

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

**Antonio Grgošević**

# **APLIKACIJA ZA VOĐENJE STATISTIKE SKLADIŠTA I PLANIRANJE ZALIHA**

**PROJEKT**

**Varaždin, 2020.**

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

**Antonio Grgošević**

**Matični broj: 44264/15–R**

**Studij: Organizacija poslovnih sustava**

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

**PROJEKT**

**Mentor:**

**Dr. sc. Bogdan Đurić Okreša**

**Varaždin, lipanj 2020.**

*Antonio Grgošević*

### **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**

Izrada aplikacije koja koristi bazu podataka s aktivnim i temporalnim komponentama. Izrađena aplikacija namijenjena je za vođenje statistike skladišta i upravljanje zalihama koje su na skladištu FIFO strategijom.

**Ključne riječi:** aktivna baza podataka, temporalna baza podataka, skladište, implementacija, FIFO strategija, PostgreSQL

# Sadržaj

|   |           |
|---|-----------|
| <b>1. Opis aplikacijske domene</b>  | <b>1</b>  |
| <b>2. Teorijski uvod</b>  | <b>2</b>  |
| 2.1. Aktivna baza podataka  | 2         |
| 2.2. Temporalna baza podataka   | 2         |
| <b>3. Model baze podataka</b>   | <b>3</b>  |
| 3.1. Zaposlenik   | 4         |
| 3.2. Dobavljač  | 4         |
| 3.3. Kupac  | 4         |
| 3.4. Artikl   | 4         |
| 3.5. Narudžba   | 4         |
| 3.6. Narudžba artikla   | 4         |
| 3.7. Radni nalog  | 5         |
| 3.8. Stavka artikla na Radni nalog  | 5         |
| 3.9. Stanje na skladištu  | 5         |
| <b>4. Implementacija</b>  | <b>6</b>  |
| 4.1. Korištenje aplikacije  | 6         |
| 4.2. Implementirane funkcije  | 10        |
| 4.2.1. Funkcija za dodavanje nove narudžbe  | 10        |
| 4.2.2. Funkcija za dodavanje novog radnog naloga i ažuriranje stanja na skladištu | 11        |
| 4.3. Implementacija okidača   | 12        |
| 4.3.1. Provjera minimalne količine  | 12        |
| 4.3.2. Provjera maksimalne količine   | 12        |
| 4.3.3. Dodavanje artikla na stanje na skladištu                                   | 13        |
| 4.3.4. Ažuriranje ukupne količine artikla   | 14        |
| 4.3.5. Provjera minimalne količine kod Radnog naloga                              | 15        |
| 4.4. Implementacija FIFO strategije   | 16        |
| <b>5. Primjeri korištenja</b>   | <b>17</b> |
| 5.1. Artikl   | 17        |
| 5.2. Narudžba   | 18        |
| 5.3. Radni nalog  | 18        |
| 5.4. Stanje na skladištu  | 20        |
| 5.4.1. Stanje nakon izdane narudžbe   | 20        |
| 5.4.2. Stanje nakon izdanog radnog naloga   | 21        |

|                                   |           |
|-----------------------------------|-----------|
| <b>6. Literatura . . . . .</b>    | <b>23</b> |
| <b>7. Zaključak . . . . .</b>     | <b>24</b> |
| <b>Popis literature . . . . .</b> | <b>24</b> |
| <b>Popis slika . . . . .</b>      | <b>25</b> |

# 1. Opis aplikacijske domene

U ovom poglavlju biti će opisana detaljnije aplikacijska domena sustava. Izrađena aplikacija je namijenjena za vođenje statistike skladišta i upravljanje zalihama koje su na skladištu. Sama struktura aplikacijske domene sastoji se od sljedećih elemenata:

- Artikl
- Zaposlenik
- Kupac
- Dobavljač
- Narudžba
- Radni nalog
- Stanje na skladištu

Prvenstveno je važno napomenuti da aplikacijom upravlja zaposlenik koji unosi određene artikle koje se može naručiti, a poslije i prodati kupcu. Skladište funkcionira na način da se prvo naruči određena količina artikala koji onda pristižu na skladište. Pri tome se vodi računa o tome koji zaposlenik je izdao određenu narudžbu, od kojeg dobavljača i na koji datum.

Jedan artikl može biti podijeljen na više paketa, pa se zbog toga artikli naručuju u paketima. Nakon što određeni artikl pristigne na skladište, a pristigne nakon što je izdana narudžba, na stanju na skladištu mu se povećava trenutna količina. Prethodno opisani način zapravo predstavlja ulazak artikla na skladište. Kako bi određeni artikl bio prodan kupcu, odnosno kako bi izašao sa skladišta, potrebno je da zaposlenik unese radni nalog. Prilikom unosa radnog naloga potrebno je da se odabere zaposlenik, koji izdaje taj radni nalog te kupac kojem se izdaje radni nalog. Osim toga potrebno je da zaposlenik unese količinu koja će biti izdana te da odabere točan datum izdavanja radnog naloga. Nakon što je radni nalog izdan, stanje na skladištu mu se smanjuje za izdanu količinu.

## **2. Teorijski uvod**

Izrađena aplikacija koristi aktivnu i temporalnu bazu podataka. U sljedeća dva potpoglavlja će iste biti opisane.

### **2.1. Aktivna baza podataka**

Aktivne baze podataka ne samo da pohranjuju podatke i informacije, već reagiraju na događaje i provode akcije prema tim događajima, npr. umetanje podataka, brisanje podataka, ažuriranje podataka i slično. Aktivno ponašanje u bazi podataka se postiže definiranjem pravila „Događaj-Uvjet-Akcija“. Pojava događaja pokreće procjenu uvjeta te ako se uvjet procjeni istinitim, tada se radnja izvršava. Možemo reći da pravovremeno reagiraju na novonastale situacije [1].

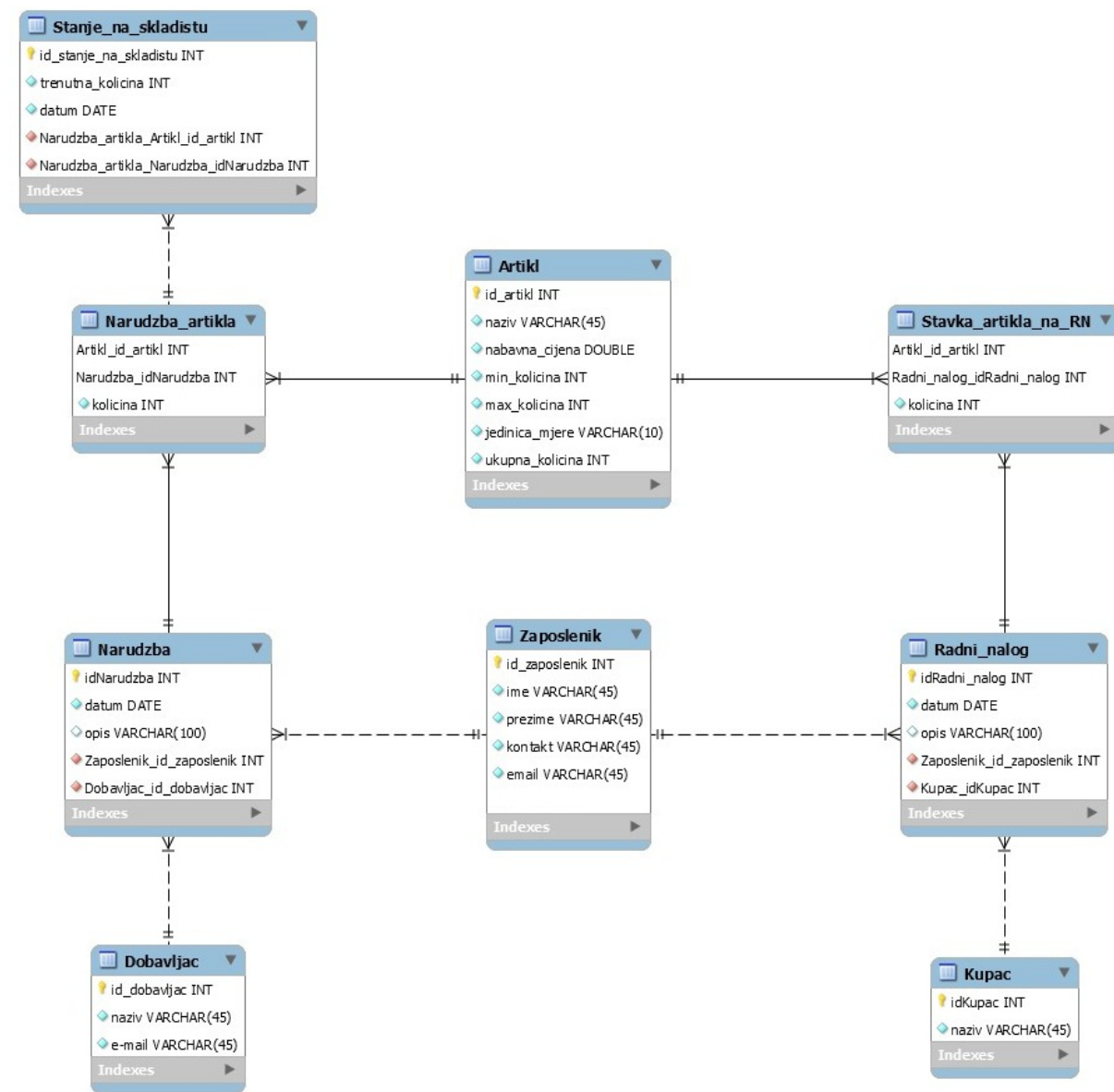
### **2.2. Temporalna baza podataka**

Temporalna baza podataka pohranjuje podatke koji se odnose na vremenske instance. Nudi vremenske tipove podataka i pohranjuje podatke koji se odnose na prošlo, sadašnje ili buduće vrijeme. Temporalna baza podataka pohranjuje zbirku vremenski povezanih podataka. Povezivanjem vremena s podacima, moguće je pohraniti različita stanja baze podataka [2].



### 3. Model baze podataka

Model baze podataka prikazan je preko ERA modela u kojem se nalaze sve potrebne tablice i veze između njih. ERA model izrađen je u MySQL Workbench aplikaciji jer je vrlo jednostavna za korištenje i kreiranje tablica. Osim toga nudi automatsko kreiranje slabih entiteta koji su prijeko potrebni u ovom projektu. Na slici 1. prikazan je izrađeni ERA model.



Slika 1: Prikaz ERA modela baze podataka

Na slici 1. vidljivo je kako se baza podataka sastoji od 7 tablica i 2 slabe entiteta. Slabe entitete predstavljaju *Narudzba artikla* i *Stavka artikla na RN*, odnosno to su tablice koje se kreiraju na vezama više na više.

### **3.1. Zaposlenik**

U tablici Zaposlenik nalaze se podaci o svim zaposlenicima koji mogu upravljati skladištem, a samim time i aplikacijom. Svaki zaposlenik ima svoje ime i prezime, kontakt broj i email. Glavna svrha zaposlenika je unos svih narudžbi i radnih naloga. Na taj način se uvijek zna koji zaposlenik je unio određenu narudžbu ili radni nalog.

### **3.2. Dobavljač**

U tablici Dobavljač nalaze se svi podaci o dobavljačima od kojih se naručuju artikli. Dobavljač ima svoj naziv te email ukoliko ga je potrebno kontaktirati.

### **3.3. Kupac**

Tablica Kupac sadrži osnovne podatke o svim mogućim kupcima, odnosno sadrži njihov naziv. Nakon što se određeni artikl naruči od dobavljača, on pristiže na skladište te se potom prodaje kupcima.

### **3.4. Artikl**

Tablica Artikl sadrži sve podatke o artiklima na skladištu. Svaki artikl ima svoj naziv, cijenu po kojoj se on nabavlja i jedinicu mjere. Također sadrži još i informacije koliko minimalno i maksimalno može biti određenog artikla na skladištu te kolika je ukupna količina svih paketa određenog artikla. Kod unosa artikla, ukupna količina je uvijek 0 te se ista mijenja prilikom izdavanje narudžbe ili radnog naloga.

### **3.5. Narudžba**

Tablica Narudžba sadrži sve potrebne informacije o pojedinim narudžbama. Najvažniji podatak kod narudžbe je datum, koji predstavlja datum kada je određena narudžba izdana. Također sadrži i opis narudžbe ukoliko je potrebno nešto posebno napomenuti. Na nju se povezuje tablica Zaposlenik i Dobavljač kako bi se znalo koji je zaposlenik izdao narudžbu i od kojeg dobavljača.

### **3.6. Narudžba artikla**

Ova tablica predstavlja slabi entitet koji služi za dodavanje artikala na narudžbu. Tu se zapisuje količina koja se naručuje za svaki artikl te se za tu količinu povećava trenutna količina artikla na stanju na skladištu. Samim time ova tablica je povezana s tablicom Stanje na skladištu i Artikl.

### **3.7. Radni nalog**

Tablica Radni nalog sadrži sve informacije o pojedinim radnim nalogima te se koristi kao izlazak artikla sa skladišta, odnosno za smanjivanje količine pojedinog artikla. Sadrži podatke o opisu i datumu izdanog radnog naloga. Povezana je s tablicom Zaposlenik i Kupac kako bi se znalo koji zaposlenik je izdao radni nalog i kojem kupcu. Osim toga povezana je i s tablicom Stavka artikla na radni nalog koja će biti detaljnije opisana u idućem potpoglavlju.

### **3.8. Stavka artikla na Radni nalog**

Ova tablica predstavlja slabi entitet koji se koristi kako bi pojedini artikl mogao izaći sa skladišta, odnosno kako bi se prodao određenom kupcu. Sukladno tome, povezana je s tablicom Artikl i Radni nalog preko vanjskih ključeva te u sebi sadrži količinu za koju se pojedini artikl smanjuje sa skladišta.

### **3.9. Stanje na skladištu**

Ova tablica služi za evidenciju trenutnih količina svakog dostupnog artikla. Trenutna količina se povećava ili smanjuje, ovisno o tome je li izdana narudžba ili radni nalog. Osim toga sadrži i datum koji predstavlja točan datum kada je pojedini artikl dospio na skladište, odnosno kada je naručen.

## 4. Implementacija

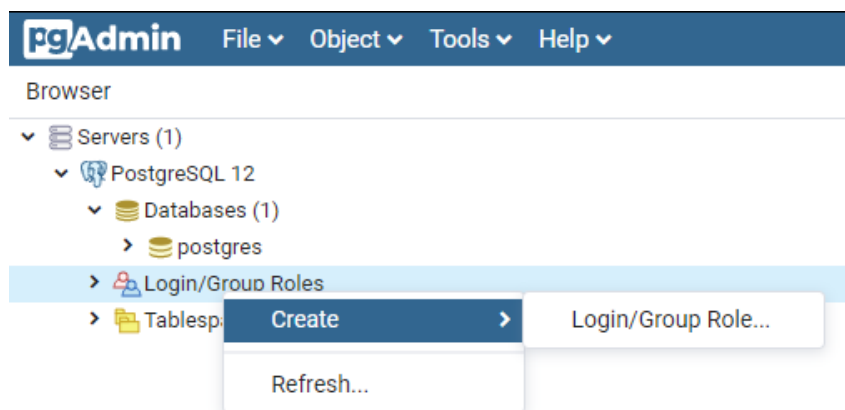
Baza podataka izrađena je u PostgreSQL-u i za to je korišten alat pgAdmin 4. Sve tablice i veze među njima se kreiraju preko grafičkog sučelja odabirom opcija. Iz tog razloga nije bilo potrebno ručno pisati SQL kod za isto.

### 4.1. Korištenje aplikacije

Kako bi se aplikacija i baza podataka instalirale potrebno je napraviti sljedeće:

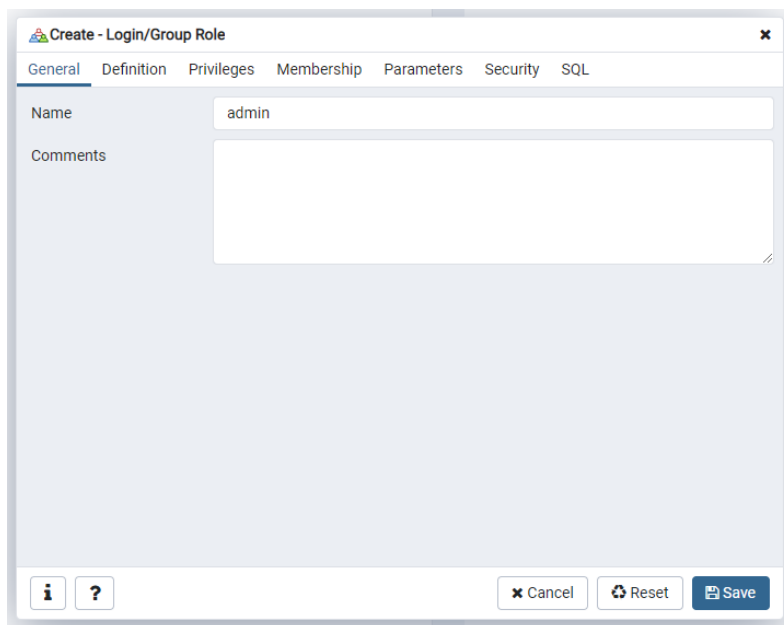
- Preko setup datoteke instalirati aplikaciju na željeno mjesto
- Na lokalno serveru od izabrane postgres baze podataka kreirati korisnika *admin* s lozinkom *admin* i postaviti ga na superuser, odnosno dati mu sva dopuštenja
- U nekoj od postgres baza podataka kreirati bazu podataka s imenom *TBP Baza* (preporučeno je koristiti pgAdmin 4)
- Otvoriti datoteku SkladisteBackUp.sql te kopirati njezin sadržaj u prozoru za izvršavanje SQL upita

U nastavku ovog poglavlja bit će prikazane slike kreiranja prethodno navedenih koraka u alatu pgAdmin 4.



Slika 2: Prikaz opcije za kreiranje novog korisnika

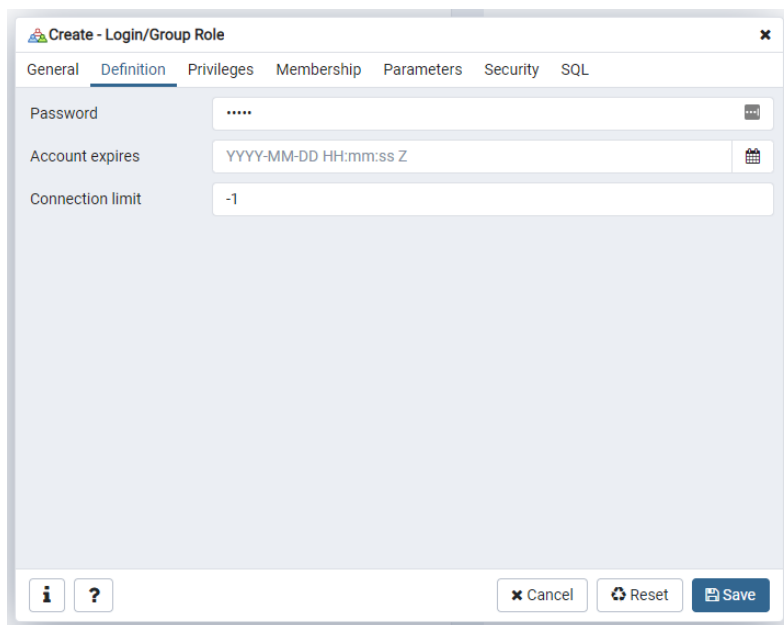
Na slici 3. vidljiv je prikaz prozora za kreiranje novog korisnika pod nazivom *admin*. U kartici privilegije dane su mu sve privilegije čime je zapravo postao superuser.



The screenshot shows a window titled "Create - Login/Group Role" with a close button (X) in the top right corner. Below the title bar are several tabs: "General", "Definition", "Privileges", "Membership", "Parameters", "Security", and "SQL". The "General" tab is selected. It contains two input fields: "Name" with the value "admin" and "Comments" which is empty. At the bottom of the window, there are three buttons: "Cancel", "Reset", and "Save".

Slika 3: Ime korisnika - admin

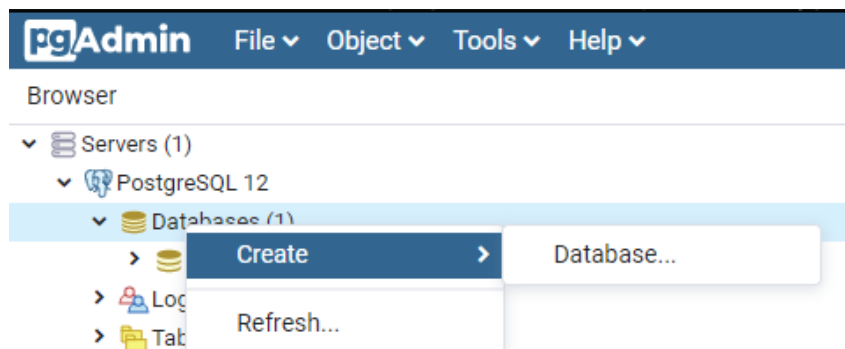
Na slici 4. vidljiv je prikaz dijela kod definiranja lozinke. Korisniku *admin* zbog jednostavnosti je dana identična lozinka imenu korisnika, odnosno lozinka je *admin*.



The screenshot shows the same "Create - Login/Group Role" window, but with the "Definition" tab selected. It contains three input fields: "Password" with the value "admin" (masked with dots), "Account expires" with the value "YYYY-MM-DD HH:mm:ss Z", and "Connection limit" with the value "-1". At the bottom of the window, there are three buttons: "Cancel", "Reset", and "Save".

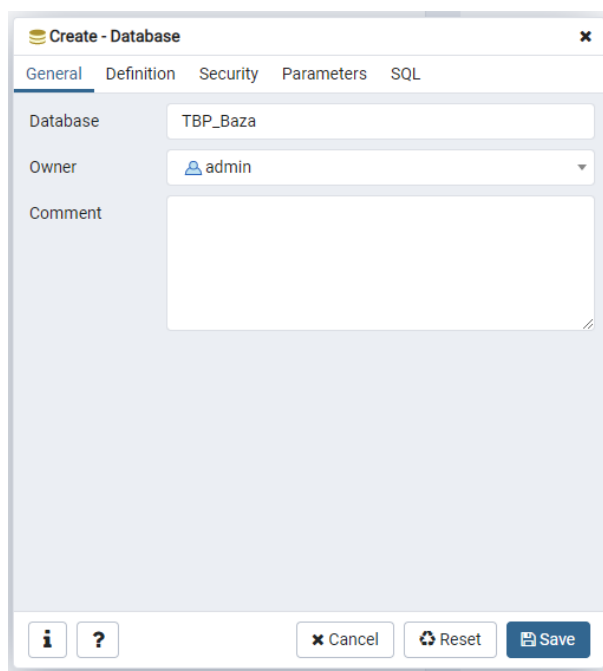
Slika 4: Lozinka korisnika - admin

Na slici 5. vidljiv je prikaz izbornika u kojem desnim klikom na Databases je moguće kreirati novu bazu.



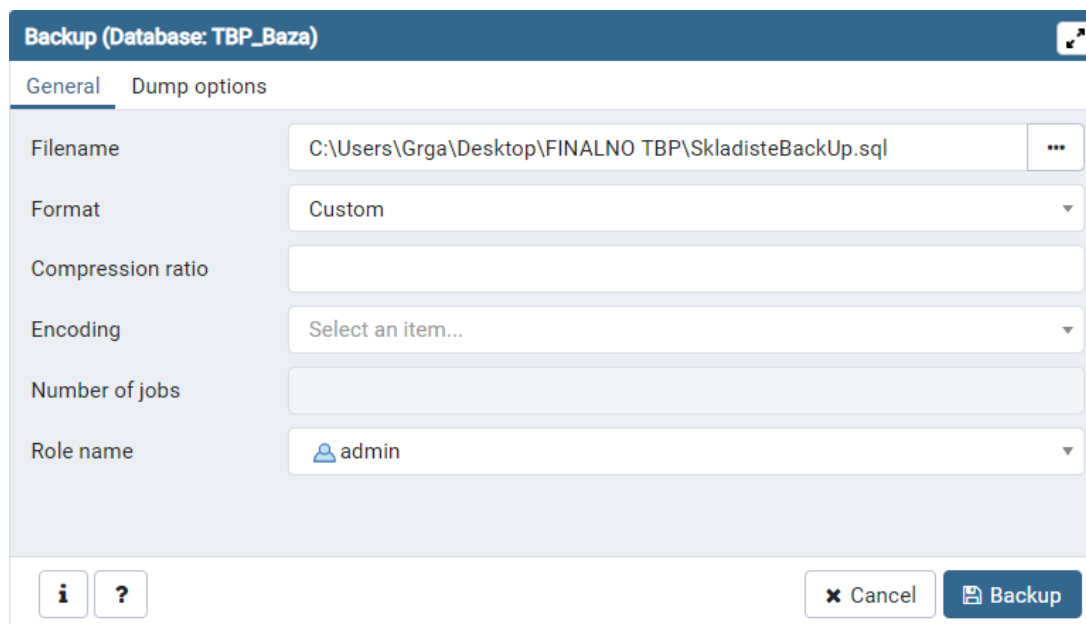
Slika 5: Kreiranje baze podataka

Na slici 6. vidljiv je prikaz danog imena baze podataka za ovaj projekt pod nazivom *TBP Baza*



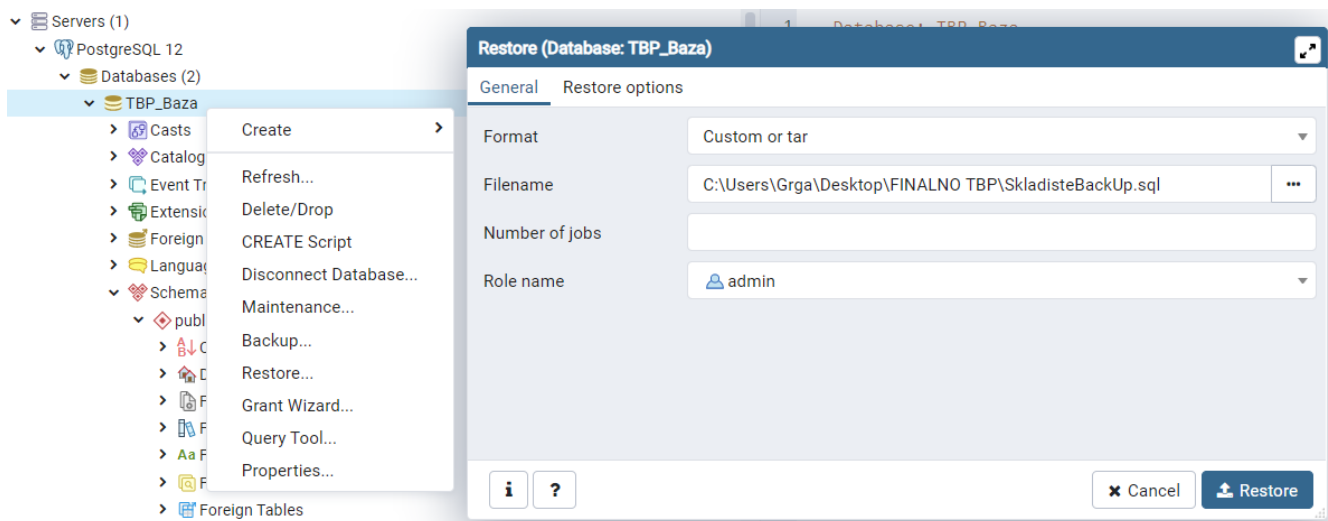
Slika 6: Ime baze podataka - TBP Baza

Na slici 7. prikaz je prozor za backup kreirane baze podataka. Potrebno je odabrati datoteku, pod nazivom *SkladisteBackUp*, u kojoj će biti pohranjene sve kreirane tablice, okidači i funkcije.



Slika 7: Backup baza podataka

Na slici 8. vidljiv je prozor za restore baze podatka. Tu je također potrebno odabrati datoteku, pod nazivom *SkladisteBackUp*. Pritiskom na gumb Restore, ubacit će se kreirana baza podataka te sve njezine povezane tablice, okidači i funkcije.



Slika 8: Restore baza podataka

## 4.2. Implementirane funkcije

U iduća dva potpoglavlja će biti prikazane dvije izrađene funkcije te će biti objašnjena njihova svrha u aplikaciji.

### 4.2.1. Funkcija za dodavanje nove narudžbe

Prvenstveno je kreirana funkcija za dodavanje nove narudžbe. Obzirom da je potrebno prvo dodati podatke u tablicu Narudžbu, a tek onda u slabi entitet, odnosno tablicu Narudžba artikla, kreirana je sljedeća funkcija:

```
CREATE OR REPLACE FUNCTION public.funkcija_dodavanje_narudzbe(  
    zaposlenik_id integer,  
    dobavljac_id integer,  
    datum_narudzbe date,  
    opis_narudzbe text,  
    kolicina_narudzbe integer,  
    artikl_id integer)  
RETURNS void  
LANGUAGE 'sql'  
  
COST 100  
VOLATILE  
  
AS $BODY$  
INSERT INTO public."Narudzba"(datum, opis, "Zaposlenik_id_zaposlenik", "  
    Dobavljac_id_dobavljac")  
VALUES (datum_narudzbe, opis_narudzbe, zaposlenik_id, dobavljac_id);  
  
INSERT INTO public."Narudzba_artikla"(kolicina, "Artikl_id_artikl", "  
    Narudzba_id_narudzba")  
VALUES (kolicina_narudzbe, artikl_id, (SELECT "idNarudzba" FROM public."Narudzba"  
    ORDER BY "idNarudzba" DESC LIMIT 1));  
$BODY$;  
  
ALTER FUNCTION public.funkcija_dodavanje_narudzbe(integer, integer, date, text,  
    integer, integer)  
OWNER TO admin;
```



### 4.2.2. Funkcija za dodavanje novog radnog naloga i ažuriranje stanja na skladištu

Druga kreirana funkcija u ovom projektu služi za dvije stvari. Prva je za dodavanje novog radnog naloga, odnosno prvo se dodaju određeni podaci u tablicu Radni nalog, a potom u slabi entitet, odnosno tablicu Stavka artikla na RN. Druga bitna stvar kod ove funkcije je da ona zapravo nakon kreiranja radnog naloga, smanjuje količinu artikla, odnosno ažurira stanje na skladištu. U nastavku je prikazana prethodno navedena funkcija.

```
CREATE OR REPLACE FUNCTION public.funkcija_oduzimanje_sa_stanja(  
    zaposlenik_id integer,  
    kupac_id integer,  
    datum_radni_nalog date,  
    opis_radni_nalog text,  
    kolicina_radni_nalog integer,  
    artikl_id integer,  
    radni_nalog_id integer,  
    stanje_id integer)  
    RETURNS void  
    LANGUAGE 'sql'  
  
    COST 100  
    VOLATILE  
  
AS $BODY$  
INSERT INTO public."Radni_nalog"(datum, opis, "Zaposlenik_id_zaposlenik", "  
    Kupac_id_kupac")  
VALUES (datum_radni_nalog, opis_radni_nalog, zaposlenik_id, kupac_id);  
  
INSERT INTO public."Stavka_artikla_na_RN"(kolicina, "Radni_nalog_id_Radni_nalog", "  
    Stavka_Artikl_id_artikl")  
VALUES (kolicina_radni_nalog, (SELECT "idRadni_nalog" FROM public."Radni_nalog"  
    ORDER BY "idRadni_nalog" DESC LIMIT 1), artikl_id);  
  
UPDATE public."Stanje_na_skladistu" SET trenutna_kolicina = trenutna_kolicina -  
    kolicina_radni_nalog  
WHERE "idStanjaNaSkladistu" = stanje_id;  
  
$BODY$;  
  
ALTER FUNCTION public.funkcija_oduzimanje_sa_stanja(integer, integer, date, text,  
    integer, integer, integer, integer)  
    OWNER TO admin;
```

## 4.3. Implementacija okidača

U ovom poglavlju će biti prikazani svi implementirani okidači koji su bili potrebni za funkcioniranje aplikacije skladišta.

### 4.3.1. Provjera minimalne količine

Prvi okidač služi za provjeravanje unesene količine prilikom naručivanja pojedinog artikla. Korisnik ne smije unijeti količinu koja je manja od definirane minimalne količine određenog artikla. Ta minimalna količina je definirana u tablici `Artikl` te istu korisnik unosi prilikom unosa artikla. U nastavku je prikazana implementacija prvog okidača.

```
CREATE FUNCTION public.provjera_min_kolicine()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$BEGIN

IF new.kolicina > (SELECT min_kolicina FROM public."Artikl" WHERE new."
    Artikl_id_artikl" = "idArtikl") THEN
    RETURN NEW;
ELSE
    RAISE EXCEPTION '%', 'Unesena_kolicina_mora_biti_ćvea_od_definirane_
        minimalne_kolicine_tog_artikla!';
END IF;

END;$BODY$;

ALTER FUNCTION public.provjera_min_kolicine()
    OWNER TO admin;

CREATE TRIGGER provjera_artikla_min_kolicine
    BEFORE INSERT
    ON public."Narudzba_artikla"
    FOR EACH ROW
    EXECUTE PROCEDURE public.provjera_min_kolicine();
```

### 4.3.2. Provjera maksimalne količine

Drugi okidač također služi za provjeravanje unesene količine prilikom naručivanja pojedinog artikla. Atribut ukupna količina za svaki artikl označava zbroj svih paketa određenog artikla. Obzirom da se jedan artikl može naručiti u više paketa, u ovom okidaču se zapravo provjerava je li unesena količina u narudžbi manja ili jednaka od razlike definirane maksimalne količine i ukupne količine artikla. Ukoliko je, tada se artikl može naručiti. U suprotnom javlja poruku o neuspješnoj narudžbi artikla. U nastavku je prikazana implementacija drugog okidača.

```

CREATE FUNCTION public.provjera_max_kolicine()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
IF new.kolicina <= (SELECT max_kolicina FROM public."Artikl" WHERE new."
    Artikl_id_artikl" = "idArtikl")-
(SELECT ukupna_kolicina FROM public."Artikl" WHERE new."Artikl_id_artikl" = "
    idArtikl") THEN
    RETURN NEW;
ELSE
    RAISE EXCEPTION '\%', 'Unesena_kolicina_mora_biti_manja_od_maksimalne_
        kolicine_tog_artikla!';
END IF;
END;
$BODY$;
ALTER FUNCTION public.provjera_max_kolicine()
    OWNER TO admin;

CREATE TRIGGER provjera_artikla_max_kolicine
    BEFORE INSERT
    ON public."Narudzba_artikla"
    FOR EACH ROW
    EXECUTE PROCEDURE public.provjera_max_kolicine();

```

### 4.3.3. Dodavanje artikla na stanje na skladištu

Sljedeći okidač izvršava se nakon dodavanja nove narudžbe te se dodaje naručeni artikl na skladište. Osim toga dodaje se njegova trenutna količina i datum kada je izdana narudžba. Ovim okidačem artikli zapravo ulaze na skladište. U nastavku je prikazana njegova implementacija.

```

CREATE FUNCTION public.dodavanje_na_stanje()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
INSERT INTO public."Stanje_na_skladistu"(trenutna_kolicina, datum, "
    Artikl_id_narudzba_artikla", "Narudzba_id_Narudzba_artikala")
VALUES(new.kolicina, (SELECT datum FROM public."Narudzba" WHERE "idNarudzba"=new."
    Narudzba_id_narudzba" limit 1), new."Artikl_id_artikl", new."
    Narudzba_id_narudzba");
RETURN NEW;
END;
$BODY$;
ALTER FUNCTION public.dodavanje_na_stanje()
    OWNER TO admin;

```

```
CREATE TRIGGER dodavanje_artikla_na_stanje
  AFTER INSERT
  ON public."Narudzba_artikla"
  FOR EACH ROW
  EXECUTE PROCEDURE public.dodavanje_na_stanje();
```

#### 4.3.4. Ažuriranje ukupne količine artikla

Sljedeći okidač služi za ažuriranje atributa ukupne količine artikla. Okidač se izvršava nakon što je ažurirano stanje na skladištu, bilo to izdanom narudžbom ili radnim nalogom. Sukladno tome, ukupna količina u tablici Artikl se povećava ili smanjuje. U nastavku je prikazana implementacija tog okidača.

```
CREATE FUNCTION public."azuriranje_ukupne_kolicine"()
  RETURNS trigger
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN

UPDATE public."Artikl" SET ukupna_kolicina = (SELECT SUM(trenutna_kolicina)
FROM public."Stanje_na_skladistu"
WHERE "Artikl_id_narudzba_artikla" =new."Artikl_id_narudzba_artikla"
GROUP BY "Artikl_id_narudzba_artikla")
WHERE new."Artikl_id_narudzba_artikla" ="idArtikl";
RETURN NEW;

END;
$BODY$;

ALTER FUNCTION public."azuriranje_ukupne_kolicine"()
  OWNER TO admin;

CREATE TRIGGER "azuriranje_ukupne_kolicine_artikla"
  AFTER INSERT OR UPDATE OF trenutna_kolicina
  ON public."Stanje_na_skladistu"
  FOR EACH ROW
  EXECUTE PROCEDURE public."azuriranje_ukupne_kolicine"();
```

### 4.3.5. Provjera minimalne količine kod Radnog naloga

Sljedeći okidač služi kao provjera prilikom izdavanja radnog naloga. Tu nije moguće izdati kolicinu artikla koja bi dovela ukupnu količinu ispod definirane minimalne količine tog artikla. U nastavku slijedi prikaz implementacije tog okidača.

```
BEGIN

IF new.kolicina <= ((SELECT ukupna_kolicina FROM public."Artikl" WHERE new."
    Stavka_Artikl_id_artikl" = "idArtikl")-
(SELECT min_kolicina FROM public."Artikl" WHERE new."Stavka_Artikl_id_artikl" = "
    idArtikl"))
THEN
    RETURN NEW;
ELSE
    RAISE EXCEPTION '\%', 'Unesena_kolicina_ne_smije_dovesti_ukupnu_kolicinu_
        artikla_ispod_minimalne_kolicine!';
END IF;
END;

CREATE TRIGGER provjera_radni_nalog_min_kolicina
    BEFORE INSERT
    ON public."Stavka_artikla_na_RN"
    FOR EACH ROW
    EXECUTE PROCEDURE public.provjera_radni_nalog_min();
```

## 4.4. Implementacija FIFO strategije

FIFO (engl. *First-In, First-Out*) strategija je strategija koja se koristi u svrhu pretpostavke protoka troškova u proračunu troškova prodane robe [3]. Korištenjem ove metode potrebno je prvo prodati one artikle koji su prvi ušli na skladište, odnosno najstarije artikle. Ulazak na skladište gleda se unesenim datumom narudžbe. U nastavku je prikazan kod u programskom jeziku CSharp gdje se prvo prikazuju artikli s najstarijim datumom te se ne prikazuju artikli koji imaju trenutnu količinu manju od definirane minimalne količine artikla.

```
string selektirani = cmbArtikli.Text;
string connectionString = "Server=127.0.0.1;_Port=5432;_User_Id=admin;_Password=
    admin;_Database=TBP_Baza";
NpgsqlConnection npgsqlConnection = new NpgsqlConnection(connectionString);
npgsqlConnection.Open();

NpgsqlDataAdapter podaci = new NpgsqlDataAdapter();

string upit = $"SELECT_sns.\"idStanjaNaSkladistu\"_AS_id_stanja,_a.\"idArtikl\"_AS_
    ID,_a.naziv_AS_Naziv,_a.nabavna_cijena_AS_Cijena,_\" +
    \"$a.min_kolicina_AS_MIN,_a.max_kolicina_AS_MAX,_\" +
    \"$sns.trenutna_kolicina_AS_Trenutna_kolicina,_sns.datum_AS_Datum_\" +
    \"$FROM_public.\"Artikl\"_\"a_\" +
    \"$JOIN_public.\"Stanje_na_skladistu\"_\"sns_ON_sns.\"Artikl_id_narudzba_artikla\"_\"
    =_a.\"idArtikl\"_\"\" +
    \"$WHERE_a.naziv=_\"{selektirani}\"_AND_sns.trenutna_kolicina_>_a.min_kolicina_\" +
    \"$ORDER_BY_sns.datum_ASC\";

podaci.SelectCommand = new NpgsqlCommand(upit, npgsqlConnection);
DataTable podaciTablice = new DataTable();
podaci.Fill(podaciTablice);

BindingSource bindingSource = new BindingSource();
bindingSource.DataSource = podaciTablice;
dgvPrikazArtikala.DataSource = bindingSource;
npgsqlConnection.Close();
```

Također implementirano je i da se za izdavanje radnog naloga može selektirati samo prvi artikl koji je prethodnim kodom sortiran po datumu. Taj dio implementacije prikazan je u nastavku.

```
if (dgvPrikazArtikala.Rows[0].Selected == true)
    btnIzdajNalog.Enabled = true;
else
    btnIzdajNalog.Enabled = false;
```

## 5. Primjeri korištenja

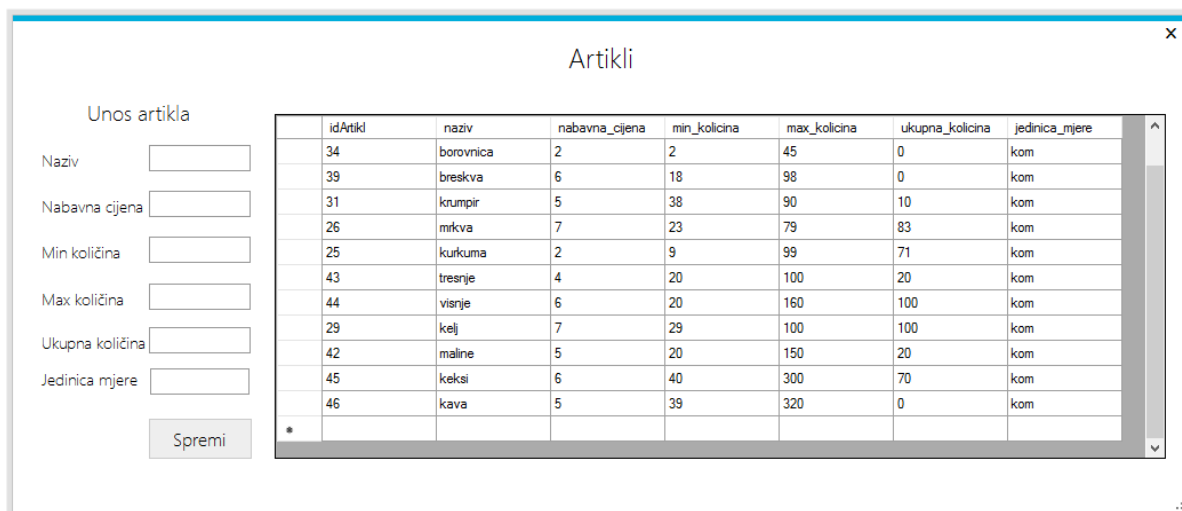
Prilikom otvaranja aplikacije, prikazuje se početni prikaz izbornika sa slike 9. U izborniku je ponuđeno 8 opcija te pritiskom na neki od tih gumba prikazat će se opcije koje on nudi. Ukoliko se ne želi pregledavati ništa od navedenog moguće je izaći iz aplikacije pritiskom na gumb "Izlaz". U nastavku ovog rada bit će prikazane najvažnije funkcionalnosti.



Slika 9: Prikaz izbornika aplikacije

### 5.1. Artikl

Prilikom unosa artikla potrebno je popuniti sva ponuđena polja, pri čemu je potrebno ukupnu količinu postaviti na 0. Kasnije će se unosom narudžbe ili radnog naloga ukupna količina povećavati, odnosno smanjivati. Na slici 10. vidljiv je prikaz svih unesenih artikala.



Slika 10: Prikaz unesenih artikala

## 5.2. Narudžba

U prvoj tablici na formi Narudžba vidljivi su svi artikli koje je moguće naručiti dok je na drugoj tablici prikazan popis svih narudžba. Selektiranjem npr. artikla kave i pritiskom na gumb "Unos narudžbe" otvara se nova forma "Unos narudžbe". Na toj formi moguće je odabrati koji zaposlenik izdaje narudžbu, od kojeg dobavljača, na koji datum i količinu koju želi naručiti, kao i opis narudžbe koji nije obavezan za unijeti. Važno je za napomenuti da se ne može unijeti količina koja je manja od definirane minimalne količine i maksimalne količine tog artikla. Isto je implementirano s 2 okidača koji je prikazan u poglavljima 4.3.1. i 4.3.2.

**Unos narudžbe**

Zaposlenik: maric

Dobavljač: Dobavljač1

Naziv artikla: kava

Količina: 70

Datum: Monday, June 15, 2022

Opis: opis

Naruči

**Narudžba**

| idArtikl | naziv   | nabavna_cijena | min_kolicina | max_kolicina | ukupna_kolicina | jedinica_mjere |
|----------|---------|----------------|--------------|--------------|-----------------|----------------|
| 26       | mikva   | 7              | 23           | 79           | 83              | kom            |
| 25       | kurkuma | 2              | 9            | 99           | 71              | kom            |
| 43       | tresnje | 4              | 20           | 100          | 20              | kom            |
| 44       | visnje  | 6              | 20           | 160          | 100             | kom            |
| 29       | kelj    | 7              | 29           | 100          | 100             | kom            |
| 42       | maline  | 5              | 20           | 150          | 20              | kom            |
| 45       | kekksi  | 6              | 40           | 300          | 70              | kom            |
| 46       | kava    | 5              | 39           | 320          | 0               | kom            |

Unos narudžbe Pogledaj stanje

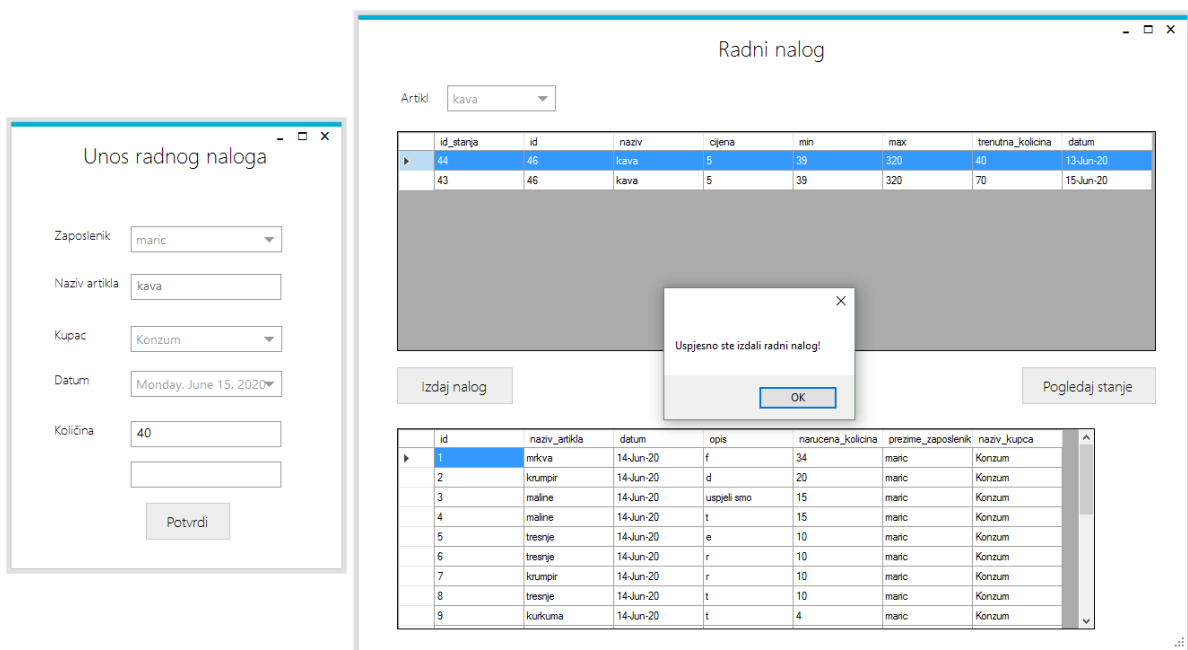
| id | naziv_artikla | datum     | opis    | narucena_kolicina | prezime_zaposlenik | naziv_dobavljacki |
|----|---------------|-----------|---------|-------------------|--------------------|-------------------|
| 43 | mikva         | 14-Jun-20 | gle sad | 20                | maric              | Dobavljač1        |
| 44 | mikva         | 14-Jun-20 |         | 35                | maric              | Dobavljač1        |
| 45 | mikva         | 12-Jun-20 | D       | 32                | maric              | Dobavljač1        |
| 47 | kurkuma       | 14-Jun-20 |         | 54                | maric              | Dobavljač1        |
| 48 | kurkuma       | 14-Jun-20 | k       | 21                | maric              | Dobavljač1        |
| 49 | krumpir       | 13-Jun-20 | opis    | 40                | ivic               | Dobavljač1        |
| 55 | maline        | 11-Jun-20 | rrr     | 35                | ivic               | Dobavljač1        |
| 56 | tresnje       | 14-Jun-20 | e       | 30                | maric              | Dobavljač1        |
| 60 | visnje        | 15-Jun-20 | nema    | 40                | maric              | Dobavljač1        |

Slika 11: Prikaz unosa narudžbe

## 5.3. Radni nalog

Na slici 12. prikazan je početni izgled forme Radni nalog. Odabirom artikla u padajućem izborniku prikazuju se svi paketi tog artikla, u ovom slučaju svi paketi kave. Osim prijašnje unesene paketa od 70 komada, izvršila se narudžba za još jedan paket od 40 komada. Obzirom da je korištena FIFO strategija, svi paketi kave su sortirani po datumu tako da je prvi prikazani paket onaj koji je prvi naručen, odnosno koji je najstariji. Na slici 12. vidljivo je kako je za prvi paket moguće izdati radni nalog, odnosno omogućen je pritisak na gumb "Izdaj nalog". Nakon unesene količine od 40 komada javlja se poruka da je nalog uspješno izdan.





Slika 12: Prikaz unosa narudžbe

Međutim, ukoliko korisnik želi izdati radni nalog za drugi paket, a prvi još nije izdan, tada mu se onemogućuje pritisak na gumb "Izdaj nalog" što je vidljivo na slici 13. Na taj način zaposlenik aplikacije je primoran izdati radni nalog za prvi paket kave. Time se poštuje korištenje FIFO strategije.



Slika 13: Prikaz unosa narudžbe

## 5.4. Stanje na skladištu

U sljedećim potpoglavljima bit će prikazano i opisano što se događa nakon što se unesu narudžba i radni nalog nekog artikla.

### 5.4.1. Stanje nakon izdane narudžbe

Nakon što se artikl naručio, on dopijeva na skladište gdje je vidljiva trenutna količina koja predstavlja količinu koja se naručila. Obzirom da je u poglavlju 5.2 prikazano da je naručeno 70 kavi, isto je evidentirano i na stanju na skladištu na slici 12.



| id | naziv   | cijena | min | max | trenutna_kolicina | datum     |
|----|---------|--------|-----|-----|-------------------|-----------|
| 31 | krumpir | 5      | 38  | 90  | 10                | 13-Jun-20 |
| 26 | mrkva   | 7      | 23  | 79  | 16                | 14-Jun-20 |
| 25 | kurkuma | 2      | 9   | 99  | 50                | 14-Jun-20 |
| 43 | tresnje | 4      | 20  | 100 | 20                | 14-Jun-20 |
| 44 | visnje  | 6      | 20  | 160 | 40                | 15-Jun-20 |
| 44 | visnje  | 6      | 20  | 160 | 60                | 12-Jun-20 |
| 29 | kelj    | 7      | 29  | 100 | 29                | 14-Jun-20 |
| 29 | kelj    | 7      | 29  | 100 | 71                | 13-Jun-20 |
| 42 | meline  | 5      | 20  | 150 | 20                | 11-Jun-20 |
| 45 | keksi   | 6      | 40  | 300 | 70                | 15-Jun-20 |
| 46 | kava    | 5      | 39  | 320 | 70                | 15-Jun-20 |

\* < >

NARUDŽBA RADNI NALOG

Slika 14: Prikaz stanja na skladištu nakon unesene narudžbe

Osim toga povećava se ukupna količina tog artikla kave na formi Artikli. Naručena su 2 paketa artikla kave, jedan od 70 komada i jedan od 40 komada pa je ukupna količina artikla kave 110. Isto je prikazano na slici 15.

Artikli

Unos artikla

Naziv

Nabavna cijena

Min količina

Max količina

Ukupna količina

Jedinica mjere

| idArtikl | naziv     | nabavna_cijena | min_kolicina | max_kolicina | ukupna_kolicina | jedinica_mjere |
|----------|-----------|----------------|--------------|--------------|-----------------|----------------|
| 34       | borovnica | 2              | 2            | 45           | 0               | kom            |
| 39       | breskva   | 6              | 18           | 98           | 0               | kom            |
| 31       | krumpir   | 5              | 38           | 90           | 10              | kom            |
| 26       | mrkva     | 7              | 23           | 79           | 83              | kom            |
| 25       | kurkuma   | 2              | 9            | 99           | 71              | kom            |
| 43       | tresnje   | 4              | 20           | 100          | 20              | kom            |
| 44       | visnje    | 6              | 20           | 160          | 100             | kom            |
| 29       | kelj      | 7              | 29           | 100          | 100             | kom            |
| 42       | maline    | 5              | 20           | 150          | 20              | kom            |
| 45       | keksi     | 6              | 40           | 300          | 70              | kom            |
| 46       | kava      | 5              | 39           | 320          | 110             | kom            |
| *        |           |                |              |              |                 |                |

Slika 15: Prikaz ukupne količine artikla

## 5.4.2. Stanje nakon izdanog radnog naloga

Nakon što se izdao radni nalog, tada mu se trenutna količina sa stanja na skladištu umanjuje za unesenu količinu u radnom nalogu. Na slici 16. vidljiva su 2 paketa, od kojih je jedan od 70 komada, a drugi ima 0 zbog izdanog radnog naloga gdje mu se smanjila količina za 40.

Stanje na skladištu

| id | naziv   | cijena | min | max | trenutna_kolicina | datum     |
|----|---------|--------|-----|-----|-------------------|-----------|
| 44 | visnje  | 6      | 20  | 160 | 60                | 12-Jun-20 |
| 29 | kelj    | 7      | 29  | 100 | 29                | 14-Jun-20 |
| 29 | kelj    | 7      | 29  | 100 | 71                | 13-Jun-20 |
| 42 | maline  | 5      | 20  | 150 | 20                | 11-Jun-20 |
| 45 | keksi   | 6      | 40  | 300 | 70                | 15-Jun-20 |
| 46 | kava    | 5      | 39  | 320 | 70                | 15-Jun-20 |
| 26 | mrkva   | 7      | 23  | 79  | 0                 | 05-Jun-20 |
| 26 | mrkva   | 7      | 23  | 79  | 22                | 12-Jun-20 |
| 32 | jabuka  | 7      | 30  | 230 | 38                | 26-May-20 |
| 25 | kurkuma | 2      | 9   | 99  | 0                 | 14-Jun-20 |
| 46 | kava    | 5      | 39  | 320 | 0                 | 13-Jun-20 |
| *  |         |        |     |     |                   |           |

Slika 16: Prikaz stanja na skladištu nakon izdanog radnog naloga

Osim smanjivanja sa stanja na skladištu, artiklu se smanjuje i ukupna količina na formi Artikl. Vidljivo je kako se s prijašnjih 110 komada, koji su prikazani na slici 15., ukupna količina smanjila za 40, odnosno za količinu koja se prodala određenom kupcu. Isto je prikazano na slici 17.

### Artikli

#### Izmjena artikla

Naziv

Nabavna cijena

Min količina

Max količina

Ukupna količina

Jedinica mjere

| idArtikl | naziv     | nabavna_cijena | min_kolicina | max_kolicina | ukupna_kolicina | jedinica_mjere |
|----------|-----------|----------------|--------------|--------------|-----------------|----------------|
| 26       | mrkva     | 7              | 23           | 79           | 39              | kom            |
| 32       | jabuka    | 7              | 30           | 230          | 38              | kom            |
| 25       | kurkuma   | 2              | 9            | 99           | 50              | kom            |
| 46       | kava      | 5              | 39           | 320          | 70              | kom            |
| 30       | MARAMA    | 10             | 20           | 200          | 0               | kom            |
| 34       | borovnica | 2              | 2            | 45           | 0               | kom            |
| 39       | breskva   | 6              | 18           | 98           | 0               | kom            |
| 31       | krumpir   | 5              | 38           | 90           | 10              | kom            |
| 43       | tresnje   | 4              | 20           | 100          | 20              | kom            |
| 44       | visnje    | 6              | 20           | 160          | 100             | kom            |
| 29       | kelj      | 7              | 29           | 100          | 100             | kom            |
| 42       | maline    | 5              | 20           | 150          | 20              | kom            |
| 45       | kekici    | 6              | 40           | 300          | 70              | kom            |

Slika 17: Prikaz ukupne količine artikla nakon izdanog radnog naloga

## 6. Literatura

[1] X.Li ,S.Chapa, J.Medina, Structural Model of ECA Rules in Active Database, 2002. godina, Dostupno: [https://www.researchgate.net/publication/220887627\\_A\\_Structural\\_Model\\_of\\_ECA\\_Rules\\_in\\_Active\\_Database](https://www.researchgate.net/publication/220887627_A_Structural_Model_of_ECA_Rules_in_Active_Database), Preuzeto: 15.6.2020.

[2] J.Patel, Temporal Database System, 2003.godina, Dostupno: <https://pdfs.semanticscholar.org/6aca/8dc7994af13f05022ce1531d99eedb64f797.pdf>, Preuzeto: 15.6.2020.

[3] FreshBooks, What is FIFO Method: Definition and Example, Dostupno: <https://www.freshbooks.com/hub/accounting/what-is-fifo> , Preuzeto: 15.6.2020.

## 7. Zaključak

Ovaj projekt izrađen je u alatu pgAdmin 4 s kojim sam se prvi put susreo. Sam alat vrlo je jednostavan za korištenje te sam se brzo snašao u njemu. Kroz samu izradu projekta prisjetio sam se PostgreSQL-a i svoje dosadašnje znanje doveo na višu razinu. Korištenje aktivnih i temporalnih baza podataka u ovoj temi ima velikog smisla jer se pomoću njih može ostvariti upravljanje zalihama. Sukladno tome okidači igraju veliku ulogu u samoj aplikaciji jer se temeljem njih događaju izračuni te se onemogućuje korisniku unos ukoliko nije zadovoljen neki uvjet.

S druge strane, zbog odabrane FIFO strategije, bili su važni datumi ulaska proizvoda na skladište kako bi se prema njima odredilo koji artikli moraju prvo izaći van sa skladišta. Artikli istog naziva mogu imati više paketa. Samim time nije određeno da artikli različitih naziva su sortirani po datumu, već su paketi koji imaju isti naziv artikla sortirani po datumu, tako da prvi artikl koji može izaći sa skladišta je onaj najstariji. Izrada same aplikacije nije mi predstavljala preveliki izazov jer sam se već susreo sa CSharpom i Visual Studiom u kojem je aplikacija izrađena. Na samom projektu naučio sam mnogo toga o okidačima što mi je do sada predstavljalo izazov.

# Popis slika

|     |  |    |
|-----|--|----|
| 1.  | Prikaz ERA modela baze podataka . . . . .                            | 3  |
| 2.  | Prikaz opcije za kreiranje novog korisnika . . . . .                 | 6  |
| 3.  | Ime korisnika - admin . . . . .                                      | 7  |
| 4.  | Lozinka korisnika - admin . . . . .                                  | 7  |
| 5.  | Kreiranje baze podataka . . . . .                                    | 8  |
| 6.  | Ime baze podataka - TBP Baza . . . . .                               | 8  |
| 7.  | Backup baza podataka . . . . .                                       | 9  |
| 8.  | Restore baza podataka . . . . .                                      | 9  |
| 9.  | Prikaz izbornika aplikacije . . . . .                                | 17 |
| 10. | Prikaz unesenih artikala . . . . .                                   | 17 |
| 11. | Prikaz unosa narudžbe . . . . .                                      | 18 |
| 12. | Prikaz unosa narudžbe . . . . .                                      | 19 |
| 13. | Prikaz unosa narudžbe . . . . .                                      | 19 |
| 14. | Prikaz stanja na skladištu nakon unesene narudžbe . . . . .          | 20 |
| 15. | Prikaz ukupne količine artikla . . . . .                             | 21 |
| 16. | Prikaz stanja na skladištu nakon izdanog radnog naloga . . . . .     | 21 |
| 17. | Prikaz ukupne količine artikla nakon izdanog radnog naloga . . . . . | 22 |