

# Razvoj programske podrške za uzgoj teladi i junadi

---

Lukić, Davor

Undergraduate thesis / Završni rad

2019

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:812183>

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

*Download date / Datum preuzimanja:* **2024-08-12**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni studij**

**RAZVOJ PROGRAMSKE PODRŠKE ZA UZGOJ  
TELADI I JUNADI**

**Završni rad**

**Davor Lukić**

**Osijek, 2019.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 01.07.2019.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada**

<b>Ime i prezime studenta:</b>	Davor Lukić
<b>Studij, smjer:</b>	Preddiplomski sveučilišni studij Računarstvo
<b>Mat. br. studenta, godina upisa:</b>	R 3859, 20.09.2018.
<b>OIB studenta:</b>	99663618223
<b>Mentor:</b>	Doc.dr.sc. Ivica Lukić
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Razvoj programske podrške za uzgoj teladi i junadi
<b>Znanstvena grana rada:</b>	<b>Informacijski sustavi (zn. polje računarstvo)</b>
<b>Predložena ocjena završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	01.07.2019.
<b>Datum potvrde ocjene Odbora:</b>	10.07.2019.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 10.07.2019.

**Ime i prezime studenta:**

Davor Lukić

**Studij:**

Prediplomski sveučilišni studij Računarstvo

**Mat. br. studenta, godina upisa:**

R 3859, 20.09.2018.

**Ephorus podudaranje [%]:**

18%

Ovom izjavom izjavljujem da je rad pod nazivom: **Razvoj programske podrške za uzgoj teladi i junadi**

izrađen pod vodstvom mentora Doc.dr.sc. Ivica Lukić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

1. UVOD .....	1
1.1. Zadatak završnog rada .....	2
2. BAZA PODATAKA .....	3
2.1. Utvrđivanje i analiza zahtjeva .....	4
2.2. Projektiranje.....	4
2.2.1. Konceptualno oblikovanje .....	4
2.2.2. Logičko oblikovanje .....	7
2.2.3. Fizičko oblikovanje.....	10
3. PROGRAMSKA PODRŠKA ZA VOĐENJE FARME .....	19
3.1. Unos podataka .....	20
3.1.1. Unos goveda.....	20
3.1.2. Vaganje .....	21
3.1.3. Liječenje.....	21
3.1.4. Lijekovi .....	23
3.1.5. Prodaja .....	23
3.1.6. Kupac .....	25
3.2. Pregled podataka.....	26
3.2.1. Detalji o govedu.....	26
3.1.3. Pregled prodaja .....	27
4. ZAKLJUČAK .....	30
LITERATURA.....	31
SAŽETAK.....	32
ABSTRACT .....	33
ŽIVOTOPIS .....	34
PRILOZI.....	35

# 1. UVOD

Kako su se uzgojne metode teladi i junadi osuvremenile i unaprijedile, u današnje vrijeme stvorila se i potreba za digitalizacijom i praćenjem uzgojnih ciljeva, prijašnjih liječenja, prodaje i prihoda tovnih farmi. Za vlasnika jednog takvog uzgoja od velike je važnosti imati sve prijašnje i sadašnje podatke vezane za tov na jednom mjestu, kako bi u bilo kojem trenutku imao uvid u stanje svog poslovanja. Time se otvara mogućnost razvoja programske podrške za uzgoj teladi i junadi, s ciljem unaprjeđenja pojedinog gospodarstva i naposljetku, doprinosa proizvodnoj industriji.

Tema završnog rada „Razvoj programske podrške za uzgoj teladi i junadi“ obuhvaća izradu programske podrške za evidenciju različitih podataka. Evidencija sadrži podatke o govedima (ime, datum rođenja, podaci o roditeljima, ušni broj markice, spol, masa i pasmina), podatke o dolasku novih goveda na farmu (datumu ulaza, kupovna cijena, masa pri početku prijevoza, masa nakon dolaska i kalo), podatke o kupcima (ime, prezime, telefonski broj, adresa, prodajna cijena i masa goveda) i podatke o liječenju goveda (ime lijeka, datum aplikacije lijeka, količina lijekova, zalihe lijekova, stanje lijekova i trajanje karence). Korisniku su ujedno dostupni podaci o junadi i teladi koje se trenutno nalaze na farmi te prikaz svih dosadašnjih ostvarenih prodaja. Fizička baza podataka farme ostvarena je na Microsoft SQL Server 2017, dok je za rad s bazom korišten LINQPad.

Zbog jednostavnijeg praćenja razvoja programske podrške, rad je podijeljen na dva poglavlja. Prvo poglavlje obuhvaća teorijski pristup izradi baze podataka i primjenu teorije na razvoju i implementaciji baze podataka za farmu. U njemu je detaljno objašnjen razvoj i struktura baze podataka, kako utvrditi i analizirati zahtjeve korisnika za izradu odabrane specifikacije te načela i primjena konceptualnog, logičkog i fizičkog oblikovanja. Cilj prvog dijela rada je približavanje zahtjevima korisnika, kako bi im personaliziranim pristupom informacijama kroz bazu podataka unaprijedili svakodnevno poslovanje. Drugo poglavlje obuhvaća programsku podršku za razvoj farme i izgled sučelja programa, koji prati zahtjeve korisnika. U drugom poglavlju izrađeno je rješenje kojim korisnik na jednostavan način može koristiti mogućnosti stvorene baze podataka. Za izradu sučelja koje omogućuje pristup korisnicima i rad s bazom podataka farme korišten je Microsoft Visual Studio Community 2015 s programskim jezikom C#.

## **1.1. Zadatak završnog rada**

Za zadatak završnog rada odabran je razvoj programske podrške za vođenje knjiga sa podacima o govedima, koji su potrebni u svakodnevnom i neometanom funkcioniranju uzgoja teladi i junadi na farmi. Vođenje knjiga obuhvaća evidenciju o trenutnim govedima na farmi, o njihovom porijeklu i boravku u tovu, prodanim govedima, detaljima izvršene prodaje, zaradi kroz prodaju, provedenim liječenjima i napretku pri uzgoju. Administrator ima mogućnost dodavanja i uređivanja svih podataka, dok se ostalim djelatnicima daje ograničeni pristup ovisno o radnom mjestu.

## 2. BAZA PODATAKA

Jedna od definicija baze podataka, prema James Martin-u, je: „Baza podataka je skup istovrsnih podataka s višestrukom namjenom. Korisnik nije zainteresiran za sve vrste podataka u bazi, već samo za one koji su mu potrebni u njegovom poslu. Korisnik može imati uvid u samo jednu, njemu potrebnu datoteku koja ima uvijek istu i to vrlo jednostavnu strukturu, iako je u biti izvedena iz mnogo kompleksnije strukture podataka. Različiti korisnici uzimaju u obzir različite datoteke izvedene iz iste baze podataka. Dakle, iako je baza podataka zajednička većem broju korisnika, različiti korisnici je različito shvaćaju.“ [1]

Bazu podataka možemo definirati kao kolekciju podataka koji su međusobno povezani, te kako se ti podaci dijele između više različitih aplikacija i korisnika uz uvjet da postoji kontrolirani i jedinstveni pristup njihovom unosu, izmjeni, brisanju i dohvaćanju. Podaci se u bazi podataka organiziraju prema određenom modelu podataka, skupu pravila koja određuju kako sve može izgledati logička struktura baze podataka. Takav model čini osnovu za projektiranje i implementiranje baze. Najčešći modeli su: hijerarhijski, mrežni, relacijski, relacijski s objektno orijentiranim proširenjima i objektni model. Za potrebe izrade ovog završnog rada korišten je relacijski model.

Prema [1], relacijski model je razradio E.F.Codd, te je jednostavan u razumijevanju i korištenju. Matematički temelji (teorija skupova) omogućavaju precizno definiranje. Spada u formalne modele jer ima skup koncepata za opis baze podataka, skup operatera za izvršavanje operacija nad podacima, skup pravila integriteta podataka. Njegovi osnovni ciljevi su nezavisnost podataka, smanjiti zalihost podataka, omogućiti razvoj jezika za obradu podataka, dati model za opis i obradu jednostavnih i kompleksnih podataka.

Prema [2], uporabom baze podataka nastoje se postići sljedeći ciljevi: fizička nezavisnost podataka, logička nezavisnost podataka, fleksibilnost pristupa podacima, istovremeni pristup do podataka, čuvanje integriteta, mogućnost oporavka nakon kvara, zaštita od neovlaštenog korištenja, zadovoljavajuća brzina pristupa, mogućnost podešavanja i kontrole.

Razvojni ciklus baze podataka može se podijeliti u pet aktivnosti: utvrđivanje i analiza zahtjeva, projektiranje, implementacija, testiranje i održavanje.



## **2.1. Utvrđivanje i analiza zahtjeva**

U razgovoru s voditeljem farme, kao rezultat utvrđivanja i analize zahtjeva dobivena je sljedeća specifikacija:

Farma za uzgoj teladi i junadi treba voditi evidenciju s podacima o govedima kao što su ime, datum rođenja, roditelji, ušni broj markice, spol, masa i pasmina. Pri dolasku goveda na farmu bilježe se podaci o datumu ulaza goveda na farmu, njegovoj cijeni, masi pri početku prijevoza i masi nakon dolaska kako bi se mogao izračunati kilo koji nastaje kao posljedica prijevoza. Također, potrebno je omogućiti kontrolna vaganja kako bi se mogao pratiti napredak pri uzgoju. Kako se uz uzgoj bave i prodajom goveda farma treba imati evidenciju o kupcima s imenom, prezimenom, telefonskim brojem i adresom. Pri prodaji dogovara se cijena goveda, te se bilježi masa goveda. Treba omogućiti pregled svih prodaja i prodaja po pojedinim kupcima, te prikazati ostvareni profit na osnovu potrošenih sredstava za nabavu i uzgoj goveda i ostvarene cijene pri prodaji.

Uz uzgoj farma treba voditi i evidenciju liječenja s opisom terapije i primijenjenim lijekovima kako bi se znalo nalazi li se govedo u karenci (zbog primjene lijeka ne smije biti poslano na klanje dok ne istekne njegovo djelovanje), koliko i kojih lijekova ima na zalihi, te detalje o lijeku kao što je njegov naziv, kratki opis i broj dana djelovanja lijeka.

Korisniku treba omogućiti prikaz goveda, teladi ili junadi koja se nalaze na farmi te prikaz ostvarenih prodaja.

## **2.2. Projektiranje**

Kako je projektiranje složena aktivnost, kojom se u skladu sa specifikacijom oblikuje građa baze, obično se provodi u tri faze: konceptualno, logičko i fizičko oblikovanje baze podataka.

### **2.2.1. Konceptualno oblikovanje**

Konceptualno oblikovanje daje konceptualnu shemu cijele baze sastavljenu od entiteta, atributa i veza, prema [2].

Prema [3], entitet je stvar, objekt, od značaja, realan ili imaginaran za koji je potrebno poznavati i čuvati određene informacije. Atribut je svaki detalj kojim se određuje, identificira, klasificira,

kvantificira ili izražava stanje entiteta, odnosno atribut predstavlja opis entiteta. Veza predstavlja povezivanje dvaju ili više entiteta u novi entitet.

Iz specifikacije dobivenom pri utvrđivanju i analizi zahtjeva otkriveni su sljedeći entiteti, veze i atributi:

**Entiteti:** GOVEDO, PASMINA, VAGANJE, PRODAJA, PRODAJA STAVKA, KUPAC, MJESTO, TERAPIJA, LIJEK.

**Veze:** JE MAJKA i JE OTAC (povratne veze) između GOVEDO i GOVEDO, PRIPADA između GOVEDO i PASMINA (veza N:1), KUPUJE između KUPAC i PRODAJA(veza 1:N), JE U između KUPAC i MJESTO(veza N:1), ZA PRODAJU između GOVEDO i PRODAJA STAVKA (veza 1:1), DIO između PRODAJA i PRODAJA STAVKA (veza 1:N), VAŽE između VAGANJE i GOVEDO(veza N:1), PRIMA između GOVEDO i TERAPIJA(veza 1:N), KORIŠTEN između LIJEK i TERAPIJA (veza M:N).

**Atributi:** UŠNI BROJ MARKICE, IME, SPOL, DATUM ROĐENJA, PASMINA, ULAZNA KILAŽA, ULAZNA CIJENA, DOLAZNA KILAŽA, IME, PREZIME, TELEFONSKI BROJ, ULICA I KUĆNI BROJ, MJESTO, PRODAJNA CIJENA, PRODAJNA KILAŽA, OPIS TERAPIJE, NAZIV LIJEKA, OPIS LIJEKA, KARENCA, ZALIHA.

Dodjela atributa entitetima (osim onih iz specifikacije dodani su i neki drugi, nužni atributi).

Entitet GOVEDO ima sljedeće attribute: UŠNI BROJ MARKICE, IME, SPOL, DATUM ROĐENJA, DATUM ULAZA, ULAZNA KILAŽA, ULAZNA CIJENA, DOLAZNA KILAŽA.

Entitet PASMINA ima sljedeće attribute: ID PASMINE, NAZIV.

Entitet VAGANJE ima sljedeće attribute: ID, DATUM VAGANJA, KILAŽA.

Entitet PRODAJA ima sljedeće attribute: ID PRODAJE, DATUM PRODAJE.

Entitet PRODAJA STAVKA ima sljedeće attribute: ID PSTAVKE, PRODAJNA CIJENA, PRODAJNA KILAŽA.

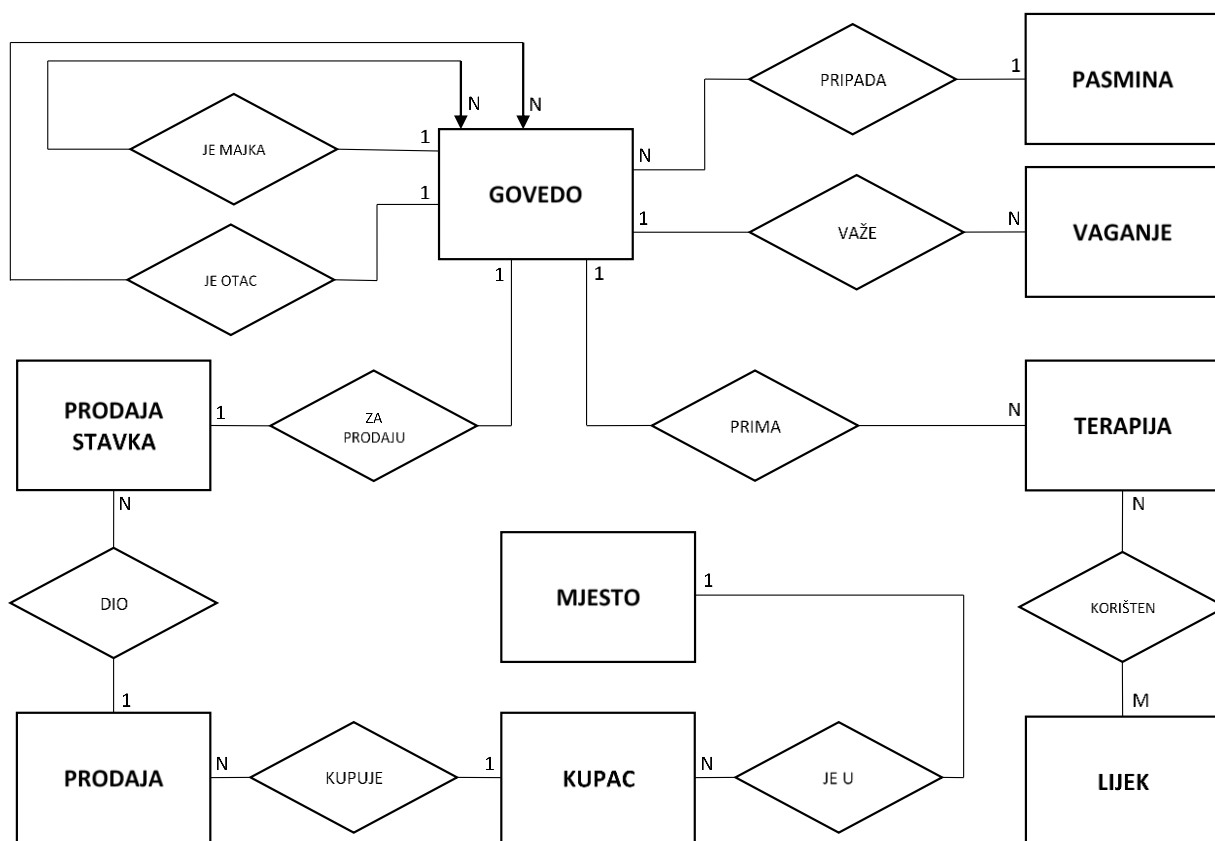
Entitet KUPAC ima sljedeće attribute: OIB, IME, PREZIME, ULICA I KUĆNI BROJ, TELEFONSKI BROJ.

Entitet MJESTO ima sljedeće atribute: POŠTANSKI BROJ, NAZIV.

Entitet TERAPIJA ima sljedeće atribute: ID TERAPIJE, DATUM TERAPIJE, OPIS, NASLOV.

Entitet LIJEK ima sljedeće atribute: ID LIJEKA, NAZIV, KARENCA, DETALJI, STANJE.

Prema entitetima, atributima i vezama dobiven je E/R dijagram (reducirani Chenov dijagram) koji je prikazan na slici 2.1., te popratni tekst prikazan tablicom 2.1. Na slici 2.1. su prikazani entiteti i veze između njih, te odgovarajuće funkcionalnosti veza. U tablici 2.1. su navedeni pripadni atributi odgovarajućih entiteta i veza gdje podvučeni atributi predstavljaju kandidate za ključ.



**Slika 2.1.** E/R dijagram za bazu podataka farme

**Tablica 2.1.** Pripadni atributi entiteta za dijagram sa slike 1.1.

<b>Entitet</b>	<b>Atributi</b>
<b>GOVEDO</b>	<u>UŠNI BROJ MARKICE</u> , IME, SPOL, DATUM ROĐENJA, DATUM ULAZA, ULAZNA KILAŽA, ULAZNA CIJENA, DOLAZNA KILAŽA
<b>PASMINA</b>	<u>ID PASMINE</u> , NAZIV
<b>VAGANJE</b>	<u>ID</u> , DATUM VAGANJA, KILAŽA
<b>PRODAJA</b>	<u>ID PRODAJE</u> , DATUM PRODAJE
<b>PRODAJA STAVKA</b>	<u>ID PSTAVKE</u> , PRODAJNA CIJENA, PRODAJNA KILAŽA
<b>KUPAC</b>	<u>OIB</u> , IME, PREZIME, ULICA I KUĆNI BROJ, TELEFONSKI BROJ
<b>MJESTO</b>	<u>POŠTANSKI BROJ</u> , NAZIV
<b>TERAPIJA</b>	<u>ID TERAPIJE</u> , DATUM TERAPIJE, OPIS, NAZIV
<b>LIJEK</b>	<u>ID LIJEKA</u> , NAZIV, KARENCA, DETALJI, STANJE

### 2.2.2. Logičko oblikovanje

Logičko oblikovanje daje logičku shemu, koja je u slučaju relacijskog modela sastavljena od relacija (tablica), prema [2].

Prema [1], logičko modeliranje podataka predstavlja izradu logičkog ili implementacijskog modela podataka, odnosno pretvaranje konceptualnog modela (gdje su podaci neovisni) izvedenog u obliku modela entiteti-veze u logičku shemu baze podataka.

U relacijskom modelu relacija predstavlja tablicu sa stupcima i redovima koju čine n-torke (redovi tablice), atribut imenovani stupac relacije, domena skup dozvoljenih vrijednosti za jedan ili više atributa. Pri preslikavanju E/R modela u relacijski model entitet i veza se preslikavaju u relaciju, dok atribut i domena ostaju isti.

Prevođenjem E/R modela (Slika 2.1. i Tablica 2.1.) u relacijski model dobivena je sljedeća relacijska shema baze podataka farme, odnosno skup relacija prikazanih u tablici 2.2. Primarni ključevi (kandidat ključevi koji su odabrani da jedinstveno odrede n-torke unutar relacije) u tablici su podcrtani, dok su strani ključevi (atribut ili skup atributa unutar jedne relacije koji odgovara kandidat ključu neke (moguće i iste) relacije) označeni kurzivom.

**Tablica 2.2.** Relacijska shema baze podataka farme

Relacija	Shema relacije
<b>MJESTO</b>	<u>POŠTANSKI BROJ</u> , NAZIV
<b>KUPAC</b>	<u>OIB</u> , IME, PREZIME, ULICA I KUĆNI BROJ, TELEFONSKI BROJ, <i>MJESTO</i>
<b>PASMINA</b>	<u>ID PASMINE</u> , NAZIV
<b>GOVEDO</b>	<u>UŠNI BROJ MARKICE</u> , IME, SPOL, DATUM ROĐENJA, DATUM ULAZA, ULAZNA KILAŽA, ULAZNA CIJENA, DOLAZNA KILAŽA, <i>PASMINA</i> , <i>OTAC</i> , <i>MAJKA</i>
<b>VAGANJE</b>	<u>ID</u> , DATUM VAGANJA, KILAŽA, <i>GOVEDO</i>
<b>PRODAJA</b>	<u>ID PRODAJE</u> , DATUM PRODAJE, <i>KUPAC</i>
<b>PRODAJA STAVKA</b>	<u>ID PSTAVKE</u> , PRODAJNA CIJENA, PRODAJNA KILAŽA, <i>PRODAJA</i> , <i>GOVEDO</i>
<b>LIJEK</b>	<u>ID LIJEKA</u> , NAZIV, DETALJI, KARENCA, STANJE
<b>TERAPIJA</b>	<u>ID TERAPIJE</u> , DATUM TERAPIJE, NASLOV, OPIS, <i>GOVEDO</i>
<b>TERAPIJA LIJEK</b>	<i>TERAPIJA</i> , <u>LIJEK</u> , DOZA

Pri pretvorbi E/R modela u relacijski dodana je nova relacija TERAPIJA LIJEK radi veze M:N KORISTI gdje primarni ključevi entiteta terapija i lijek zajedno čine složeni primarni ključ te nove relacije. Dodan je i atribut doza koji predstavlja koliko je lijeka potrošeno u terapiji.

Rječnik podataka za bazu podataka farme, odnosno za relacijsku shemu iz tablice 2.2., je dan u tablici 2.3.

**Tablica 2.3.** Rječnik podataka za bazu podataka farme

<b>Ime atributa</b>	<b>Tip</b>	<b>Opis</b>
<b>POŠTANSKI BROJ</b>	Niz od točno 5 znakova	Šifra koja jednoznačno određuje mjesto
<b>NAZIV</b>	Niz znakova	Naziv mjesta
<b>OIB</b>	Niz od točno 11 znakova	Šifra koja jednoznačno određuje kupca
<b>IME</b>	Niz znakova	Ime kupca
<b>PREZIME</b>	Niz znakova	Prezime kupca
<b>ULICA I KUĆNI BROJ</b>	Niz znakova	Adresa kupca
<b>TELEFONSKI BROJ</b>	Niz znakova	Telefonski broj kupca
<b>ID PASMINE</b>	Niz od točno 2 znaka	Šifra koja jednoznačno određuje pasminu
<b>NAZIV</b>	Niz znakova	Naziv pasmine goveda
<b>UŠNI BROJ MARKICE</b>	Niz od točno 12 znakova	Šifra koja jednoznačno određuje govedo
<b>IME</b>	Niz znakova	Ime goveda
<b>SPOL</b>	1 znak	Spol goveda
<b>DATUM ROĐENJA</b>	Datum	Datum rođenja goveda
<b>DATUM ULAZA</b>	Datum	Datum kada je govedo ušlo na farmu
<b>ULAZNA KILAŽA</b>	Cjelobrojni broj	Masa goveda pri početku prijevoza
<b>ULAZNA CIJENA</b>	Decimalni broj	Kupovna cijena goveda po kilogramu
<b>DOLAZNA KILAŽA</b>	Cjelobrojni broj	Masa goveda nakon prijevoza
<b>ID</b>	Cjelobrojni broj	Šifra koja jednoznačno određuje vaganje
<b>DATUM VAGANJA</b>	Datum	Datum vaganja goveda
<b>KILAŽA</b>	Cjelobrojni broj	Masa goveda na dan vaganja
<b>ID PRODAJE</b>	Niz od točno 5 znakova	Šifra koja jednoznačno određuje prodaju
<b>DATUM PRODAJE</b>	Datum	Datum stvaranja prodaje
<b>ID PSTAVKE</b>	Cjelobrojni broj	Šifra koja jednoznačno određuje stavku prodaje
<b>PRODAJNA CIJENA</b>	Decimalni broj	Prodajna cijena goveda po kilogramu
<b>PRODAJNA KILAŽA</b>	Cjelobrojni broj	Masa goveda pri prodaji
<b>NAZIV</b>	Niz znakova	Naziv lijeka
<b>DETALJI</b>	Niz znakova	Dodatni opis lijeka
<b>KARENCA</b>	Cjelobrojni broj	Broj dana karence nakon primitka lijeka
<b>STANJE</b>	Cjelobrojni broj	Količina lijeka u skladištu
<b>ID TERAPIJE</b>	Cjelobrojni broj	Šifra koja jednoznačno određuje terapiju
<b>DATUM TERAPIJE</b>	Datum	Datum kada je primijenjena terapija
<b>NASLOV</b>	Niz znakova	Naslov terapije
<b>OPIS</b>	Niz znakova	Dodatni opis terapije
<b>DOZA</b>	Cjelobrojni broj	Količina utrošenog lijeka u terapiji

### 2.2.3. Fizičko oblikovanje

Fizičko oblikovanje daje fizičku shemu baze, odnosno opis njezine fizičke građe. Prema [2], fizička shema je tekst sastavljen od naredbi u SQL-u (jezik za relacijske baze koji služi za definiranje podataka, manipuliranje i pretraživanje) ili nekom drugom jeziku koji razumije DBMS (engleski Data Base Management System, sustav za upravljanje bazom podataka).

Fizička baza podataka farme ostvorena je na Microsoft SQL Server 2017, dok je za rad s bazom korišten LINQPad.

Prema [4], sintaksa za stvaranje tablice u Transact-SQL je:

CREATE TABLE

```
{ database_name.schema_name.table_name. | schema_name.table_name | table_name }  
( { <column_definition> } [ ,...n ] )
```

[ ; ]

U procesu stvaranja tablica pazilo se na očuvanje integriteta baze, odnosno pazilo se na korektnost i konzistentnost podataka. To je omogućeno korištenjem ograničenja kojima se uspostavlja integritet domene (provjera vrijednosti atributa), ograničenja za čuvanje integriteta u relaciji (primarni ključevi) i ograničenja kojima se čuva referencijalni integritet (strani ključevi).

Fizička shema (SQL naredbe za implementaciju baze podataka) baze podataka farme prikazana je na slikama 2.2., 2.3., 2.4. i 2.5.

```
CREATE TABLE mjesto (  
    postanski_broj CHAR(5),  
    naziv VARCHAR(30),  
    CONSTRAINT mjesto_pk PRIMARY KEY(postanski_broj));  
  
CREATE TABLE kupac (  
    oib CHAR(11),  
    ime VARCHAR(30) NOT NULL,  
    prezime VARCHAR(30) NOT NULL,  
    telefonski_broj VARCHAR(20),  
    ulica_i_kucni_broj VARCHAR(50),  
    mjesto_kupca CHAR(5),  
    CONSTRAINT kupac_pk PRIMARY KEY(oib),  
    CONSTRAINT mjesto_kupca_fk FOREIGN KEY(mjesto_kupca)  
    REFERENCES mjesto(postanski_broj));
```

**Slika 2.2.** Stvaranje tablica mjesto i kupac

```

CREATE TABLE pasmina (
  id_pasmine CHAR(2),
  naziv VARCHAR(30) NOT NULL,
  CONSTRAINT pasmina_pk PRIMARY KEY(id_pasmine));

CREATE TABLE govedo (
  usni_broj_markice CHAR(12),
  ime VARCHAR(30),
  spol CHAR(1) NOT NULL,
  datum_rodjenja DATE,
  datum_ulaza DATE NOT NULL,
  ulazna_kilaza SMALLINT NOT NULL CHECK (ulazna_kilaza > 0),
  ulazna_cijena DECIMAL(5,2) NOT NULL CHECK (ulazna_cijena > 0),
  dolazna_kilaza SMALLINT CHECK (dolazna_kilaza > 0),
  pasmina CHAR(2) NOT NULL,
  majka CHAR(12),
  otac CHAR(12),
  CONSTRAINT govedo_pk PRIMARY KEY(usni_broj_markice),
  CONSTRAINT pasmina_fk FOREIGN KEY(pasmina)
    REFERENCES pasmina(id_pasmine),
  CONSTRAINT otac_fk FOREIGN KEY(otac)
    REFERENCES govedo(usni_broj_markice),
  CONSTRAINT majka_fk FOREIGN KEY(majka)
    REFERENCES govedo(usni_broj_markice),
  CONSTRAINT chk_spol CHECK (spol IN ('M','F')));

```

**Slika 2.3.** Stvaranje tablica pasmina i govedo

```

CREATE TABLE vaganje (
  id_vaganja INT IDENTITY(1,1),
  datum_vaganja DATE NOT NULL,
  kilaza SMALLINT NOT NULL CHECK (kilaza > 0),
  govedo CHAR(12) NOT NULL,
  CONSTRAINT id_vaganja_pk PRIMARY KEY(id_vaganja),
  CONSTRAINT govedo_fk FOREIGN KEY(govedo)
    REFERENCES govedo(usni_broj_markice));

CREATE TABLE prodaja (
  id_prodaje CHAR(5),
  datum_prodaje DATE NOT NULL,
  kupac CHAR(11) NOT NULL,
  CONSTRAINT prodaja_pk PRIMARY KEY(id_prodaje),
  CONSTRAINT kupac_fk FOREIGN KEY(kupac) REFERENCES kupac(oib));

CREATE TABLE prodaja_stavka (
  id_pstavke INT IDENTITY(1,1),
  prodajna_cijena DECIMAL(5,2) NOT NULL CHECK (prodajna_cijena > 0),
  prodajna_kilaza SMALLINT NOT NULL CHECK (prodajna_kilaza > 0),
  prodaja CHAR(5),
  govedo CHAR(12),
  CONSTRAINT id_pstavke_pk PRIMARY KEY(id_pstavke),
  CONSTRAINT prodaja_fk FOREIGN KEY(prodaja)
    REFERENCES prodaja(id_prodaje)
  CONSTRAINT goved_fk FOREIGN KEY(govedo)
    REFERENCES govedo(usni_broj_markice));

```

**Slika 2.4.** Stvaranje tablica vaganje, prodaja i prodaja\_stavka



```

CREATE TABLE lijek (
    id_lijeka INT IDENTITY(1,1),
    naziv VARCHAR(20),
    detalji VARCHAR(200),
    karenca SMALLINT,
    stanje SMALLINT,
    CONSTRAINT id_lijeka_pk PRIMARY KEY(id_lijeka));

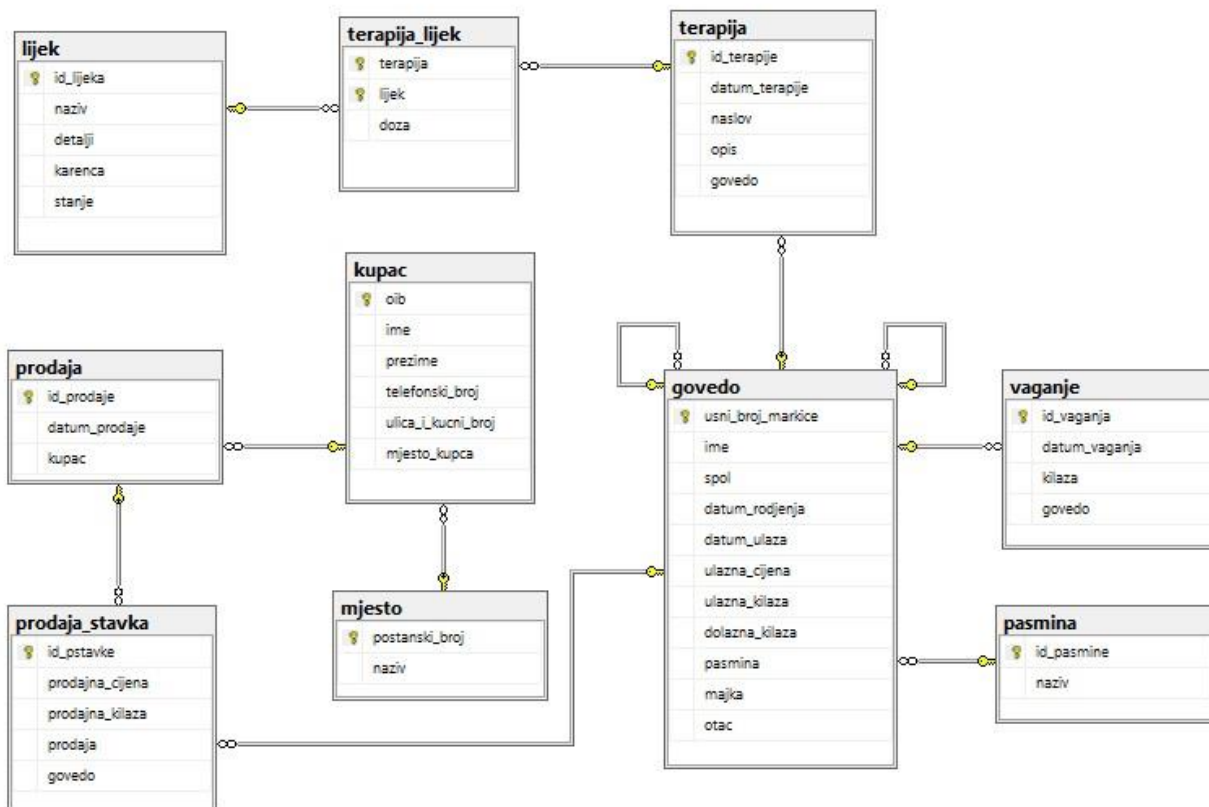
CREATE TABLE terapija (
    id_terapije INT IDENTITY(1,1),
    datum_terapije DATE NOT NULL,
    naslov VARCHAR(30),
    opis VARCHAR(200),
    CONSTRAINT id_terapije_pk PRIMARY KEY(id_terapije));

CREATE TABLE terapija_lijek (
    terapija INT,
    lijek INT,
    doza SMALLINT CHECK (doza > 0),
    CONSTRAINT terapija_lijek_pk PRIMARY KEY(terapija, lijek),
    CONSTRAINT lijek_fk FOREIGN KEY (lijek) REFERENCES lijek(id_lijeka),
    CONSTRAINT terapija_fk FOREIGN KEY (terapija) REFERENCES terapija(id_terapije));

```

Slika 2.5. Stvaranje tablica lijek, terapija i terapija\_lijek

Na slici 2.6. prikazan je dijagram baze farme dobiven u Microsoft SQL Server Management Studio.



Slika 2.6. Dijagram baze podataka farme

Osim stvaranja tablica, za izvršenje zadatka završnog rada, korištene su različite funkcije, procedure, pogledi i okidači, a u daljnjem tekstu su prikazane samo neke od njih.

Funkcija predstavlja potprogram koji obavlja određenu funkcionalnost nad podacima, pohranjen je u rječniku podataka i izvršava se u trenutku kad je to korisniku potrebno. Kao rezultat vraća jednu vrijednost ili tablicu.

Prema [5], sintaksa za stvaranje funkcije je:

```
CREATE [ OR ALTER ] FUNCTION [ schema_name. ] function_name
( [ { @parameter_name [ AS ][ type_schema_name. ] parameter_data_type
    [ = default ] [ READONLY ] }
  [ ,...n ]
]
)
RETURNS return_data_type
[ WITH <function_option> [ ,...n ] ]
[ AS ]
BEGIN
    function_body
    RETURN scalar_expression
END
[ ; ]
```

Slika 2.7. prikazuje funkciju koja računa ostvareni profit na osnovu potrošenih sredstava za nabavu i uzgoj goveda i ostvarene cijene pri prodaji. Kao parametar prima ušni broj markice goveda, a kao rezultat vraća decimalni broj. Kako se unutar te funkcije koristi i funkcija koja računa koliko je dana govedo provelo u tovu, njezin primjer vidljiv je na slici 2.8.

```

CREATE FUNCTION profit (@govedo CHAR(12))
RETURNS DECIMAL(10,2)
BEGIN
    DECLARE @profit DECIMAL(10,2)
    DECLARE @prod_kilaza SMALLINT
    DECLARE @prod_cijena DECIMAL(5,2)
    DECLARE @dana_u_tovu SMALLINT
    DECLARE @cijena_dana DECIMAL(5,2)
    DECLARE @ulaz_cijena DECIMAL(5,2)
    DECLARE @ulaz_kilaza SMALLINT

    SELECT @prod_kilaza = prodajna_kilaza, @prod_cijena = prodajna_cijena
    FROM prodaja_stavka
    WHERE govedo = @govedo
    SELECT @dana_u_tovu = dbo.dana_u_tovu_id(@govedo)
    SELECT @ulaz_cijena = ulazna_cijena, @ulaz_kilaza = ulazna_kilaza
    FROM govedo
    WHERE usni_broj_markice = @govedo
    SET @cijena_dana = 14.77
    SET @profit = (@prod_kilaza * @prod_cijena) - (@dana_u_tovu * @cijena_dana) - (@ulaz_cijena * @ulaz_kilaza)

    RETURN @profit
END

```

**Slika 2.7.** Funkcija profit

```

CREATE FUNCTION dana_u_tovu_id (@govedo CHAR(12))
RETURNS SMALLINT
BEGIN
    DECLARE @pocetak DATE
    DECLARE @kraj DATE

    SELECT @pocetak = datum_ulaza FROM govedo
        WHERE usni_broj_markice = @govedo
    SELECT @kraj = datum_prodaje FROM prodaja
        WHERE id_prodaje = (SELECT prodaja FROM prodaja_stavka
            WHERE govedo = @govedo)

    RETURN DATEDIFF(DAY, @pocetak, @kraj)
END

```

**Slika 2.8.** Funkcija dana u tovu id

Za razliku od funkcije, pohranjena procedura je potprogram koji u pozivajući program ne vraća rezultat kao povratnu vrijednost.

Prema [6], sintaksa za stvaranje procedure je:

```
CREATE [ OR ALTER ] { PROC | PROCEDURE }  
    [schema_name.] procedure_name [ ; number ]  
    [ { @parameter [ type_schema_name. ] data_type }  
      [ VARYING ] [ = default ] [ OUT | OUTPUT | [READONLY]  
    ] [ ,...n ]  
  [ WITH <procedure_option> [ ,...n ] ]  
  [ FOR REPLICATION ]  
  AS { [ BEGIN ] sql_statement [;] [ ...n ] [ END ] }  
  [;]
```

Na slici 2.9. prikazana je procedura koja prikazuje detalje vezane za razvoj određenog goveda na farmi.

```
CREATE PROCEDURE govedo_vaganja @usni_broj_markice CHAR(12)  
AS  
SELECT datum_vaganja "Datum vaganja", kilaza "Kilaža",  
       dbo.dana_u_tovu(datum_ulaza, datum_vaganja) "Dana u tovu",  
       dbo.prirast(dolazna_kilaza, kilaza) "Kontrolni prirast",  
       dbo.dnevni_prirast(dbo.prirast(dolazna_kilaza, kilaza),  
                          dbo.dana_u_tovu(datum_ulaza, datum_vaganja)) "Kontrolni dnevni prirast",  
       dbo.prirast(ulazna_kilaza, kilaza) "Ukupni prirast",  
       dbo.dnevni_prirast(dbo.prirast(ulazna_kilaza, kilaza),  
                          dbo.dana_u_tovu(datum_ulaza, datum_vaganja)) "Dnevni prirast"  
FROM govedo, vaganje  
WHERE govedo = @usni_broj_markice and  
       usni_broj_markice = @usni_broj_markice  
RETURN(0)
```

**Slika 2.9.** *Procedura govedo vaganja*

Pogled je relacija čija su shema i sadržaj definirani izrazom relacijske algebre pomoću SQL upita i određuje se tek pri obavljanju SQL upita. Upotrebljava se za ograničenje pristupa podacima, pojednostavljenje složenih upita, omogućavanje neovisnosti podataka ili prikaz različitih pogleda na iste podatke.

Prema [7], sintaksa za stvaranje pogleda je:

```
CREATE [ OR ALTER ] VIEW [ schema_name . ] view_name [ ( column [ ,...n ] ) ]  
[ WITH <view_attribute> [ ,...n ] ]  
AS select_statement  
[ WITH CHECK OPTION ]  
[ ; ]
```

Slika 2.10. prikazuje stvaranje pogleda koji daje pregled prodanih goveda na farmi.

```
CREATE VIEW pregled_prodanih_goveda  
( "JŽB goveda", Spol, Pasma, "Datum ulaza", "Ulazna cijena/kg",  
  "Ulazna kilaža", "Cijena kupnje", "Šifra prodaje", "Datum izlaza",  
  "Prodajna cijena/kg", "Prodajna kilaža", "Cijena prodaje",  
  "Dana u tovu", Profit, "Oib kupca" )  
AS  
SELECT usni_broj_markice, spol, pasmina, datum_ulaza, ulazna_cijena,  
       ulazna_kilaza, ulazna_cijena * ulazna_kilaza, id_prodaje, datum_prodaje,  
       prodajna_cijena, prodajna_kilaza, prodajna_cijena * prodajna_kilaza,  
       dbo.dana_u_tovu(datum_ulaza, datum_prodaje), dbo.profit(usni_broj_markice),  
       kupac  
FROM   govedo g, prodaja_stavka s, prodaja p  
WHERE  g.usni_broj_markice = s.govedo AND  
       s.prodaja = p.id_prodaje
```

**Slika 2.10.** Pogled za pregled prodanih goveda

Okidač je programski kod koji se izvršava automatski prilikom izvođenja specificirane DML (engleski Data Manipulation Language, jezik za manipuliranje podacima) operacije nad specificiranom tablicom.

Prema [8], sintaksa za stvaranje okidača je:

```
CREATE [ OR ALTER ] TRIGGER [ schema_name . ]trigger_name
ON { table }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS { sql_statement [ ; ] [ ,...n ] }
```

Na slici 2.11. nalazi se primjer okidača koji prije umetanja vrijednosti u tablicu terapija lijek uzima vrijednost doze te je oduzima od vrijednosti stanja u tablici lijek.

```
CREATE TRIGGER okidac_lijek_terapija_kolicina
ON terapija_lijek
FOR INSERT, UPDATE
AS
    DECLARE @lijek INT
    DECLARE @kolicina SMALLINT
    DECLARE @stanje SMALLINT
    SELECT @lijek = lijek, @kolicina = doza
    FROM INSERTED
    SELECT @stanje = stanje FROM lijek WHERE id_lijeka = @lijek;
    UPDATE lijek SET stanje = @stanje - @kolicina
    WHERE id_lijeka = @lijek;
```

**Slika 2.11.** Okidač na tablici terapija lijek

Kako bi se podaci zaštitili od neovlaštene uporabe ili kako bi im se ograničio pristup moguće je koristiti SQL naredbe grant i revoke koje daju ili oduzimaju ovlaštenja određenim korisnicima ili ulogama. Primjer stvaranja korisnika, uloge, davanja i oduzimanja ovlaštenja prikazan je na slici 2.12.

```
CREATE LOGIN upravitelj WITH PASSWORD = '123',
DEFAULT_DATABASE = farm;
CREATE USER Upravitelj FOR LOGIN upravitelj;

CREATE LOGIN tehnicar WITH PASSWORD = '111',
DEFAULT_DATABASE = farm;
CREATE USER Tehnicar FOR LOGIN tehnicar;

CREATE ROLE UPRAVITELJI_ULOGA;

GRANT SELECT, INSERT, UPDATE, EXECUTE TO
UPRAVITELJI_ULOGA;

EXEC sp_addrolemember UPRAVITELJI_ULOGA, Upravitelj;
EXEC sp_addrolemember UPRAVITELJI_ULOGA, Tehnicar;

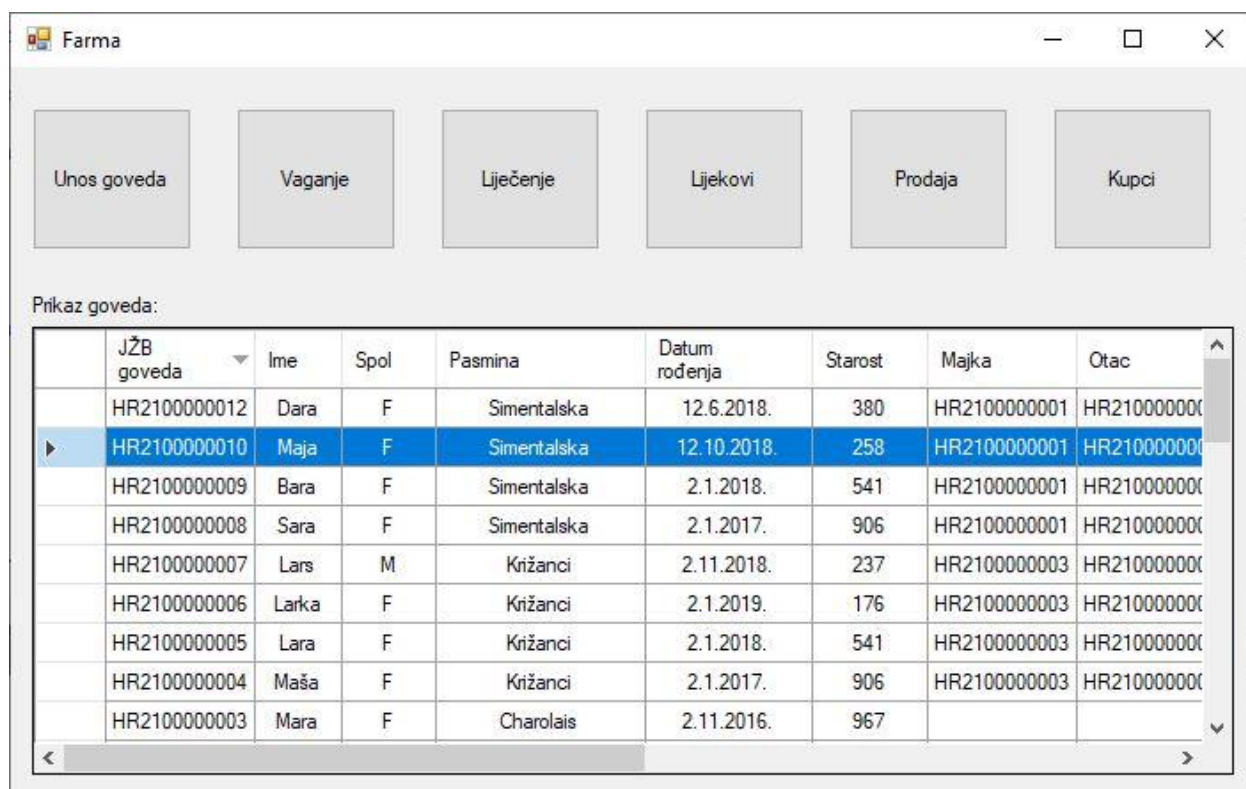
REVOKE SELECT, INSERT, UPDATE ON prodaja FROM Tehnicar;
REVOKE SELECT, INSERT, UPDATE ON prodaja_stavka FROM Tehnicar;
```

**Slika 2.12.** *Primjer davanja i oduzimanja ovlaštenja*

### 3. PROGRAMSKA PODRŠKA ZA VOĐENJE FARME

Kako bi se olakšalo vođenje farme bilo je potrebno izraditi rješenje preko kojeg bi osoba mogla na jednostavni način koristiti mogućnosti stvorene baze podataka. Kako bi se to izvelo iskorišteno je znanje stečeno na kolegijima Objektno orijentirano programiranje, Razvoj programske podrške objektno orijentiranim načelima i Baze podataka. Za izradu takvog sučelja koje omogućuje pristup korisnicima i rad s bazom podataka farme korišten je Microsoft Visual Studio Community 2015 s programskim jezikom C#. Povezivanje s bazom ostvareno je klasom SqlConnection, dohvaćanje podataka klasom SqlDataAdapter i slanje podataka klasom SqlCommand [9, 10, 11]. Grafičko sučelje je izrađeno korištenjem Windows Form-i [12] koje se u daljnjem tekstu spominju kao forme.

Slika 3.1. prikazuje početno grafičko sučelje koje omogućuje vođenje farme. Sučelje je izrađeno tako da bude što jednostavnije i učinkovitije, dok se ostale forme pozivaju kao dijaloški okviri.



Slika 3.1. *Farma*

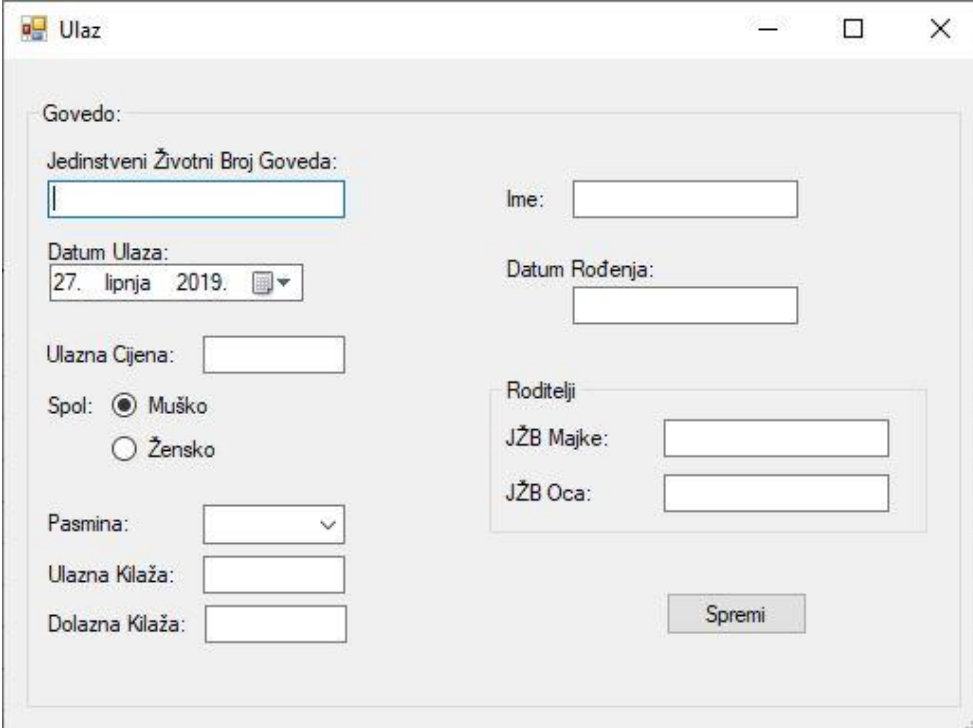


## 3.1. Unos podataka

### 3.1.1. Unos goveda

S početnog sučelja klikom na „Unos goveda“ otvara se forma Ulaz koja daje mogućnost unosa novog goveda.

Slika 3.2. daje prikaz forme za unos svih podataka o govedu. Podaci koji se moraju popuniti prilikom unosa kupljenog goveda su jedinstveni životni broj, odnosno ušni broj markice, datum ulaza na farmu, kupovna cijena po kilogramu, spol, pasmina, ulazna masa i dolazna masa u kilogramima. U slučaju da je govedo rođeno na farmi unose se i podaci o datumu rođenja koji je u tom slučaju jednak datumu ulaza, imenu te jedinstvenim životnim brojevima majke i oca.



The image shows a software window titled "Ulaz" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form for entering cow data. The form is organized into several sections:

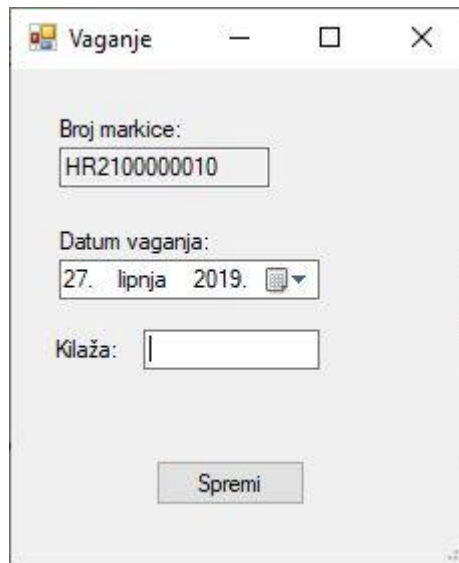
- Govedo:** A sub-section containing:
  - Jedinstveni Životni Broj Goveda:** A text input field.
  - Datum Ulaza:** A date picker showing "27. lipnja 2019." with a calendar icon.
  - Ulazna Cijena:** A text input field.
  - Spol:** Radio buttons for "Muško" (selected) and "Žensko".
  - Pasmina:** A dropdown menu.
  - Ulazna Kilaža:** A text input field.
  - Dolazna Kilaža:** A text input field.
- Ime:** A text input field.
- Datum Rođenja:** A text input field.
- Roditelji:** A sub-section containing:
  - JŽB Majke:** A text input field.
  - JŽB Oca:** A text input field.

At the bottom right of the form is a button labeled "Spremi".

Slika 3.2. Ulaz

### 3.1.2. Vaganje

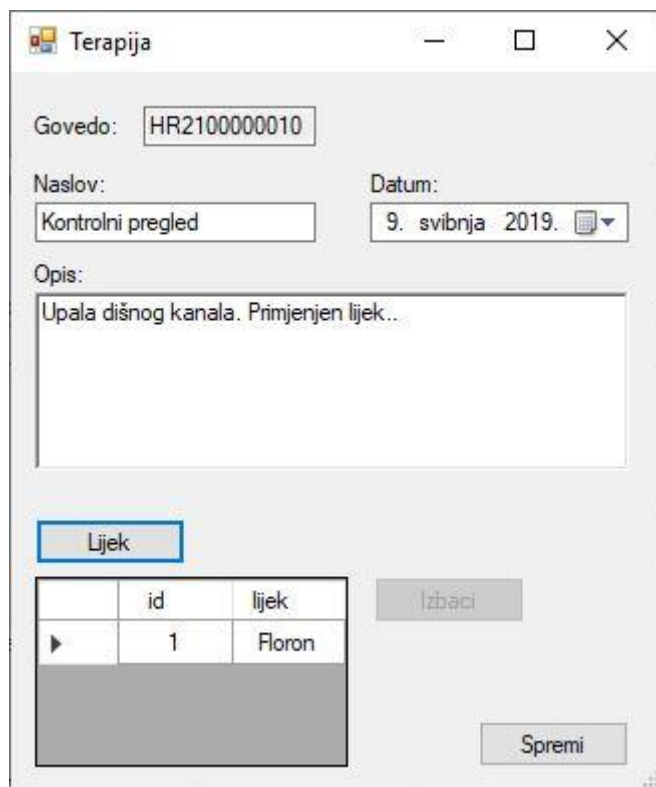
Do forme za unos podataka o vaganju goveda može se doći klikom na „Vaganje“ koje se nalazi na početnom sučelju (Slika 3.1.) ili klikom na „Izvaži“ koja se nalazi na formi Detalji o govedu na kartici Vaganja (Slika 3.10). Slika 3.3. prikazuje formu Vaganje gdje se unosi ušni broj markice goveda koje se važe, datum vaganja i izvagana masa u kilogramima.



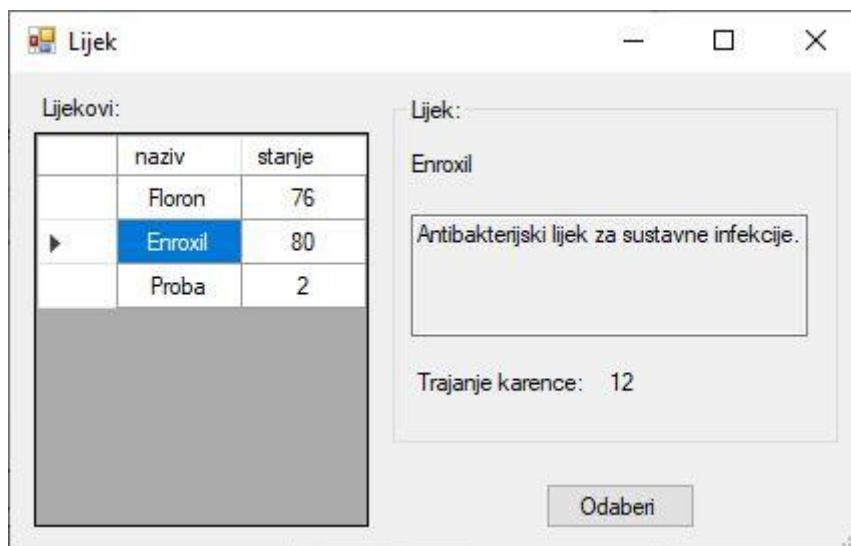
**Slika 3.3.** *Vaganje*

### 3.1.3. Liječenje

Za unos podataka o liječenju goveda koriste se forma Terapija (Slika 3.4.) i forma Lijek (Slika 3.5.). Do forme Terapija dolazi se klikom na „Liječenje“ na početnom sučelju (Slika 3.1.) ili klikom na „Liječi“ koje se nalazi u formi Detalji o govedu na kartici Terapije (Slika 3.12.). U formi Terapije unosi se ušni broj markice goveda, naslov koji ukratko opisuje terapiju, datum liječenja, opširniji opis terapije ako je to potrebno, te se dodaju lijekovi korišteni u terapiji. Također, postoji i mogućnost izbacivanja lijeka ako je izabran pogrešan. Lijek se unosi klikom na „Lijek“ koji otvara formu Lijek koja prikazuje tablicu dostupnih lijekova s njihovim stanjem na skladištu te naziv, opis i trajanje karence odabranog lijeka u tablici. Klikom na „Odaberi“ označeni lijek se dodaje u terapiju.



Slika 3.4. Terapija

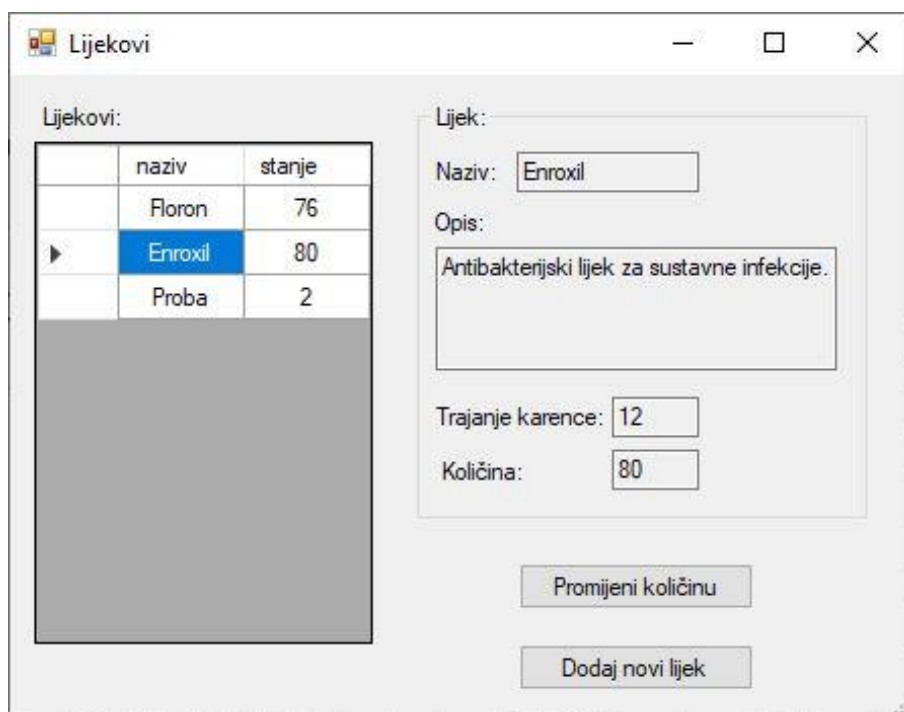


Slika 3.5. Lijek

### 3.1.4. Lijekovi

Forma Lijekovi, kao što je vidljivo na slici 3.6., omogućuje dodavanje novog lijeka i mijenjanje količine pojedinog lijeka na skladištu. Osim toga prikazuje i tablicu dostupnih lijekova s njihovim stanjem na skladištu. Klikom na „Promijeni količinu“ polje za unos količine se otključava i nudi mogućnost izmjene trenutnog stanja označenog lijeka u tablici. Klik na „Dodaj novi lijek“ pak otključava ostala polja za unos kako bi se mogli unijeti podaci o nazivu, opisu, trajanju karence u danima i količini lijeka.

Do ove forme može se doći klikom na „Lijekovi“ na početnom sučelju (Slika 3.1.).



Lijekovi:	
	stanje
Floron	76
▶ Enroxil	80
Proba	2

Lijek: Enroxil

Naziv: Enroxil

Opis: Antibakterijski lijek za sustavne infekcije.

Trajanje karence: 12

Količina: 80

Promijeni količinu

Dodaj novi lijek

Slika 3.6. Lijekovi

### 3.1.5. Prodaja

Za unošenje podataka o prodaji koriste se tri forme, a do same forme Prodaja pristupa se s početnog sučelja klikom na „Prodaja“ (Slika 3.1.) i klikom na „Nova Prodaja“ u formi Pregled Prodaja na kartici Prodaje (Slika 3.12.).

	Stavka	Spol	Pasmina	Prodajna kilaža	Prodajna cijena	Ukupno	
▶	010	17	F	SI	230	21,19	4873,70

**Slika 3.7. Prodaja**

U formi Prodaja (Slika 3.7.) unosi se šifra prodaje i datum prodaje, klikom na „Dodaj Kupca“ otvara se forma Kupac (Slika 3.9.) gdje je moguće odabrati kupca za prodaju. Nakon odabira kupca ponovno se vraća na formu Prodaja gdje se prikazuju detalji o kupcu pri čemu se gumb „Dodaj Kupca“ zaključava jer se prodaja radi samo prema jednom kupcu. Klikom na „Dodaj Govedo“ otvara se forma Govedo-Prodaja (Slika 3.8.) gdje se odabire govedo za prodaju i unose vrijednosti o masi u kilogramima na dan prodaje i dogovorena cijena po kilogramu.

	JŽB goveda	Spol	Pasmina	Kilaža zadnjeg vaganja	Datum zadnj vaganja
	HR2100000006	F	Križanci		
	HR2100000008	F	Simentalska		
▶	HR2100000010	F	Simentalska	190	8.4.2019.

**Slika 3.8. Prodaja – Govedo**

Forma Govedo-Prodaja prikazuje i tablicu svih dostupnih goveda s osnovnim podacima bitnim za prodaju i omogućuje pretraživanje goveda prema djelomičnom ušnom broju markice. Klikom na „Prodaj“ osvježava se tablica s dodanim govedima u formi Prodaja. U formi Prodaja pomoću „Izbaci Govedo“ poništava se odabrano govedo za prodaju u slučaju pogreške, a prodaja se zaključuje sa „Spremi Prodaju“.

### 3.1.6. Kupac

Do forme Kupac (Slika 3.9.) može se doći klikom na „Kupac“ na početnom sučelju (Slika 3.1.) ili na način koji je opisan u prethodnom odjeljku.

The screenshot shows a window titled "Kupac" with the following fields and buttons:

- OIB: 12312312313
- Ime: Ivan
- Prezime: Horvat
- Tel. broj: 0933333333
- Adresa: Glavna 20
- Pošanski broj: 34000
- Mjesto: Požega
- Buttons: "Prikaži sve" and "Spremi"

Below the fields is a table labeled "Kupci":

	OIB	Ime	Prezime	Telefon	Adresa	Pošanski broj
	12312312311	Ivo	Ivić	0911111111	Ivana Zajca 10	31500
	12312312312	Marija	Marić	0922222222	Kralja Tomislava 56	31000
▶	12312312313	Ivan	Horvat	0933333333	Glavna 20	34000
	12312312314	Ivo	Ivić	0944444444	Ivana Zajca 10	34350
	12312312315	Marko	Pejić	0955555555	Kneza Trpimira	31400

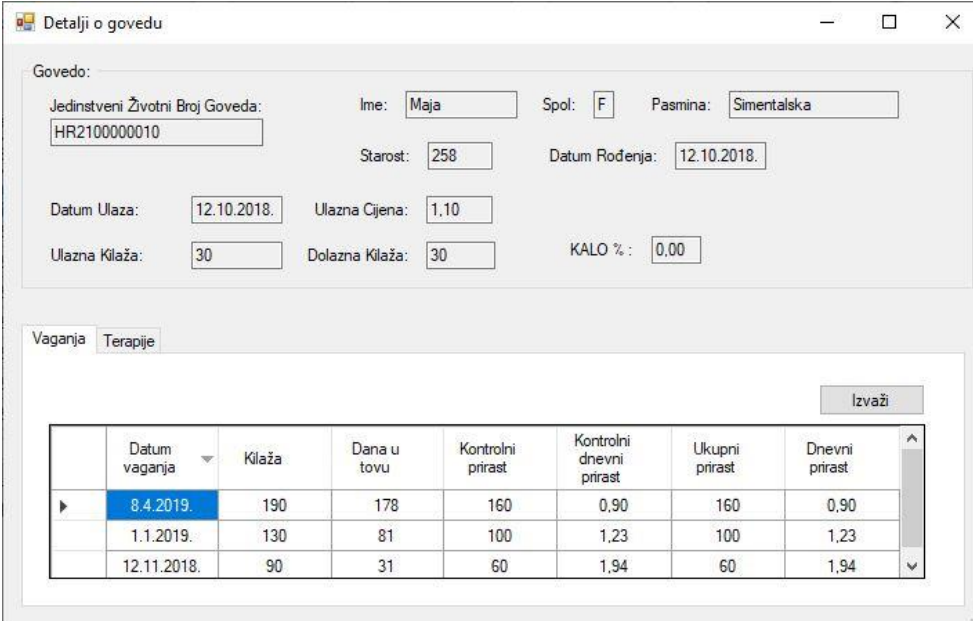
Slika 3.9. Kupac

Osim prikaza detalja o kupcima nudi i mogućnost njihove izmjene i dodavanja novog kupca u bazu. U slučaju kada je ova forma pozvana iz forme Prodaja, klikom na „Spremi“ se uz spremanje ili izmjenu kupca ujedno odabire i za prodaju.

## 3.2. Pregled podataka

### 3.2.1. Detalji o govedu

Forma Detalji o govedu otvara se dvostrukim klikom na redak tablice koja prikazuje goveda na farmi u početnom sučelju (Slika 3.1.). Osim što prikazuje detalje o odabranom govedu sadrži i dvije dodatne kartice.



The screenshot shows a window titled "Detalji o govedu" with a "Govedo:" section containing the following fields:

- Jedinstveni Životni Broj Goveda: HR210000010
- Ime: Maja
- Spol: F
- Pasmina: Simentalska
- Starost: 258
- Datum Rođenja: 12.10.2018.
- Datum Ulaza: 12.10.2018.
- Ulazna Cijena: 1,10
- Ulazna Kilaža: 30
- Dolazna Kilaža: 30
- KALO %: 0,00

Below the form are two tabs: "Vaganja" (selected) and "Terapije". The "Vaganja" tab contains a table with the following data:

	Datum vaganja	Kilaža	Dana u tovu	Kontrolni prirast	Kontrolni dnevni prirast	Ukupni prirast	Dnevni prirast
▶	8.4.2019.	190	178	160	0,90	160	0,90
	1.1.2019.	130	81	100	1,23	100	1,23
	12.11.2018.	90	31	60	1,94	60	1,94

An "Izvaži" button is located above the table.

Slika 3.10. Detalji o govedu – kartica Vaganja

Kartica Vaganje, kao što je vidljivo iz slike 3.10., sadrži tablicu koja prikazuje vrijednosti vezane za napredak goveda u uzgoju i gumb „Izvaži“ pomoću kojeg se mogu unijeti nove vrijednosti vaganja.

Druga kartica je kartica Terapije (Slika 3.11.) koja u tablici prikazuje vrijednosti vezane za liječenje goveda. Osim tablice, sadrži i gumb „Liječi“ kojim poziva formu Terapija za unos podataka o liječenju.

Detalji o govedu

Govedo:

Jedinstveni Životni Broj Goveda: HR210000010    Ime: Maja    Spol: F    Pasmina: Simentalska

Starost: 258    Datum Rođenja: 12.10.2018.

Datum Ulaza: 12.10.2018.    Ulazna Cijena: 1,10

Ulazna Kilaža: 30    Dolazna Kilaža: 30    KALO % : 0,00

Vaganja    Terapije

Liječi

ID	Datum	Naslov	Opis	Lijek
5	1.11.2018.	Prvi pregled	Govedo se normalno razvija.	
6	16.1.2019.	Drugi pregled	Upala dišnog kanala. Započeto lije...	Floron

Slika 3.11. Detalji o govedu – kartica Terapije

### 3.1.3. Pregled prodaja

Do prikaza ostvarenih prodaja dolazi se klikom na „Prodaja“ u početnom sučelju (Slika 3.1.). Forma Pregled Prodaja sadrži dvije kartice koje prikazuju sve ostvarene prodaje i prodaje filtrirane prema kupcima.

Kartica Prodaje (Slika 3.12.) u tablici prikazuje podatke o datumu, šifri, kupcu, broju prodanih goveda i prodajne vrijednosti pojedine prodaje. Uz tablicu sadrži i gumb „Nova Prodaja“ za unos nove podatka nove prodaje.

Pregled Prodaja

Prodaje    Kupci

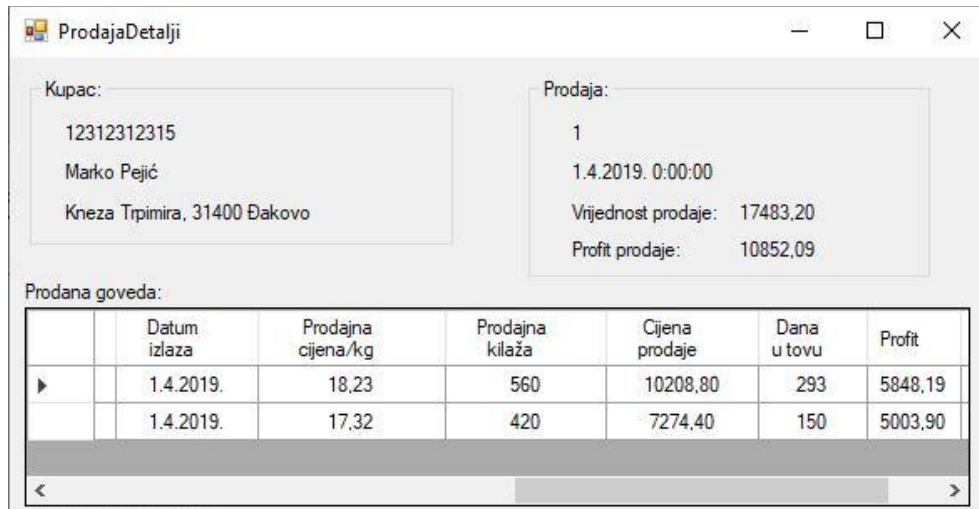
Nova Prodaja

Datum	Šifra prodaje	Oib kupca	Kupac	Broj goveda	Vrijednost
27.6.2019.	2	12312312315	Marko Pejić	1	5575,20
1.4.2019.	1	12312312315	Marko Pejić	2	17483,20

Slika 3.12. Pregled Prodaja – kartica Prodaje



Dvostrukim klikom na redak tablice otvara se nova forma s detaljima odabrane prodaje čiji je izgled prikazan na slici 3.13. Nova forma prikazuje podatke o kupcu, šifri, datumu, vrijednosti i ostvarenom profitu prodaje. Osim navedenoga, sadrži i tablicu sa svim govedima i popratnim podacima o pojedinom govedu koja su prodana u toj prodaji.



**Slika 3.13.** *Detalji odabrane prodaje*

Kartica Kupci (Slika 3.14.) u tablici prikazuje podatke o kupcima, kolika je vrijednost njihovih kupovina, broju goveda koja su kupili i broju ostvarenih prodaja.



**Slika 3.14.** *Pregled Prodaja – kartica Kupci*

Do detalja o prodajama po pojedinom kupcu dolazi se dvostrukim klikom na redak tablice prikazane u kartici Kupci. Slika 3.15. prikazuje kako ti detalji izgledaju. Prikazuju se podaci o kupcu, broj prodaja s tim kupcem, ukupni broj goveda koja je kupio, ukupnoj vrijednosti prodaja s kupcem i ostvaren ukupni profit na tom kupcu. Tablica prikazuje sa sva goveda i popratne podatke o pojedinom govedu koja su prodana kupcu.

	Ulazna kilaža	Cijena kupnje	Šifra prodaje	Datum izlaza	Prodajna cijena/kg	Prodajna kilaža
▶	30	33,00	1	1.4.2019.	18,23	560
	50	55,00	1	1.4.2019.	17,32	420
	30	33,00	2	27.6.2019.	23,23	240

**Slika 3.15.** *Detalji prodaja po kupcu*

## 4. ZAKLJUČAK

Razvoj programske podrške za uzgoj junadi i teladi, izrada baze podataka i sučelja kojim se krajnji korisnik koristi uvelike pomaže u svakodnevnom poslovanju farme. Kroz različite evidencije i pristupe podacima korisnik brzo i efikasno dolazi do potrebnih informacija. Time korisnik učinkovito iskorištava svoje vrijeme, povećava produktivnost farme i svoje poslovanje.

Da bi baza podataka bila što savršenija i uspješnija, potrebno je cjelokupan proces i postupak koji želimo predočiti bazom detaljno proučiti na licu mjesta, posjetom farmi i dogovorom s klijentima. Najbolja idejna rješenja se ostvaruju u suradnji s klijentima, kako bi im osigurali bazu podataka koja u potpunosti odgovara njihovim zahtjevima i potrebama.

Nakon postignutog dogovora, potrebno je izraditi bazu podataka i sučelje. Uvijek je dobro pojednostaviti sučelje kako bi se korisnik u potpunosti mogao koristiti mogućnostima programa. Pojednostavljenje sučelja će naposljetku pomoći korisniku kako bi vrlo lako i brzo došao do potrebnih evidencija, vodio određene knjige i pratio goveda. Time mu olakšavamo rad i svakodnevicu.

Razvojem novih baza podataka na temelju određenih specifikacija te njihovom primjenom u raznim industrijama i proizvodnim pogonima, dajemo naš doprinos u unaprjeđenju i razvoju gospodarstva. Bilo da se radi o stočarskoj proizvodnji, ili bilo kojoj drugoj vrsti proizvodnje, digitalizacija informacija je postala osnova, odnosno „alfa i omega“ svakog poslovanja.

## LITERATURA

- [1] M. Varga, BAZE PODATAKA; Konceptualno, logičko i fizičko modeliranje podataka, DRIP, Zagreb, 1994.
- [2] R. Manger, Osnove projektiranja baza podataka, Srce, 2010, dostupno na: [https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310\\_polaznik.pdf](https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310_polaznik.pdf) [18.5.2019.]
- [3] P. Chen, The Entity Relationship Model – Toward a Unified View of Data, dostupno na: <https://dspace.mit.edu/bitstream/handle/1721.1/47432/entityrelationshx00chen.pdf> [21.5.2019]
- [4] Microsoft, CREATE TABLE (Transact-SQL), <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-2017#simple-syntax> [26.5.2019]
- [5] Microsoft, CREATE FUNCTION (Transact-SQL), <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-function-transact-sql?view=sql-server-2017> [26.5.2019]
- [6] Microsoft, CREATE PROCEDURE (Transact-SQL), <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-procedure-transact-sql?view=sql-server-2017> [26.5.2019]
- [7] Microsoft, CREATE VIEW (Transact-SQL), <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-2017> [26.5.2019]
- [8] Microsoft, CREATE TRIGGER (Transact-SQL), <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-2017> [26.5.2019]
- [9] Microsoft, SqlConnection Class, <https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlconnection?view=netframework-4.8> [1.6.2019]
- [10] Microsoft, SqlDataAdapter Class, <https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqldataadapter?view=netframework-4.8> [1.6.2019]
- [11] Microsoft, SqlCommand Class, <https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlcommand?view=netframework-4.8> [1.6.2019]
- [12] Microsoft, Windows Forms, <https://docs.microsoft.com/en-us/dotnet/framework/winforms/windows-forms-overview> [1.6.2019]

## SAŽETAK

Kako su se uzgojne metode teladi i junadi osuvremenile i unaprijedile, u današnje vrijeme stvorila se potreba za digitalizacijom i praćenjem uzgojnih ciljeva, prijašnjih liječenja, prodaje i prihoda tovnih farmi. Da bi se to ostvarilo, razvijena je programska podrška i baza podataka za vođenje knjiga i evidencije s podacima o govedima na farmi, o njihovom porijeklu, boravku u tovu i prodaji, detaljima izvršene prodaje, zarade kroz prodaju, vođenje knjige liječenja i napretka pri uzgoju. Fizička baza podataka farme ostvarena je na Microsoft SQL Server 2017, dok je za rad s bazom korišten LINQPad. Za izradu sučelja koje omogućuje pristup korisnicima i rad s bazom podataka farme korišten je Microsoft Visual Studio Community 2015 s programskim jezikom C#.

**Ključne riječi:** baza podataka, C#, programska podrška, sql, uzgoj teladi i junadi.

## **ABSTRACT**

### **Software development for breeding calves and heifers**

As the breeding methods of calves and heifers have been developed and improved, the need for digitalization and monitoring of breeding goals, previous medical treatments, sales, and revenues has been created. To achieve this, software and database for management and keeping records with information about the cattle on the farm were developed. Database stores data about cattle origin, cattle fattening progress, cattle medical treatments, details on sales and earnings through sales. The farm database was realized on Microsoft SQL Server 2017, while the LINQPad was used to work with the database. To create interfaces that allow users to access and work with the farm database, Microsoft Visual Studio Community 2015 was used with the C# programming language.

**Keywords:** breeding calves and heifers, C#, database, software, sql.

## **ŽIVOTOPIS**

Davor Lukić rođen je 30.10.1982. u Požegi. 2001. godine završava Prirodoslovno-matematičku gimnaziju u Srednjoj školi Isidora Kršnjavoga Našice. Sveučilišni preddiplomski studij, smjer Računarstvo upisuje 2015. godine na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. 2019. godine završava Android Dev Academy u organizaciji Osijek Software City-a.

---

Davor Lukić

## **PRILOZI**

CD s izvornim kodom programske podrške.