

# **REŠENJE PROBLEMA**

## **IMPLEMENTACIJA SISTEMA UPRAVLJANJA ZALIHAMA**

### **1. TEHNOLOGIJA I OKRUŽENJE**

Za realizaciju sistema koristio sam Embarcadero RAD Studio (Delphi), sa FireMonkey (FMX) framework-om za cross-platform razvoj. Baza podataka je simulirana kroz memorijske strukture (nizovi record-a) jer projekat ne zahteva punu bazu, ali je dizajniran tako da se lako može prebaciti na SQLite ili Firebird.

Verzija Delphi-ja je 10.3+, sa fokusom na Windows platformu za testiranje. Ovo omogućava jednostavan interfejs sa tabovima za svaki modul, bez potrebe za eksternim bibliotekama. Izbor tehnologije obezbeđuje brz razvoj prototipa uz mogućnost lage migracije na produkcijski sistem.

### **2. MAPIRANJE DIJAGRAMA NA IMPLEMENTACIJU**

IDEF0 dijagram (A0: Upravljanje zalihama) mapiran je na glavni TabControl sa pet tabova, gde svaki tab odgovara potprocesu (A1 do A5). Na primer, A1 je TabItem1 sa poljima za unos robe, A4 je TabItem4 sa EOQ kalkulatorom.

#### **Mapiranje DFD skladišta:**

- D1 Proizvodi → array of TProizvod
- D2 Transakcije → array of TStavkaZaliha
- D3 Dobavljači → array of TDobavljac
- D4 Porudžbine → array of TPorudzbina

MOV dijagram preveden je u record strukture sa poljima koja odgovaraju atributima iz skripte (npr. ProizvodID kao PK, relacije preko stranih ključeva simulirane indeksima). Kardinalnosti (1:N) održane su kroz petlje za pretragu.

### 3. IMPLEMENTACIJA MODULA

Sistem je razvijen modularno, sa procedurama za svaku akciju. Inicijalizacija se vrši u FormCreate, gde se postavljaju nizovi i dodaju test podaci (dva dobavljača i admin korisnik). Svaki modul koristi dugmad za pokretanje funkcija.

#### 3.1. MODUL A1 - PRIJEM I SKLADIŠTENJE

Klik na ButtonPrijem proverava obavezna polja, traži postojeći proizvod po šifri. Ako postoji, ažurira količinu; ako ne, kreira novi. Dodaje transakciju ulaza i generiše broj dokumenta (PR-YYYYMMDD-ID). Osvežava prikaz zaliha u MemoProizvodi.

##### Algoritam:

1. Validacija obaveznih polja (šifra, naziv, količina)
2. Pretraga postojećeg proizvoda po šifri
3. If postoji: Update količine, Else: Create novi record
4. Dodavanje StavkaZaliha sa tipom "ULAZ"
5. Generisanje broja prijemnice
6. Refresh prikaza

#### 3.2. MODUL A2 - KONTROLA ZALIHA

ButtonProveriKriticne iterira kroz proizvode, proverava min/max nivoe i prikazuje upozorenja u MemoKontrola sa statistikom (broj kritičnih, prekomernih, normalnih).

##### Algoritam:

```
For each proizvod in niz:
  If TrenutnaZaliha < MinZaliha: Status := "KRITIČNO"
  Else If TrenutnaZaliha > MaxZaliha: Status :=
"PREKOMERNO"
  Else: Status := "OK"
End For
```

Prikaz statistike i upozorenja

### 3.3. MODUL A3 - IZDAVANJE MATERIJALA

ButtonIzdaj proverava izabrani proizvod i količinu, ako je dovoljno na zalihi, smanjuje stanje, dodaje transakciju izlaza i generiše izdatnicu (IZ-YYYYMMDD-ID).

#### Algoritam:

1. Validacija izbora proizvoda i količine
2. Provera: If TrenutnaZaliha >= KoličinaIzdavanja
3. If OK: Smanjenje zalihe, Create StavkaZaliha  
"IZLAZ"
4. Generisanje broja izdatnice
5. Else: ShowMessage("Nedovoljna zaliha")

### 3.4. MODUL A4 - OPTIMIZACIJA PORUDŽBINA

ButtonEOQ uzima parametre (potražnja, trošak narudžbe/čuvanja), računa EOQ po Wilsonovoj formuli, broj porudžbina i period (365 dana/godina). Prikazuje rezultate i uštede u MemoOptimizacija.

#### Wilson-ova formula:

$$EOQ = \sqrt{(2 * D * S) / H}$$

gde je: D = godišnja potražnja, S = trošak narudžbe, H = trošak čuvanja

### 3.5. MODUL A5 - IZVEŠTAVANJE

ButtonStandardniIzvestaj prikazuje listu proizvoda sa stanjem i sumarnim podacima (ukupna količina/vrednost). ButtonKPIIzvestaj analizira transakcije (broj ulaza/izlaza, koeficijent obrta) i statistiku baze.

#### KPI pokazatelji:

- Koeficijent obrta zaliha = Ukupni izlaz / Prosečna zaliha
- Ukupna vrednost zaliha =  $\sum$  Količina \* Cena)
- Statistika transakcija po tipu

#### 4. STRUKTURA BAZE PODATAKA (SIMULACIJA)

Baza je simulirana nizovima record-a za efikasnost, sa brojačima za ID-eve (kao AUTO\_INCREMENT). Ovo omogućava brzu obradu bez eksterne baze, ali je kompatibilno sa SQL-om.

##### Strukture podataka:

```
type
  TProizvod = record
    ProizvodID: Integer;
    Sifra: string;
    Naziv: string;
    JedinicaMere: string;
    MinZaliha: Double;
    MaxZaliha: Double;
    TrenutnaZaliha: Double;
    Cena: Double;
    DatumKreiranja: TDateTime;
    Aktivan: Boolean;
  end;

  TStavkaZaliha = record
    StavkaID: Integer;
    ProizvodID: Integer;
    DobavljacID: Integer;
    TipTransakcije: string; // "ULAZ" ili "IZLAZ"
    Kolicina: Double;
    DatumTransakcije: TDateTime;
    BrojDokumenta: string;
  end;
```

Relacije se održavaju kroz strane ključeve (npr. StavkaZaliha.ProizvodID pokazuje na Proizvodi[index].ProizvodID), sa brzom pretragom kroz petlje.

## 5. KORISNIČKI INTERFEJS

Interfejs je jednostavan i intuitivan: glavni form sa TabControl-om za module, edit polja za unos (šifra, naziv, količina...), combo box-ovi za izbor (dobavljači, proizvodi), memo za prikaz rezultata i izveštaja. Dodat je MemoProizvodi za trenutno stanje zaliha na svakom tabu.

### Komponente interfejsa:

- TabControl sa pet tabova (A1-A5)
- Edit komponente za unos podataka
- ComboBox za izbor iz lista
- Memo komponente za prikaz rezultata
- Button komponente za akcije

FormResize osigurava responsivnost. Test podaci se učitavaju pri startu za demonstraciju funkcionalnosti sistema.

## 6. TOKOVI PODATAKA

### Osnovni tokovi sistema:

#### 6.1. Tok prijema:

- Unos podataka
  - validacija
  - ažuriranje niza Proizvodi
    - dodavanje u StavkeZaliha
    - osvežavanje prikaza

#### 6.2. Tok izdavanja:

- Izbor i količina
  - provera dostupnosti
  - ažuriranje Proizvoda
    - dodavanje transakcije
    - poruka o uspehu

### 6.3. Tok optimizacije:

Unos parametara → računanje formule → prikaz preporuka

### 6.4. Tok izveštaja:

Prolaz kroz nizove → generisanje statistike → prikaz u memo

Svi tokovi poštuju CRUD matricu, sa fokusom na read/update za zalihe.

## 7. ALGORITMI

### 7.1. Status zaliha

```
function ProverStatusZaliha(proizvod: TProizvod): string;
begin
    if proizvod.TrenutnaZaliha < proizvod.MinZaliha then
        Result := 'KRITIČNO'
    else if proizvod.TrenutnaZaliha > proizvod.MaxZaliha
then
        Result := 'PREKOMERNO'
    else
        Result := 'OK';
end;
```

### 7.2. EOQ kalkulacija

```
function IzracunajEOQ(D, S, H: Double): Double;
begin
    Result := Sqrt((2 * D * S) / H);
end;
```

```
function BrojPorudzbinaGodisnje(D, EOQ: Double): Integer;
begin
    Result := Round(D / EOQ);
end;
```

```
function PeriodIzmedjuPorudzbina(BrojPorudzbina:
Integer): Integer;
begin
```

```
    Result := Round(365 / BrojPorudzbina);  
end;
```

### 7.3. Generisanje dokumenata

```
function GenerisiBrojDokumenta(tip: string; ID: Integer):  
string;  
var  
    datum: string;  
begin  
    datum := FormatDateTime('YYYYMMDD', Now);  
    Result := tip + '-' + datum + '-' + IntToStr(ID);  
end;
```

## 8. VALIDACIJA I SIGURNOST

### Implementirane validacije:

- Obavezna polja proverena if-uslovima, sa ShowMessage za greške
- Numeričke vrednosti parsirane StrToFloatDef sa default-ovima
- Provera jedinstvenosti šifre proizvoda petljom
- Nema brisanja, samo ažuriranje aktivan flag-a za deaktivaciju
- Testirano sa pozitivnim/negativnim scenarijima

```
// Primer validacije  
if (EditSifra.Text = '') or (EditNaziv.Text = '') then  
begin  
    ShowMessage('Sifra i naziv su obavezni!');  
    Exit;  
end;  
  
kolicina := StrToFloatDef(EditKolicina.Text, 0);  
if kolicina <= 0 then  
begin  
    ShowMessage('Kolicina mora biti veća od nule!');  
    Exit;  
end;
```

## 9. TESTIRANJE

Sistem je testiran kroz sledeće scenarije:

### Test scenario:

- **Prijem novog proizvoda:** unos, provera stanja, prikaz
- **Ažuriranje postojećeg:** povećanje količine, transakcija
- **Kontrola:** simulacija min/max, upozorenja
- **Izdavanje:** uspešno i neuspešno (nedovoljno zaliha)
- **EOQ:** sa različitim ulazima, provera formule
- **Izveštaji:** prazna baza, sa podacima, KPI računice

## 10. BUDUĆA PROŠIRENJA

### Planirana pobojšanja:

- **Integracija prave baze:** SQLite za persistenciju podataka
- **Grafički prikazi:** TChart komponenta za izveštaje
- **Automatizacija:** automatsko generisanje porudžbina
- **Više korisnika:** podrška za login sistem
- **Export funkcije:** izveštaji u PDF/Excel formatu
- **API integracija:** eksterni API-ji za cene dobavljača

## 11. ZAKLJUČAK

Sistem je uspešno implementiran prema funkcionalnim zahtevima, sa potpunom podrškom za sve IDEF0 potprocese (A1-A5). Korišćenje Delphi FMX framework-a omogućilo je brz razvoj i testiranje, dok simulirana baza podataka obezbeđuje fleksibilnost za buduće proširenje na prave RDBMS sisteme.

Interfejs je jednostavan za korišćenje, a svi moduli rade u skladu sa projektovanim tokovima podataka iz dijagrama. Implementirani algoritmi za EOQ optimizaciju, kontrolu zaliha i generisanje dokumenata obezbeđuju potpunu funkcionalnost sistema upravljanja zalihama.