

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

Valentina Požar

**APLIKACIJA ZA VOĐENJE STATISTIKE
SKLADIŠTA I PLANIRANJE ZALIHA –
AKTIVNE I TEMPORALNE BAZE
PODATAKA**

PROJEKT

Varaždin, 2019.

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

Valentina Požar

JMBAG: 0016110738

Studij: Baze podataka i baze znanja

**APLIKACIJA ZA VOĐENJE STATISTIKE SKLADIŠTA I PLANIRANJE
ZALIHA – AKTIVNE I TEMPORALNE BAZE PODATAKA**

PROJEKT

Mentor:

Izv. prof. dr. sc. Markus Schatten

Varaždin, lipanj 2019.

Izjava o izvornosti

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

Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Tema ovog projektnog zadatka je izrada desktop aplikacije za vođenje statistike skladišta i planiranje zaliha. Glavni je cilj prikazati kombinaciju aktivnih i temporalnih baza podataka koje će biti korištene za funkcioniranje projektnog zadatka. Baza podataka je izrađena u Navicatu za PostgreSQL. PostgreSQL je sustav za upravljanje relacijskim bazama podataka otvorenog koda koji podržava veliki broj SQL standarada, ali nudi i mogućnosti kao što su složeni upiti, vanjski ključevi (eng. foreign keys), okidači (eng. triggers), pogledi (eng. views) i ostalo. Za implementaciju rješenja korišteno je Microsoft Visual Studio razvojno okruženje, a kod je napisan u programskom jeziku C#. U nastavku će biti prikazani svi koraci izrade projekta pomoću slika i opisa.

Ključne riječi: skladište; aktivne baze; temporalne baze; PostgreSQL; trigger

Sadržaj

1. Teorijski uvod	1
2. Opis aplikacijske domene	2
3. Model baze podataka	4
4. Implementacija	6
4.1. Kreiranje tablica	6
4.2. Spajanje na bazu podataka	9
4.3. Čitanje iz baze podataka	9
4.4. Zapisivanje, ažuriranje i brisanje u/iz baze podataka	10
4.5. Okidači	10
5. Primjeri korištenja	15
5.1. Prijava i registracija korisnika	15
5.2. Glavni izbornik	16
5.3. Pregled, unos, ažuriranje i brisanje zaposlenika	16
5.4. Pregled, unos, ažuriranje i brisanje dobavljača	18
5.5. Pregled, unos, ažuriranje i brisanje artikla	19
5.6. Pregled, ažuriranje i brisanje primke	21
5.7. Pregled, ažuriranje i brisanje narudžbenice	22
5.8. Pregled, ažuriranje i brisanje računa	23
5.9. Pregled stanja na skladištu	25
5.10. Izrada primke	25
6. Zaključak	27
Popis literature	28
Popis slika	30

1. Teorijski uvod

U današnje vrijeme se pod pojmom baze podataka podrazumijeva kompjutorizirana baza podataka. To znači da su podaci smješteni na disku u obliku koji nije razumljiv krajnjem korisniku, pa je potreban neki sustav za upravljanje bazama podataka (SUBP) kako bi korisnici mogli koristiti te podatke i raditi s njima. Jednostavno se može reći da je baza podataka skup povezanih, organiziranih podataka. Korisnici obično bazu podataka doživljavaju kao skup povezanih tablica [1].

Cilj ovog projekta je prikazati uporabu aktivnih i temporalnih baza podataka kroz izradu aplikacije.

Temporalne baze podataka imaju bitnu ulogu u današnjem „digitalnom“ svijetu. To su tradicionalne baze podataka koje imaju vremensku odnosno temporalnu komponentu. Vremenski zapis je vezan za podatak i govori kada je taj podatak bio važeći ili kada je bio pohranjen u bazi podataka. Dva glavna vremenska tipa su Datum (eng. date) i Vremenski pečat (eng. timestamp). Ostali vremenski tipovi su varijacije ova dva tipa [2]. Također, temporalne baze podataka se koriste kako bi mogli vidjeti prijašnja stanja vrijednosti nekih atributa jer omogućuju povijesni pregled objekata. Isto tako, vremenska komponenta omogućuje spremanje više stanja istovremeno. Temporalne relacije, temporalna relacijska algebra i temporalni uvjeti integriteta čine temporalni relacijski model [3]. Temporalni podaci se mogu prikazivati na različite načine, npr. pomoću povijesne relacije (eng. historical relation), pomoću povratne relacije (eng. rollback relation), pomoću temporalne relacije (eng. temporal relation) i kao snimka trenutnog stanja (eng. snapshot) [4].

Aplikacije koje se izrađuju na temelju aktivnih baza podataka mogu sadržavati: sigurnost baze podataka, statističke analize, sustave temeljene na znanju, poglede, uvijete integriteta i ostalo. Aktivne baze podataka se temelje na relacijskim bazama podataka. Također uključuju aktivna pravila koja su najčešće u obliku DUA modela (DOGAĐAJ – UVJET – AKCIJA). Okidači su glavni mehanizmi aktivnih baza podataka. Oni omogućuju automatiziranje nekih radnji koje se izvode nakon unosa podataka u neku tablicu, ali i prilikom ažuriranja ili brisanja podataka iz nekih tablica [1].

2. Opis aplikacijske domene

Tema odabranog projektnog zadatka je vođenje statistike skladišta i planiranje zaliha. Domena na kojoj će se rješavati problematika je skladište građevinskog materijala. Svako prodajno poduzeće mora pružiti mogućnost izrade računa, primki, narudžbenica i ostalih dokumenata. U današnje vrijeme se javlja problem što neka poduzeća još uvijek ručno ispunjavaju i arhiviraju navedene dokumente. Upravo zbog toga je potrebno pronaći efikasnije rješenje. Implementirana aplikacija ima više ponuđenih mogućnosti, ovisno o korisniku aplikacije. Korisnik aplikacije može biti administrator, zaposlenik ili kupac.

Administrator ima mogućnost pregledavanja, ažuriranja i brisanja svih izrađenih narudžbenica koje su izrađene za potrebe naručivanja materijala od dobavljača te ima mogućnost pregledavanja, ažuriranja i brisanja svih izrađenih primki i računa. Također, administrator može pregledavati koji artikli su trenutno dostupni na skladištu, može ih ažurirati i brisati, a može pregledavati i stanje na skladištu. Isto tako može pregledavati, ažurirati i brisati zaposlenike kao i dobavljače poduzeća, a ima i mogućnost izrade primke.

Zaposlenici poduzeća putem aplikacije imaju mogućnost izrade primke i upravljanja artiklima na skladištu, dok kupci imaju mogućnost pregleda dostupnih artikala.

Koncepti koji se koriste u aplikaciji su sljedeći:

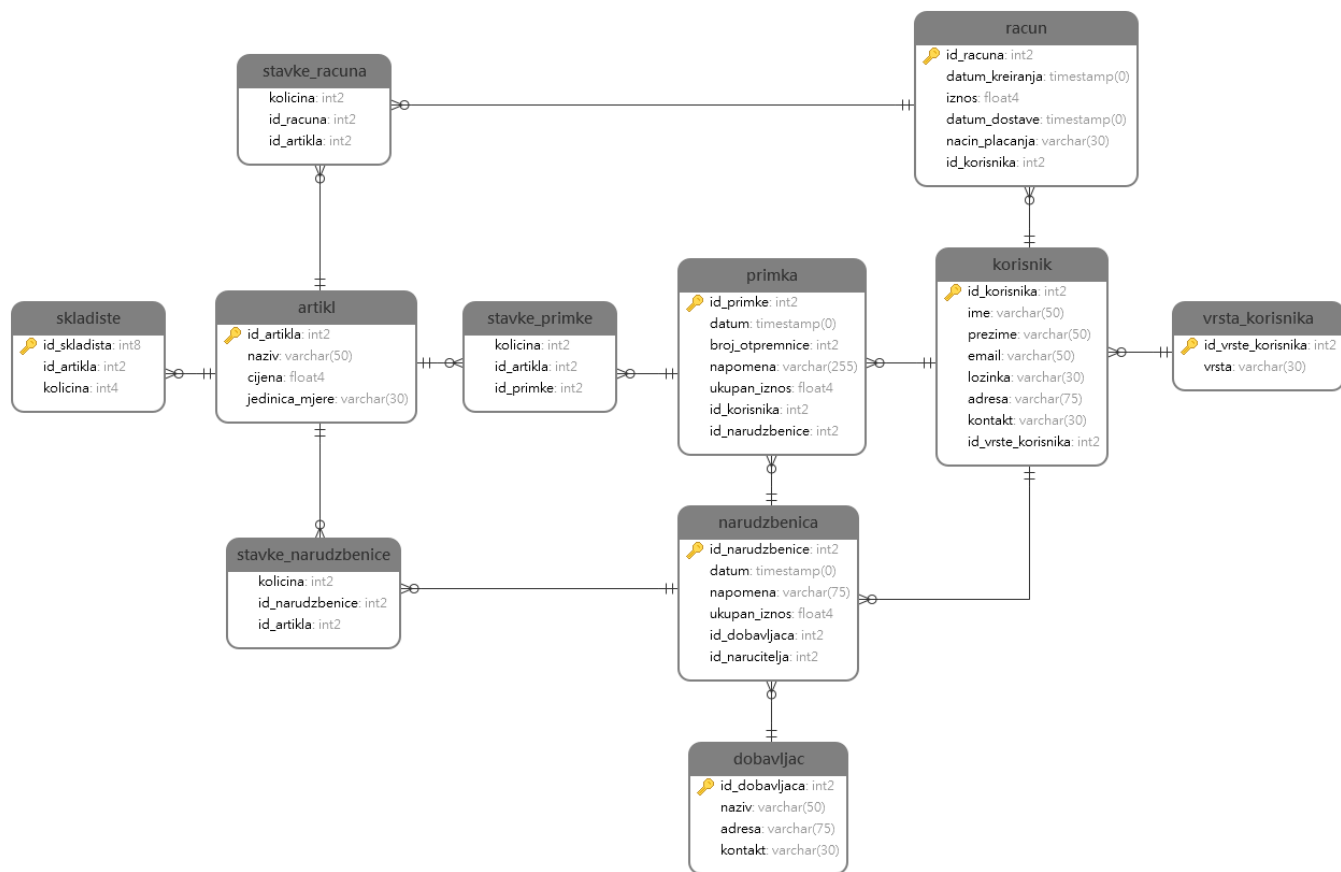
- **artikl** – vrsta građevinskog materijala koja se može kupiti u poduzeću, sadrži sljedeće attribute: id artikla, naziv, cijena, jedinica mjere
- **dobavljač** – pravna osoba koja surađuje s poduzećem i dobavlja proizvode, sadrži sljedeće attribute: id dobavljača, naziv, adresa, kontakt
- **korisnik** – osoba koja koristi aplikaciju, sadrži sljedeće attribute: id korisnika, ime, prezime, email, lozinka, adresa, kontakt, id vrste korisnika
- **narudzbenica** – dokument koji služi za naručivanje artikala od dobavljača, sadrži sljedeće attribute: id narudzbenice, datum, napomena, ukupan iznos, id dobavljača, id narucitelja
- **primka** – dokument koji nastaje na temelju narudzbenice te ga ispunjava administrator ili zaposlenik poduzeća po primitku artikala, sadrži sljedeće attribute: id primke, datum, broj otpremnice, napomena, ukupan iznos, id korisnika, id narudzbenice
- **skladiste** – predstavlja trenutno stanje svih artikala na skladištu, sadrži sljedeće attribute: id skladišta, id artikla, kolicina
- **racun** – isprava koja nastaje nakon kupnje artikala, sadrži sljedeće attribute: id racuna, datum kreiranja, iznos, datum dostave, nacin placanja, id korisnika
- **stavke narudzbenice** – popis stavki koje se nalaze na narudzbenici, sadrži sljedeće attribute: kolicina, id narudzbenice, id artikla
- **stavke primke** – popis stavki koje se nalaze na primki, sadrži sljedeće attribute: kolicina, id artikla, id primke

- stavke racuna – popis stavki koje se nalaze na računu, sadrži sljedeće attribute: kolicina, id racuna, id artikla
- vrsta korisnika – predstavlja vrstu korisnika koji mogu koristiti aplikaciju, sadrži sljedeće attribute: id vrste korisnika, vrsta

3. Model baze podataka

U nastavku će biti opisan model baze podataka u obliku ERA modela. ERA model je izrađen u Navicatu, a sastoji se od jedanaest tablica. Model se sastoji od tri tablice koje predstavljaju jake objekte koji ne ovise o drugim objektima, a to su: artikl, dobavljač i vrsta korisnika. Ostale tablice u bazi predstavljaju slabe objekte te ovise o jakim objektima u bazi podataka.

Relacija „vrsta korisnika“ povezana je s relacijom „korisnik“ vezom 1:N (jedan prema više) što znači da jedna vrsta korisnika može pripadati većem broju korisnika, dok jedan korisnik može biti ili administrator ili zaposlenik ili kupac. Relacija „korisnik“ je povezana s relacijom „racun“ vezom 1:N što znači da jednom korisniku odnosno kupcu može pripadati više računa ako je više puta kupovao u poduzeću, dok jedan račun može pripadati samo jednom korisniku. Također je relacija „korisnik“ povezana s relacijom „narudžbenica“ vezom 1:N odnosno jedan korisnik (administrator ili zaposlenik) može izraditi više narudžbenica, a jedna narudžbenica može pripadati samo jednom korisniku. Isto tako, relacija „korisnik“ je povezana s relacijom „primka“ vezom 1:N, a to znači da jedan korisnik može izraditi više primki, dok jedna primka može biti izrađena od strane samo jednog korisnika. Relacija „narudžbenica“ povezana je s relacijom „primka“ vezom 1:N jer se po jednoj narudžbenici može izraditi više primki, a jedna primka je izrađena samo po jednoj narudžbenici. Nadalje, relacija „dobavljač“ je povezana s relacijom „narudžbenica“ vezom 1:N što govori da jedan dobavljač može primiti više različitih narudžbenica, a jedna narudžbenica može biti izrađena samo za jednog dobavljača. Osim toga, relacija „artikl“ je povezana s relacijama „stavke racuna“, „stavke primke“ i „stavke narudžbenice“, te s relacijom „skladiste“ vezama 1:N što znači da se jedan artikl može nalaziti na više različitih stavki, a jedna stavka može sadržavati samo jedan isti artikl. U nastavku će biti prikazan ERA model koji je prethodno opisan.



Slika 1: ERA model baze podataka

4. Implementacija

Prije izrade same aplikacije, bilo je potrebno kreirati odgovarajuću bazu podataka. U ovom slučaju, baza podataka je izrađena u Navicatu za PostgreSQL. Baza se sastoji od jedanaest tablica, šest funkcija i šest triggera.

4.1. Kreiranje tablica

U nastavku će biti prikazane SQL naredbe pomoću kojih su kreirane relacije u bazi podataka koje odgovaraju navedenom ERA modelu.

```
CREATE TABLE "public"."artikl" (  
    "id_artikla" int2 NOT NULL GENERATED BY DEFAULT AS IDENTITY (  
        INCREMENT 1  
        MINVALUE 1  
        MAXVALUE 32767  
        START 1  
    ),  
    "naziv" varchar(50) COLLATE "pg_catalog"."default",  
    "cijena" float4,  
    "jedinica_mjere" varchar(30) COLLATE "pg_catalog"."default"  
)  
;  
  
CREATE TABLE "public"."dobavljac" (  
    "id_dobavljacka" int2 NOT NULL GENERATED BY DEFAULT AS IDENTITY (  
        INCREMENT 1  
        MINVALUE 1  
        MAXVALUE 32767  
        START 1  
    ),  
    "naziv" varchar(50) COLLATE "pg_catalog"."default",  
    "adresa" varchar(75) COLLATE "pg_catalog"."default",  
    "kontakt" varchar(30) COLLATE "pg_catalog"."default"  
)  
;
```

```

CREATE TABLE "public"."korisnik" (
    "id_korisnika" int2 NOT NULL GENERATED BY DEFAULT AS IDENTITY (
INCREMENT 1
MINVALUE 1
MAXVALUE 32767
START 1
),
    "ime" varchar(50) COLLATE "pg_catalog"."default",
    "prezime" varchar(50) COLLATE "pg_catalog"."default",
    "email" varchar(50) COLLATE "pg_catalog"."default",
    "lozinka" varchar(30) COLLATE "pg_catalog"."default",
    "adresa" varchar(75) COLLATE "pg_catalog"."default",
    "kontakt" varchar(30) COLLATE "pg_catalog"."default",
    "id_vrste_korisnika" int2 NOT NULL
)
;

CREATE TABLE "public"."narudzbenica" (
    "id_narudzbenice" int2 NOT NULL GENERATED BY DEFAULT AS IDENTITY (
INCREMENT 1
MINVALUE 1
MAXVALUE 32767
START 1
),
    "datum" timestamp(0),
    "napomena" varchar(75) COLLATE "pg_catalog"."default",
    "ukupan_iznos" float4,
    "id_dobavljacka" int2 NOT NULL,
    "id_narucitelja" int2 NOT NULL
)
;

CREATE TABLE "public"."primka" (
    "id_primke" int2 NOT NULL GENERATED BY DEFAULT AS IDENTITY (
INCREMENT 1
MINVALUE 1
MAXVALUE 32767
START 1
),
    "datum" timestamp(0),
    "broj_otpremnice" int2,
    "napomena" varchar(255) COLLATE "pg_catalog"."default",
    "ukupan_iznos" float4,
    "id_korisnika" int2 NOT NULL,
    "id_narudzbenice" int2 NOT NULL
)
;

```

```

CREATE TABLE "public"."racun" (
    "id_racuna" int2 NOT NULL GENERATED BY DEFAULT AS IDENTITY (
        INCREMENT 1
        MINVALUE 1
        MAXVALUE 32767
        START 1
    ),
    "datum_kreiranja" timestamp(0),
    "iznos" float4,
    "datum_dostave" timestamp(0),
    "nacin_placanja" varchar(30) COLLATE "pg_catalog"."default",
    "id_korisnika" int2 NOT NULL
)
;

CREATE TABLE "public"."skladiste" (
    "id_skladista" int8 NOT NULL DEFAULT nextval('skladiste_id_skladista_seq'::
        regclass),
    "id_artikla" int2 NOT NULL,
    "kolicina" int4 NOT NULL
)
;

CREATE TABLE "public"."stavke_narudzbenice" (
    "kolicina" int2,
    "id_narudzbenice" int2 NOT NULL,
    "id_artikla" int2 NOT NULL
)
;

CREATE TABLE "public"."stavke_primke" (
    "kolicina" int2,
    "id_artikla" int2 NOT NULL,
    "id_primke" int2 NOT NULL
)
;

CREATE TABLE "public"."stavke_racuna" (
    "kolicina" int2,
    "id_racuna" int2 NOT NULL,
    "id_artikla" int2 NOT NULL
)
;

CREATE TABLE "public"."vrsta_korisnika" (
    "id_vrste_korisnika" int2 NOT NULL GENERATED BY DEFAULT AS IDENTITY (
        INCREMENT 1
        MINVALUE 1
        MAXVALUE 32767
        START 1
    ),
    "vrsta" varchar(30) COLLATE "pg_catalog"."default"
)
;

```

4.2. Spajanje na bazu podataka

Kako bi aplikacija mogla funkcionirati, odnosno kako bi se mogli unositi, čitati, ažurirati i brisati podaci iz baze podataka, potrebno je bilo dodati kod pomoću kojeg se sama aplikacija povezuje s bazom podataka. Sljedeći dio koda vrši spajanje na bazu podataka, a on se nalazi u klasi „DB“.

```
private static DB instance;
private string connectionString;
private NpgsqlConnection connection;

private DB()
{
    connectionString = Properties.Settings.Default.ConnectionSkladiste;
    Connection = new NpgsqlConnection(connectionString);
    Connection.Open();
}
```

Slika 2: Spajanje na bazu podataka

4.3. Čitanje iz baze podataka

Korištenjem aplikacije u pozadini se najčešće odvija operacija čitanja podataka iz baze. U nastavku je prikazan kod pomoću kojeg se podaci čitaju iz baze podataka.

```
public static DB Instance
{
    get
    {
        if (instance == null)
        {
            instance = new DB();
        }
        return instance;
    }
}
```

Slika 3: DB instanca

```
public NpgsqlDataReader DohvatiDataReader(string sqlUpit)
{
    NpgsqlCommand command = new NpgsqlCommand(sqlUpit, Connection);
    return command.ExecuteReader();
}
```

Slika 4: Čitanje iz baze podataka

4.4. Zapisivanje, ažuriranje i brisanje u/iz baze podataka

Na slici 5 je prikazan dio koda pomoću kojeg se podaci zapisuju u bazu podataka, a isto tako se ažuriraju i brišu iz baze podataka.

```
public int IzvrsiUpit(string sqlUpit)
{
    NpgsqlCommand command = new NpgsqlCommand(sqlUpit, Connection);
    return command.ExecuteNonQuery();
}
```

Slika 5: Zapisivanje, ažuriranje i brisanje u/iz baze podataka

4.5. Okidači

Za implementaciju određenih funkcionalnosti aplikacije korišteni su okidači s obzirom da aktivne baze podataka za svoj rad koriste upravo njih. Prvi okidač postavlja datum i vrijeme u tablici „narudzbenica“ na trenutni datum i trenutno vrijeme prilikom izrade i ažuriranja bilo koje narudžbenice.

```
CREATE OR REPLACE FUNCTION "public"."tr_datum"()
    RETURNS "pg_catalog"."trigger" AS $BODY$
BEGIN
    new.datum = NOW();
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;

CREATE TRIGGER "tr_azuriranje_datum" BEFORE INSERT OR UPDATE ON "public"."
    narudzbenica"
FOR EACH ROW
EXECUTE PROCEDURE "public"."tr_datum"();
```

Drugi okidač postavlja datum i vrijeme u tablici „primka“ na trenutni datum i trenutno vrijeme prilikom izrade i ažuriranja bilo koje primke.

```
CREATE TRIGGER "tr_datum" BEFORE INSERT OR UPDATE ON "public"."primka"
FOR EACH ROW
EXECUTE PROCEDURE "public"."tr_datum"();
```

Treći okidač se izvršava nad tablicom „racun“. Računi se pohrane u bazu podataka kada neki korisnik aplikacije odnosno kupac kupi nešto od ponuđenih artikala. Prilikom izrade i ažuriranja bilo kojeg računa, navedeni okidač postavlja datum i vrijeme kreiranja računa na trenutni datum i trenutno vrijeme. Također, isti okidač postavlja datum i vrijeme dostave artikala na datum koji je točno tri dana nakon datuma kreiranja.

```

CREATE OR REPLACE FUNCTION "public"."tr_datumi_racun"()
  RETURNS "pg_catalog"."trigger" AS $BODY$
BEGIN
    new.datum_kreiranja = NOW();
    new.datum_dostave = new.datum_kreiranja+INTERVAL'3 DAY';
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;

CREATE TRIGGER "tr_azuriranje_datuma" BEFORE INSERT OR UPDATE ON "public"."racun"
FOR EACH ROW
EXECUTE PROCEDURE "public"."tr_datumi_racun"();

```

Četvrti okidač se izvršava nad tablicom „primka“. Prilikom izrade nove primke, provjerava jesu li sva polja ispunjena, odnosno jesu li ispunjena polja id narudžbenice, napomena, broj otpremnice i ukupan iznos. Ukoliko bilo koje polje nije ispunjeno, okidač ispisuje upozorenje koje je definirano unutar samog okidača te se primka ne može izraditi i ne unosi se u bazu podataka. Ako se prilikom izrade primke ispune sva potreba polja, primka se unosi u bazu podataka.

```

CREATE OR REPLACE FUNCTION "public"."tr_unos_primka"()
  RETURNS "pg_catalog"."trigger" AS $BODY$
DECLARE
    v_napomena VARCHAR(255);
BEGIN
    SELECT TRIM(BOTH FROM NEW.napomena)
    INTO    v_napomena;

    CASE
        WHEN NEW.id_narudzbenice = -1 THEN
            RAISE EXCEPTION
            'Morate popuniti polje id_narudzbenice!';
        WHEN NEW.broj_otpremnice = -1 THEN
            RAISE EXCEPTION
            'Morate popuniti polje broj_otpremnice!';
        WHEN v_napomena = TRIM(CHR(32)) THEN
            RAISE EXCEPTION
            'Morate popuniti polje napomena!';
        WHEN NEW.ukupan_iznos = -1 THEN
            RAISE EXCEPTION
            'Morate popuniti polje ukupan_iznos!';
        ELSE
            RETURN NEW;
        END CASE;
END
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;

```



```
CREATE TRIGGER "tr_unos_primka" BEFORE INSERT OR UPDATE ON "public"."primka"
FOR EACH ROW
EXECUTE PROCEDURE "public"."tr_unos_primka"();
```

Peti okidač se izvršava nad tablicom „artikl“. Okidač radi sljedeće:

- prilikom unosa novog artikla u tablicu „artikl“, dodaje se novi redak u tablicu „skladiste“
- prilikom ažuriranja postojećeg artikla u tablici „artikl“, ažurira se vrijednost u tablici „skladiste“
- prilikom brisanja artikla iz tablice „artikl“, briše se i redak s tim artiklom iz tablice „skladiste“

```
CREATE OR REPLACE FUNCTION "public"."tr_unos_artikla"()
RETURNS "pg_catalog"."trigger" AS $BODY$
BEGIN
    CASE
        WHEN (TG_OP = 'INSERT') THEN
            INSERT INTO skladiste(id_artikla, kolicina)
            VALUES (NEW.id_artikla, 0);

            RETURN NEW;
        WHEN (TG_OP = 'UPDATE') THEN
            IF(NEW.id_artikla <> OLD.id_artikla) THEN
                UPDATE skladiste
                SET id_artikla = NEW.id_artikla
                WHERE id_artikla = OLD.id_artikla;
            END IF;

            RETURN NEW;
        WHEN (TG_OP = 'DELETE') THEN
            DELETE FROM skladiste
            WHERE id_artikla = OLD.id_artikla;

            RETURN NEW;
        ELSE
            RETURN NEW;
    END CASE;
END
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```

```
CREATE TRIGGER "tr_unos_novog" AFTER INSERT OR UPDATE OR DELETE ON "public"."artikl"
FOR EACH ROW
EXECUTE PROCEDURE "public"."tr_unos_artikla"();
```

Šesti okidač se izvršava nad tablicom „stavke primke“. Okidač radi sljedeće:

- prilikom unosa stavke primke u tablicu „stavke primke“, povećava se količina artikla u tablici „skladiste“ za onu količinu koja je navedena u stavci
- prilikom ažuriranja postojeće stavke u tablici „stavke primke“, ažurira se i stanje tog artikla u tablici „skladiste“
- prilikom brisanja stavke primke iz tablice „stavke primke“, smanjuje se količina tog artikla u tablici „skladiste“

```
CREATE OR REPLACE FUNCTION "public"."tr_azuriranje_skladista"()
RETURNS "pg_catalog"."trigger" AS $BODY$
DECLARE
    v_kolicina INTEGER;
BEGIN
    CASE
        WHEN (TG_OP = 'INSERT') THEN
            SELECT kolicina
            INTO    v_kolicina
            FROM    skladiste
            WHERE   id_artikla = NEW.id_artikla;

            UPDATE skladiste
            SET     kolicina = v_kolicina
                   + NEW.kolicina
            WHERE  id_artikla = NEW.id_artikla;

            RETURN NEW;
        WHEN (TG_OP = 'UPDATE') THEN
            SELECT kolicina
            INTO    v_kolicina
            FROM    skladiste
            WHERE   id_artikla = NEW.id_artikla;

            UPDATE skladiste
            SET     kolicina = v_kolicina
                   + (NEW.kolicina - OLD.kolicina)
            WHERE  id_artikla = NEW.id_artikla;

            RETURN NEW;
        WHEN (TG_OP = 'DELETE') THEN
            SELECT kolicina
            INTO    v_kolicina
            FROM    skladiste
            WHERE   id_artikla = NEW.id_artikla;
```

```

        UPDATE skladiste
        SET
            kolicina = v_kolicina
            - OLD.kolicina
        WHERE id_artikla = NEW.id_artikla;

        RETURN NEW;

    ELSE

        RETURN NEW;

    END CASE;

END
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;

CREATE TRIGGER "tr_unos_stavke" AFTER INSERT OR UPDATE OR DELETE ON "public"."
stavke_primke"
FOR EACH ROW
EXECUTE PROCEDURE "public"."tr_azuriranje_skladista"();

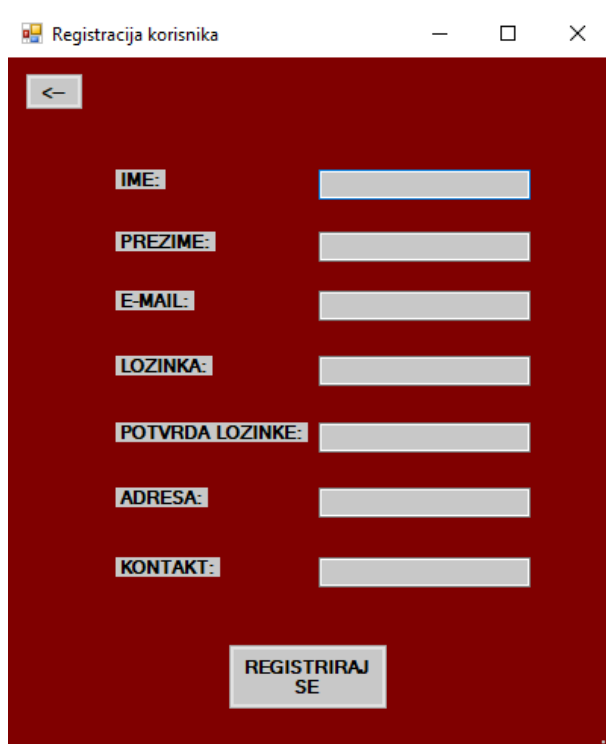
```

5. Primjeri korištenja

Kao što je već gore navedeno, korisnik aplikacije može biti administrator, zaposlenik ili kupac. Primjeri korištenja aplikacije bit će prikazani iz perspektive administratora, pošto administrator ima pristup svim mogućnostima.

5.1. Prijava i registracija korisnika

Nakon ulaska u aplikaciju, administrator i zaposlenici se moraju prijaviti, dok se kupci moraju prvo registrirati, pa tek onda prijaviti u aplikaciju.



Registracija korisnika

<

IME:

PREZIME:

E-MAIL:

LOZINKA:

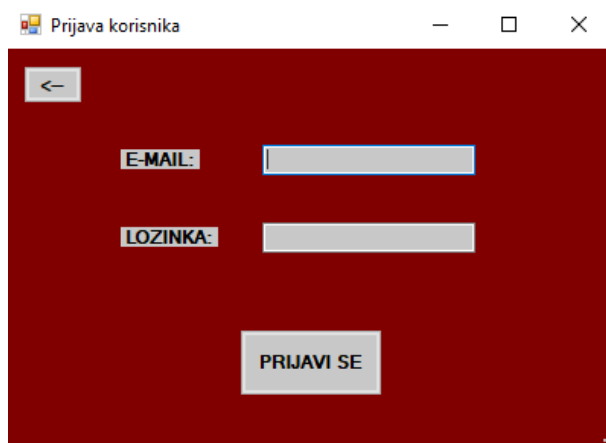
POTVRDA LOZINKE:

ADRESA:

KONTAKT:

REGISTRIRAJ SE

Slika 6: Registracija korisnika



Prijava korisnika

<

E-MAIL:

LOZINKA:

PRIJAVI SE

Slika 7: Prijava korisnika

5.2. Glavni izbornik

Na slici 5 je prikazan glavni izbornik aplikacije. Klikom na pojedini gumb otvaraju se prozori koji će biti objašnjeni u nastavku.



Slika 8: Glavni izbornik

5.3. Pregled, unos, ažuriranje i brisanje zaposlenika

Na formi „Prikaz zaposlenika“ mogu se vidjeti podaci o svim zaposlenicima koji su trenutno uneseni u bazu podataka. Također je moguće pretraživati zaposlenike po emailu. Trenutno odabrani zaposlenik je označen plavom bojom. Klikom na gumb „Dodaj zaposlenika u bazu“ otvara se nova forma „Unos zaposlenika“ pomoću koje se može unijeti novi zaposlenik tako da se popune svi podaci o zaposleniku, a to su: ime, prezime, e-mail, lozinka, potvrda lozinke, adresa i kontakt. Klikom na gumb „Ažuriraj podatke zaposlenika“ otvara se forma „Ažuriranje zaposlenika“. Polja za unos e-maila, lozinke, adrese i kontakta su popunjena s podacima onog zaposlenika koji je prethodno odabran te se oni mogu promijeniti i spremiti. Klikom na gumb „Izbriši zaposlenika iz baze“ briše se prethodno odabrani zaposlenik iz baze podataka te se on više ne prikazuje.

Prikaz zaposlenika

Pretraga po emailu:

	IDKorisnik	Ime	Prezime	Email	Lozinka	Adresa	Kontakt	IDVrsteKo
▶	3	Marina	Krik	markrik...	968TRbg	Đ. Kuha...	097886...	2
	6	Luka	Lukić	luka996...	we98Wz	Šibensk...	099110...	2
	10	Katja	Vuković	katja.vu...	VUKzec...	Prisavlje...	092784...	2

DODAJ ZAPOSLENIKA U BAZU AŽURIRAJ PODATKE ZAPOSLENIKA IZBRIŠI ZAPOSLENIKA IZ BAZE

Slika 9: Prikaz zaposlenika

Unos zaposlenika

IME:

PREZIME:

E-MAIL:

LOZINKA:

POTVRDA LOZINKE:

ADRESA:

KONTAKT:

UNESI ZAPOSLENIKA

Slika 10: Unos zaposlenika

Ažuriranje zaposlenika

Marina Krik

E-MAIL: markrik@gmail.com

LOZINKA: 968TRbg

ADRESA: Đ. Kuhara 49, Varaždin

KONTAKT: 0978865982

AŽURIRAJ

Slika 11: Ažuriranje zaposlenika

5.4. Pregled, unos, ažuriranje i brisanje dobavljača

Na formi „Prikaz dobavljača“ mogu se vidjeti podaci o svim dobavljačima koji su trenutno uneseni u bazu podataka. Također je moguće pretraživati dobavljače po nazivu. Trenutno odabrani dobavljač je označen plavom bojom. Klikom na gumb „Dodaj dobavljača u bazu“ otvara se nova forma „Unos dobavljača“ pomoću koje se može unijeti novi dobavljač tako da se popune svi podaci o dobavljaču, a to su: naziv, adresa i kontakt. Klikom na gumb „Ažuriraj podatke dobavljača“ otvara se forma „Ažuriranje dobavljača“. Polja za unos naziva, adrese i kontakta su popunjena s podacima onog dobavljača koji je prethodno odabran te se oni mogu promijeniti i spremati. Klikom na gumb „Izbriši dobavljača iz baze“ briše se prethodno odabrani dobavljač iz baze podataka te se on više ne prikazuje.

Prikaz dobavljača

Pretraga po nazivu:

IDDobavljača	Naziv	Adresa	Kontakt
2	Ria Export-Import d...	Zelena ul. 44, 4200...	042642120
3	STIL-PLET	104. BRIGADE HR...	042781036
4	Gram	Ul. Zrinsko Franko...	040328720
5	Građevinski materij...	Celine 2, 10340 Vr...	012792265
6	REMEX d.o.o.	Remetinec 115b, 4...	042601166
7	Trgoing d.o.o.	Ul. Matka Laginje 3...	047646041
8	Solliss	Banija ul. 9, 47000 ...	0914949445
9	BEGAMONT	Priobalna cesta, 10...	013498193
10	TO M...	Glavni ul. 1, 10...	013498193

DODAJ DOBAVLJAČA U BAZU AŽURIRAJ PODATKE DOBAVLJAČA IZBRIŠI DOBAVLJAČA IZ BAZE

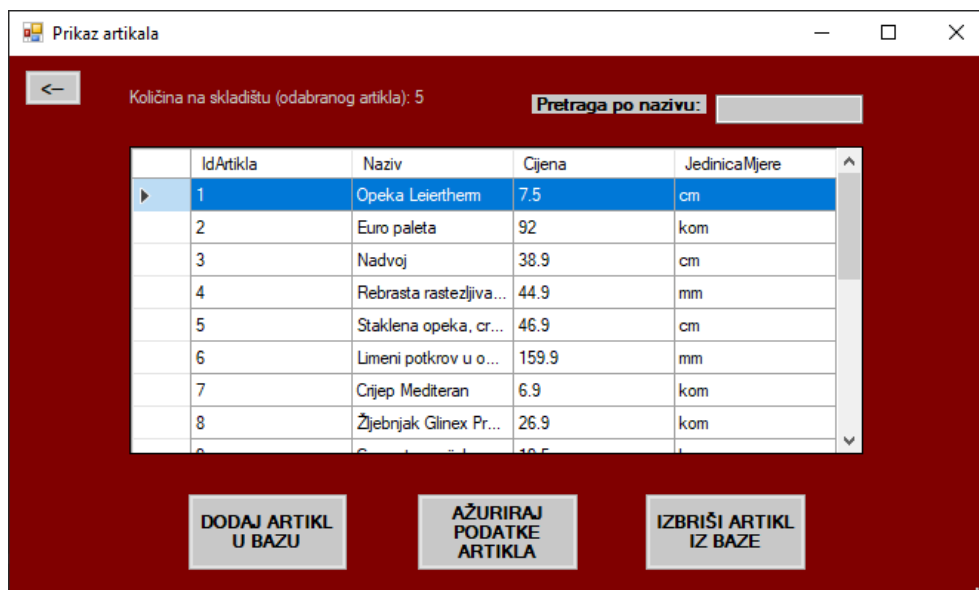
Slika 12: Prikaz dobavljača

Slika 13: Unos dobavljača

Slika 14: Ažuriranje dobavljača

5.5. Pregled, unos, ažuriranje i brisanje artikla

Na formi „Prikaz artikala“ mogu se vidjeti podaci o svim artiklima koji su trenutno uneseni u bazu podataka. Također je moguće pretraživati artikle po nazivu. Trenutno odabrani artikl je označen plavom bojom. Klikom na gumb „Dodaj artikl u bazu“ otvara se nova forma „Unos artikala“ pomoću koje se može unijeti novi artikl tako da se popune svi podaci o artiklu, a to su: naziv, cijena i jedinica mjere. Klikom na gumb „Ažuriraj podatke artikla“ otvara se forma „Ažuriranje artikala“. Polja za unos naziva, cijene i jedinice mjere su popunjena s podacima onog artikla koji je prethodno odabran te se oni mogu promijeniti i spremiti. Klikom na gumb „Izbriši artikl iz baze“ briše se prethodno odabrani artikl iz baze podataka te se on više ne prikazuje.



Slika 15: Prikaz artikala

Unos artikala

←

NAZIV:

CIJENA:

JEDINICA MJERE:

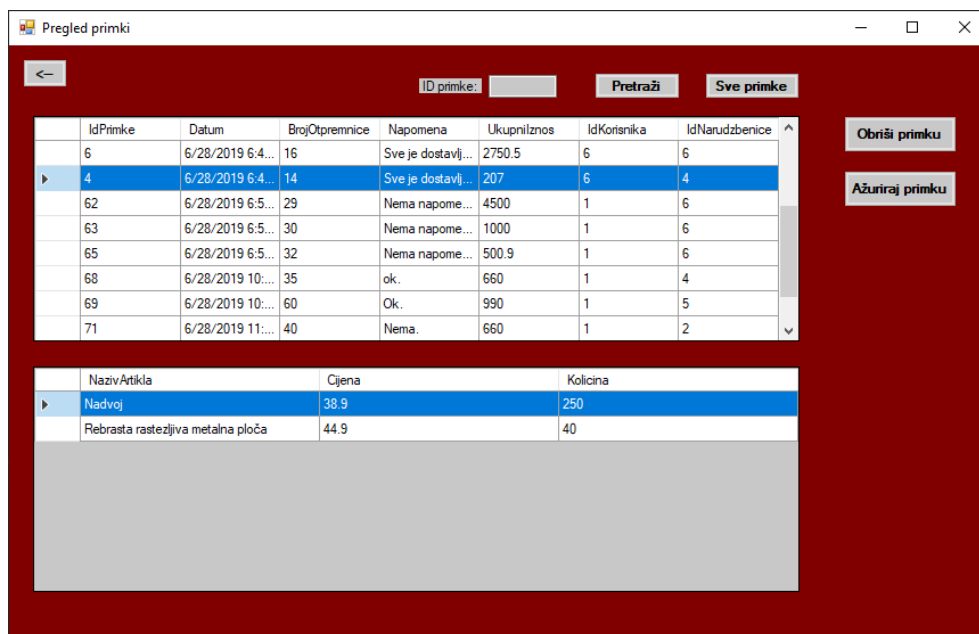
UNESI ARTIKL

Slika 16: Unos artikala

Slika 17: Ažuriranje artikala

5.6. Pregled, ažuriranje i brisanje primke

Na formi „Pregled primki“ mogu se vidjeti podaci o svim primkama koje su trenutno unesene u bazu podataka. Također je moguće pretraživati primke po id-u. Trenutno odabrana primka je označena plavom bojom. Kada se odabere primka u prvom DataGridView-u, u drugom DataGridView-u se prikazuju stavke te primke odnosno naziv artikla, cijena i količina. Klikom na gumb „Ažuriraj primku“ otvara se forma „Ažuriranje primke“. Polja za unos broja otpremnice, napomene, ukupnog iznosa, id korisnika i id narudžbenice su popunjena s podacima one primke koja je prethodno odabrana te se oni mogu promijeniti i spremiti. Polje datum se prilikom ažuriranja primke automatski postavlja na trenutni datum i trenutno vrijeme pomoću okidača. Klikom na gumb „Obriši primku“ briše se prethodno odabrana primka iz baze podataka te se ona više ne prikazuje.



Slika 18: Pregled primki

Primka: 4
BROJ OTPREMNICE: 14
NAPOMENA: Sve je dostavljeno.
UKUPAN IZNOS: 207
ID KORISNIKA: 6
ID NARUDŽBENICE: 4
DATUM: Friday, June 28

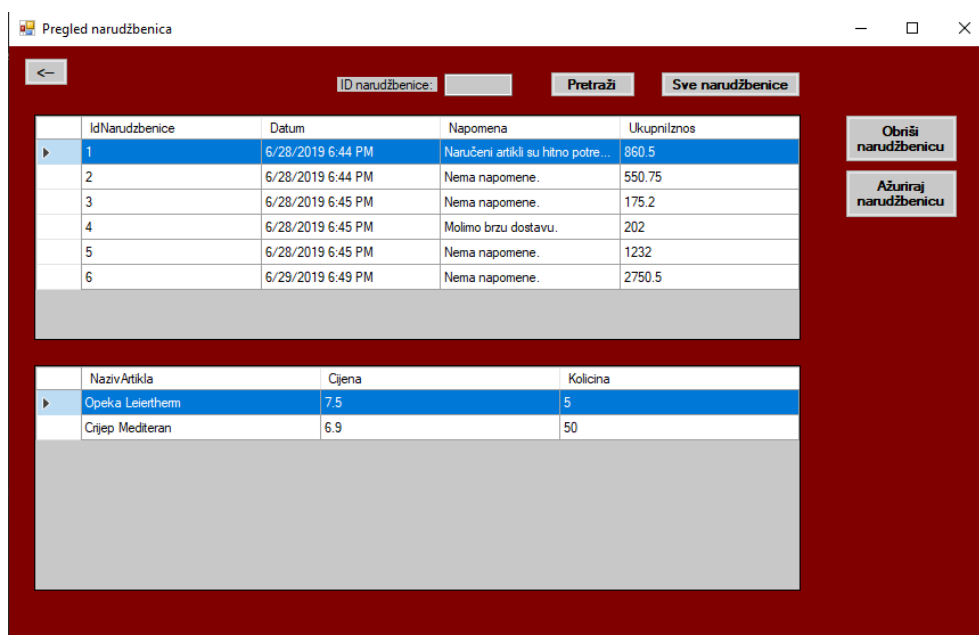
AŽURIRAJ

Slika 19: Ažuriranje primke

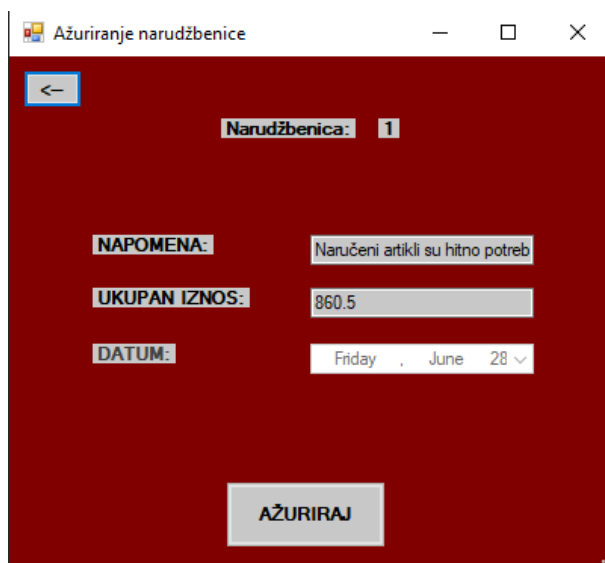
5.7. Pregled, ažuriranje i brisanje narudžbenice

Na formi „Pregled narudžbenica“ mogu se vidjeti podaci o svim narudžbenicama koje su trenutno unesene u bazu podataka. Također je moguće pretraživati narudžbenice po id-u. Trenutno odabrana narudžbenica je označena plavom bojom. Kada se odabere narudžbenica u prvom DataGridView-u, u drugom DataGridView-u se prikazuju stavke te narudžbenice odnosno naziv artikla, cijena i količina. Klikom na gumb „Ažuriraj narudžbenicu“ otvara se forma „Ažuriranje narudžbenice“. Polja za unos napomene i ukupnog iznosa su popunjena s podacima one narudžbenice koja je prethodno odabrana te se oni mogu promijeniti i spremiti. Polje

datum se prilikom ažuriranja narudžbenice automatski postavlja na trenutni datum i trenutno vrijeme pomoću okidača. Klikom na gumb „Obriši narudžbenicu“ briše se prethodno odabrana narudžbenica iz baze podataka te se ona više ne prikazuje.



Slika 20: Pregled narudžbenica

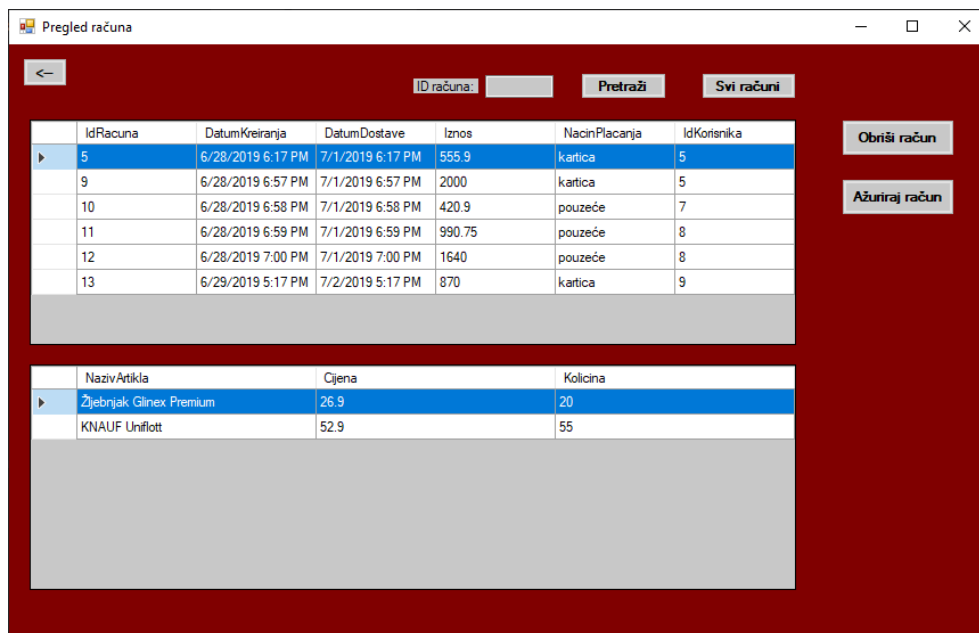


Slika 21: Ažuriranje narudžbenice

5.8. Pregled, ažuriranje i brisanje računa

Na formi „Pregled računa“ mogu se vidjeti podaci o svim računima koji su trenutno uneseni u bazu podataka. Također je moguće pretraživati račune po id-u. Trenutno odabrani račun je označen plavom bojom. Kada se odabere račun u prvom DataGridView-u, u drugom DataGridView-u se prikazuju stavke tog računa odnosno naziv artikla, cijena i količina. Klikom

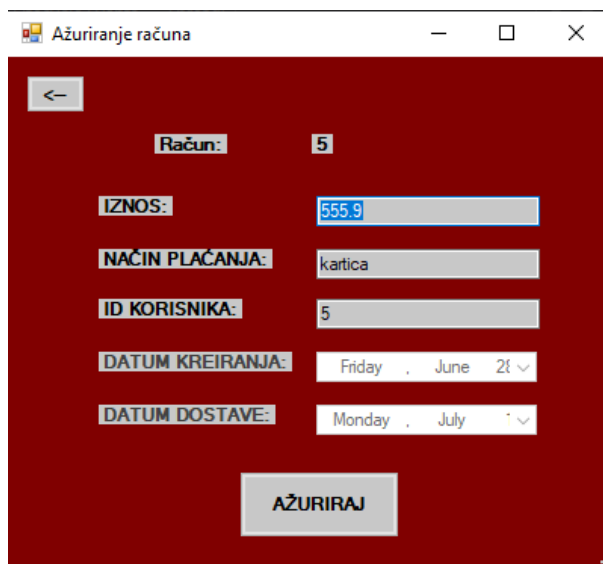
na gumb „Ažuriraj račun“ otvara se forma „Ažuriranje računa“. Polja za unos iznosa, načina plaćanja i id korisnika su popunjena s podacima onog računa koji je prethodno odabran te se oni mogu promijeniti i spremati. Polje datum kreiranja se prilikom ažuriranja računa automatski postavlja na trenutni datum i trenutno vrijeme pomoću okidača, dok se polje datum dostave prilikom ažuriranja računa automatski postavlja na datum i vrijeme koji su točno tri dana nakon datuma kreiranja. Klikom na gumb „Obriši račun“ briše se prethodno odabrani račun iz baze podataka te se on više ne prikazuje.



IdRacuna	DatumKreiranja	DatumDostave	Iznos	NacinPlacanja	IdKorisnika
5	6/28/2019 6:17 PM	7/1/2019 6:17 PM	555.9	kartica	5
9	6/28/2019 6:57 PM	7/1/2019 6:57 PM	2000	kartica	5
10	6/28/2019 6:58 PM	7/1/2019 6:58 PM	420.9	pouzeće	7
11	6/28/2019 6:59 PM	7/1/2019 6:59 PM	990.75	pouzeće	8
12	6/28/2019 7:00 PM	7/1/2019 7:00 PM	1640	pouzeće	8
13	6/29/2019 5:17 PM	7/2/2019 5:17 PM	870	kartica	9

NazivArtikla	Cijena	Kolicina
Žljebnjak Glinex Premium	26.9	20
KNAUF Uniflott	52.9	55

Slika 22: Pregled računa



Račun: 5

IZNOS: 555.9

NAČIN PLAĆANJA: kartica

ID KORISNIKA: 5

DATUM KREIRANJA: Friday, June 28

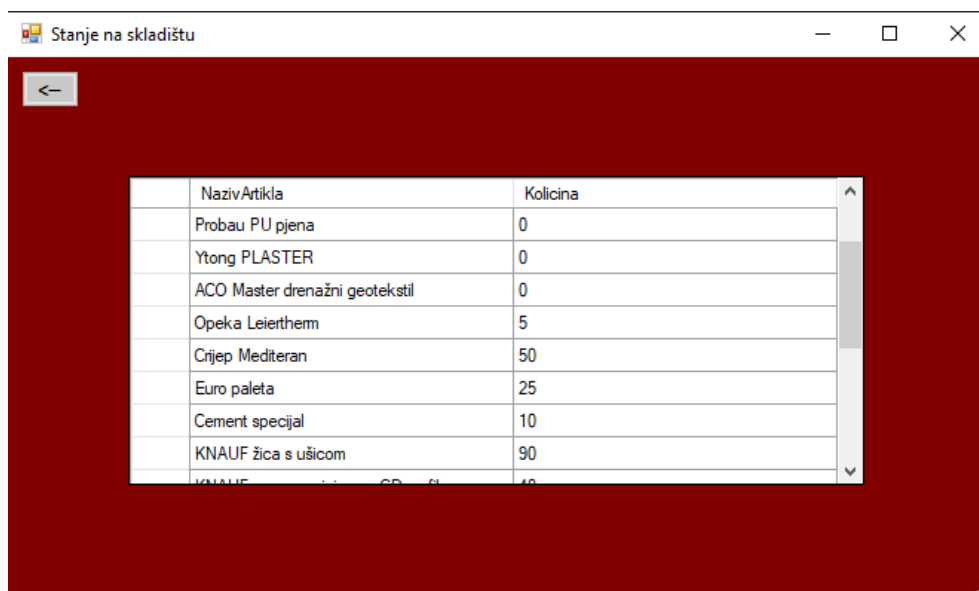
DATUM DOSTAVE: Monday, July 1

AŽURIRAJ

Slika 23: Ažuriranje računa

5.9. Pregled stanja na skladištu

Na formi „Stanje na skladištu“ može se u svakom trenutku vidjeti kakvo je stanje na skladištu, odnosno može se vidjeti koji sve artikli postoje na skladištu i koja je njihova količina. Stanje na skladištu se ažurira pomoću triggera.



The screenshot shows a window titled "Stanje na skladištu" with a red background. In the top-left corner of the window is a button with a left-pointing arrow. Centered on the window is a table with two columns: "NazivArtikla" and "Kolicina". The table contains the following data:

NazivArtikla	Kolicina
Probau PU pjena	0
Ytong PLASTER	0
ACO Master drenažni geotekstil	0
Opeka Leierthem	5
Crijep Mediteran	50
Euro paleta	25
Cement specijal	10
KNAUF žica s ušicom	90
KNAUF	10

Slika 24: Stanje na skladištu

5.10. Izrada primke

Na formi „Izrada primke“ mogu se izraditi primke na temelju narudžbenica te stavke primki. U gornjem lijevom DataGridView-u su prikazane sve narudžbenice, dok su u gornjem desnom DataGridView-u prikazane stavke svake narudžbenice kako bi administrator ili zaposlenici mogli odabrati narudžbenicu po kojoj će izraditi primku. Polja id narudžbenice, napomena, broj otpremnice i ukupan iznos se moraju obavezno popuniti jer će se u suprotnom ispisati upozorenje, a to je napravljeno okidačem. Kako bi se unijele stavke primke mora se prvo odabrati id primke te popuniti polja količina i id artikla. Kako bi se mogli pravilno unijeti artikli, u donjem DataGridView-u su prikazani svi artikli iz baze podataka te je ispisano njihovo stanje na skladištu koje se povećava za onu količinu koja je unesena na stavku primke.

Izrada primke

ID narudžbenice:

Pretraži

Sve narudžbenice

	IdNarudzenice	Datum	Napomena	UkupniIznos
▶	1	6/28/2019 6:44 ...	Naručeni artikli su...	860.5
	2	6/28/2019 6:44 ...	Nema napomene.	550.75
	3	6/28/2019 6:45 ...	Nema napomene.	175.2
	4	6/28/2019 6:45 ...	Molimo brzu dosta...	202
	6	6/28/2019 6:45 ...	Nema napomene.	2750.5
	5	6/28/2019 6:45 ...	Nema napomene.	1232

Stavke odabrane narudžbenice

	NazivArtikla	Cijena	Kolicina
▶	Opeka Leierth...	7.5	5
	Crijep Mediteran	6.9	50

Primka

ID narudžbenice:

Napomena:

Broj otpremnice:

Ukupan iznos:

ID primke: 50

Količina:

ID artikla:

Dodaj stavku

Naziv artikla:

Pretraži

Svi artikli

	IdArtikla	Naziv	Cijena	JedinicaMj
▶	1	Opeka L...	7.5	cm
	2	Euro pal...	92	kom
	3	Nadvoj	38.9	cm
	4	Rebrast...	44.9	mm
	5	Staklena...	46.9	cm
	6	Limeni p...	159.9	mm

Količina na skladištu (odabranog artikla): 5

Izradi primku

Slika 25: Izrada primke

6. Zaključak

Sama domena ovog projekta vezana je uz skladište građevinskog materijala te praćenje stanja zaliha na skladištu. Aplikacija ima jako puno mogućnosti, a njome se mogu koristiti administrator, zaposlenici te kupci. Dio online prodaje nije implementiran u ovoj aplikaciji, ali se kupci mogu registrirati i prijaviti u aplikaciju te pregledavati dostupne artikle, a administrator može pregledavati, ažurirati i brisati račune koji su pohranjeni u bazu podataka nakon online kupnje. Baza podataka sadrži jedanaest relacija i šest okidača koji omogućuju da se automatiziraju neke radnje nad bazom podataka.

Glavna tema ovog projektnog zadatka bile su aktivne i temporalne baze podataka. Kako bi se implementirale aktivne i temporalne baze podataka, koristila se PostgreSQL tehnologija. Sama aplikacija je izrađena u Visual Studiu. Aktivne baze podataka nude mogućnost korištenja okidača, pa je pomoću njih lako implementirano automatsko postavljanje datuma, praćenje stanja na skladištu te ograničenja prilikom izrade primke. Osim toga, temporalne baze podataka su se pokazale jako korisnima za implementaciju vremenske komponente.

Popis literature

- [1] K. Rabuzin, „SQL - napredne teme”, *Fakultet organizacije i informatike, Zagreb*, 2014.
- [2] TimeConsult, *What are Temporal Databases?*, 2005. adresa: <http://www.timeconsult.com/TemporalData/TemporalDB.html> (pogledano 25. 5. 2019).
- [3] M. Maleković, „Teorija baza podataka - skripta”, 2008.
- [4] R. T. Snodgrass i I. Ahn, „Temporal databases - Computer science press”, *Rockville*, 1986.

Popis slika

1.	ERA model baze podataka	5
2.	Spajanje na bazu podataka	9
3.	DB instanca	9
4.	Čitanje iz baze podataka	9
5.	Zapisivanje, ažuriranje i brisanje u/iz baze podataka	10
6.	Registracija korisnika	15
7.	Prijava korisnika	15
8.	Glavni izbornik	16
9.	Prikaz zaposlenika	17
10.	Unos zaposlenika	17
11.	Ažuriranje zaposlenika	18
12.	Prikaz dobavljača	18
13.	Unos dobavljača	19
14.	Ažuriranje dobavljača	19
15.	Prikaz artikala	20
16.	Unos artikala	20
17.	Ažuriranje artikala	21
18.	Pregled primki	22
19.	Ažuriranje primke	22
20.	Pregled narudžbenica	23
21.	Ažuriranje narudžbenice	23
22.	Pregled računa	24
23.	Ažuriranje računa	24

24. Stanje na skladištu	25
25. Izrada primke	26