

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Karlo Pavleka

**FUTURESTOCK – APLIKACIJA ZA
VOĐENJE STATISTIKE SKLADIŠTA I
PLANIRANJE ZALIHA**

PROJEKT

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Karlo Pavleka

Matični broj: 44128/15–R

Studij: Baze podataka i baze znanja

**FUTURESTOCK – APLIKACIJA ZA VOĐENJE STATISTIKE
SKLADIŠTA I PLANIRANJE ZALIHA**

PROJEKT

Mentor:

Izv. prof. dr. sc. Markus Schatten

Varaždin, siječanj 2019.

Karlo Pavleka

Izjava o izvornosti

Izjavljujem da je moj projekt izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Projekt FutureStock je aplikacija za vođenje statistike skladišta i planiranje zaliha. Na početku je objašnjen razlog odabira projekta te je dan sažet osvrt na temu. Opisuje se čvrsta veza između aktivnih baza podataka i skladišnog sustava, te također kako se temporalne baze podataka nadovezuju na navedeno. U sljedećem poglavlju opisuje se izrađeni model baze podataka - jezgra cijele aplikacije. Model baze podataka izrađen je u Navicat besplatnom alatu koji služi za grafičko modeliranje i upravljanje bazom podataka. Detaljno su pojašnjeni odnosi između tablica te su spomenuti ključni atributi unutar tablica. Govori se o ključnoj ulozi okidača te brojnim funkcionalnostima za koje su zaslužni u FutureStock aplikaciji. Nakon toga slijedi prikaz i pojašnjenje grafičkog sučelja koje je izrađeno u Microsoftovom Visual Studiu. Pokazane su sve funkcionalnosti koje aplikacija posjeduje te su spomenuti određeni problemi prilikom implementacije programskog koda. U zaključku je dan osvrt na rad i korištenje navedenih tehnologija te isto tako i značaj kvalitetnog sustava za upravljanje skladišnim zalihama.

Ključne riječi: aktivne baze podataka, skladište, temporalne baze podataka, postgresql, Visual Studio, sustav

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	2
3. Razrada teme	3
3.1. Model baze podataka	3
3.1.1. Aktivne baze podataka - okidači	5
3.1.2. Grafičko sučelje	8
4. Zaključak	15
5. Literatura	17
Popis slika	18

1. Uvod

Kao temu izrade projektnog rada iz kolegija Teorija baza podataka odabrao sam izradu aplikacije za vođenje statistike skladišta i planiranja zaliha. Za ovu temu sam se prvestveno opredjelio zato jer sam prilikom istraživanja ostalih tema imao u vidu koncept kojega bih se pridržavao vezano uz ovu temu te mi je djelovala interesantno. Također sam se htio okušati u PostgreSQL-u zbog toga što sam se u prošlosti često susretao sa MySQL-om te nisam imao priliku detaljnije istražiti ovaj sustav za upravljanje bazama podataka.

Sustav za upravljanje skladištem treba omogućiti konstantno praćenje cjeloukupnog stanja skladišta te pravovremeno reagirati na novonastale situacije. Upravo zbog navedene potrebe za preciznošću i pravovremenošću koriste se aktivne baze podataka koje su uz temporalne baze podataka nužne za provedbu kvalitetne aplikacije koja se bavi navedenom domom. Naime, skladišne aktivnosti znaju biti izrazito dinamične te u određenim trenucima može doći do iznenadnog pada količine zaliha što može ugroziti čitavu firmu. Ovdje na scenu dolaze aktivne baze podataka koje korištenjem specifičnih mehanizama poput okidača sprečavaju neželjene situacije. Okidači ovdje prvenstveno služe kako bi vodili računa o niskim ili ekstremno visokim količinama zaliha na skladištu te je njihov zadatak normalizirati ih ovisno o potrebi poduzeća - automatsko naručivanje kritičnih količina zaliha, informiranje o nedostatku prostora na skladištu i slično.

Uz sposobnost aktivnih baza podataka da proširuju baze podataka sposobnošću reaktivnog ponašanja ovdje su ključne i temporalne baze podataka. Temporalne baze služe kako bi pravovremeno reagirale na određene događaje, Vremenska točnost i preciznost koju temporalne baze podataka nude sustavu za upravljanje zalihama na skladištu je od izrazite važnosti kako bi određena poslovnica čitavo vrijeme bilo osposobljena optimalnom količinom robe. Isto tako uz temporalne baze usko su vezane narudžbenica i otpremnica koje imaju važan vremenski aspekt - vođenje evidencije o potražnji te potrebi za zalihama može se steći određena vizija kako će se tržište kretati, odnosno što treba napraviti kako bi poduzeće bilo korak ispred konkurencije.

U nastavku rada slijedi popis metoda i tehnika rada kojima sam se koristio prilikom izrade ovog projekta, te ću također detaljno prikazati cijeli proces kojim se razvoj aplikacije razvijao - od izrade početnog modela baze podataka pa sve do izrade grafičkog sučelja i finalne realizacije aplikacije.

2. Metode i tehnike rada

Za izradu ovog projekta koristio sam nekoliko ključnih alata bez kojih izrada aplikacije ne bi bila moguća. Koristio sam se PostgreSQL objektno-relacijskim sustavom za upravljanje bazama podataka. U početku izrade projekta proveo sam određeno vrijeme proučavajući sintaksu PostgreSQL-a i istražujući dokumentaciju na njihovim službenim internetskim stranicama. Korištenjem njihove dokumentacije brzo sam se snašao te je savladavanje određenih aspekata sintakse bilo jednostavno.

Prije konkretne izrade modela podataka koristio sam se web alatom Draw.io u kojem sam izradio nekoliko skica ERA modela kako bih unaprijed imao ideju kako pristupiti izradi što kvalitetnije baze podataka na koju se cijela aplikacija i nadovezuje. Draw.io, iako ne sadrži velik broj mogućnosti, koristio je za crtanje osnovnih komponenata modela baze podataka, kao što su tablice i njihov sadržaj, određeni odnosi između tablica i slično.

Alat u kojemu sam dizajnirao i izradio model baze podataka je Navicat - izuzetno kvalitetan softver za grafičko dizajniranje i modeliranje baze podataka. S njime se nisam susreo po prvi put te sam već imao priliku raditi u ovom alatu. Navicat je besplatan softver koji nudi velik izbor pogodnosti i mogućnosti (iako postoji i premium izdanje softvera, besplatno je i više nego dovoljno).

Još jedan ključan alat kojim sam se služio za modeliranje i izradu baze podataka je pgAdmin - najpopularnija open-source administrativna i razvojna platforma za PostgreSQL, koja je bogata širokom ponudom mogućnosti. pgAdminom služio sam se kako bih napravio svoju postgres bazu te imao bolji pregled nad svim mogućnostima i specifikacijama modela baze podataka.

Grafičko sučelje izradio sam u Microsoftovom Visual Studio 2017. alatu, a aplikaciju sam pisao u programskom jeziku C.Sharp s kojim sam već dobro upoznat. Microsoft Visual Studio je integrirano razvojno okruženje (IDE) koje se prvenstveno koristi za razvoj Desktop aplikacija za Windows, za razvoj web stranica te usluga. Visual Studio svojim korisnicima nudi kvalitetnu podršku u pisanju koda u obliku IntelliSensea (Intelligent code completion) koji na temelju konteksta predlaže određena programska rješenja te znatno ubrzava pisanje koda. Grafičko sučelje rađeno je koristeći Windows Forms GUI koji je sadržan unutar Microsoftovog .NET Frameworka.

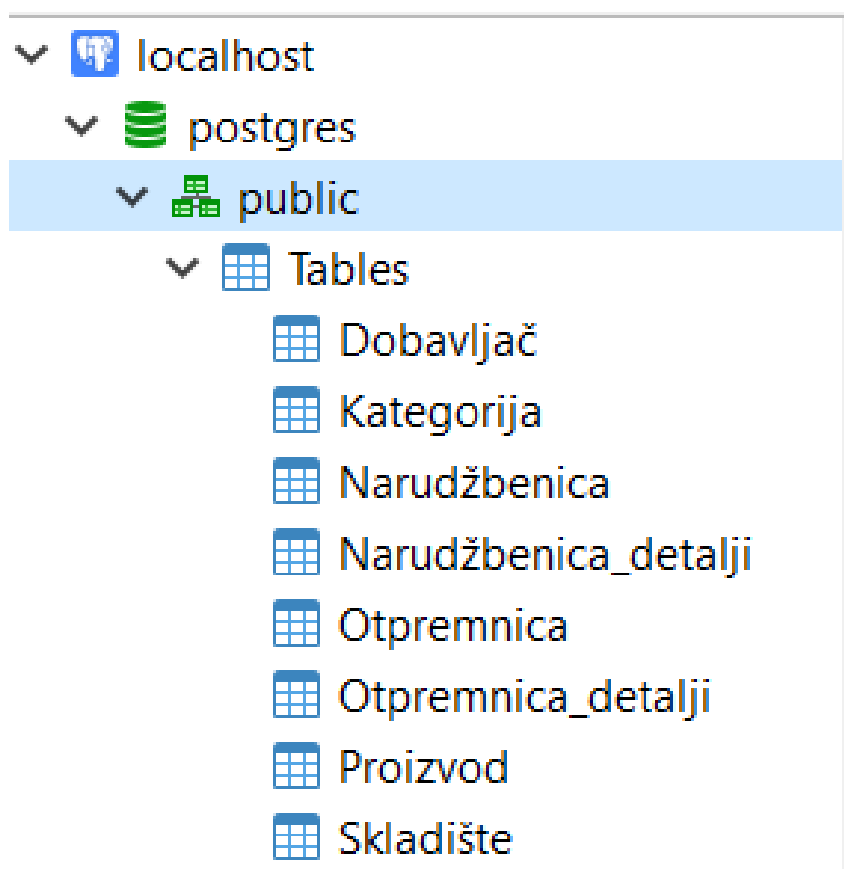
3. Razrada teme

Došli smo do glavnog poglavlja u kojem ću detaljno prikazati kroz koje korake je ovaj projekt realiziran - počevši od izrade početnih tablica unutar baze podataka sve do izrade grafičkog sučelja koje je realizirano u C.Sharp programskom jeziku. U prvom poglavlju detaljno ću predstaviti model baze podataka kojim se aplikacija koristi za realizaciju sustava za održavanje zaliha i vođenje statistike skladišta. Nakon toga slijedi prikaz aktivnih baza podataka koji je realiziran kroz nekoliko različitih okidača od kojih svaki ima važnu ulogu. U sljedećem poglavlju ukratko ću pojasniti neke najznačajnije pojedinosti važne za izradu grafičkog sučelja, a na kraju ću demonstrirati i prikazati sve funkcionalnosti koje aplikacija nudi.

3.1. Model baze podataka

Kao što sam već i ranije spomenuo, početne skice modela baze podataka izrađene su šturo koristeći se web alatom Draw.io koji služi za crtanje skica različitih vrsta grafova. Nakon toga je uslijedila provedba nacrtane skice u alatu za grafičko dizajniranje i modeliranje baze podataka - Navicatu.

U nastavku slijedi prikaz osnovnih tablica modela baze podataka te ću svaku od njih objasniti na ERA modelu gdje se vidi njihov međusoban odnos

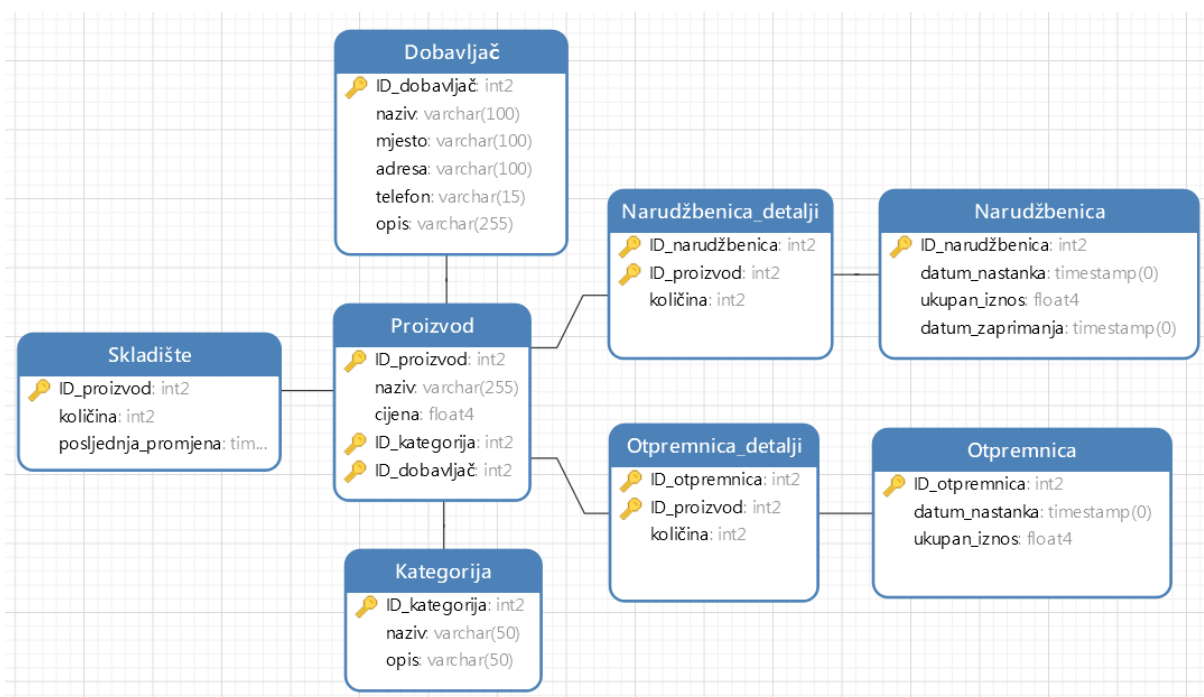


Slika 1: Prikaz tablica modela baze podataka (Izvor: vlastita izrada, 2019.)

Baza podataka kojom se koristi aplikacija sastoji se od 8 tablica:

- Dobavljač
- Kategorija
- Narudžbenica
- Narudžbenica detalji
- Otpremnica
- Otpremnica detalji
- Proizvod
- Skladište

Ovih 8 tablica čini funkcionalan model baze podataka koji obavlja sve funkcionalnosti koje mu aplikacija može zadati. Prikaz modela baze podataka je sljedeći:



Slika 2: Prikaz modela baze podataka (Izvor: vlastita izrada, 2019.)

U objašnjavanje baze podataka krenuti ću od najmanje zavisnih tablica. Tablica Kategorija sastoji se od 3 atributa te ona obilježava određenu kategoriju kojoj proizvod pripada (npr. posuđe). Jednostavna tablica kod koje se prilikom stvaranja entiteta mora unijeti ID kategorija i naziv, dok je opis kategorije opcionalan. Sljedeća po zavisnosti je tablica Dobavljač, koja se sastoji od nešto više atributa, gdje je opis također opcionalan. Svakom proizvodu pripada točno jedan dobavljač, dok određeni dobavljač može imati više proizvoda koje isporučuje. Tablica Skladište sastoji se od tri atributa te je važno napomenuti da ona nema vlastiti identifikacijski ključ, nego ovisi o identifikacijskom ključu proizvoda koji se na skladištu nalazi. Također

se u tablici nalazi i količina, jedan od najvažnijih atributa u ovom projektu, te isto tako i atribut posljednjaPromjena, koji nam govori točno vrijeme zadnjeg ažuriranja na skladištu.

Slijede dvije tablice sličnog sadržaja: Narudžbenica i Otpremnica. Obje tablice imaju vlastiti identifikacijski ključ, dijele attribute datum nastanka koji govori kad je navedeni dokument nastao, te ukupan iznos, odnosno sumu svih cijena proizvoda koji se na dokumentu nalaze. Jedina razlika je u tome što tablica narudžbenica sadrži atribut datumZaprimanja koji je važan kako bi se znalo kad je naručena roba stigla na skladište, dok se kod otpremnice ta evidencija ne vodi.

Tablice narudžbenicaDetalji i otpremnicaDetalji nastali su kao posljedica veze više-više sa tablicom Proizvod - na narudžbenici se može nalaziti više proizvoda, ali isto tako jedan proizvod se može nalaziti na više različitih narudžbenica. Navedene tablice preuzele su identifikacijske ključeve od susjednih tablica, te također imaju atribut količina, koji označava koliko se određenih komada specifičnog proizvoda nalazi na narudžbenici, odnosno otpremnici. Za kraj je ostala tablica Proizvod koja se sastoji od svog identifikacijskog ključa, naziva proizvoda te jedinične cijene pojedinog komada robe. Ovdje se također nalaze i ključevi od tablica Dobavljač i Kategorija.

3.1.1. Aktivne baze podataka - okidači

Aktivne baze podataka imaju ključnu ulogu u upravljanju skladišnim sustavom za održavanje zaliha. Kako ne bi došlo do nepredvidivih situacija gdje određeni proizvod nije raspoloživ ili do situacije da je skladište prepuno te nema dovoljno mjesta, koriste se okidači čiji je zadatak izvršiti određenu akciju prilikom specifičnog događaja. U projektu se nalazi 7 okidača, koje ću u nastavku objasniti.

```
CREATE OR REPLACE FUNCTION skladiste_naruci()
RETURNS TRIGGER
AS $$
BEGIN
    IF EXISTS (SELECT * FROM "Skladište" WHERE "ID_proizvod" = NEW."ID_proizvod")
    THEN
        UPDATE "Skladište"
        SET "količina" = "količina" + NEW."količina",
            "posljednja_promjena" = date_trunc('second'::text, CURRENT_TIMESTAMP)
        WHERE "ID_proizvod" = NEW."ID_proizvod";
    ELSE
        INSERT INTO "Skladište" VALUES (NEW."ID_proizvod", NEW."količina", date_trunc('second'::text, CURRENT_TIMESTAMP));
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER skladiste_narudzba
AFTER INSERT ON "Narudžbenica_detalji"
FOR EACH ROW EXECUTE PROCEDURE skladiste_naruci();
```

Slika 3: Okidač br. 1 - Skladište narudžba (Izvor: vlastita izrada, 2019.)

Prvi okidač povezuje dvije tablice: NarudžbenicaDetalji te Skladište. Naime, kada se putem narudžbenice naruče određeni proizvodi, okidač navedene proizvode pridodaje na skladište kako bi bili raspoloživi. Okidač funkcionira na način da se nakon unosa u tablicu NarudžbenicaDetalji za svaki redak izvrši funkcija skladisteNaruci() koja provjerava postoji li već

navedeni proizvod na skladištu. Ukoliko postoji, količini tog proizvoda će se samo pridodati naručena količina, a ako tog proizvoda još nema na skladištu, tada će se unijeti novi redak u tablicu Skladište sa navedenim proizvodom.

```
CREATE OR REPLACE FUNCTION skladiste_otpremi()
RETURNS TRIGGER
AS $$
DECLARE cijena2 FLOAT;
] BEGIN
    UPDATE "Skladište"
    SET "količina" = "količina" - NEW."količina",
        "posljednja_promjena" = date_trunc('second'::text, CURRENT_TIMESTAMP)
    WHERE "ID_proizvod" = NEW."ID_proizvod";
    SELECT cijena INTO cijena2 FROM "Proizvod" WHERE "ID_proizvod" = NEW."ID_proizvod";
    IF EXISTS (SELECT * FROM "Skladište" WHERE "ID_proizvod" = NEW."ID_proizvod" AND "količina" < 10)
] THEN INSERT INTO "Narudžbenica"
VALUES (nextval('narudžbenica_seq'), date_trunc('second'::text, CURRENT_TIMESTAMP), cijena2*50, NULL);
INSERT INTO
"Narudžbenica_detalji" ("ID_narudžbenica", "ID_proizvod", "količina")
VALUES (currval('narudžbenica_seq'), NEW."ID_proizvod", 50);
- END IF;
RETURN NEW;
- END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER skladiste_otprema
AFTER INSERT ON "Otpremnica_detalji"
FOR EACH ROW EXECUTE PROCEDURE skladiste_otpremi();
```

Slika 4: Okidač br. 2 - Skladište otprema (Izvor: vlastita izrada, 2019.)

Drugi okidač također je povezan sa tablicom Skladište, ali ovoga puta radi se o otpremnici. Prilikom stvaranja nove otpremnice okidač sa skladišta oduzima otpremljenu količinu proizvoda. Nakon toga slijedi provjera preostale količine otpremljenog proizvoda. Ovdje je uvedena automatizacija naručivanja proizvoda. Naime, ako je nakon otpremanja proizvoda preostala količina manja od 10, sustav će automatski naručiti 50 komada proizvoda kako bi se osiguralo da ne dođe do nedostatka zaliha. Kreira se automatska narudžbenica koja se pohranjuje u bazu, a samim time se aktivira i prethodni okidač koji tu količinu ponovno pohranjuje na skladište.

```
CREATE OR REPLACE FUNCTION provjeri_min_kolicinu()
RETURNS TRIGGER
AS $$
] BEGIN
] IF NEW."količina" < 0
] THEN
    RAISE EXCEPTION 'Nedovoljan broj komada proizvoda na skladištu! Na raspolaganju je % komada!', OLD."količina";
    RETURN OLD;
] END IF;
RETURN NEW;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER provjera_min_kolicine
AFTER UPDATE ON "Skladište"
FOR EACH ROW EXECUTE PROCEDURE provjeri_min_kolicinu();
```

Slika 5: Okidač br. 3 - Provjera minimalne količine (Izvor: vlastita izrada, 2019.)

Treći okidač vrlo je jednostavan, ali efikasan. Nakon što dođe do određenog ažuriranja količine na skladištu provjerava se količina koja je nakon tog događaja preostala. Ako je došlo do pokušaja otpremanja količine proizvoda veće od one koja na skladištu postoji, korisnika će

se obavijestiti da na skladištu nema dovoljno komada proizvoda te mu se neće dozvoliti stvaranje takve otpremnice. Također će ga se upoznati sa preostalom količinom s kojom raspolaže. Okidač sličan ovome je okidač za provjeru maksimalne količine koji vodi računa o tome da je moguće imati maksimalno 300 komada istog proizvoda na skladištu, te prilikom narudžbe količine koja bi prekoračila tu brojku korisnika se obavještava te mu se ne dozvoljava narudžba.

```
CREATE OR REPLACE FUNCTION provjeri_ukupnu_kolicinu()
RETURNS TRIGGER
AS $$
DECLARE pomocna INT;
BEGIN
    SELECT SUM(kolicina) INTO pomocna FROM "Skladište";
    IF NEW."količina" + pomocna > 2000
    THEN
        RAISE EXCEPTION 'Na skladištu se može nalaziti najviše 2000 komada proizvoda!';
        RETURN OLD;
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER provjera_ukupne_kolicine
AFTER UPDATE ON "Skladište"
FOR EACH ROW EXECUTE PROCEDURE provjeri_ukupnu_kolicinu();
```

Slika 6: Okidač br. 4 - Provjera ukupne količine (Izvor: vlastita izrada, 2019.)

Četvrti okidač koji ću spomenuti je okidač koji provjerava maksimalni kapacitet skladišta. Skladište dozvoljava maksimalno 2000 komada proizvoda sveukupno, a prilikom korištenja aplikacije korisnika se zanimljivim ProgressBarom obavještava o tome koliko još ima slobodnog mjesta na skladištu. Okidač poziva funkciju koja zbraja sve količine svih proizvoda na skladištu, te ukoliko prilikom narudžbe količina prelazi 2000, korisniku se neće dozvoliti narudžba.

```
CREATE OR REPLACE FUNCTION obrisi_narudzbenicu()
RETURNS TRIGGER
AS $$
BEGIN
    DELETE FROM "Narudžbenica_detalji"
    WHERE "Narudžbenica_detalji"."ID_narudžbenica" = OLD."ID_narudžbenica";
    RETURN OLD;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER narudzbenica_obrisi
BEFORE DELETE ON "Narudžbenica"
FOR EACH ROW EXECUTE PROCEDURE obrisi_narudzbenicu();
```

Slika 7: Okidač br. 5 - Brisanje detalja narudžbe (Izvor: vlastita izrada, 2019.)

Za kraj je ostao jedan jednostavan okidač čija je funkcija ta da kada korisnik obriše određenu narudžbenicu, okidač briše sve detalje narudžbe koji su bili povezani sa navedenom

narudžbenicom. Identičan okidač postoji i za otpremnicu. To je ujedno bio i zadnji, sedmi okidač u ovoj bazi podataka. U nastavku slijedi prikaz grafičkog sučelja aplikacije.

3.1.2. Grafičko sučelje

Grafičko sučelje izrađeno je u Microsoft Visual Studio razvojnom okruženju, dok je kod pisan u programskom jeziku C.Sharp. Kroz aplikaciju ću proći koristeći više slika kako bi se bolje shvatila realizacija projekta. Aplikacija je građena objektno-orijentiranim pristupom te se u njoj nalazi 14 klasa. Na slici ispod može se vidjeti forma dobrodošlice koja se pojavi prilikom otvaranja aplikacije.



Slika 8: Forma dobrodošlice (Izvor: vlastita izrada, 2019.)

Nakon inicijalne forme dobrodošlice u aplikaciju, ovo je početna, glavna forma aplikacije. S ove aplikacije se pristupa svim funkcionalnostima. Možemo odabrati raditi s entitetima gdje se može dodavati novi sadržaj u svaku tablicu, te se isto tako entiteti mogu i brisati. Na glavnoj formi nalazi se prikaz sadržaja na skladištu, odnosno vide se svi proizvodi koji su na skladištu dostupni te njihova količina. U lijevom dijelu forme nalazi se ProgressBar koji prikazuje koliko je skladište popunjeno. S druge strane forme nalazi se mogućnost zaprimanja pošiljke, odnosno narudžbenice gdje se vide sve narudžbenice koje nisu zaprimljene, odnosno koje jesu.

U aplikaciji je također bio zamišljen i grafički prikaz stanja na skladištu u obliku grafikona, ali zbog određenih bugova nisam uspio realizirati navedeni grafički prikaz. Postoji mogućnost nadogradnje navedene funkcionalnosti u budućnosti.


Na sljedećoj slici (Slika 9.) vidi se što se dogodi nakon klika na gumb Entiteti na početnoj stranici. Ovdje ima nekoliko tabova gdje se mogu izabrati sadržaji svih tablica. Gumbom Povratak vraćamo se na početnu formu, gumbom Kreiraj otvara se nova forma gdje se stvara novi entitet tablice. Gumb Briši obrisati će označeni red u tablici te ažurirati bazu podataka.

Dashboard

Rad s entitetima

Entiteti

Popunjenost skladišta



Zaprimanje pošiljke

Pošiljke

Skladišna statistika

	Naziv proizvoda	Količina	Posljednja promjena
▶	Čokolada s rižom	22	20.1.2019. 11:30
	Zrnati sir	33	16.1.2019. 14:53
	Bombonjera	52	20.1.2019. 11:29
	Edamer sir	54	20.1.2019. 13:50
	Lubenica	57	20.1.2019. 13:52
	Jabuka	88	20.1.2019. 12:02
	Juha od gljiva	186	20.1.2019. 12:02

Slika 9: Početna forma aplikacije (Izvor: vlastita izrada, 2019.)

Rad s entitetima

Dobavljači

Proizvodi

Kategorije

Narudžbenice

Otpremnice

	Šifra	Naziv	Mjesto	Adresa	Telefon	Opis
▶	1	Vindija	Varaždin	Međimurska 6	042 399 999	Prehrambena industrija ...
	2	Koka	Varaždin	Trg bana Jelačića 8a	043 585 928	Koka d.o.o
	3	Mobis	Zagreb	Branimirova ulica 17	01 0483 483	Mobis d.d.
	4	Podravka	Zagreb	Slavonska avenija 6	01 4891 910	Podravka d.d.
	5	Kandit	Osijek	Vukovarska cesta 239	031 512 512	Kandit d.o.o.
	6	KB	Sračinec	Vilka Novaka 6	042 215 777	KB d.o.o.
	7	Agrodalm	Zagreb	Blizno 13	01 2392 617	Agrodalm d.o.o

Povratak

Kreiraj

Briši

Slika 10: Rad s entitetima - Dobavljači (Izvor: vlastita izrada, 2019.)

Na slici br.10 vidi se forma na koju korisnik može otići pritiskom na gumb Kreiraj na prošloj formi. Korisnik ovdje može ispuniti polja te kreirati željenog dobavljača te pritiskom na gumb Kreiraj dobavljač se pohranjuje u bazu. Ukoliko se korisnik ipak predomisli, može se vratiti na prethodnu formu klikom na gumb Povratak.

Ova slika sličnog je sadržaja kao i popis dobavljača, no razlika je u tome što na ovoj formi postoji gumb Detalji. Klikom na navedeni gumb korisnik može pregledati detalje narudžbenice koja je selektirana kako bi se bolje upoznao sa sadržajem navedenog dokumenta.

Kreiranje dobavljača

Novi dobavljač

Naziv dobavljača:

Mjesto:

Adresa:

Telefon:

Opis (opcionalno):

Kreiraj

Povratak

Slika 11: Rad s entitetima - Kreiranje dobavljača (Izvor: vlastita izrada, 2019.)

Rad s entitetima

Dobavljači Proizvodi Kategorije **Narudžbenice** Otpremnice

	Šifra	Datum nastanka	Ukupan iznos (kn)	Datum zaprimanja
▶	1	3.1.2019. 14:55	142	
	2	15.1.2019. 5:13	310	19.1.2019. 20:33
	3	15.1.2019. 23:47	354	20.1.2019. 2:29
	4	24.1.2019. 21:53	555	20.1.2019. 2:24
	5	9.1.2019. 22:40	13	
	7	20.1.2019. 11:34	780	
	8	20.1.2019. 12:02	1029	

Povratak Kreiraj Briši Detalji

Slika 12: Rad s entitetima - Narudžbenice (Izvor: vlastita izrada, 2019.)

Na sljedećoj slici (slika br.13) nalazi se prikaz forme kojoj se može pristupiti sa početne forme. Klikom na gumb Pošiljke korisnik ima uvid u sve narudžbenice koje su zaprimljene i koje nisu - odvojene su u dvije tablice. Korisnik ima mogućnost zaprimiti narudžbenicu koja tada dobije status zaprimljene te se pojavi u drugoj tablici.

Na gornje dvije slike prikazano je kako se zaprimanjem narudžbenice postavi datum zaprimanja te se narudžbenice više ne nalaze sa nezaprimljenim narudžbenicama.

Narudžbenice

Nezaprmljene

	Šifra	Nastalo	Ukupan iznos (kn)
▶	5	9.1.2019. 22:40	13
	7	20.1.2019. 11:34	780
	8	20.1.2019. 12:02	1029

Zaprmljene

	Šifra	Nastalo	Ukupan iznos (kn)	Zaprmljeno
▶	1	3.1.2019. 14:55	142	20.1.2019. 17:36
	2	15.1.2019. 5:13	310	19.1.2019. 20:33
	3	15.1.2019. 23:47	354	20.1.2019. 2:29
	4	24.1.2019. 21:53	555	20.1.2019. 2:24

Povratak
Zaprmi

Slika 15: Nezaprmljene narudžbenice 2/2 (Izvor: vlastita izrada, 2019.)

Kreiranje narudžbenice

Nova narudžbenica Stvori predložak

Proizvod:

Količina:

Jedinična cijena (kn):

Ukupno (kn):

Dodaj proizvod

Povratak
Kreiraj

Slika 16: Kreiranje narudžbenice 1/3 (Izvor: vlastita izrada, 2019.)

naziv proizvoda, ako korisnik napusti polje a naziv proizvoda ne postoji u bazi, doći će do exceptiona te će aplikacija javiti grešku).

Nakon upisivanja količine aplikacija će ispisati ukupan iznos navedene stavke narudžbenice, te tada korisnik treba stisnuti gumb Dodaj kako bi se proizvod dodao na narudžbenicu (korisnik će biti obaviješten nakon što klikne na gumb). Korisnik unosi koliko hoće proizvoda, a kada završi treba stisnuti gumb Kreiraj kako bi potvrdio kreiranje narudžbenice.

Kreiranje narudžbenice

Nova narudžbenica Stvori predložak

Proizvod: Edamer sir

Količina: 14

Jedinična cijena (kn): 24

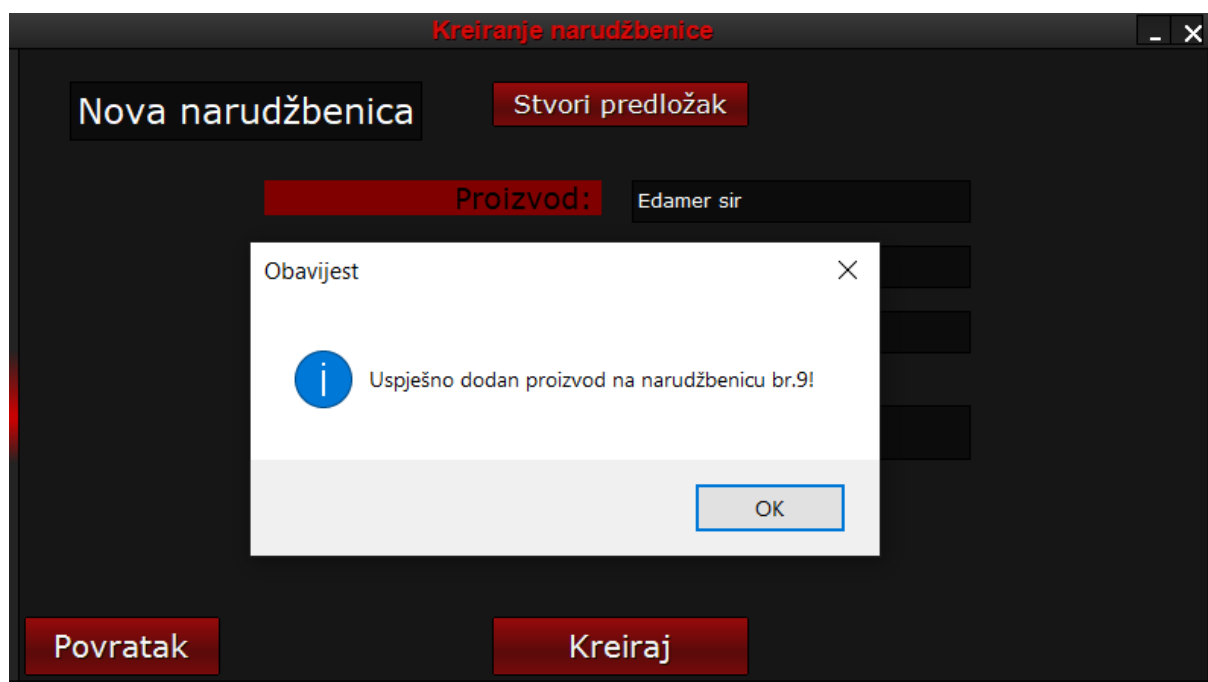
Ukupno (kn): **336**

Dodaj proizvod

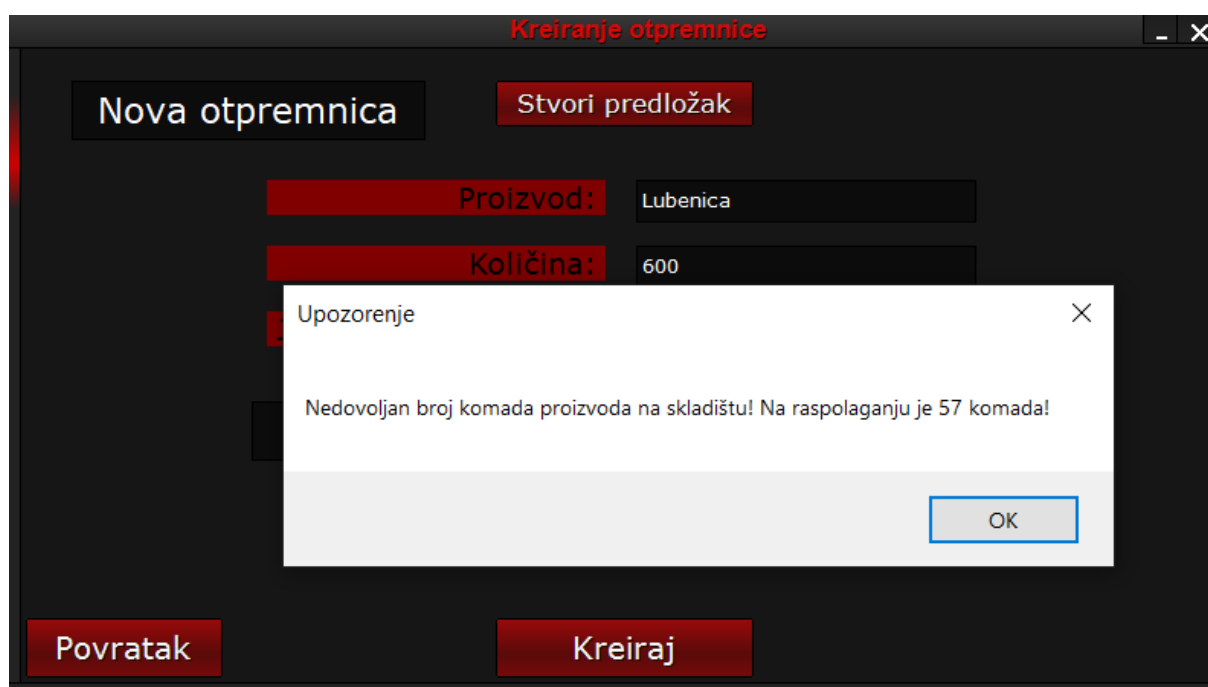
Povratak Kreiraj

Slika 17: Kreiranje narudžbenice 2/3 (Izvor: vlastita izrada, 2019.)

Ovdje se jasno vidi kako izgleda popunjena forma za kreiranje jedne stavke narudžbenice. Aplikacija je izračunala umnožak jedinične cijene i količine, a pritiskom na gumb Dodaj proizvod sva polja će se isprazniti, stavka će se dodati te će korisnik ponovno moći dodati proizvod na narudžbenicu. Ako korisnik za vrijeme kreiranja narudžbenice stisne gumb Povratak, ako je stvorio prazan predložak narudžbenica će se pohraniti kao prazna, a ako stisne gumb Povratak nakon što je dodao jedan proizvod na narudžbenicu, pohraniti će se narudžbenica sa postojećim stavkama. Na sljedećoj slici može se vidjeti obavijest korisniku da je uspješno dodao proizvod.



Slika 18: Kreiranje narudžbenice 3/3 (Izvor: vlastita izrada, 2019.)

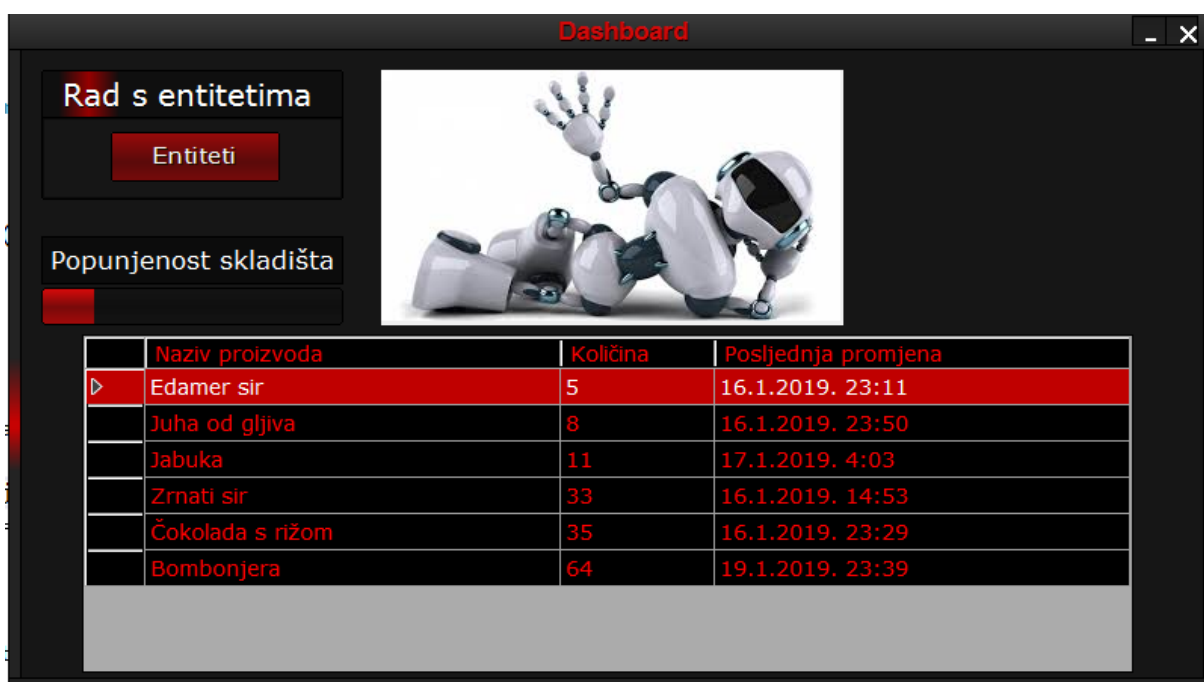


Slika 19: Nedovoljna količina proizvoda (Izvor: vlastita izrada, 2019.)

Ukoliko korisnik pokuša otpremiti više proizvoda nego što se na skladištu nalazi, okidač će mu javiti da je odabrao preveliku količinu proizvoda te će ga obavijestiti o raspoloživoj količini.

4. Zaključak

Skladišni sustav od velike je važnosti u poduzećima koja se bave proizvodnjom ili prodajom. Pravilna strategija upravljanja skladišnim sustavom može biti ključna u vrijeme kad je tržište veliko i rasprostranjeno. Iako smatram Navicat kvalitetnim alatom te se znam dobro njime služiti, najnovija verzija softvera ima određeni bug koji ne dozvoljava da se otvore funkcije unutar Navicata, nego je jedina opcija otvoriti ih preko pgAdmina. Također sam prilikom uređivanja dizajna aplikacije naišao na problem sa izmjenom boje na datagridview kontrolama. Iako je bila postavljena default boja za ćeliju, prilikom ponovnog pokretanja aplikacija svaki put bi se boje resetirale te je dizajn datagridviewa ostao bijele boje. Na sljedećoj slici može se vidjeti kako je aplikacija izgledala.



Slika 20: Inicijalni izgled aplikacije (Izvor: vlastita izrada, 2019.)

Iako sam zadovoljan radom te naučenim unutar PostgreSQL-a, ne sviđa mi se to što je unutar C.Sharpa nezgodno rukovati sa Npgsql komandama za razliku od drukčijih oblika povezivanja. Korištenje Npgsql.dll datoteke za povezivanje bilo je u određenim aspektima programiranja ograničavajuće (primjerice izrada statističkog grafa za skladište). Aktivne baze podataka pokazale su se izrazito korisnima pogotovo u ovom projektu. Nemoguće je zamisliti rad skladišnog sustava bez automatizma kao što je okidač - reopskrbljivanje zaliha okidači izuzetno efikasno rješavaju te omogućavaju fluidnost proizvoda. Iako se prilikom posjeta određenim trgovinama nema kontakta sa skladištem, može se vrlo lako prepoznati koja trgovina ima dobar plan upravljanja skladištem, što daje znatnu prednost. Temporalne baze podataka daju dodatnu dimenziju skladišnom sustavu jer je vrijeme izuzetno važno - vremenski zastoј značio bi velike gubitke za poduzeće, te je važno da skladišni sustav upravljanja bude čvrst oslonac poslovanju tvrtke.

Može se zaključiti da je upravljanje skladišnim aktivnostima jedna od najvažnijih trgo-

vačkih aktivnosti jer skladištenje ima važnu ulogu za proizvodnju i kupce. Nužna je prilagodba tržištu te isto tako i tehnologiji - uvođenje automatizacije u sustav skladištenja korak je dalje u pravom smjeru.

5. Literatura

1. Navicat GUI | DB Admin Tool for PostgreSQL - <https://www.navicat.com/en/>
2. Microsoft Visual Studio IDE - <https://visualstudio.microsoft.com/>
3. PostgreSQL | The world's most advanced open source database - <https://www.postgresql.org/>
4. TaOPis Wiki - <http://autopoiesis.foi.hr/wiki.php?name=Baze+Podataka+-+FOI>

Popis slika

1.	Prikaz tablica modela baze podataka (Izvor: vlastita izrada, 2019.)	3
2.	Prikaz modela baze podataka (Izvor: vlastita izrada, 2019.)	4
3.	Okidač br. 1 - Skladište narudžba (Izvor: vlastita izrada, 2019.)	5
4.	Okidač br. 2 - Skladište otprema (Izvor: vlastita izrada, 2019.)	6
5.	Okidač br. 3 - Provjera minimalne količine (Izvor: vlastita izrada, 2019.)	6
6.	Okidač br. 4 - Provjera ukupne količine (Izvor: vlastita izrada, 2019.)	7
7.	Okidač br. 5 - Brisanje detalja narudžbe (Izvor: vlastita izrada, 2019.)	7
8.	Forma dobrodošlice (Izvor: vlastita izrada, 2019.)	8
9.	Početna forma aplikacije (Izvor: vlastita izrada, 2019.)	9
10.	Rad s entitetima - Dobavljači (Izvor: vlastita izrada, 2019.)	9
11.	Rad s entitetima - Kreiranje dobavljača (Izvor: vlastita izrada, 2019.)	10
12.	Rad s entitetima - Narudžbenice (Izvor: vlastita izrada, 2019.)	10
13.	Rad s entitetima - Detalji narudžbenice (Izvor: vlastita izrada, 2019.)	11
14.	Nezaprimljene narudžbenice 1/2 (Izvor: vlastita izrada, 2019.)	11
15.	Nezaprimljene narudžbenice 2/2 (Izvor: vlastita izrada, 2019.)	12
16.	Kreiranje narudžbenice 1/3 (Izvor: vlastita izrada, 2019.)	12
17.	Kreiranje narudžbenice 2/3 (Izvor: vlastita izrada, 2019.)	13
18.	Kreiranje narudžbenice 3/3 (Izvor: vlastita izrada, 2019.)	14
19.	Nedovoljna količina proizvoda (Izvor: vlastita izrada, 2019.)	14
20.	Inicijalni izgled aplikacije (Izvor: vlastita izrada, 2019.)	15