

Razvoj sustava za kupovinu radničkih mjesečnih karata za pravne osobe

Babić, Kristijan

Master's thesis / Diplomski rad

2020

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

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:137:694832>

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

Download date / Datum preuzimanja: **2024-04-19**



Repository / Repozitorij:

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



Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

KRISTIJAN BABIĆ

RAZVOJ SUSTAVA ZA KUPOVINU RADNIČKIH MJESEČNIH KARATA ZA
PRAVNE OSOBE

Diplomski rad

Pula, siječanj, 2020 godine

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

KRISTIJAN BABIĆ

RAZVOJ SUSTAVA ZA KUPOVINU RADNIČKIH MJESEČNIH KARATA ZA
PRAVNE OSOBE

Diplomski rad

JMBAG: 0303046229, redoviti student

Studijski smjer: Informatika

Kolegij: Napredni algoritmi i strukture podataka

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijsko komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i Informatologija

Mentor: Izv. prof. dr. sc. Tihomir Orehovački

Pula, siječanj, 2020 godine



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisan Kristijan Babić, 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.

Student

Kristijan Babić

U Puli, siječanj, 2020. godine



IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, Kristijan Babić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Razvoj sustava za kupovinu radničkih mjesečnih karata za pravne osobe“ 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.

Student

Kristijan Babić

U Puli, siječanj, 2020. godine

Zahvala

Zahvaljujem se svom mentoru Izv. prof. dr. sc. Tihomiru Orehovačkom na ukazanom povjerenju i pruženoj pomoći tijekom izrade diplomskog rada.

Zahvaljujem se tvrtci Penta d.o.o. na pruženoj pomoći i savjetima tijekom izrade diplomskog rada.

Od srca se zahvaljujem svojoj obitelji i bližnjima na potpori tijekom akademskog studija.

Sadržaj

UVOD	1
1. KORIŠTENE TEHNOLOGIJE	2
1.1 C sharp (C#).....	2
1.2 .Net Framework.....	2
1.3 .ASP.NET Core	3
1.4 WPF	4
1.5 MVVM	5
2. Baza podataka	6
2.1 Opis tablica	7
2.1.1 Tablica dbo.NarudzbenicaKarata	7
2.1.2 Tablica dbo.NarudzbenicaKarataStavka	9
2.1.3 Tablica dbo.FizickaOsoba	11
2.1.4 Tablica dbo.PravnaOsoba.....	13
2.1.5 Tablica dbo.Kartica	14
2.1.6 Tablica dbo.KarticaProdaja	16
2.1.7 Tablica dbo.Artikal.....	19
2.1.8 Tablica dbo.Relacija.....	20
2.1.9 Tablica dbo.Radnik	21
2.1.10 Tablica dbo.PravnaOsobaLogin	23
2.1.11 Tablica dbo.KategorijaPutnika.....	25
2.1.12 Tablica dbo.Mjesto	26
2.2 Dijagram baze podataka	27
3. Web aplikacija	28
3.1 Prikaz datotečnog i programskog dijela web aplikacije	28
3.2 Prikaz grafičkog dijela web aplikacije	39
4. Desktop aplikacija.....	42
4.1 Prikaz datotečnog i programskog dijela desktop aplikacije	43
4.2 Prikaz grafičkog dijela desktop aplikacije	45
ZAKLJUČAK.....	50
LITERATURA	51
Popis slika	53
Popis tablica	54
Sažetak.....	55

Abstract	55
----------------	----

UVOD

Tema ovog diplomskog rada je razvoj sustava za kupovinu radničkih mjesečnih karata za pravne osobe. Ovaj rad izrađen je u tvrtki „Penta d.o.o.“ koja je dala suglasnost da se kopira baza podataka i dio programskog koda u cilju formiranja logičke cjeline sustava. Također tvrtka „Penta d.o.o.“ dala je suglasnost za javno objavljivanje programskog koda, dok su produkcijski podaci tajni i ne smiju biti javno objavljeni. Ovaj rad je jedan dio cjelokupnog sustava pod nazivom „BusCARD“. „BusCARD“ je višefunkcijski integrirani informatički sustav čiji je osnovni cilj automatiziranje naplate, povećanje efikasnosti upravljanja sustavom javnog gradskog i prigradskoga prijevoza, nadzor nad troškovima i povećanje kvalitete usluge korisnicima javnog prijevoza.¹

Cilj ovog sustava je olakšati korisnicima kupovinu radničkih mjesečnih karata te smanjiti redove čekanja na šalterima jer se korištenjem aplikacije omogućuje kupovina većeg broja karata istovremeno.

Tehnologije koje su korištene tijekom razvijanja ovog sustava su ASP.NET Core 2.2 Razor Pages, MVVM, WPF, Entity Framework, Microsoft SQL Server.

U prvom poglavlju ovog rada bit će opisane tehnologije koje su korištene tijekom razvijanja ovog sustava. Zatim u drugom poglavlju opisana je baza podataka i tablice u bazi podataka. U trećem poglavlju opisana je web aplikacija, prikazani su isječci programskog koda, te korisnička sučelja aplikacije. Dok u posljednjem, četvrtom poglavlju, opisana je desktop aplikacija, te su također prikazani isječci koda i korisnička sučelja aplikacije.

1 „Automatska naplata“. BusCARD. <https://www.buscard.biz/hr/sustavi/automatska-naplata/>

1. KORIŠTENE TEHNOLOGIJE

1.1 C sharp (C#)

C# je moderni objektno orijentirani programski jezik opće namjene kojeg je kreirala i razvila multinacionalna kompanija Microsoft, te kojega je odobrila Europska udruga proizvođača računala (ECMA) i Međunarodna organizacija za standarde (ISO). Ovaj programski jezik radi na .Net (dot net) radnom okruženju. Pomoću programskog jezika C# mogu se razviti različite vrste aplikacija poput programskih aplikacija, web aplikacija, distribuiranih aplikacija, aplikacija za web usluge, aplikacija za baze podataka i dr. C# razvijen je od strane Anders Hejlsberg-a i njegovog tima tijekom razvijanja .Net okruženja. Neki od razloga zbog kojih je C# postao široko korišteni programski jezik su sljedeći: ²

- C# je moderan programski jezik opće namjene
- Objektno je orijentiran
- Komponentno je orijentiran
- Lagan je za naučiti
- Pomoću njega moguće je razviti učinkovite programe
- Moguće ga je pokrenuti na različitim računalnim platformama
- Dio je .Net okruženja

1.2 .Net Framework

.Net Framework je softverski okvir koji služi kao pomoć za razvijanje Windows aplikacija, web aplikacija i web usluga (web service). Ovaj softverski okvir dizajniran je da se može koristiti i s drugim programskim jezicima poput C++, Visual Basic, Jscript, COBOL i dr. Svi ovi jezici mogu pristupiti softverskom okviru kao i komunicirati jedni s drugima. Prva verzija .Net softverskog okvira bila je 1.0 koja se je počela koristiti 2002.

² Tutorials point (2014), C# Programming object-oriented programming str.1

godine. .Net Framework podržava više od 60 programskih jezika. Glavne komponente .Net softverskog okvira su Common Language Runtime (CLR) i Framework Class Library (FCL). CLR je osnovna i virtualna komponenta .Net okvira koja je u osnovi odgovorna za upravljanje izvršavanjem .Net programa bez obzira na bilo koji .Net programski jezik. Prednosti CLR-a su te da: poboljšava performanse, podržava automatsko upravljanje memorijom uz pomoć Garbage Collector-a, omogućava neovisnost o jeziku, platformi i arhitekturi, omogućava stvaranje skalabilnih i višeslojnih aplikacija na lakši način jer programer nema potrebu razmišljati o upravljanju memorijom i pitanjima sigurnosti i dr.³ Framework Class Library pruža sistemsku funkcionalnost budući da ima: različite klase, tipove podataka, sučelja i dr. FLC integrirana je s CLR-om i koriste ga svi .Net jezici.

1.3 .ASP.NET Core

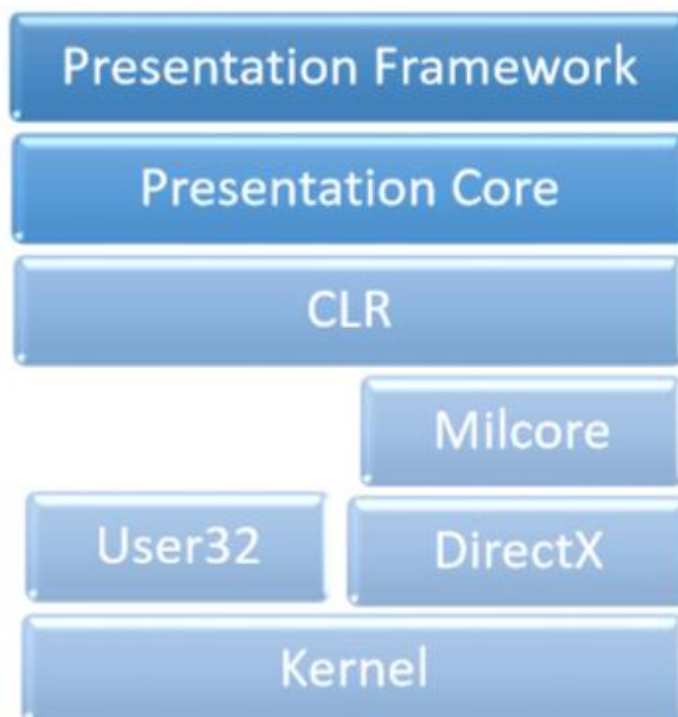
ASP.NET Core je cross-platforma, odličnih performansi, open-source softverski okvir za razvijanje modernih aplikacija utemeljenih na oblaku i internetu. Pomoću ASP.NET Core moguće je razvijati web aplikacije, IoT aplikacije i mobilne aplikacije. Neke od prednosti ASP.NET Core-a je ta što je prilagođen za testiranja, objedinjen je za razvijanje web sučelja i web API-a, umjesto korištenja klasičnog pristupa MVC-a (Model-View-Controller) s Razor Pages kodiranje je lakše i produktivnije zbog toga što je programski kod usmjeren na stranice.⁴

3 „Common Language Runtime (CLR) in C#“. GeeksforGeeks. <https://www.geeksforgeeks.org/common-language-runtime-clr-in-c-sharp/>

4 „Introduction to ASP.NET Core“. Microsoft. <https://docs.microsoft.com/en-us/aspnet/core/index?view=aspnetcore-3.0#recommended-learning-path>

1.4 WPF

WPF je skraćenica od Windows Presentation Foundation. On je moćan okvir za izradu Windows aplikacija. Ovaj okvir predstavljen je u .NET Framework-u 3.0 te se je u kasnijim verzijama poboljšavao. Glavne komponente WPF arhitekture prikazane su na slici 1.



Slika 1. Arhitektura Windows Presentation Foundation-a

Izvor: WPF – Overview, https://www.tutorialspoint.com/wpf/wpf_overview.htm

Presentation Framework, Presentation Core i Milcore glavni su dijelovi WPF-a. Od ove tri komponente Milcore je jedina neupravljiva komponenta. Milcore je napisan neupravljanim kodom kako bi se omogućila integracija s DirectX-om.⁵ Cjelokupni prikaz u WPF-u vrši se putem DirectX-a koji omogućava učinkovito prikazivanje. CLR kod WPF-a čini razvojni proces produktivnijim na način da nudi mnoge značajke poput:

⁵ „WPF Architecture“. Microsoft. <https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/wpf-architecture>

upravljanja memorijom, rukovanjem pogreškama i dr. U ovom okviru elementi korisničkog sučelja dizajnirani su u Extensible Application Markup Language-u odnosno XAML-u. Nakon dizajniranja korisničkog sučelja ponašanje istog moguće je implementirati pomoću programskog jezika C# ili VB.Net. Zbog ove karakteristike vrlo je jednostavno odvojiti korisničko sučelje i samo ponašanje istog. Jedna od glavnih prednosti XAML-a je ta što programeri mogu paralelno raditi s dizajnerima korisničkog sučelja.⁶

1.5 MVVM

Model View ViewModel je obrazac koji omogućuje odvajanje prezentacijskog dijela aplikacije od same aplikacije.⁷ Ova tehnologija je osmišljena da ide zajedno sa WPF tehnologijom. Kao što glasi naziv ove tehnologije, ona je podijeljena na sljedeća tri dijela:⁸

1. Model predstavlja stvarne podatke i/ili podatke s kojima se bavimo te predstavlja poslovnu logiku. Model je zbirka objekata sa svojstvima. Svaki od objekata može se odnositi na drugi objekt te se taj objekt naziva objektnim modelom.
2. View sadrži prilagođene podatke odnosno View je korisničko sučelje. Ono prikazuje podatke prema korisniku i modificira ih u skladu sa sadnjama korisnika aplikacije.
3. View Model predstavlja glavni dio obrasca MVVM-a. Ovaj dio omogućuje korisniku da vrši interakciju s aplikacijskim podacima i modificira ih prema izvršenim radnjama.⁹

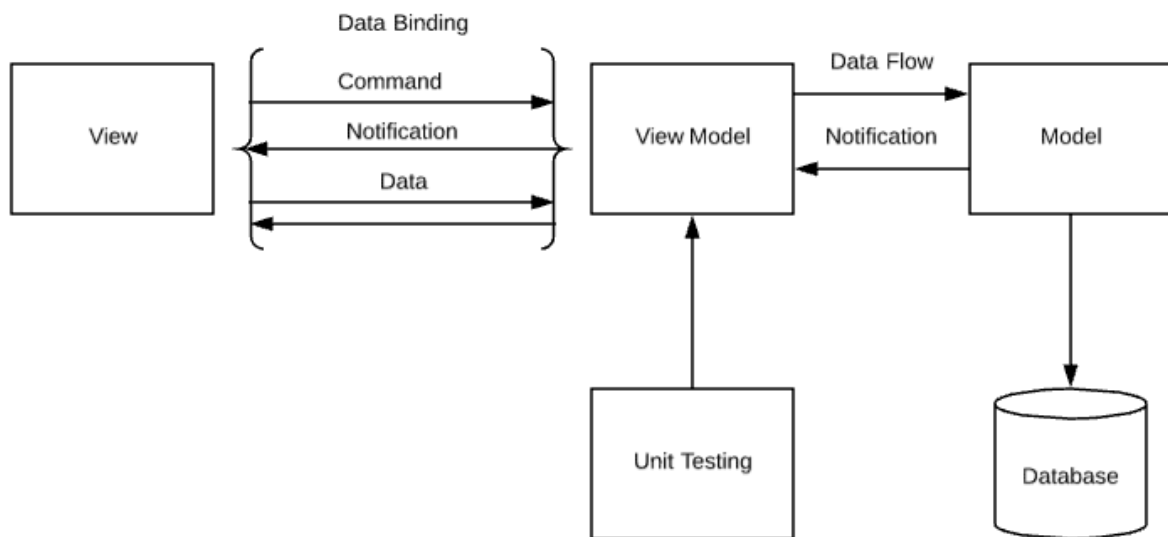
6 „WPF-XAML Overview“. Tutorialspoint. https://www.tutorialspoint.com/wpf/wpf_xaml_overview.htm

7 Knok, Ž i Marčec, M „WPF i MVVM“. Stručni članak. Čakovec: Međimursko veleučilište, 2015. str. 452.

8 Likness, J „Model-View-ViewModel (MVVM) Explained“, 2014. <https://www.wintellect.com/model-view-viewmodel-mvvm-explained/>

9 „MVVM Design Pattern for IOS and Android“. 2019. Codewave insights <https://insights.codewave.com/mvvm-design-pattern-for-ios-and-android/>

Sljedeća slika prikazuje arhitekturu MVVM obrasca.



Slika 2. Arhitektura MVVM

Izvor: MVVM Design Pattern for iOS and Android, <https://insights.codewave.com/mvvm-design-pattern-for-ios-and-android/>

2. Baza podataka

Za izradu tablica korišten je Microsoft SQL Server. To je sustav za upravljanje relacijskim bazama podataka odnosno RDBMS. Ovaj sustav izgrađen je na vrhu SQL-a, standardiziranog programskog jezika kojeg koriste administratori baza podataka i drugi IT profesionalci za upravljanje istih, te za izradu samih upita nad podacima. SQL server povezan je s Transact-SQL-om (T-SQL) koji je proširenje standardnog SQL-a.¹⁰

Unutar Microsoft SQL-a definirano je dvanaest tablica koje su potrebne za razvoj aplikacije. Primarne tablice su:

- Tablica `dbo.NarudzbenicaKarata`: sadrži podatke o narudžbenicama koje su izradile pravne osobe putem web aplikacije

¹⁰ Rouse, M „Microsoft SQL Server“, 2017. <https://searchsqlserver.techtarget.com/definition/SQL-Server>

- Tablica dbo.NarudzbenicaKarataStavka: sadrži podatke o stavkama same narudžbenice
- Tablica dbo.FizickaOsoba: sadrži podatke o fizičkoj osobi
- Tablica dbo.PravnaOsoba: sadrži podatke o pravnoj osobi
- Tablica dbo.Kartica: sadrži podatke o kartici koju posjeduje fizička osoba
- Tablica dbo.KarticaProdaja: sadrži podatke o prodaji artikala i samoj nadopuni prava na kartici
- Tablica dbo.Artikal: sadrži podatke o artiklima
- Tablica dbo.Relacija: sadrži podatke o relaciji
- Tablica dbo.Radnik: sadrži podatke o radnicima
- Tablica dbo.PravnaOsobaLogin: sadrži podatke koji su potrebni kako bi pravna osoba mogla pristupiti korištenju web aplikacije
- Tablica dbo.KategorijaPutnika: sadrži podatke o kategoriji putnika
- Tablica dbo.Mjesto: sadrži podatke o mjestima

2.1 Opis tablica

U ovom poglavlju opisane su tablice koje se koriste u „Sustavu za kupovinu radničkih mjesečnih karata za pravne osobe“. Dok u svakom potpoglavlju opisana je pojedina tablica odnosno atributi koji se koriste u ovom sustavu, te u tablicama prikazani su svi atributi neovisno da li se koriste u ovom sustavu ili ne. Atributi koji se ne koriste u ovom sustavu koriste se u drugim segmentima „BusCARD“ aplikacije.

2.1.1 Tablica dbo.NarudzbenicaKarata

Tablica dbo.NarudzbenicaKarata sastoji se od 10 atributa. Prvi atribut je „Id“ koji je identifikacijski broj odnosno primarni ključ tablice te glavno njegovo svojstvo je to što ne može doći do dupliranja zapisa unutar tablice. Atribut „PRAVNAOSOBA_Id“ je tipa integer odnosno cjelobrojni broj. Sljedeći atribut je MMGG te on sadrži zapis znakovnog niza duljine od četiri znaka koji označava mjesec i godinu za koji vrijedi

narudžbenica. MM predstavlja mjesec npr. 01 predstavlja Siječanj, dok GG predstavlja posljednja dva broja godine npr. 19 predstavlja 2019. godinu. „Rbr“ je 16bitni cjelobrojni broj, atribut „Status“ tipa je bit koji označava status narudžbenice, ukoliko ima vrijednost 0 onda narudžbenica ima status Naručeno što znači da je pravna osoba napravila zahtjev za nadopunama, ukoliko ima vrijednost 1 onda narudžbenica ima status Obrađeno što znači da je narudžbenica obrađena i poslana na fakturiranje i te da je poslano na bijelu listu. Atribut „Aktivan“ također je tipa bit te može imati vrijednost 0 ili 1, vrijednost 1 ima kada se narudžbenica kreira, dok vrijednost 0 ima kada se narudžbenica deaktivira odnosno ukoliko se prilikom obrade ista odbije te to rezultira slanjem kartica na crnu listu. Atributi „LASTUSER_Id“, „LastAction“ i „LastUpdate“ nalaze se u svim tablicama unutar baze podataka. „LASTUSER_Id“ tipa je cjelobrojnog broja te označava „Id“ radnika odnosno korisnika aplikacije, LastAction tipa je nchar te unutar aplikacije ima ograničenje te može imati samo vrijednosti „I“-uneseno (Insert), „D“-obrisano (Delete), i „U“-ažurirano (Update). Sljedeći atribut „LastUpdate“ je datumskog tipa te označava vrijeme posljednje izmjene podatka.

Tablica 1. Popis atributa u tablici dbo.NarudzbenicaKarata

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
PRAVNAOSOBA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.PravnaOsoba
DatumNarudzbenice	Date	Obavezno polje	Datum kreiranja narudžbenice
MMGG	Nchar	Obavezno polje, duljine od 4 znaka	Mjesec i godina za koju vrijedi narudžbenica
Rbr	Smallint	Obavezno polje	Redni broj narudžbenice za MMGG
Status	Bit	Obavezno polje	Status narudžbenice
Aktivan	Bit	Obavezno polje	Aktivnost narudžbenice

LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D","I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.2 Tablica dbo.NarudzbenicaKarataStavka

Tablica dbo.NarudzbenicaKarataStavka sastoji se od 15 atributa. Prvi atribut „Id“ identifikacijski je broj odnosno primarni ključ tablice te glavno njegovo svojstvo je to što ne može doći do dupliranja zapisa unutar tablice, tipa je cjelobrojni broj. Atribut „NARUDZBENICAKARATA_Id“ označava kojoj narudžbenici stavka pripada, tipa je cjelobrojni broj. „KARTICA_Id“ označava za koju karticu se stavka odnosi također je tipa cjelobrojnog broja. Sljedeći atribut „KARTICAPRODAJA_Id“ povezuje ovu tablicu s tablicom dbo.KarticaProdaja, tipa je cjelobrojni broj. „ARTIKAL_Id“ povezuje tablicu dbo.NarudzbenicaKarataStavka sa dbo.Artikal također je tipa cjelobrojnog broja. Atribut „RELACIJA_Id“ tip je cjelobrojni broj te povezuje tablicu dbo.Relacija s ovom tablicom. Nadalje atribut „Cijena“ predstavlja cijenu artikla odnosno mjesečne karte i iznos kojeg će pravna osoba trebati platiti, tipa je decimalnog broja. „Porez“ označava iznos poreza u trenutku naplate te je također tipa decimalnog broja. Atributi „PravoOd“ i „PravoDo“ datumskog su tipa te označava od kojeg do kojeg datuma nadopuna vrijedi. Posljednji atributi „Status“ i „Aktivan“ tipa su bit.

Tablica 2. Popis atributa u tablici dbo.NarudzbenicaKarataStavka

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj

NARUDZBENICAKARATA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.NarudzbenicaKarata
KARTICA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Kartica
KARTICAPRODAJA_Id	Int	Obavezno polje,	Povezivanje sa tablicom dbo.KarticaProdaja
ARTIKAL_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Artikal
RELACIJA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Relacija
Cijena	decimal	Obavezno polje	Cijena nadopune kartice
Porez	decimal	Obavezno polje	Iznos poreza u trenutku naplate
PravoOd	date	Obavezno polje	Datum od kada važi nadopune
PravoDo	date	Obavezno polje	Datum do kada važi nadopune do
Status	Bit	Obavezno polje	Status stavke narudžbenice
Aktivan	Bit	Obavezno polje	Aktivnost stavke narudžbenice
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.3 Tablica dbo.FizickaOsoba

Tablica dbo.FizickaOsoba sastoji se od 29 atributa. Prvi atribut „Id“ identifikacijski je broj odnosno primarni ključ tablice. „PRAVNAOSoba_Id“ označava kojoj pravnoj osobi fizička osoba pripada, odnosno ukoliko je osoba zaposlena, upisuje se naziv tvrtke/ustanove u kojoj radi, a ukoliko se radi o učeniku ili studentu, upisuje se naziv obrazovne ustanove koju pohađa. Sljedeći atribut „SKOLA_Id“ predstavlja koju školu fizička osoba pohađa, dok atribut „RELACIJA_Id“ označava na koju relaciju fizička osoba ima pravo. „MJESTO_Id“ govori iz kojeg mjesta osoba dolazi. Atributi „Ime“, „Prezime“, „OIB“, „Adresa“, „Spol“ „DatumRodjenja“ predstavljaju podatke o fizičkoj osobi. „Telefon“, „Mobitel“ i „Email“ su atributi pomoću kojih se može kontaktirati fizičku osobu u slučaju koje kakvih problema.

Tablica 3. Popis atributa u tablici dbo.FizickaOsoba

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
PRAVNAOSoba_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.PravnaOsoba
SKOLA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Skola
RELACIJA_Id	Int	Obavezno polje,	Povezivanje sa tablicom dbo.Relacija
MJESTO_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Mjesto
Ime	Nvarchar	Neobavezno polje	Ime fizičke osobe
Prezime	Nvarchar	Obavezno polje	Prezime fizičke osobe
OIB	Nvarchar	Neobavezno polje	OIB fizičke osobe

Adresa	Nvarchar	Obavezno polje	Adresa fizičke osobe
Telefon	Nvarchar	Neobavezno polje	Broj telefona fizičke osobe
Mobitel	Nvarchar	Neobavezno polje	Broj mobitela fizičke osobe
Email	Nvarchar	Neobavezno polje	Email adresa fizičke osobe
Spol	Nchar	Neobavezno polje	Spol fizičke osobe
DatumRodnja	Date	Obavezno polje	Datum rođenja fizičke osobe
Roditelj	Nvarchar	Neobavezno polje	Ime roditelja
PravoNaPratioca	Bit	Neobavezno polje	
Stranac	Bit	Neobavezno polje	
Slijepac	Bit	Neobavezno polje	
Razred	Tinyint	Neobavezno polje	
Semestar	Tinyint	Neobavezno polje	
KmDoSkole	Decimal	Neobavezno polje	
Drzava	Nvarchar	Neobavezno polje	
SIF_Osoba	Int	Neobavezno polje	
Aktivan	Bit	Obavezno polje	Oznaka da li je osoba aktivna u sustavu

Povlasteni	Bit	Obavezno polje	
LINIJAMATICNAGRUPA_ID	Int	Neobavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.4 Tablica dbo.PravnaOsoba

Tablica dbo.PravnaOsoba sastoji se od 18 atributa. Prvi atribut je „Id“ koji je identifikacijski broj pravne osobe. Atributi „Naziv“ i „Adresa“ predstavljaju naziv firme/tvrtke odnosno adresu sjedišta.

Tablica 4. Popis atributa u tablici dbo.PravnaOsoba

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
PRAVNAOSOBA_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.PravnaOsoba
PRAVNAOSOBATIP_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.PravnaOsobaTip
MJESTO_Id	Int	Neobavezno polje,	Povezivanje sa tablicom dbo.Mjesto
Naziv	Nvarchar	Obavezno polje	Naziv pravne osobe

Adresa	Nvarchar	Obavezno polje	Adresa pravne osobe
Telefon	Nvarchar	Neobavezno polje	Telefon pravne osobe
Mobitel	Nvarchar	Neobavezno polje	Mobitel pravne osobe
Fax	Nvarchar	Neobavezno polje	Fax pravne osobe
Email	Nvarchar	Neobavezno polje	Email adresa pravne osobe
KontaktOsoba	Nvarchar	Neobavezno polje	Kontakt osoba pravne osobe
IBAN	Nvarchar	Neobavezno polje	IBAN pravne osobe
PDVBroj	Nvarchar	Neobavezno polje	PDV broj pravne osobe
IsPrijevoznik	Bit	Neobavezno polje	Da li je pravna osoba prijevoznik
OIB	Nvarchar	Neobavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.5 Tablica dbo.Kartica

Tablica dbo.Kartica sastoji se od 15 atributa. Prvi atribut „Id“ je identifikacijski broj kartice. „FIZICKAOSOBA_Id“ povezuje tablicu dbo.Kartica sa dbo.FizickaOsoba, dok Atribut „UID“ je jedinstveni broj kartice.

Tablica 5. Popis atributa u tablici dbo.Kartica

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
FIZICKAOSOBA_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.FizickaOsoba
KARTICAVRSTA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.KarticaVrsta
KARTICA_Id_Reizdavanje	Int	Neobavezno polje	
UID	Nvarchar	Obavezno polje	Jedinstveni broj kartice
DatumIzrade	Datetime	Obavezno polje	Datum izrade kartice
DatumAktivacije	Datetime	Neobavezno polje	Datum aktivacije kartice
DatumStorna	Datetime	Neobavezno polje	Datum storniranja kartice
VrijediOd	nchar	Obavezno polje	Valjanost kartice od
VrijediDo	Nchar	Obavezno polje	Valjanost kartice do
Status	Nchar	Obavezno polje	Status kartice
RADNIK_Id	Int	Neobavezno polje	
SIF_Kartica	Int	Neobavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")

LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene
-------------------	----------	----------------	----------------------------

Izvor: Autor

2.1.6 Tablica dbo.KarticaProdaja

Tablica dbo.KarticaProdaja sastoji se od 37 atributa. Prvi atribut „Id“ je identifikacijski broj pojedinog zapisa unutar tablice. Atribut „KARTICA_Id“ povezuje tablicu dbo.Kartica sa tablicom dbo.KarticaProdaja. Nadalje atribut „KATEGORIJAPUTNIKA_Id“ povezuje tablicu dbo.KategorijaPutnika sa tablicom dbo.KarticaProdaja. Atribut „ARTIKAL_Id“ povezuje tablicu dbo.Artikal sa dbo.KarticaProdaja. Sljedeći atribut „RELACIJA_Id“ povezuje tablicu dbo.Relacija sa tablicom dbo.KarticaProdaja, dok atribut „NACINPLACANJA_Id“ povezuje tablicu dbo.NacinPlacanja sa tablicom dbo.KarticaProdaja. Sljedeći atribut „Datum“ sadrži zapis vremena kreiranja zapisa, dok atributi „PravoOd“ i „PravoDo“ sadrže informaciju od kada do kada vrijedi pravo nadopune. Atribut „Kolicina“ sadrži zapis o jediničnoj količini kupljenog prava. Posljednji atribut „Cijena“ definira novčani iznos nadopune kartice.

Tablica 6. Popis atributa u tablici dbo.KarticaProdaja

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
KARTICA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Kartica
KATEGORIJAPUTNIKA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.KategorijaPutnika
ARTIKAL_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Artikal

RELACIJA_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.Relacija
ZONA_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.Zona
KARTICAPRODAJASTATUS_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.KarticaProdajaStatus
POSLOVNIPROSTOR_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.PoslovniProstor
POS_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.POS
NACINPLACANJA_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.NacinPlacanja
RACUNSTAVKA_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.KategorijaPutnika
KARTICAPRODAJA_Id_Stornirana	Int	Neobavezno polje	
Datum	Datetime	Obavezno polje	Datum kreiranja zapisa
PravoOd	Date	Neobavezno polje	Vrijeme od kada pravo vrijedi
PravoDo	Date	Neobavezno polje	Vrijeme do kada pravo vrijedi
Kolicina	Decimal	Neobavezno polje	Kolicina na dokupljenog prava
Cijena	Decimal	Neobavezno polje	Cijena na dokupljenog prava
SubvencioniranaCijena	Decimal	Neobavezno polje	
Nadopuna	Decimal	Neobavezno polje	

Saldo	Decimal	Neobavezno polje	
Valuta	Smallint	Neobavezno polje	
PravoPrepisano	Bit	Neobavezno polje	
Napomena	Nvarchar	Neobavezno polje	
MjesecPrihoda	Date	Obavezno polje	
BrojMjeseciPrihoda	Tinyint	Neobavezno polje	
TipDCC	Int	Neobavezno polje	
OperaterKioska	Nvarchar	Neobavezno polje	
KIOSKPRODAJA_Id	Int	Neobavezno polje	
KioskPotvrđeno	Bit	Neobavezno polje	
Z_Odjava	Int	Neobavezno polje	
NeogranicenBrojVožnji	Bit	Neobavezno polje	
BrojVoznji	Smallint	Neobavezno polje	
ARTIKALSLOZENI_Id	Int	Neobavezno polje	
BARKODKARTA_Id	Int	Neobavezno polje	

LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.7 Tablica dbo.Artikal

Tablica dbo.Artikal sastoji se od 21 atributa. Prvi atribut „Id“ je identifikacijski broj pojedinog zapisa unutar tablice, dok atribut „Naziv“ sadrži naziv artikla.

Tablica 7. Popis atributa u tablici dbo.Artikal

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
FIZICKAOSOBA_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.FizickaOsoba
KARTICAVRSTA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.KarticaVrsta
KARTICA_Id_Reizdavanje	Int	Neobavezno polje	
UID	Nvarchar	Obavezno polje	Jedinstveni broj kartice
DatumIzrade	Datetime	Obavezno polje	Datum izrade kartice
DatumAktivacije	Datetime	Neobavezno polje	Datum aktivacije kartice

DatumStorna	Datetime	Neobavezno polje	Datum storniranja kartice
VrijediOd	nchar	Obavezno polje	Valjanost kartice od
VrijediDo	Nchar	Obavezno polje	Valjanost kartice do
Status	Nchar	Obavezno polje	Status kartice
RADNIK_Id	Int	Neobavezno polje	
SIF_Kartica	Int	Neobavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.8 Tablica dbo.Relacija

Tablica dbo.Relacija sastoji se od 11 atributa. Prvi atribut „Id“ je identifikacijski broj pojedinog zapisa unutar tablice, dok atribut „Naziv“ predstavlja naziv relacije.

Tablica 8. Popis atributa u tablici dbo.Relacija

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
Naziv	Nvarchar	Neobavezno polje	Naziv relacije
TarifnaUnija	Bit	Obavezno polje	
Aktivan	Bit	Obavezno polje	

ZONA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.Zona
StaraSifra	Int	Neobavezno polje	
VrijediZaCijeluZonu	Bit	Obavezno polje	
LastActionSubmitted	Bit	Obavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.9 Tablica dbo.Radnik

Tablica dbo.Radnik sastoji se od 33 atributa. U svakoj tablici unutar sustava „BusCARD“ nalazi se atribut „LASTUSER_Id“ koji je referenciran sa atributom „Id“ unutar tablice dbo.Radnik. Atribut „Id“ je identifikacijski broj pojedinog zapisa unutar tablice.

Tablica 9. Popis atributa u tablici dbo.Radnik

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
MB	Nvarchar	Neobavezno polje	Matični broj radnika
Aktivan	Bit	Obavezno polje	Aktivnost radnika
Ime	Nvarchar	Obavezno polje	Ime radnika
Prezime	Nvarchar	Obavezno polje	Prezime radnika

ImeRoditelja	Nvarchar	Obavezno polje	Ime radnikovog roditelja
Oib	Nvarchar	Obavezno polje	Osobni identifikacijski broj radnika
DatumRodjenja	Nvarchar	Obavezno polje	Datum rođenja radnika
Adresa	Nvarchar	Obavezno polje	Adresa radnika
MJESTO_Id	Int	Obavezno polje	Mjesto prebivališta
VozackaDozvola	Nvarchar	Obavezno polje	
VozackaDozvolaVrijediDo	Date	Neobavezno polje	
VozackaDozvolaIzdanaOd	Nvarchar	Neobavezno polje	
DigitalnaKartica	Nvarchar	Neobavezno polje	
DigitalnaKarticaVrijediDo	Date	Neobavezno polje	
Osobnalskaznica	Nvarchar	Obavezno polje	
OsobnalskaznicaVrijediDo	Date	Obavezno polje	
IsVozacAutobus	Bit	Obavezno polje	
IsVozacTramvaj	Bit	Obavezno polje	
DatumPocetkaRada	Datetime	Obavezno polje	
DatumPrestankaRada	Datetime	Neobavezno polje	
BrojKartice	Datetime	Neobavezno polje	
SIF_Radnik	Int	Neobavezno polje	

Token	Uniqueidentifier	Neobavezno polje	
TokenDtm	Datetime	Neobavezno polje	
PRAVNAOSOBA_Id	Int	Neobavezno polje	Pravna osoba kojoj radnik pripada
PIN	Nchar	Neobavezno polje	
LastActionSubmitted	Bit	Neobavezno polje	
StariBrojKartice	Int	Neobavezno polje	
FiskSifraOperatera	Nvarchar	Neobavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.10 Tablica dbo.PravnaOsobaLogin

Tablica dbo.PravnaOsobaLogin sastoji se od osam atributa. Ova tablica sadrži podatke koji su potrebni za prijavu na web aplikaciju sustava za kupovinu radničkih mjesečnih karata za pravne osobe. Prvi atribut „Id“ je identifikacijski broj pojedinog zapisa unutar tablice. Atribut „PRAVNAOSOBA_Id“ predstavlja kojoj pravnoj osobi namijenjeni podaci za prijavu. Sljedeći atribut „LoginName“ sadrži korisničko ime s kojim se pravna osoba prijavljuje u web aplikaciju, dok atribut „PasswordHash“ sadrži

lozinku u neizvornom obliku odnosno u kriptiranom formatu. Algoritam koji se koristi za kriptiranje je MD5. Atribut „Aktivan“ označava da li pravna osoba koristi web aplikaciju.

Tablica 10. Popis atributa u tablici dbo.PravnaOsobaLogin

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
PRAVNAOSOBA_Id	Int	Obavezno polje	Povezivanje sa tablicom dbo.PravnaOsoba
LoginName	Nvarchar	Obavezno polje	Korisničko ime za prijavu
PasswordHash	Varbinary	Obavezno polje	Kriptirani format lozinke za prijavu
Aktivan	Bit	Obavezno polje	Aktivnost korištenja web aplikacije
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.11 Tablica dbo.KategorijaPutnika

Tablica dbo.KategorijaPutnika sastoji se od sedam atributa. Prvi atribut „Id“ je identifikacijski broj pojedinog zapisa unutar tablice. Atribut „Naziv“ predstavlja naziv kategorije putnika

Tablica 11. Popis atributa u tablici dbo.KategorijaPutnika

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
Naziv	Nvarchar	Obavezno polje	Naziv kategorije putnika
Aktivan	Bit	Obavezno polje	
StariTipKartice	Int	Neobavezno polje	
IspisNaKartici	Nvarchar	Neobavezno polje	
SifraNaDCC	Int	Obavezno polje	
LastActionSubmitted	Bit	Obavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.1.12 Tablica dbo.Mjesto

Tablica dbo.Mjesto sastoji se od osam atributa. Prvi atribut „Id“ je identifikacijski broj pojedinog zapisa unutar tablice. Atribut „Naziv“ predstavlja naziv mjesta.

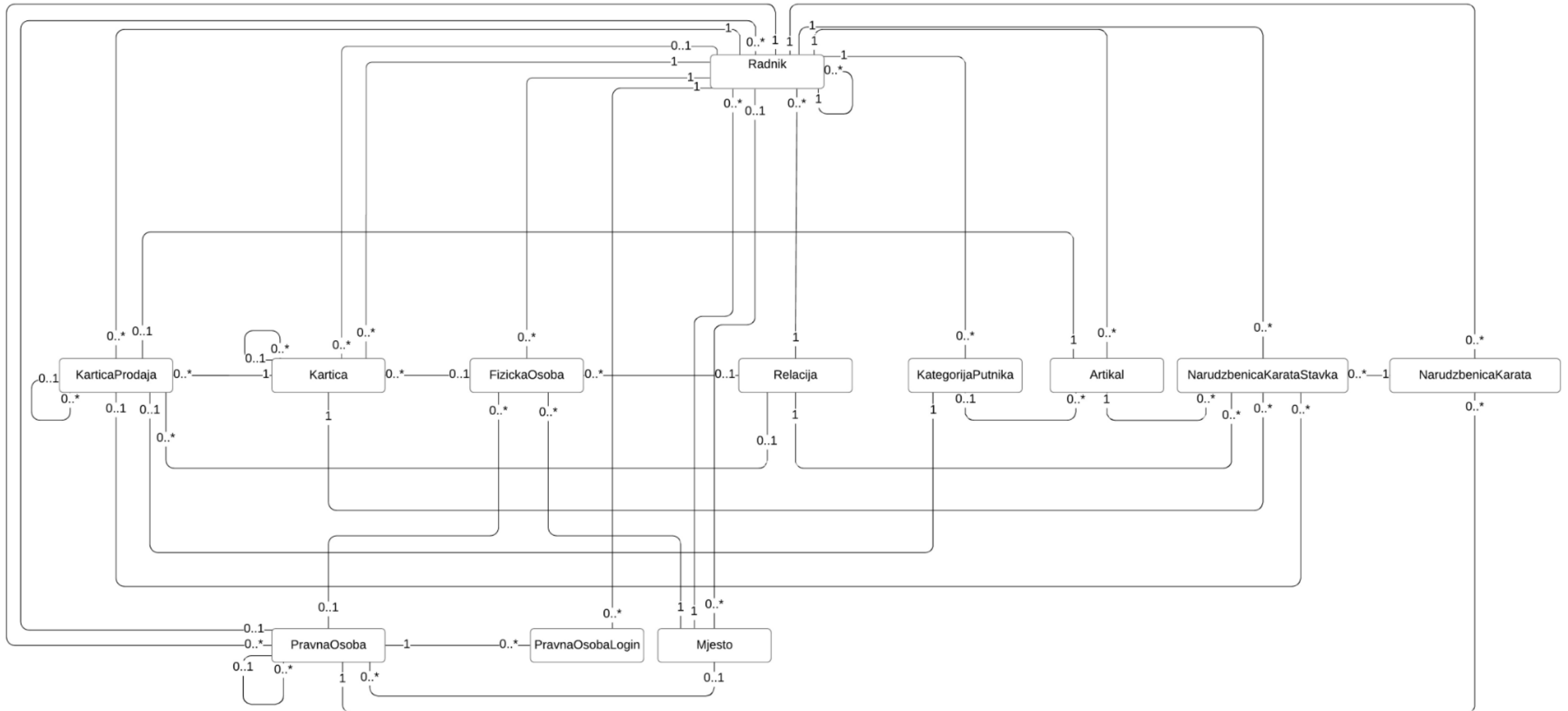
Tablica 12. Popis tablica u tablici dbo.Mjesto

Naziv atributa	Tip podatka	Specifičnosti	Opis
Id	Int	Primarni ključ, obavezan unos	Jedinstveni identifikacijski broj
Sifra	Int	Neobavezno polje	
MaticniBroj	Nvarchar	Neobavezno polje	Matični broj mjesta
Opcina_Id	Int	Neobavezno polje	Povezivanje sa tablicom dbo.Opcina
PPT	Nvarchar	Neobavezno polje	
Naziv	Nvarchar	Obavezno polje	Naziv mjesta
Zona	Int	Neobavezno polje	
LastActionSubmitted	Bit	Obavezno polje	
LASTUSER_Id	Int	Obavezno polje	Id korisnika dodavanja/ brisanja/ izmjenjivanja
LastAction	Nchar	Obavezno polje	Oznaka posljednje izmjene zapisa ("D", "I", "U")
LastUpdate	datetime	Obavezno polje	Vrijeme posljednje izmjene

Izvor: Autor

2.2 Dijagram baze podataka

U ovom potpoglavlju prikazan je grafički prikaz tablica, te veze među njima. Grafički prikaz kreiran je u LucidChart-u.



Slika 3. Grafički prikaz tablica

Izvor: Autor

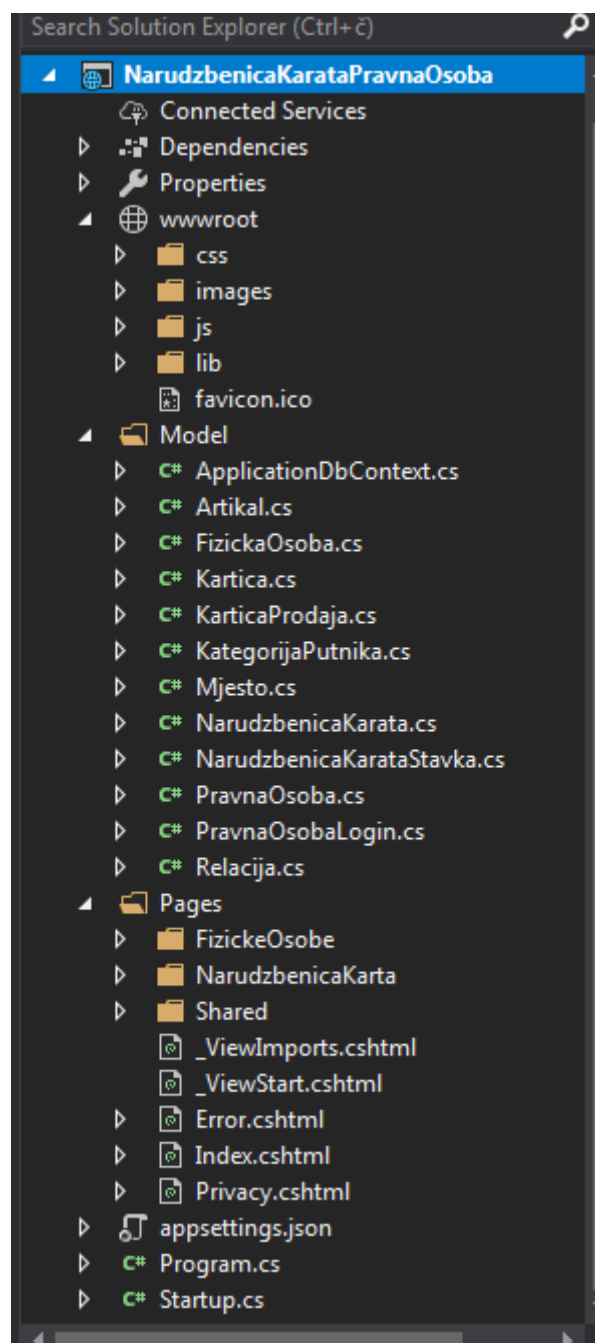
3. Web aplikacija

Web aplikacija sustava za kupovinu radničkih mjesečnih karata izrađena je u tehnologiji ASP.NET Core 2.2 Razor Pages. Ova tehnologija je softverski okvir koji je fokusiran na izradu dinamičkih web stranica koje se temelje na podacima s jasnim razdvajanjem problema. Razor Pages podržava da se prilikom razvoja aplikacija ista može implementirati u različite operative sustave poput Windows, Unix, Mac i dr. Razor Pages okvir je jednostavan i fleksibilan, te programeru pruža potpunu kontrolu nad HTML-om. Okvir Razor Pages izgrađen je na vrhu ASP.NET Core MVC softverskog okvira. Prednost razvijanja aplikacija u ASP.NET Core Razor Pages je ta da nije potrebno razumijevanje klasičnog MVC-a kako bi mogli raditi sa ovom tehnologijom. Razor Pages koristi C# programski jezik na poslužiteljskoj strani, te glavna njegova osobina je umetanje C# programskog koda unutar HTML koda za stvaranje dinamičkog sadržaja na web stranici.¹¹ Prilikom izrade web aplikacije nije korišten nikakav sustav za upravljanjem sadržajem odnosno Content management system (CMS).

3.1 Prikaz datotečnog i programskog dijela web aplikacije

Web aplikacija sustava za kupovinu radničkih mjesečnih karata sastoji se od više datoteka. Datoteke su prikazane na slici 4. Sve Razor datoteke završavaju s ekstenzijom .cshtml.

11 „Learn Razor Pages“. 2020. Learn Razor Pages. <https://www.learnrazorpages.com/>

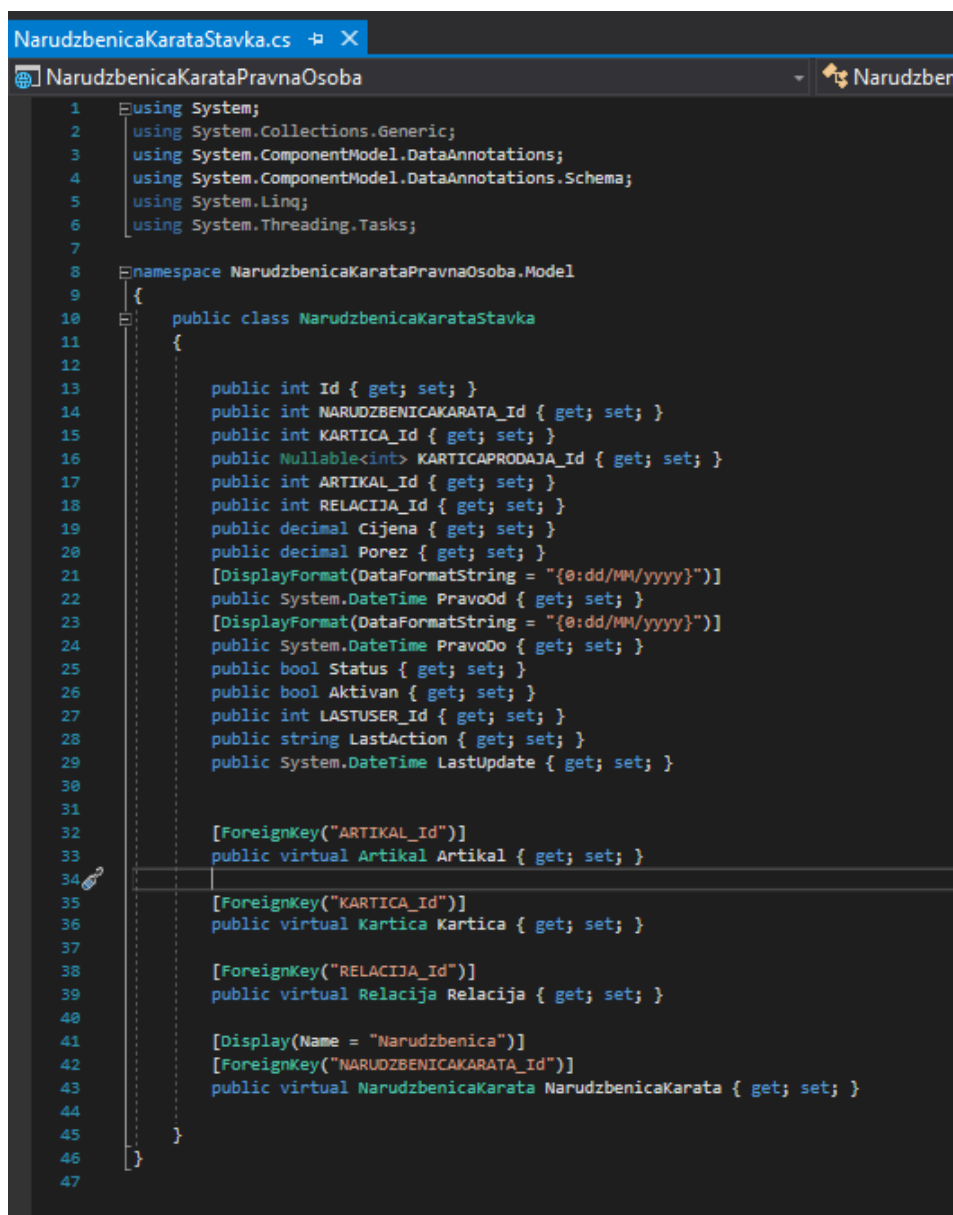


Slika 4. Datoteke web aplikacije

Izvor: Autor

Web aplikacija sustava za kupovinu radničkih mjesečnih karata sastoji se od osam glavnih datoteka i mapa. „Dependencies“ sadrži neophodne pakete koji su potrebni za normalno izvršavanje aplikacije. „Properties“, postavke projekta organizirana su u grupe i svaka grupa ima svoje svojstvo. Svako pojedino svojstvo može biti definirano za različite jezike i vrstu projekta. Mapa „wwwroot“ predstavlja statičke datoteke koje se nalaze unutar projekta. Ova mapa sadrži sve stilske attribute (CSS), ukoliko oni nisu

direktno napisani unutar HTML-a, također sadrži sve statičke datoteke poput fotografija, animacija i dr., nadalje također sadrži i cjelokupan JavaScript kod ukoliko nije napisan direktno unutar web stranica kao i CSS. Ova mapa sadrži druge okvire poput „Bootstrap-a“, „jQuery-a“ i sl. Mapa „Models“ služi kako bi mogli raditi s podacima. Atributi tablica unutar baze podataka koji se koriste u web aplikaciji sustava za kupovinu radničkih mjesečnih karata organizirani su u zasebne datoteke odnosno klase. Na sljedećoj slici prikazan je primjer klase za tablicu dbo.NarudzbenicaKarataStavka.



```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.ComponentModel.DataAnnotations.Schema;
5  using System.Linq;
6  using System.Threading.Tasks;
7
8  namespace NarudzbenicaKarataPravnaOsoba.Model
9  {
10     public class NarudzbenicaKarataStavka
11     {
12
13         public int Id { get; set; }
14         public int NARUDZBENICAKARATA_Id { get; set; }
15         public int KARTICA_Id { get; set; }
16         public Nullable<int> KARTICAPRODAJA_Id { get; set; }
17         public int ARTIKAL_Id { get; set; }
18         public int RELACIJA_Id { get; set; }
19         public decimal Cijena { get; set; }
20         public decimal Porez { get; set; }
21         [DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}")]
22         public System.DateTime PravoOd { get; set; }
23         [DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}")]
24         public System.DateTime PravoDo { get; set; }
25         public bool Status { get; set; }
26         public bool Aktivan { get; set; }
27         public int LASTUSER_Id { get; set; }
28         public string LastAction { get; set; }
29         public System.DateTime LastUpdate { get; set; }
30
31
32         [ForeignKey("ARTIKAL_Id")]
33         public virtual Artikal Artikal { get; set; }
34
35         [ForeignKey("KARTICA_Id")]
36         public virtual Kartica Kartica { get; set; }
37
38         [ForeignKey("RELACIJA_Id")]
39         public virtual Relacija Relacija { get; set; }
40
41         [Display(Name = "Narudzbenica")]
42         [ForeignKey("NARUDZBENICAKARATA_Id")]
43         public virtual NarudzbenicaKarata NarudzbenicaKarata { get; set; }
44     }
45 }
46
47

```

Slika 5. Primjer klase za tablicu dbo.NarudzbenicaKarataStavka

Izvor: Autor

Svaka linija koda unutar klase naziva se svojstvo (engl. Property). To je proširenje varijable klase i pruža mehanizam za čitanje, pisanje ili izmjenu vrijednosti varijable klase. Svojstvo može sadržavati jedno ili dva bloka programskog koda. Blokovi programskog koda nazivaju se funkcije dohvaćanja (Get) i postavljanja (Set) (engl. Accessors). Svojstva unutar klase definirani su kao javni.¹²

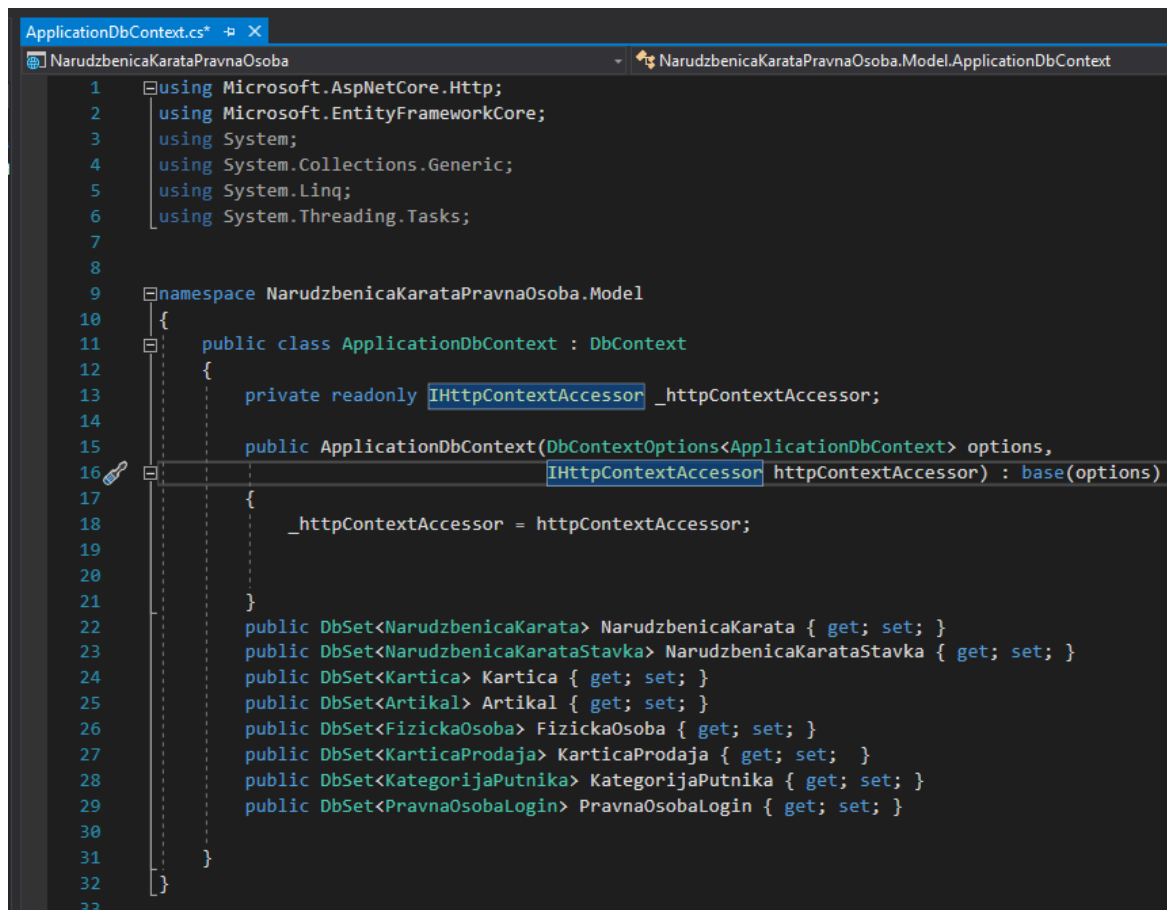
U C# programskom jeziku svojstva su kategorizirana u tri tipa:

Tip	Opis
Dohvaćanje-Postavljanje	Svojstvo sadrži i funkciju dohvaćanja i funkciju postavljanja
Dohvaćanje	Svojstvo sadrži samo funkciju dohvaćanja
Postavljanje	Svojstvo sadrži samo funkciju postavljanja

Izvor: Autor

Klasa „ApplicationDbContext“ nasljeđuje klasu „DbContext“. Ova klasa koristi je se s izvedbenim tipom koji sadrži svojstvo DbSet<TEntity>. Ovo svojstvo može se koristiti za izradu upita nad podacima i za spremanje samih podataka u bazu podataka. Da bi vršili manipulaciju nad podacima moramo navesti svojstva odnosno klase iz modela koje su prikazane na slici 6.

¹² „C# Properties (GET, SET)“,2019. Tutlane. <https://www.tutlane.com/tutorial/csharp/csharp-properties-get-set>



```

1  using Microsoft.AspNetCore.Http;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Threading.Tasks;
7
8
9  namespace NarudzbenicaKarataPravnaOsoba.Model
10 {
11     public class ApplicationDbContext : DbContext
12     {
13         private readonly IHttpContextAccessor _httpContextAccessor;
14
15         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options,
16             IHttpContextAccessor httpContextAccessor) : base(options)
17         {
18             _httpContextAccessor = httpContextAccessor;
19
20         }
21
22         public DbSet<NarudzbenicaKarata> NarudzbenicaKarata { get; set; }
23         public DbSet<NarudzbenicaKarataStavka> NarudzbenicaKarataStavka { get; set; }
24         public DbSet<Kartica> Kartica { get; set; }
25         public DbSet<Artikal> Artikal { get; set; }
26         public DbSet<FizickaOsoba> FizickaOsoba { get; set; }
27         public DbSet<KarticaProdaja> KarticaProdaja { get; set; }
28         public DbSet<KategorijaPutnika> KategorijaPutnika { get; set; }
29         public DbSet<PravnaOsobaLogin> PravnaOsobaLogin { get; set; }
30     }
31 }
32
33

```

Slika 6. Prikaz klase „ApplicationDbContext“

Izvor: Autor

Mapa „Pages“ sadrži sve web stranice u aplikaciji, te sadrži tri podmape i pet stranica. Podmapa „FizickeOsobe“ sadrži stranicu Index.cshtml koja prikazuje sve fizičke osobe koje su povezane sa pravnom osobom. Na slikama 7 i 8 prikazan je programski kod za dohvat i prikaz fizičkih osoba na ekranu.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Security.Claims;
5 using System.Threading.Tasks;
6 using Microsoft.AspNetCore.Authentication;
7 using Microsoft.AspNetCore.Authentication.Cookies;
8 using Microsoft.AspNetCore.Http;
9 using Microsoft.AspNetCore.Mvc;
10 using Microsoft.AspNetCore.Mvc.RazorPages;
11 using Microsoft.EntityFrameworkCore;
12 using NarudzbenicaKarataPravnaOsoba.Model;
13
14 namespace NarudzbenicaKarataPravnaOsoba.Pages.FizickeOsobe
15 {
16     public class IndexModel : PageModel
17     {
18         private readonly ApplicationDbContext _db;
19         private readonly IHttpContextAccessor _httpContextAccessor;
20
21         public IndexModel(ApplicationDbContext db, IHttpContextAccessor httpContextAccessor)
22         {
23             _db = db;
24             _httpContextAccessor = httpContextAccessor;
25         }
26
27         public IEnumerable<FizickaOsoba> FizickaOsoba { get; set; }
28
29         public async Task OnGetAsync()
30         {
31             var user = _httpContextAccessor.HttpContext;
32             string userIdString = user.User.FindFirst(ClaimTypes.Actor).Value;
33             int userId = Int32.Parse(userIdString);
34
35             if(userId == 0)
36             {
37                 RedirectToPage("/NarudzbenicaKarta/Login");
38             }
39             else
40             {
41                 FizickaOsoba = await _db.FizickaOsoba.Where(m => m.PRAVNAOSOBA_Id == userId && m.LastAction != "D").ToListAsync();
42             }
43         }
44
45         public async Task<IActionResult> OnPostLogoutAsync()
46         {
47             await HttpContext.SignOutAsync();
48             HttpContext.Session.Clear();
49             foreach (var cookieKey in Request.Cookies.Keys)
50             {
51                 Response.Cookies.Delete(cookieKey);
52             }
53             await HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
54             return RedirectToPage("/NarudzbenicaKarta/Login");
55         }
56     }
57 }
58

```

Slika 7. Prikaz programskog koda za dohvaćanje fizičkih osoba

Izvor: Autor

Konstruktor IndexModel prima dva parametra; prvi parametar je ApplicationDbContext koji služi da bi imali konekciju na bazu podataka, dok pomoću drugog parametara dohvaćamo podatke o prijavi korisnika. Prva metoda OnGetAsync dohvaća sve fizičke osobe koje su povezane sa prijavljenom pravnom osobom, te koje imaju različit LastAction od „D“ odnosno one fizičke osobe koje nisu obrisane. Ukoliko dolaskom u funkciju userId nije pronađen aplikacija se preusmjerava na početnu formu za prijavu u aplikaciju. Preusmjeravanje se također događa kada sesija aplikacije istekne. Druga metoda OnPostLogoutAsync poziva se prilikom pritiska na dugme Odjava, gdje se

pritom briše aktualna sesija i kolačići aplikacije te na posljepku se vrši preusmjerenje na početnu stranicu za prijavu u aplikaciju.

```
1 @page
2 @model NarudzbenicaKarataPravnaOsoba.Pages.FizickeOsobe.IndexModel
3 @{
4     Layout = "_Layout";
5 }
6 <br />
7
8 <div class="row container-fluid">
9     <h2 class="text-info">Popis fizičkih osoba</h2>
10 </div>
11 <br />
12 <div class="container-fluid">
13
14     <if (Model.FizickaOsoba.Count() > 0)>
15     {
16
17
18         <table id="tablica" class="table table-striped border table-fit table-hover">
19             <tr class="table-secondary">
20                 <th>@Html.DisplayNameFor(m => m.FizickaOsoba.FirstOrDefault().Id)</th>
21                 <th>@Html.DisplayNameFor(m => m.FizickaOsoba.FirstOrDefault().Ime)</th>
22                 <th>@Html.DisplayNameFor(m => m.FizickaOsoba.FirstOrDefault().Prezime)</th>
23                 <th>@Html.DisplayNameFor(m => m.FizickaOsoba.FirstOrDefault().OIB)</th>
24                 <th>@Html.DisplayNameFor(m => m.FizickaOsoba.FirstOrDefault().Adresa)</th>
25                 <th>@Html.DisplayNameFor(m => m.FizickaOsoba.FirstOrDefault().Telefon)</th>
26                 <th>@Html.DisplayNameFor(m => m.FizickaOsoba.FirstOrDefault().Mobitel)</th>
27
28             </tr>
29
30             <foreach (var item in Model.FizickaOsoba)>
31             {
32                 <tr>
33                     <td>@Html.DisplayFor(m => item.Id)</td>
34                     <td>@Html.DisplayFor(m => item.Ime)</td>
35                     <td>@Html.DisplayFor(m => item.Prezime)</td>
36                     <td>@Html.DisplayFor(m => item.OIB)</td>
37                     <td>@Html.DisplayFor(m => item.Adresa)</td>
38                     <td>@Html.DisplayFor(m => item.Telefon)</td>
39                     <td>@Html.DisplayFor(m => item.Mobitel)</td>
40
41                 </tr>
42             }
43         </table>
44     }
45     else
46     {
47         <p>Nema fizičkih osoba</p>
48     }
49
50 </div>
```

Slika 8. Prikaz programskog koda za prikaz fizičkih osoba na ekranu

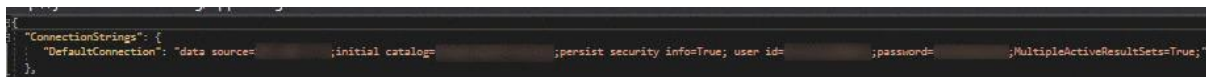
Izvor: Autor

Na slici 8 prikazan je programski kod za prikaz podataka o fizičkim osobama. Ukoliko lista fizičkih osoba sadrži fizičke osoba, odnosno kada lista nije prazna, tada se ide

kreirati zaglavlje tablice prikaza. Nakon toga za svaku fizičku osobu iz liste kreira se zaseban redak u tablici.

Podmapa „NarudzbenicaKarta“ sadrži tri stranice, stranica Login.cshtml sadrži formu za prijavu u aplikaciju. Korisnik odnosno pravna osoba treba unijeti korisničko ime i lozinku koje su dodijeljeni od referenta u tvrtci. Druga stranica je Index.cshtml koja je početna stranica aplikacije odnosno stranica koja se pojavljuje nakon uspješne prijave u aplikaciju, te sadrži popis svih kreiranih narudžbenica i stavki. Treća stranica je Create.cshtml koja služi za kreiranje narudžbenice i stavaka narudžbenice. Podmapa „Shared“ sadrži tri djelomične stranice koje se koriste na drugim stranicama unutar aplikacije. Prvi djelomični prikaz je _CookieConsentPartial.cshtml koji se pojavljuje prilikom prvog puta pokretanja aplikacije. Ova stranica prikazuje skočni prozor u kojemu se pita korisnika da li prihvaća kolačiće koji su neophodni za ispravni rad aplikacije. Drugi djelomični prikaz u „Shared“ mapi je _Layout. Kako sve stranice osim stranice za prijavu u aplikaciju imaju isto zaglavlje, podnožje i navigacijski sustav, to je definirano u ovom djelomičnom prikazu, te to rezultira smanjenjem količine programskog koda. Posljednji djelomični prikaz je _ValidationScriptsPartial.cshtml koji se koristi za provjeru na klijentskoj strani. Na kraju programskog koda stranice _Layout nalazi se kod: „@RenderSection("Scripts", required:false)“ u kojemu je definirano da nije potrebna dodatna provjera valjanosti klijenta. Datoteka _ViewImports.cshtml sadrži tri direktive, @using, @namespace i @addTagHelper. Direktiva @using predstavlja okruženje aplikacije odnosno ukoliko je u ovoj datoteci definirano neko okruženje istog nije potrebno pozivati u ostalim stranicama, direktiva @namespace služi za definiranje korijenskog okruženja za Razor stranice. Ukoliko je u ovoj datoteci definirano više okruženja u obzir se uzima posljednje definirano. Okruženje @addTagHelper služi kako bi programski kod stranice mogao sudjelovati u kreiranju prikaza HTML elemenata u Razor datotekama. Datoteka _ViewStart.cshtml koristi se za postavljanje izgleda stranice za svaku stranicu, odnosno onaj sadržaj koji je definiran u datoteci _Layout.cshtml. Stranice Error.cshtml, Index.cshtml i

Privacy.cshtml automatsko su kreirane stranice prilikom izrade projekta.¹³ U datoteci appsettings.json definiran je znakovni niz za spajanje na izvor podataka odnosno bazu podataka. Znakovni niz prikazan je na slici 9. Neki dijelovi su skriveni zbog sigurnosti i integriteta baze podataka.

The image shows a code editor window with a dark background. It displays a JSON configuration object named "ConnectionStrings". Inside this object, there is a key "DefaultConnection" whose value is a SQL Server connection string. The string is partially obscured by redaction boxes (grey rectangles). The visible parts of the string are: "data source=", ";initial catalog=", ";persist security info=True; user id=", ";password=", and ";MultipleActiveResultSets=True;". The string is enclosed in double quotes and the entire JSON object is enclosed in curly braces.

Slika 9. Znakovni niz za spajanje aplikacije na bazu podataka

Izvor: Autor

U „ConnectionStrings“ definiran je „DefaultConnection“ u kojemu su svi podaci potrebni kako bi aplikacija mogla imati komunikaciju s izvorom podataka. Data source predstavlja lokaciju baze podataka. Initial catalog je naziv same baze podataka. Persist security info služi kako bi Windows zapamtio lozinku koja je specificirana u znakovnom nizu tj. unutar ConnectionString-a. User Id i password sadrže korisničko ime i lozinku koji su potrebni kako bi se aplikacija uspješno spojila na bazu podataka. Nadalje nakon appsettings.json, aplikacija sadrži klasu Program.cs. čiji sadržaj je prikazan na slici 10.

¹³ „Learn Razor Pages“. 2020. Learn Razor Pages. <https://www.learnrazorpages.com/>


```

1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Threading.Tasks;
6  using Microsoft.AspNetCore;
7  using Microsoft.AspNetCore.Hosting;
8  using Microsoft.Extensions.Configuration;
9  using Microsoft.Extensions.Logging;
10
11 namespace NarudzbenicaKarataPravnaOsoba
12 {
13     public class Program
14     {
15         public static void Main(string[] args)
16         {
17             CreateWebHostBuilder(args).Build().Run();
18         }
19
20         public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
21             WebHost.CreateDefaultBuilder(args)
22                 .UseStartup<Startup>();
23     }
24 }

```

Slika 10. Sadržaj klase Program.cs

Izvor: Autor

Aplikacija prilikom pokretanja gradi „host-a“. „Host“ obuhvaća sve resurse aplikacije poput implementacije HTTP poslužitelja, konfiguracije i Dependency injection-a i dr. Dependency injection je princip dodavanja ili dodjeljivanja instance objekta varijabli nekog drugog objekta.¹⁴ U ovoj klasi konfiguracija se postavlja kao dio `WebHost.CreateDefaultBuilder` metode koja je ulazna točka aplikacije. Aplikacija se sastoji još i od `Startup.cs` klase u kojoj su konfigurirane usluge i mehanizam za obradu zahtjeva (engl. Request pipeline). ASP.NET Core aplikacija mora sadržavati `Startup` klasu. `Startup` klasa je kao `Global.asax` u tradicionalnim .NET aplikacijama. Ova klasa kao što joj naziv govori izvršava se prva prilikom pokretanja aplikacije.¹⁵ Na slici 9 prikazan je sadržaj klase `Startup`.

14 „Dependency injection (inversion of control)“ 2013. Učim programiranje. <http://www.ucim-programiranje.com/2013/02/dependency-injection-inversion-of-control/>

15 „ASP.NET Core – Startup Class“, 2019. TutorialTeacher <https://www.tutorialsteacher.com/core/aspnet-core-startup>

```

using System;
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using NarudbenicaKartaPravnaOsoba.Model;
namespace NarudbenicaKartaPravnaOsoba
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMemoryCache();
            services.AddAuthentication(options =>
            {
                options.DefaultSignInScheme = CookieAuthenticationDefaults.AuthenticationScheme;
                options.DefaultAuthenticateScheme = CookieAuthenticationDefaults.AuthenticationScheme;
                options.DefaultChallengeScheme = CookieAuthenticationDefaults.AuthenticationScheme;
            }).AddCookie(options => { options.LoginPath = "/NarudbenicaKarta/Login"; });
            services.AddMvc().AddRazorPagesOptions(options =>
            {
                options.Conventions.AuthorizeFolder("/");
                options.Conventions.AuthorizePage("/NarudbenicaKarta");
                options.Conventions.AllowAnonymousPage("/NarudbenicaKarta/Login");
            });
            services.Configure<CookiePolicyOptions>(options =>
            {
                // This lambda determines whether user consent for non-essential cookies is needed for a given request.
                options.CheckConsentNeeded = context => false;
                options.MinimumSameSitePolicy = SameSiteMode.None;
            });
            services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));
            services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
            services.AddMvc();
            services.AddDistributedMemoryCache();
            services.AddSession(options =>
            {
                options.Cookie.Name = "SessionCookie";
                options.IdleTimeout = TimeSpan.FromMinutes(60);
                options.Cookie.IsEssential = false;
            });
            services.AddSingleton<HttpContextAccessor, HttpContextAccessor>();
        }

        // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
        public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }
            else
            {
                app.UseExceptionHandler("/error");
                app.UseHsts();
            }
            app.UseAuthentication();
            app.UseHttpsRedirection();
            app.UseStaticFiles();
            app.UseCookiePolicy();
            app.UseSession();
            app.UseMvc();
        }
    }
}

```

Slika 11. Startup klasa

Izvor: Autor

Startup klasa ima dvije metode, ConfigureServices i Configure metodu. ConfigureServices metoda je opcionalna. Unutar ove funkcije vrši se podešavanja putem Add{Service} metode poput AddDbContext, AddSession i dr. Configure metoda

koristi se za određivanje na koji će način aplikacija reagirati na HTTP zahtjeve. Mehanizam za obradu zahtjeva konfigurira se dodavanjem middleware komponenti na instancu `IApplicationBuilder`-a. Middleware je komponenta odnosno klasa koja se izvršava na svaki zahtjev ASP.NET Core aplikacije.¹⁶

3.2 Prikaz grafičkog dijela web aplikacije

U ovom poglavlju prikazane su sve stranice web aplikacije Sustava za kupovinu radničkih mjesečnih karata za pravne osobe.

Slika 12. Forma za prijavu u web aplikaciju

Izvor: Autor

Forma sadrži dva polja za upisivanje podataka i gumb za kontrolu upisanih podataka. U prvo polje upisuje se korisničko ime, dok u drugo polje lozinka. Na slici 12. „dipl“ predstavlja korisničko ime, dok točke predstavljaju lozinku. Ako je korisnik uspješno unio korisničko ime i lozinku te nakon pritiska na gumb PRIJAVA, korisnik će biti

16 „ASP.NET Core – Middleware Class“, 2019. TutorialTeacher.
<https://www.tutorialsteacher.com/core/aspnet-core-middleware>

preusmjeren na stranicu koja je prikazana na slici 13, u protivnom to se neće dogoditi, već će ostati na trenutnoj stranici za prijavu u aplikaciju (slika 12.).

PULAPROMET

Narudžbenica karata Kreiranje narudžbenice Fizičke osobe

Odjava

Lista narudžbenica

Kreiraj novu narudžbenicu

0320_1_ U
Obradi

0120_2_ Odobreno

0120_1_ U
Obradi

1219_1_ Odobreno

UID	Ime	Prezime	OIB	Relacija	Naziv	Cijena	PravoOd	PravoDo
CC3EDC95	Ajla	Nikolić		ZONA 1+2	Opća mjesečna	400,00	01.03.2020	31.03.2020
DC2CCB95	Mislav	Ivanović		ZONA 1+2	Opća mjesečna	400,00	01.03.2020	31.03.2020

Ukupno : 800 KN

© 2020 - PulaPromet

Slika 13. Web stranica sa pregledom narudžbenica i stavki narudžbenice

Izvor: Autor

Stranica prikazana na gornjoj slici prikazuje sve kreirane narudžbenice od strane pravne osobe. Klikom na jednu od narudžbenica koje se nalaze na lijevoj strani ispod liste narudžbenica, prikažu se sve stavke. Ako odabrana narudžbenica ima više stavki, njih je moguće pretraživati tražilicom koja se nalazi na vrhu tablice. Na desnoj strani sučelja nalazi se gumb „Kreiraj novu narudžbenicu“, klikom na njega preusmjerava se na stranicu koja je prikazana na slici 14. Također to preusmjeravanje je moguće ostvariti i klikom na „Kreiraj narudžbenicu“ koja se nalazi u navigacijskoj traci na vrhu sučelja. Navigacijska traka nalazi se na svim stranicama web aplikacije, osim na stranici prijave u sustav. Ovisno na koji element se klikne, isti preusmjerava na

određenu stranicu. U navigacijskoj traci se nalazi još i logotip Pulapromet poduzeća za kojeg je i ovaj sustav trenutno namijenjen, nadalje u navigacijskoj traci nalazi se i gumb „Odjava“ pomoću kojeg se korisnik odjavljuje iz aplikacije.

PULAPROMET Narudžbenica karata Kreiranje narudžbenice Fizičke osobe [Odjava](#)

Kreiraj novu narudžbenicu karata

01.2020 ▼

Id	UID	Ime	Prezime	OIB	Relacija	Adresa	Mobitel	Kupovina
41234	8C0D0295	Ivan	Mirković		ZONA 1+2	Stankovićeve 26	0998785456	☐
43078	FC873295	Martina	Vuković		ZONA 1+2	Ul. Ivo Lole Ribara 22		☐
43345	EC39D394	Nikola	Beriša		ZONA 1	Jadreški 15		☐

[Spremi narudžbenicu](#)

© 2020 - PulaPromet

Slika 14. Web stranica za kreiranje narudžbenice i stavki narudžbenice

Izvor: Autor

Stranica prikazana na gornjoj slici prikazuje sučelje za kreiranje narudžbenice. Odabirom elementa u padajućem izborniku odabire se za koji mjesec se želi kreirati narudžbenica. Pravo od kada do kada vrijedi narudžbenica i njezine stavke definirana je za period od početka odabranog mjeseca do kraja odabranog mjeseca. Ako je već kreirana jedna ili više narudžbenica za odabrani mjesec, prikazat će se samo one fizičke osobe koje nisu navedene u već dosadašnjim kreiranim narudžbenicama. Time se sprječava kupovina dvostrukog prava za odabrani mjesec. Odabir za koje fizičke osobe se želi kupiti pravo ostvaruje se klikom na potvrdni okvir. Nakon odabira fizičkih osoba korisnik aplikacije treba kliknuti na dugme „Spremi narudžbenicu“. Klikom na dugme kreira se narudžbenica i njezine stavke, te zapis o kupovini prava koji se bilježi

u tablici „dbo.KarticaProdaja“. Nakon spremanja automatski se dogodi preusmjerenje na početnu stranicu aplikacije. Na slici 15 nalazi se popis svih fizičkih osoba koje pripadaju pravnoj osobi.

PULAPROMET

[Narudžbenica karata](#) [Kreiranje narudžbenice](#) [Fizičke osobe](#)

Odjava

Popis fizičkih osoba

Id	Ime	Prezime	OIB	Adresa	Telefon	Mobitel
29157	Emily	Batelić		Jadreški 28 A		
29862	Patrik	Knezović		Varda Cologne 19, Ližnjan		
30083	Lejla	Jahja		centar 9 ŠišAN		097 660 33 66
31386	Zoran	Simetić		Plekuti 60		0989014631
31917	Ilenia	Cinkopan		Pod vrh 43, Ližnjan		0989600572
31918	Nicole	Cinkopan		Pod vrh 43, LižN.JAN		0989600572
32872	Ridvan	Beriša	52855607307	Jadreški 60		

© 2019 - PulaPromet

Slika 15. Web stranica sa popisom fizičkih osoba

Izvor: Autor

Stranica prikazana na gornjoj slici sadrži popis svih fizičkih osoba koje pripadaju pravnoj osobi. Fizičke osobe moguće je pretraživati filterom na vrhu tablice.

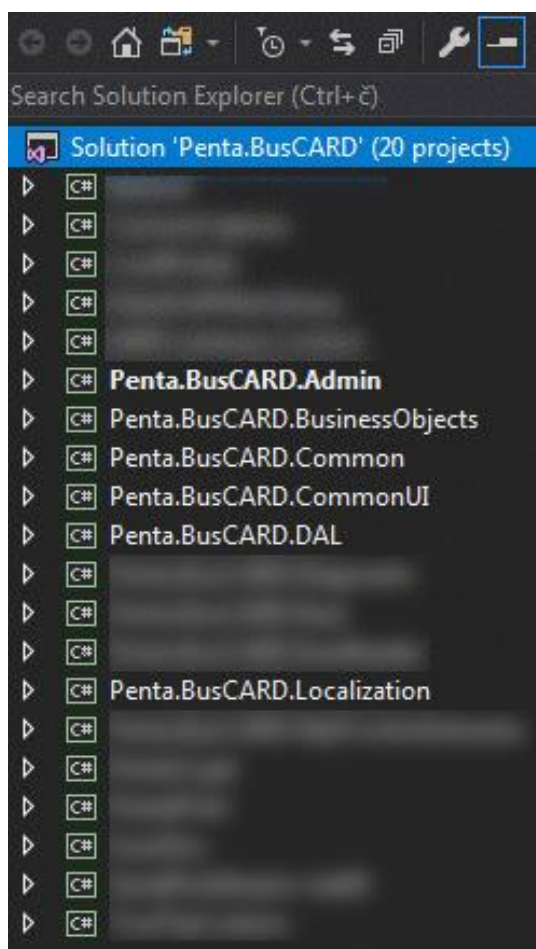
4. Desktop aplikacija

Desktop aplikacija sustava za kupovinu radničkih mjesečnih karata realizirana je kroz Microsoft-ovu tehnologiju MVVM koja je objašnjena na početku ovog rada. Kao dodatak MVVM korišten je Entity Framework odnosno ADO.NET. ADO.NET pruža

dosljedan pristup izvoru podataka odnosno SQL Serveru. ADO.NET sastoji se od skupa klasa koje se nalaze u biblioteci „Using System.Data“.

4.1 Prikaz datotečnog i programskog dijela desktop aplikacije

BusCARD sustav sastoji se od 20 manjih projekata koji su prikazani na slici 16. Neki od projekata sakriveni su zbog sigurnost tvrtke „Penta d.o.o.“.

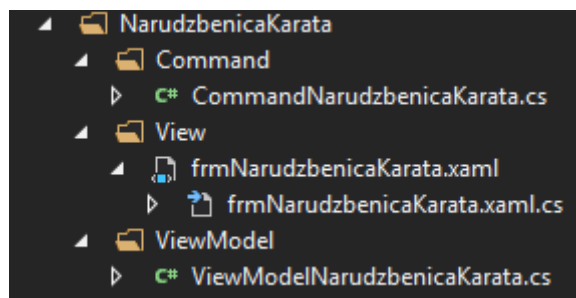


Slika 16. Popis projekata u BusCARD sustavu.

Izvor: Autor

U projektu Penta.BusCARD.Admin nalazi se mapa Forms u kojoj se nalaze podmape NarudzbenicaKarata i PravnaOsobaLogin, te frmFizickaOsoba, dok izvan mape Forms nalazi se MainWindow koji predstavlja glavni zaslon aplikacije. Podmape

NarudzbenicaKarata i PravnaOsobaLogin sastoje se od tri podmape Command, View i ViewModel. Na slici 17 prikazan je sadržaj mape NarudzbenicaKarata.



Slika 17. Prikaz sadržaja mape NarudzbenicaKarata

Izvor: Autor

U mapi Command nalazi se klasa u kojoj su definirane naredbe za gumbove, U mapi View definiran je izgled i struktura zaslona, dok u mapi ViewModel nalazi se klasa u kojoj su definirana svojstva prikaza na ekranu. U projektu Penta.BusCARD.BusinessObjects nalaze se klase NarudzbenicaKarata i NarudzbenicaKarataStavka u kojima su definirani business i basic objekti. Pomoću business objekata dohvaćamo sve attribute, dok s basic objektima samo one koje prikazujemo na ekranu. Unutar projekta Penta.BusCARD.CommonUI nalazi se mapa Resources u kojemu se nalaze sve ikone i fontovi slova. Nadalje projekt Penta.BusCARD.DAL sastoji se od 3 podmape, Mappers, Model, Repositories. U podmapi Mappers nalazi se NarudzbenicaKarataMapper, NarudzbenicaKarataStavkaMapper i PravnaOsobaLoginMapper. Mapperi služe za povezivanje objekata s entitetima u bazi podataka. U podmapi Model nalazi se BusCARDModel.edmx koji predstavlja kontainer za Entity data model, u podmapi Repositories nalaze se klase koje sadrže metode za manipulaciju s podacima. Jedna od metoda u NarudzbenicaKarataRepository je GetAllByPravnaOsoba koja je prikazana na slici 18. Ova metoda dohvaća sve narudžbenice karata od određene pravne osobe, te ju vraća kao listu.


```

public List<NarudzbenicaKarataObject> GetAllByPravnaOsoba(Int32 PravnaOsobaId)
{
    BusCARDEntities context = null;

    try
    {
        context = new MyBusCARDEntities();

        return context.NarudzbenicaKarata
            .Where(arg => arg.LastAction != "D" && arg.PRAVNAOSOBA_Id == PravnaOsobaId)
            .AsEnumerable()
            .Select(row => NarudzbenicaKarataMapper.ToBusinessObject(row))
            .ToList();
    }
    catch
    {
        throw new BusCARDCustomException(LocalizationProvider.GetLocalizedValue("NEOCEKIVANA_GRESKA"));
    }
    finally
    {
        if (context != null)
            context.Dispose();
    }
}

```

Slika 18. Funkcija za dohvaćanje narudžbenica karata pojedine pravne osobe

Izvor: Autor

U projektu Penta.BusCARD.Localization definirana je lokalizacija za aplikaciju. Lokalizacija služi zbog jednostavnijeg prijevoda aplikacije. Na slici 19 prikazan je primjer lokalizacije. Lokalizacija se sastoji od imena (ključa) i vrijednosti. Svaki ključ ima svoju vrijednost. Promjenom vrijednosti mijenja se prikaz na svim mjestima unutar aplikacije gdje je pozvan ključ.

<pre> </border> <Label Content="{lex:Loc NARUDZBENICE}" Foreground="CorflwoseBlue" </pre>	<table border="1"> <thead> <tr> <th>Name</th><th>Value</th></tr> </thead> <tbody> <tr> <td>NARUDZBENICA</td><td>Narudžbenica</td></tr> </tbody> </table>	Name	Value	NARUDZBENICA	Narudžbenica
Name	Value				
NARUDZBENICA	Narudžbenica				

Slika 19. Primjer lokalizacije teksta

Izvor: Autor

4.2 Prikaz grafičkog dijela desktop aplikacije

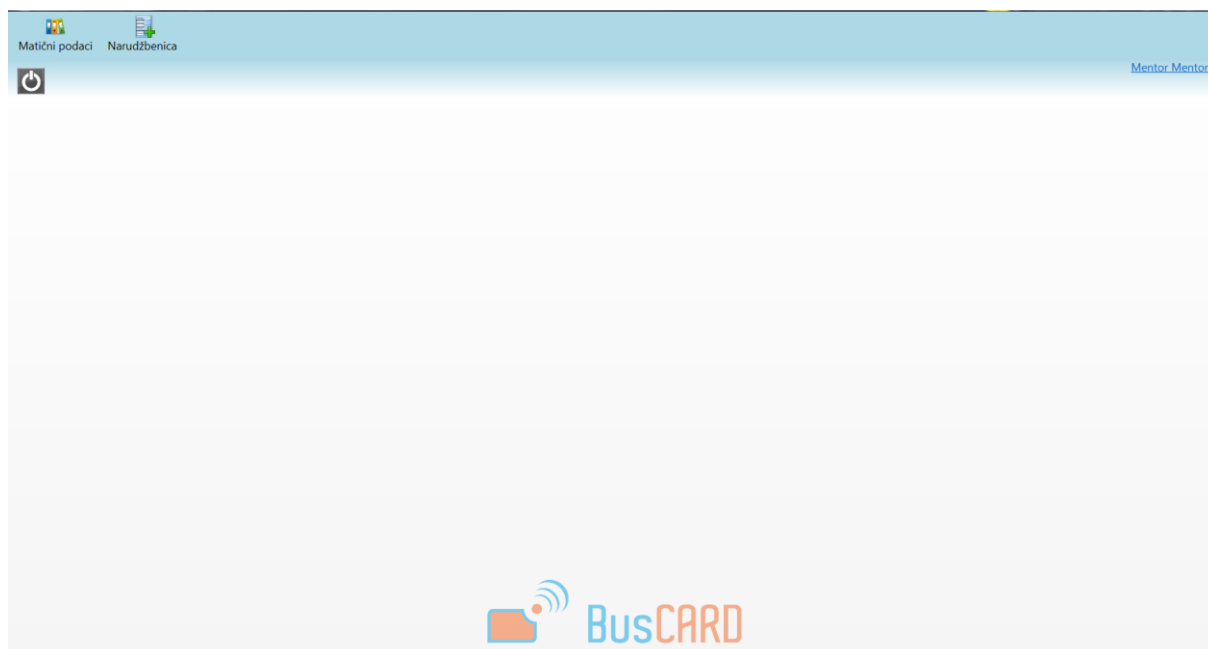
U ovom pod poglavlju prikazani su sučelja desktop aplikacije sustava za kupovinu radničkih mjesečnih karata za pravne osobe. Na slici 19 prikazana je forma za prijavu u BusCARD sustav.



Slika 20. Ekran prijave u "BusCARD" sustav

Izvor: Autor

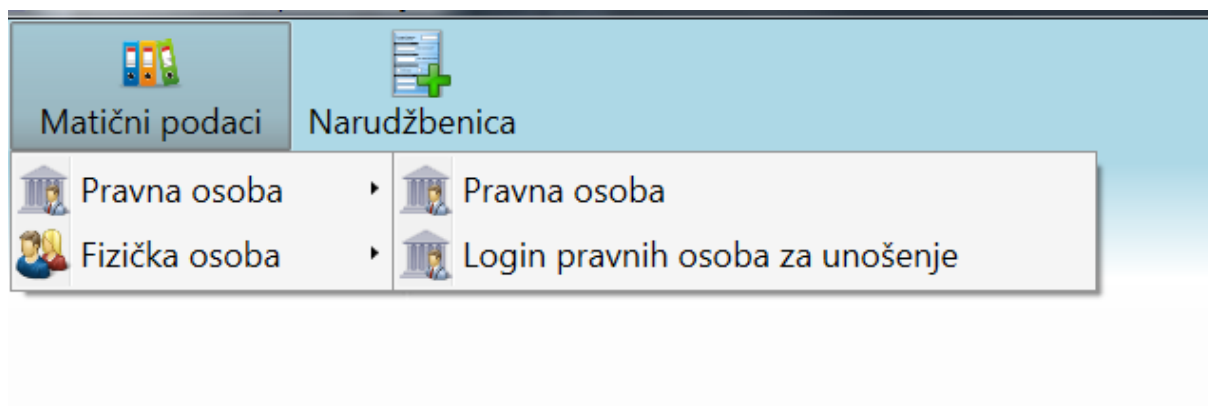
Na slici 20 prikazan je početno sučelje koje se prikazuje nakon uspješne prijave u „BusCARD“ sustav. Na vrhu sučelja nalaze se izbornici „Matični podaci“ i „Narudžbenica“.



Slika 21. Početni ekran "BusCARD" sustava

Izvor: Autor

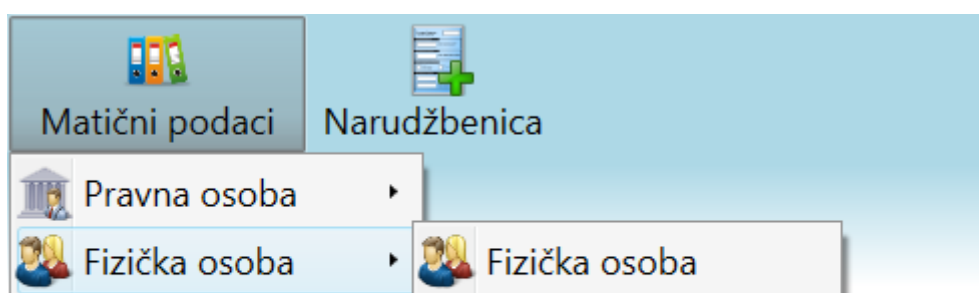
Izbornik „Matični podaci“ sastoji se od dva pod izbornika „Pravna osoba“ i „Fizička osoba“ (slika 21.). Pod izbornik „Pravna osoba“ sastoji se od dva pod izbornika „Pravna osoba“ koji sadrži popis svih pravnih osoba, te formu za unošenje novih pravnih osoba u sustav. Drugi pod izbornik „Login pravnih osoba za unošenje“ prikazuje popis svih pravnih osoba koje koriste web aplikaciju sustava za kupovinu radničkih mjesečnih karata, te formu za unošenje novih pravnih osoba (slika 24).



Slika 22. Izbornik Matičnih podataka i pod izbornik „Pravna osoba“ u "BusCARD" sustavu

Izvor: Autor

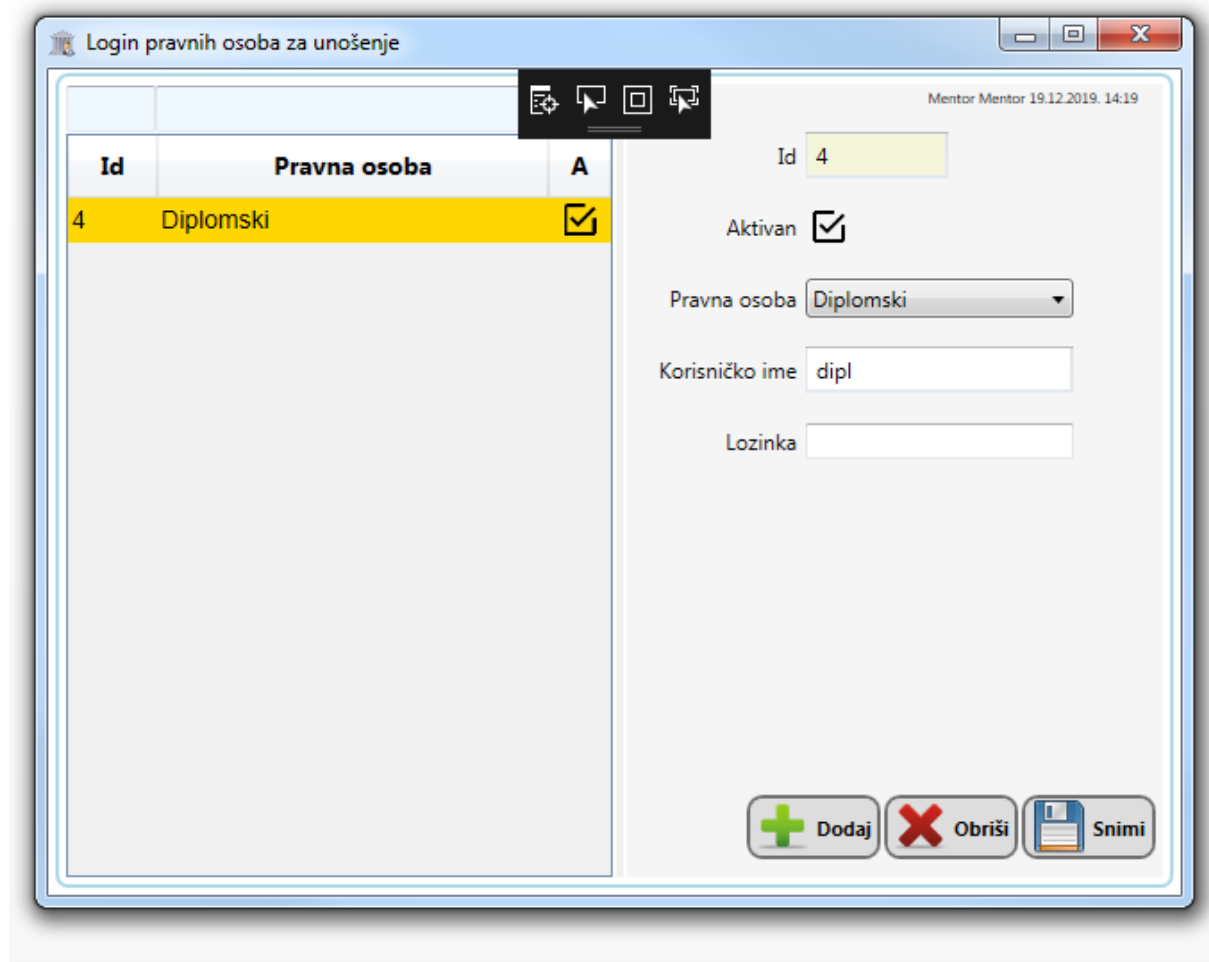
Pod izbornik „Fizička osoba“ sastoji se od pod izbornika „Fizička osoba“ (slika 22.) koja prikazuje popis svih fizičkih osoba, te formu za unošenje istih.



Slika 23. Izbornik "Matični podaci" i pod izbornik "Fizička osoba" u "BusCARD" sustavu

Izvor: Autor

Na slici 24 prikazan je na naziv pravne osobe koja ima kreirane korisničke podatke za prijavu u web aplikaciju. Na lijevoj strani nalazi se popis pravnih osoba, dok su na desnoj strani prikazani podaci pravne osobe s detaljima za prijavu u web aplikaciju.. Lozinka nije prikazana zbog nemogućnosti mijenjanja iste. Ako korisnik izgubi ili zaboravi lozinku potrebno je obrisati korisničke podatke, te kreirati nove.

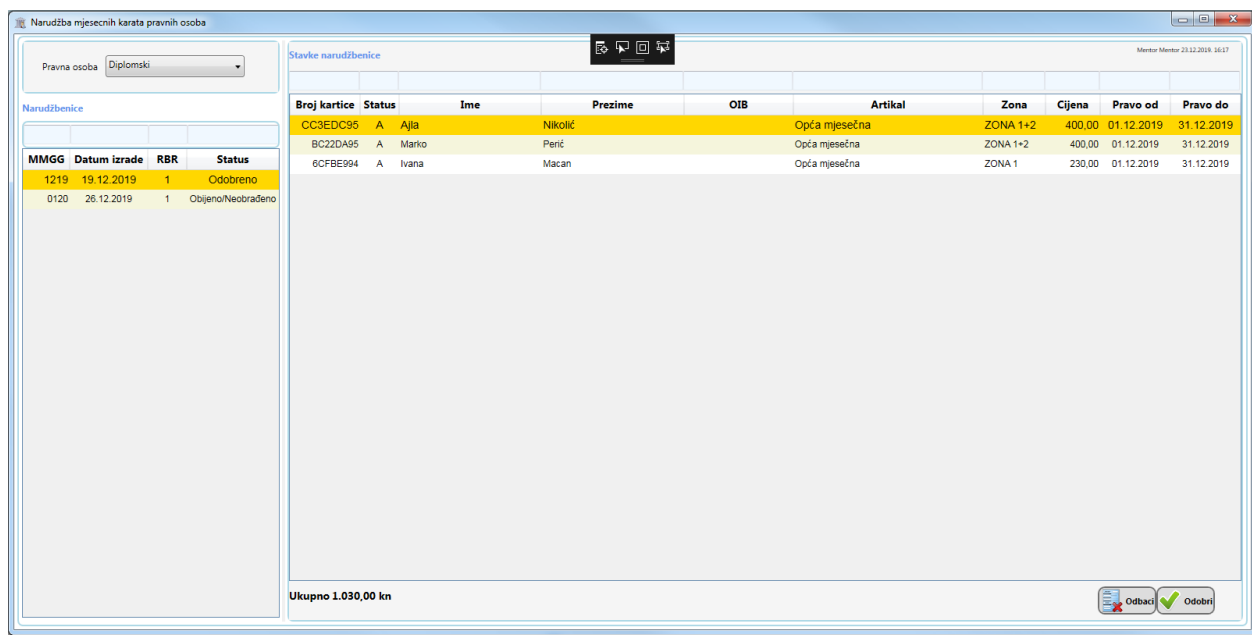


Slika 24. Ekran za kreiranje korisničkih podataka za prijavu u web aplikaciju

Izvor: Autor

Na slici 25 prikazan je popis narudžbenica, te njihove stavke za pojedinu narudžbenicu. Promjenom pravne osobe u padajućem izborniku prikažu se sve kreirane narudžbenice, te odabirom narudžbenice prikažu se sve njezine stavke. Referent pregledom narudžbenice i njezinih stavaka, odobrava ili odbacuje istu. Ako

se narudžbenica odobri ona odlazi u proces naplate i dodjeljivanja prava za svaku stavku narudžbenice.



Narudžba mjesečnih karata pravnih osoba

Pravna osoba: Diplomski

Narudžbenice

MMGG	Datum izrade	RBR	Status
1219	19.12.2019	1	Odobreno
0120	26.12.2019	1	Ocijenjeno/Neobrađeno

Stavke narudžbenice

Broj kartice	Status	Ime	Prezime	OIB	Artikal	Zona	Cijena	Pravo od	Pravo do
CC3EDC95	A	Ajla	Nikolić		Opća mjesečna	ZONA 1+2	400,00	01.12.2019	31.12.2019
BCZZDA95	A	Marko	Perić		Opća mjesečna	ZONA 1+2	400,00	01.12.2019	31.12.2019
6CFBE994	A	Ivana	Macan		Opća mjesečna	ZONA 1	230,00	01.12.2019	31.12.2019

Ukupno 1.030,00 kn

Odbaci Odobri

Slika 25. Ekran za prikaz narudžbenica i njezinih stavaka

Izvor: Autor

ZAKLJUČAK

Cilj ovog diplomskog rada je izrada sustava za kupovinu radničkih mjesečnih karata za pravne osobe. Ideja ovog sustava je smanjivanje redova čekanja na šalterima prijevoznčkih poduzeća, ubrzati i povećati prihod prodaje tvrtke, te takav sustav omogućava uštedu vremena. Ovaj sustav rađen je u tvrtki Penta d.o.o. iz Pule kao dio projekta BusCARD.

Sustav za kupovinu radničkih mjesečnih karata za pravne osobe sastoji se od web i desktop aplikacije. One su u većini izrađene u Microsoftovim tehnologijama ASP.Net Core 2.2 Razor Pages, WPF-a te Entity Frameworka uz korištenje Microsoft Visual Studija i Microsoft SQL Management Studija 2018. Ostali programski jezici koji su korišteni u izradi sustava su HTML, CSS, TSQL i dr.

Funkcionalnosti desktop dijela aplikacije BusCARD potrebne za sustav kupovine radničkih mjesečnih karata sastoji se od forme za pregled i kreiranje korisničkih podataka za prijavu u web aplikaciju, pregled narudžbenica i njenih stavaka, te odobravanje odnosno odbijanje istih.

Funkcionalnosti web aplikacije su pregled narudžbenica i njenih stavaka, kreiranje novih narudžbenica i stavaka, te pregled fizičkih osoba.

Trenutno na tržištu postoje slične aplikacije, ali kako trenutno razvijeni sustav ima puno mjesta za napredak, te ako želi opstati i dalje konkurirati na tržištu potrebno je raditi na poboljšanjima i uvođenjima novih funkcionalnosti.

LITERATURA

Knjige i radovi:

Knok, Ž., Marčec, M. (2015.), WPF i MVVM, Čakovec: Međimursko veleučilište u Čakovcu

Tutorials point (2014) C# Programing object-oriented programming
https://www.tutorialspoint.com/csharp/csharp_tutorial.pdf (Pristupljeno: 10. studenog 2019).

Tutorials point (2018) WPF https://www.tutorialspoint.com/wpf/wpf_tutorial.pdf
(Pristupljeno: 20. studenog 2019).

Internet stranice:

„Automatska naplata“. 2019. BusCARD. Dostupno na:
<https://www.buscard.biz/hr/sustavi/automatska-naplata/> (Pristupljeno: 1. studenog 2019)

„ASP.NET Core - Middleware“. 2015. Tutorialsteacher. Dostupno na:
<https://www.tutorialsteacher.com/core/aspnet-core-middleware> (Pristupljeno: 15. prosinca 2019)

„ASP.NET Core – Startup Class“. 2015. Tutorialsteacher. Dostupno na:
<https://www.tutorialsteacher.com/core/aspnet-core-startup> (Pristupljeno: 15. prosinca 2019)

„Common Language Runtime (CLR) in C#“. GeeksforGeeks. Dostupno na:
<https://www.geeksforgeeks.org/common-language-runtime-clr-in-c-sharp/>
(Pristupljeno: 12. studenog 2019)

„C# Properties (GET, SET)“. 2018. Tutkane. Dostupno na:
<https://www.tutlane.com/tutorial/csharp/csharp-properties-get-set> (Pristupljeno: 2. prosinca 2019)

„Dependency injection (inversion of control)“. 2013. Učim programiranje. Dostupno na: <http://www.ucim-programiranje.com/2013/02/dependency-injection-inversion-of-control/> (Pristupljeno: 30. studenog 2019)

„Introduction to ASP.NET Core“. 2019. Microsoft. Dostupno na: <https://docs.microsoft.com/en-us/aspnet/core/index?view=aspnetcore-3.0#recommended-learning-path> (Pristupljeno: 12. studenog 2019)

„Learn Razor Pages“. 2019. Learn Razor Pages. Dostupno na: <https://www.learnrazorpages.com/> (Pristupljeno: 30. studenog 2019)

Likness, J. (2014) „Model-View-ViewModel (MVVM) Explained“, Dostupno na: <https://www.wintellect.com/model-view-viewmodel-mvvm-explained/> (Pristupljeno: 26. studenog. 2019)

Rouse, M. (2019) „Microsoft SQL Server“ Dostupno na: <https://searchsqlserver.techtarget.com/definition/SQL-Server> (Pristupljeno: 2. prosinca 2019)

Sharma, N. (2019) „MVVM Design Pattern for iOS and Android“, Dostupno na: <https://insights.codewave.com/mvvm-design-pattern-for-ios-and-android/> (Pristupljeno: 29. studenog. 2019)

„WPF Architecture“. 2017. Microsoft. Dostupno na: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/wpf-architecture> (Pristupljeno: 15. studenog 2019)

„WPF - Overview“. Tutorialspoint. Dostupno na: https://www.tutorialspoint.com/wpf/wpf_overview.htm (Pristupljeno: 14. studenog. 2019)

„WPF-XAML Overview“. Tutorialspoint. Dostupno na: https://www.tutorialspoint.com/wpf/wpf_xaml_overview.htm (Pristupljeno: 25. studenog 2019)

Popis slika

Slika 1. Arhitektura Windows Presentation Foundation-a.....	4
Slika 2. Arhitektura MVVM.....	6
Slika 3. Grafički prikaz tablica.....	27
Slika 4. Datoteke web aplikacije	29
Slika 5. Primjer klase za tablicu dbo.NarudzbenicaKarataStavka.....	30
Slika 6. Prikaz klase „ApplicationDbContext“	32
Slika 7. Prikaz programskog koda za dohvaćanje fizičkih osoba.....	33
Slika 8. Prikaz programskog koda za prikaz fizičkih osoba na ekranu.....	34
Slika 9. Znakovni niz za spajanje aplikacije na bazu podataka	36
Slika 10. Sadržaj klase Program.cs	37
Slika 11. Startup klasa.....	38
Slika 12. Forma za prijavu u web aplikaciju	39
Slika 13. Web stranica sa pregledom narudžbenica i stavki narudžbenice	40
Slika 14. Web stranica za kreiranje narudžbenice i stavki narudžbenice.....	41
Slika 15. Web stranica sa popisom fizičkih osoba	42
Slika 16. Popis projekata u BusCARD sustavu.....	43
Slika 17. Prikaz sadržaja mape NarudzbenicaKarata.....	44
Slika 18. Funkcija za dohvaćanje narudžbenica karata pojedine pravne osobe.....	45
Slika 19. Primjer lokalizacije teksta.....	45
Slika 20. Ekran prijave u "BusCARD" sustav.....	46
Slika 21. Početni ekran "BusCARD" sustava.....	46
Slika 22. Izbornik Matičnih podataka i pod izbornik „Pravna osoba“ u "BusCARD" sustavu	47
Slika 23. Izbornik "Matični podaci" i pod izbornik "Fizička osoba" u "BusCARD" sustavu	47
Slika 24. Ekran za kreiranje korisničkih podataka za prijavu u web aplikaciju	48
Slika 25. Ekran za prikaz narudžbenica i njezinih stavaka	49

Popis tablica

Tablica 1. Popis atributa u tablici dbo.NarudzbenicaKarata	8
Tablica 2. Popis atributa u tablici dbo.NarudzbenicaKarataStavka.....	9
Tablica 3. Popis atributa u tablici dbo.FizickaOsoba	11
Tablica 4. Popis atributa u tablici dbo.PravnaOsoba	13
Tablica 5. Popis atributa u tablici dbo.Kartica	15
Tablica 6. Popis atributa u tablici dbo.KarticaProdaja.....	16
Tablica 7. Popis atributa u tablici dbo.Artikal	19
Tablica 8. Popis atributa u tablici dbo.Relacija	20
Tablica 9. Popis atributa u tablici dbo.Radnik	21
Tablica 10. Popis atributa u tablici dbo.PravnaOsobaLogin	24
Tablica 11. Popis atributa u tablici dbo.KategorijaPutnika	25
Tablica 12. Popis tablica u tablici dbo.Mjesto	26

Sažetak

Cilj ovog rada je prikazati i objasniti na koji način je moguće izraditi narudžbenicu karata i njenih stavaka za kupovinu radničkih mjesečnih karata za pravne osobe. U radu su prikazane i ukratko objašnjene tehnologije u kojemu je sustav izrađen. Također u radu su prikazane i objašnjene tablice i atributi tablica u bazi podataka. Nakon toga prikazani su isječci programskog koda i sučelja desktop i web aplikacije.

Ključne riječi: web aplikacija, desktop aplikacija, ASP.Net Core 2.2 Razor Pages, WPF, MVVM, EF Core, Microsoft Visual Studio, Microsoft SQL Management Studio 2018

Abstract

The aim of this paper is to present and explain how is possible to create ticket order and it's items for buying workers monthly cards by legal entities. In this paper are shown and briefly explained the technologies in which is system made. The tables and attributes of tables in the database are also presented in this paper. The code snippets and interfaces of the desktop and web application are displayed.

Keywords: web application, desktop application, ASP.Net Core 2.2 Razor Pages, WPF, MVVM, EF Core, Microsoft Visual Studio, Microsoft SQL Management Studio 2018