9.6.2019. 13:11:27 - Penza Admin - DOKUMENT: Fizička osoba unesena od strane web zahtjeva). At the bottom right, there are buttons: "Poništi FO", "Osvježi", "Dodaj", "Obriši", and "Snimi".

Slika 45. Prozor fizičke osobe

7.2.4. Prozor zahtjev za fizičku osobu

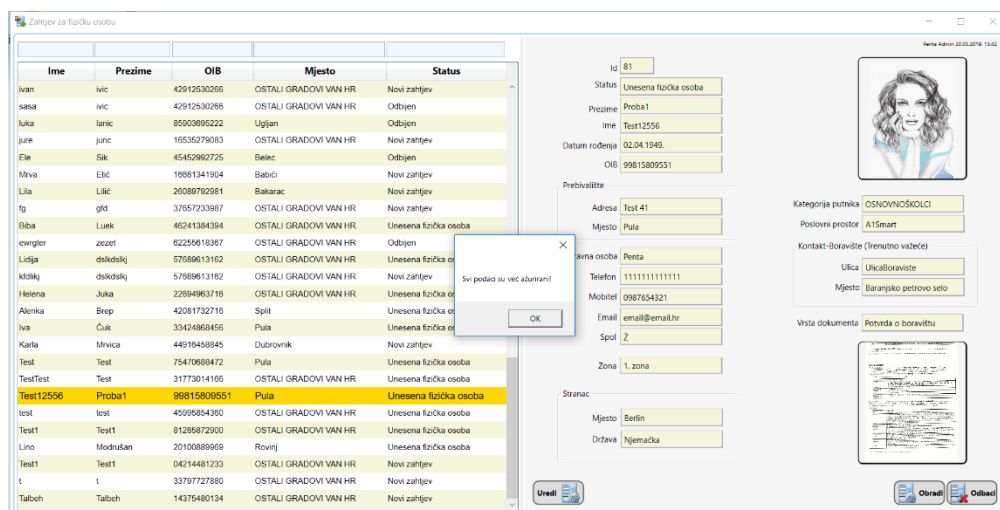
U prozoru zahtjev koji je prikazan na slici 46., korisnik može pregledati zahtjev koji je došao putem mreže te ima status 'Novi zahtjev'. Pomoću gumba 'Obradi' može promijeniti status u 'Unesena fizička osoba', pomoću gumba 'Odbij' može promijeniti status u 'Odbijen' te pomoću gumba 'Uredi' može izmijeniti podatke fizičke osobe koja je poslala zahtjev preko internetskog preglednika, a već postoji u sustavu, odnosno već je unesena fizička osoba. Ako fizička osoba ima status 'Unesena fizička osoba', a korisnik klikne na gumb 'Uredi' pojavljuje mu se obavijest da su podaci ažurirani, to znači da se podaci u fizičkoj osobi ne razlikuju od novog zahtjeva.



Slika 46. Prozor zahtjev za fizičku osobu

7.2.5. Prozor ažuriranje podataka fizičke osobe pomoću predanog zahtjeva preko mreže

Kad korisnik označi osobu koja ima status 'Unesena fizička osoba' i klikne na gumb 'Uredi', otvara mu se novi prozor, prikazan na slici 48., u kojem su na lijevoj strani prozora podaci iz fizičke osobe, a s desne strane podaci koji su došli putem web zahtjeva. Ako su kojim slučajem podaci isti u fizičkoj osobi i predani putem zahtjeva, klikom na gumb 'Uredi' prikazuje se prozor sa porukom koji je prikazan na slici 47., na kojem piše 'Svi podaci su već ažurirani'.



Slika 47. Prozor s porukom o ažuriranim podacima

Pokraj podataka koji su došli putem web zahtjeva se nalaze potvrdni okviri pomoću kojih korisnik označava one podatke koje želi ažurirati u fizičkoj osobi, odnosno one podatke koji se razlikuju. Na dnu tog prozora se nalazi gumb 'spremi' pomoću kojeg se označeni podaci ažuriraju u fizičkoj osobi te se prozor nakon pritiska na gumb 'Spremi' zatvara.

Ažuriranje podataka

Fizička osoba

Ime: Marica
Prezime: Lenić
Datum rođenja: 16.6.1950.
OIB: 99815809551
Adresa: Test 53
Mjesto: Pula
Telefon:
Mobitel: 0987654321
Email:
Spol: Ž
Zona: 1. ZONA

Stranac
Država:

Kategorija putnika: OSNOVNOŠKOLCI
Vrsta dokumenta:

Kontakt-Boravište (Trenutno važeće)
Ulica: UlicaBoraviste
Mjesto: Baranjsko petrovo selo

Zahtjev za fizičku osobu sa weba

Ime: Test12556
Prezime: Proba1
Datum rođenja: 2.4.1949.
OIB: 99815809551
Adresa: Test 41
Mjesto: Pula
Pravna osoba: Penta
Telefon: 1111111111111111
Mobitel: 0987654321
Email: email@email.hr
Spol: Ž
Zona: 1. zona

Stranac
Mjesto: Berlin
Država: Njemačka

Kategorija putnika: OSNOVNOŠKOLCI
Poslovni prostor: A1Smart
Vrsta dokumenta: Potvrda o boravištu

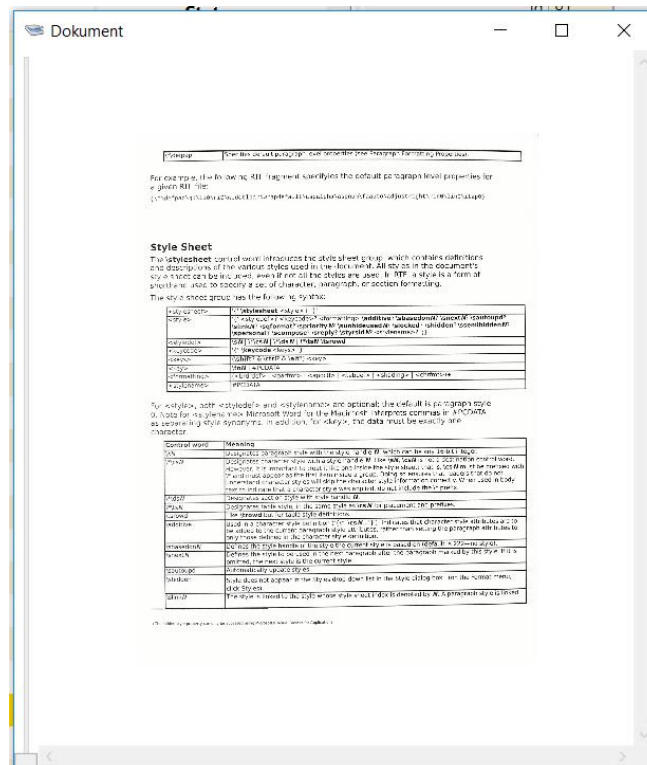
Kontakt-Boravište (Trenutno važeće)
Ulica: UlicaBoraviste
Mjesto: Baranjsko petrovo selo

Snimi

Slika 48. Prozor za ažuriranje podataka

7.2.6. Dodatne funkcionalnosti

U prozoru zahtjev, dvoklikom na fizičku osobu koja je predala zahtjev putem internetskog preglednika, otvara se prozor fizičke osobe te je točno ta fizička osoba označena i prikazana na zaslonu. Također u prozoru zahtjev, klikom na sken/sliku dokumenta se otvara odgovarajući prozor koji je prikazan na slici 49., koji ima mogućnost povećavanja skena/slike dokumenta.



Slika 49. Prozor za detaljniji prikaz dokumenta

ZAKLJUČAK

Cilj diplomskog rada bio je razviti sustav za predaju i obradu zahtjeva za izdavanje beskontaktnih personaliziranih kartica. Sustav se razvijao u tvrtki Penta d.o.o. iz Pule unutar njihovog projekta BusCARD, kao nova dodatna funkcionalnost. Web aplikacija služi za predaju zahtjeva za izdavanje beskontaktnih personaliziranih kartica i razvijala se u programskom jeziku C#, HTML, CSS, JavaScript. Funkcionalnosti web aplikacije su provjera OIB-a, odabir jedne ili više zona, odabir dodatnog polja ako je korisnik stranac, odabir datoteke s računala (fotografija i sken dokumenta), jednostavan odabir kategorije putnika, mjesta prebivališta i boravišta, dostave kartice.

Web usluge za razliku od web aplikacije pružaju samo informacije i nisu zainteresirane za to kako izgleda korisničko sučelje. Najpoznatiji tip web API-ja na poslužiteljskoj strani je REST API.

Desktop aplikacija služi za obradu zahtjeva za izdavanje beskontaktnih personaliziranih kartica i razvijala se u programskom jeziku C#. Neke od funkcionalnosti desktop aplikacije su brz unos osobe iz web zahtjeva u fizičku osobu, dvoklik za ulaz u ekran fizičke osobe, promjenjiv status zahtjeva (novi zahtjev, odbijen zahtjev, unesena fizička osoba), ažuriranje unesene fizičke osobe, dvoklik za detaljni prikaz skena dokumenta.

Aplikacije su realizirane u programskom okruženju .NET-a, u programu Microsoft Visual Studio uz korištenje tehnologija ASP.NET MVC, MVVM, Entity Framework-a. Korištenjem navedenih tehnologija i pristupa programski je ostvarena web i desktop aplikacija i baza podataka za odgovarajuću primjenu.

Sigurno postoje slične aplikacije na tržištu ali one nisu primjenjive za domaće tržište te u tome upravo leži inovativnost ovog dijela aplikacije. Kako bi opstali na tržištu potrebno je biti drugačiji od ostalih, a za to je potreban odličan timski rad i dobro planirani koraci razvoja projekta. Timski rad je ključan dio rada na velikim projektima gdje su aplikacije uvijek pouzdanije, više puta testirane i daleko kvalitetnije. Za daljnji razvoj je planirano ostvariti zadnja dva statusa zahtjeva, izrađena kartica i preuzeta kartica, razviti predaju i obradu zahtjeva za pravne osobe i poboljšati vizualni prikaz desktop aplikacije.

LITERATURA

- [1] Brdar, L. (2016) REST API kao poslužiteljski dio jednostranične web aplikacije. Završni rad. Rijeka: Sveučilište u Rijeci, Tehnički fakultet
- [2] C-sharpcorner, MVVM in WPF (2019). URL: <https://www.c-sharpcorner.com/> (2019-04-15)
- [3] Domagojpa blog. URL: <http://domagoj.eu/oib/> (2019-01-10)
- [4] Fajković, S. (2014) Programski jezik C# & Windows Forms. URL: <http://carpediem.hr/PublikacijeCarpeDiem/Publikacije/C%23%20programiranje.pdf> (2019-04-20)
- [5] Jennings, R. (2009) Professional ADO.NET 3.5 with LINQ and the Entity Framework. Indiana: Wiley Publishing, Inc.
- [6] Knok, Ž.; Marčec, M. (2015) WPF AND MVVM. Tehnički glasnik 9. str. 452. URL: <https://hrcak.srce.hr/file/220481>
- [7] Larry, T.; Zeeshan, H. (2010) Entity Framework 4.0 Recipes. United States of America: Apress
- [8] Mueller, J. P. (2013) Microsoft ADO.NET Entity Framework: Step by Step. United States of America : O'Reilly Media
- [9] Medium (2018). URL: <https://medium.com/@gabrymartinez/how-to-create-your-own-little-restful-web-api-and-do-not-get-lost-in-the-process-part-1-cf8db6833ae4> (2019-05-22)
- [10] Microsoft, ASP.NET MVC (2009). URL: <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview> (2019-04-15)
- [11] Microsoft (2009). URL: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview> (2019-04-15)
- [12] Microsoft (2018). URL: <https://docs.microsoft.com/hr-hr/visualstudio/designers/getting-started-with-wpf?view=vs-2019> (2019-04-17)

- [13] Microsoft (2017). URL: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm> (2019-04-25)
- [14] .NET Framework (2018). URL: https://hr.wikipedia.org/wiki/.NET_Framework (2019-04-15)
- [15] Objektno orijentirano programiranje: Auditorne vježbe. URL: http://www.etfos.unios.hr/~lukic/oop/Auditorne_vje%C5%BEbe_5.pdf (2019-04-15)
- [16] Penta d.o.o. URL: <https://www.penta.hr/hr/> (2019-04-10)
- [17] Solis, D. (2012) Illustrated C# 2012. New York: Apress Media LLC
- [18] Studytonight. URL: <https://www.studytonight.com/rest-web-service/types-of-webservices> (2019-05-22)
- [19] Škola koda. URL: <https://skolakoda.org/web-servisi> (2019-05-22)
- [20] Zekić-Sučac, M.; Mitrović S. (2013) Rad s bazom podataka u .NET okruženju. Seminarski rad. Osijek: Ekonomski fakultet u Osijeku

POPIS SLIKA

| | |
|--|----|
| Slika 1. Odnos između modela, pogleda i upravljača | 11 |
| Slika 2. MVVM arhitektura | 13 |
| Slika 3. Dijagram obrasca upotrebe | 14 |
| Slika 4. Klasni dijagram | 15 |
| Slika 5. Prototip web aplikacije | 16 |
| Slika 6. Prototip web aplikacije | 17 |
| Slika 7. Prototip desktop aplikacije | 18 |
| Slika 8. Prototip desktop aplikacije | 19 |
| Slika 9. Dijagram entiteta i veza glavnih tablica | 31 |
| Slika 10. Kompletan dijagram entiteta i veza | 39 |
| Slika 11. Mapiranje u Entity Framework-u – Zahtjev i DokumentTip | 42 |
| Slika 12. Kontekst klasa | 43 |
| Slika 13. Konekcija u App.Config datoteci | 43 |
| Slika 14. SOAP omotnica | 45 |
| Slika 15. RESTful WEB API | 46 |
| Slika 16. Dodavanje metode POST | 47 |
| Slika 17. Kreiranje klase ZonaDTO | 48 |
| Slika 18. Klasa GetZonesOut.cs | 49 |
| Slika 19. Dodavanje Zone u model | 49 |
| Slika 20. Klasa ZonaRepo.cs | 50 |
| Slika 21. Klasa ZonaDTO.cs | 50 |
| Slika 22. Klasa GetZoneOut.cs | 51 |
| Slika 23. Dodavanje u klasu Zahtjev atribut ZONA_Id | 51 |

| | |
|--|----|
| Slika 24. Instanciranje zone | 52 |
| Slika 25. Komponente desktop aplikacije | 53 |
| Slika 26. Klasa Zahtjev - funkcija getZahtjev | 54 |
| Slika 27. Datoteke pogleda | 55 |
| Slika 28. Funkcija za provjeru OIB-a | 56 |
| Slika 29. Funkcija za neispravan OIB | 56 |
| Slika 30. Funkcija za označavanje zona | 57 |
| Slika 31. Funkcije za stranca i kontakt podatke | 57 |
| Slika 32. Funkcija za pravnu osobu | 57 |
| Slika 33. Komponente desktop aplikacije | 59 |
| Slika 34. MVVM model Zahtjeva | 59 |
| Slika 35. Metoda BeginTransaction() | 60 |
| Slika 36. Stanje entiteta dodano i metoda spremanja | 60 |
| Slika 37. Stanje entiteta modificirano i metoda spremanja | 61 |
| Slika 38. Metode Commit() i Rollback() | 61 |
| Slika 39. Dio funkcije za ažuriranje statusa zahtjeva | 61 |
| Slika 40. Zahtjev za izdavanje kartica | 62 |
| Slika 41. Prikaz potvrdnog okvira i dodatnog polja za unos stranca | 63 |
| Slika 42. Prikaz potvrdnog okvira i dodatnog polja za unos boravišta | 64 |
| Slika 43. Prijava u aplikaciju | 65 |
| Slika 44. Glavni zaslona aplikacije | 66 |
| Slika 45. Prozor fizičke osobe | 67 |
| Slika 46. Prozor zahtjev za fizičku osobu | 68 |
| Slika 47. Prozor s porukom o ažuriranim podacima | 68 |

| | |
|---|----|
| Slika 48. Prozor za ažuriranje podataka | 69 |
| Slika 49. Prozor za detaljniji prikaz dokumenta | 70 |

POPIS TABLICA

| | |
|--|----|
| Tablica 1. Popis atributa koji se nalaze u tablici dbo.Zahtjev | 22 |
| Tablica 2. Popis atributa koji se nalaze u tablici dbo.FizickaOsoba | 24 |
| Tablica 3. Popis atributa koji se nalaze u tablici dbo.ZahtjevStatus | 26 |
| Tablica 4. Popis atributa koji se nalaze u tablici dbo.Zona | 26 |
| Tablica 5. Popis atributa koji se nalaze u tablici dbo.Mjesto | 27 |
| Tablica 6. Popis atributa koji se nalaze u tablici dbo.DokumentTip | 28 |
| Tablica 7. Popis atributa koji se nalaze u tablici dbo.KategorijaPutnika | 29 |
| Tablica 8. Popis atributa koji se nalaze u tablici dbo.PoslovniProstor | 30 |
| Tablica 9. Popis atributa koji se nalaze u tablici dbo.Drzava | 32 |
| Tablica 10. Popis atributa koji se nalaze u tablici dbo.Kartica | 33 |
| Tablica 11. Popis atributa koji se nalaze u tablici dbo.Oblast | 34 |
| Tablica 12. Popis atributa koji se nalaze u tablici dbo.Opcina | 35 |
| Tablica 13. Popis atributa koji se nalaze u tablici dbo.Parametar | 36 |
| Tablica 14. Popis atributa koji se nalaze u tablici dbo.Radnik | 38 |
| Tablica 15. Prikaz osnovnih razlika između dvije vrste web usluga | 46 |

SAŽETAK

Cilj ovog rada je pobliže objasniti razvoj sustav za predaju i obradu zahtjeva za izdavanje beskontaktnih kartica. Rad se sastoji od 6 glavnih cjelina. Najprije opisuje korištene tehnologije. Druga cjelina prikazuje razradu funkcionalnosti. Sljedeća cjelina se sastoji od relacijske baze podataka koja je izrađena u Microsoft SQL Server Management Studio-u. Četvrta cjelina se bavi Web uslugama koje koristi .NET okruženje. Peta cjelina se sastoji od implementacije Web aplikacije koja koristi ASP.NET MVC programski okvir koji implementira model – pogled – upravljač (eng. Model – View – Controller) oblikovni obrazac. Šesta cjelina se sastoji od implementacije desktop aplikacije koja koristi ASP.NET MVVM programski okvir koji implementira model – pogled – model pogleda (eng. Model – View – ViewModel). Posljednja cjelina opisuje korištenje web i desktop aplikacije za predaju i obradu zahtjeva.

Ključne riječi: MVC, MVVM, ASP.NET, RELACIJSKA BAZA PODATAKA, ENTITY FRAMEWORK, #C, MICROSOFT VISUAL STUDIO

ABSTRACT

The basic idea of this paper is to describe the development of a system for submitting and processing request for issuing contactless cards. The paper consists of six main parts. First part describes the technology used. The second part shows the elaboration of functionality. Next part consists of a relational database that was created in Microsoft SQL Server Management Studio. The fourth part deals with Web services using the .NET environment. The fifth part describes implementation of a Web application uses the ASP.NET MVC programming framework that implements the model – view – controller design pattern. Next part describes implementation of a desktop application that uses the ASP.NET MVVM programming framework that implements the model – view – viewmodel. The last part describes the use of web and desktop applications for submission and processing request.

Keywords: MVC, MVVM, ASP.NET, RELIABLE DATEBASE, ENTITY FRAMEWORK, C#, MICROSOFT VISUAL STUDIO