

BAZA PODATAKA ZA DOBROVOLJNO DARIVANJE KRVI

BLOOD DONATION DATABASE

Nikolina Jurak¹, Željko Kovačević²

¹Tehničko veleučilište u Zagrebu, Vrbik 8, 10000 Zagreb, Hrvatska, Student

²Tehničko veleučilište u Zagrebu, Vrbik 8, 10000 Zagreb, Hrvatska

SAŽETAK

Ovaj rad opisuje razvoj sustava s bazom podataka za dobrovoljno darivanje krvi. Rješenje je razvijeno u svrhu poboljšanja postojećeg sustava, kao dodatni modul namijenjen medicinskom osoblju koje sudjeluje u samom procesu donacije. Kroz rad su izdvojeni ključni koraci koji moraju biti zadovoljeni prilikom razvoja tražene baze podataka, s naglaskom na korištenje Microsoft SQL Servera 2019. Osim baze podataka, rad opisuje i klijent aplikaciju pisanu u programskom jeziku C# koja uz odgovarajuća prava i dozvole omogućuje obradu podataka.

Ključne riječi: baza podataka, SQL, SQL Server, dobrovoljno darivanje krvi

ABSTRACT

This paper describes the development of a system with a database for voluntary blood donation. The solution was developed for the purpose of improving the existing system, as an additional module intended for medical staff participating in the donation process itself. The paper highlights the key steps that must be taken when developing the required database, with emphasis on the use of Microsoft SQL Server 2019. In addition to the database, this paper describes the client application written in C# programming language that with appropriate privileges and permissions allows data processing.

Keywords: database, SQL, SQL Server, blood, donations

1. UVOD

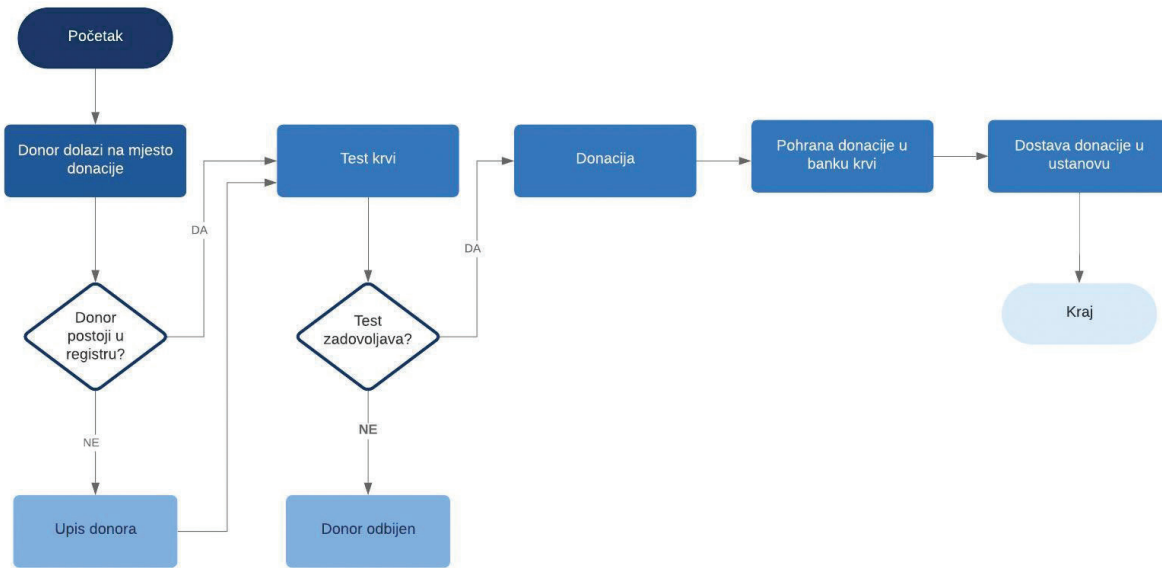
1. INTRODUCTION

Sadržaj ovog rada prvenstveno je orijentiran na bazu podataka dok je klijent aplikacija razvijena kako bi se korisnicima omogućila komunikacija s njom. Kao ključni koraci koji moraju biti zadovoljeni prilikom razvoja nove baze podataka kroz rad su izdvojeni analiza poslovnog procesa, dizajn baze podataka, kreiranje objekata na osnovi dizajna te zaštita osjetljivih podataka korištenjem kriptografskih algoritama i funkcija za maskiranje. Kroz nastavak rada prikazani su navedeni koraci na primjeru prethodno razvijene baze podataka za dobrovoljno darivanje krvi. Za razvoj baze podataka korišten je Microsoft SQL Server. [1] Potrebno je napomenuti da novo rješenje nije razvijeno kao zamjena postojećeg rješenja već kao dodatni modul u svrhu njegova poboljšanja.

2. ANALIZA POSLOVNOG PROCESA

2. BUSINESS PROCESS ANALYSIS

Kako bi se razvilo kvalitetno rješenje potrebno je upoznati se s organizacijom i poslovanjem sustava za koje je rješenje namijenjeno. Potrebno je utvrditi poslovne procese koji se odvijaju te na temelju njih odrediti zahtjeve i omogućiti pravilnu pohranu podataka. Isto tako, potrebno je odrediti granice sustava odnosno dijelove koji će mu pripadati kako bi sustav bio optimalan. Transfuzijska služba trenutno na nacionalnoj razini koristi sustav e-Delphyn. [2] Sustav ima dva modula: "Modul Krvne banke" i "Modul Klinička transfuzija". Postojeći sustav pokriva sve zahtjeve vezane uz donaciju, međutim pokriva i dodatne procese poput proizvodnje krvnih pripravaka, dodatnih testiranja donacija i slično. [3] [4]



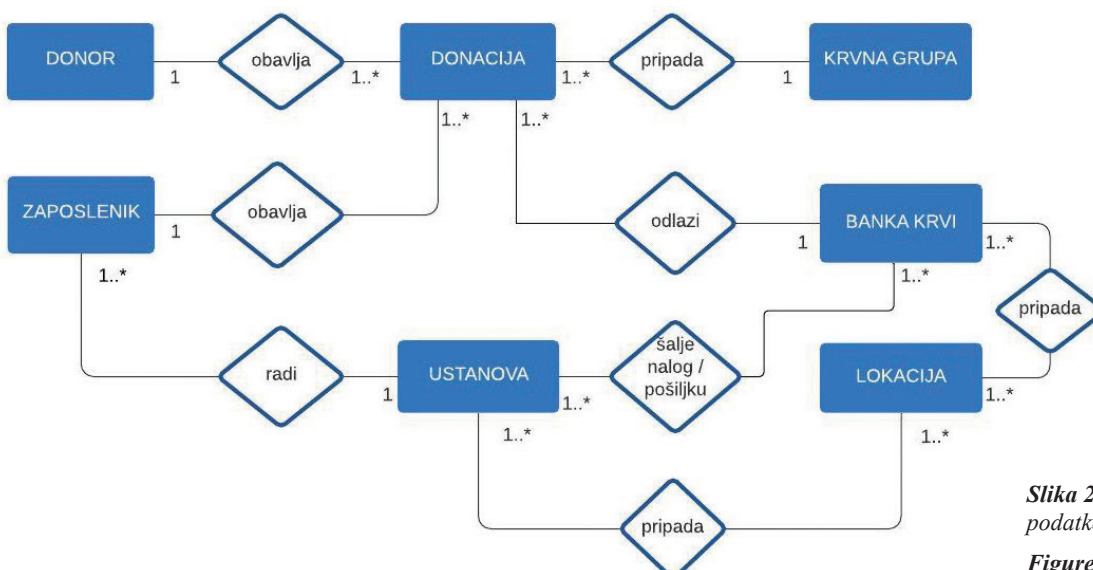
Slika 1 Dijagram procesa darivanja krvi

Figure 1 Blood donation process diagram

Kako bi se rasteretio dio medicinskog osoblja od poznavanja i učenja svih komponenti sustava u kojima ne sudjeluju aktivno razvijen je dodatan modul, "Modul donacija". Cilj je iz postojećeg sustava u manji izdvojiti proces donacije u najjednostavnijem obliku, a opet osigurati ispravan put krvi od donora do pacijenta.

Prema neposredno prikazanom dijagramu procesa (Slika 1) vidimo da se uvođenjem "Modula donacija" zapravo ostvaruje i poveznica između postojećih modula jer sadrži podatke o bankama krvi koje koriste "Modul Krvne banke", te podatke o ustanovama koje koriste "Modul Klinička transfuzija".

Proces darivanja krvi (Slika 1) započinje dolaskom donora na mjesto donacije, prilikom čega se prvo provjerava postoje li podaci donora u registru. Ukoliko ne postoje, potrebno ih je upisati. Sljedeći korak je testiranje donorove krvi. Ukoliko su svi parametri testa zadovoljeni uzima se donacija, u suprotnome donor biva privremeno ili trajno odbijen. Uzeta donacija pohranjuje se u banku krvi koja je sudjelovala u organiziranju dobrovoljnog darivanja. Kada se unutar neke od ustanova pojavi potreba za određenim dozama krvi šalje se nalog prema banci krvi za njihovu dostavu.



Slika 2 Konceptualni podatkovni model

Figure 2 Conceptual data model

3. DIZAJN BAZE PODATAKA

3. DATABASE DESIGN

U nastavku je prikazan konceptualni podatkovni model (Slika 2) koji sadrži sve entitete potrebne za opis i implementaciju procesa dobrovoljnog darivanja krvi. [5]

Iz prikazanog konceptualnog podatkovnog modela (Slika 2) možemo pročitati da je entitet *DONOR* u vezi s entitetom *DONACIJA*. Jedan donator može izvršiti više donacija, dok jedna donacija može biti od samo jednog donora. Nadalje, jedan zaposlenik može obaviti jednu ili više donacija dok jednu donaciju može obaviti samo jedan zaposlenik. Čitajući ostatak modela na identičan način, tj. promatrajući odnose među entitetima u oba smjera (s lijeva nadesno, zatim s desna nalijevo), moguće je vidjeti odnose među svim povezanim entitetima te dobiti još jasniji uvid u sam način izvođenja procesa.

4. RAZVOJ BAZE PODATAKA

4. DATABASE DEVELOPMENT

Kao najvažniji objekti istaknute su uskladištene procedure i okidači. Uskladištene procedure korištene su kao objekti koji pružaju mogućnost centralizacije poslovne logike, smanjenje mrežnog prometa i zaštitu podataka, a okidači kao dodatni sigurnosni mehanizmi koji pridonose očuvanju konzistentnosti podataka i očuvanju referencijalnog integriteta.

4.1. USKLADIŠTENE PROCEDURE

4.1. STORED PROCEDURES

Uskladištene procedure korištene su radi manjeg prometa mrežom (klijent šalje samo naziv procedure i parametre), ali i iz sigurnosnih razloga (sprječavanje korisnika da vidi tijelo unutar procedure, a time i eventualnu strukturu i objekte unutar baze podataka). Na taj način osigurana je komunikacija između klijent aplikacije i baze podataka te operacije poput dohvata, unosa, izmjena i brisanje podataka. [6]

Unutar uskladištene procedure (Primjer 1) implementiran je poziv skalarne funkcije koja izračunava trenutno stanje donacija u pojedinoj banci krvi prema krvnim grupama.

Korisno je imati uvid u stanje kako bi se znalo u koju banku krvi poslati nalog za potrebnim dozama.

Primjer 1. Uskladištena procedura za dohvat stanja donacija

Example 1. Stored procedure for fetching donation status

```
CREATE PROCEDURE usp_ukupno_donacija_prema_
tipu_krg

    @krg_id int,
    @bnk_id int

AS
BEGIN
    DECLARE @sum int
    SELECT @sum = [IO].[fn_broj_
donacija_tip_krg](@krg_id, @bnk_id)
    PRINT @sum
END
```

4.3. OKIDAČI

4.3. TRIGGERS

Kako bi se spriječila neovlaštena manipulacija postojećim objektima baze ili kreiranje novih objekata korišten je DDL (eng. *Data Definition Language*) okidač na razini baze podataka. [7] Isti okidač implementiran je kako bi u tablicu spremio podatke o korisniku koji je pokušao napraviti promjenu te upit koji je pokušao izvršiti. Osim DDL okidača, korišteni su i DML (eng. *Data Manipulation Language*) okidači kako bi bazu podataka zaštitili od unosa podataka koji ne zadovoljavaju postojeća pravila nad podacima. [8] Primjerice, okidač (Primjer 2) neće dozvoliti unos OIB-a koji nije duljine od 11 znakova, niti će dozvoliti unos spola koji nije muško ili žensko.

Primjer 2. DML (instead of) okidač za provjeru duljine OIB-a i vrijednosti spola

Example 2. OIB lenght and gender check DML (instead of) trigger

```
CREATE TRIGGER tr_oib_spolON [IO].[adk_
zaposlenik]
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS(SELECT * FROM Inserted
                WHERE LEN(oib) <> 11 or spol
                <> 'ž' and spol <> 'm')
    BEGIN
        RAISERROR('OIB mora sadržavati
```

```

11 znamenki, a spol mora biti
M/Ž!', 16, 1);
    ROLLBACK;
END
ELSE
    INSERT INTO [IO].[adk_zaposlenik]
    SELECT * FROM Inserted;
END CREATE TRIGGER tr_oib_spolON [IO].[adk_
zaposlenik]

INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS(SELECT * FROM Inserted
              WHERE LEN(oib) <> 11 or spol
                 <> 'ž' and spol <> 'm')
    BEGIN
        RAISERROR('OIB mora sadržavati
11 znamenki, a spol mora biti
M/Ž!', 16, 1);
        ROLLBACK;
    END
    ELSE
        INSERT INTO [IO].[adk_zaposlenik]
        SELECT * FROM Inserted;
    END
END

```

Također je implementiran i DML okidač koji sprječava daljnji unos podataka ukoliko je status testiranja "ne", odnosno donor je odbijen. Kako bi podaci unutar baze bili čitljivi kreirani su i DML okidači koji nakon unosa podataka poput naziva ustanova pretvaraju mala slova u velika.

5. ZAŠTITA PODATAKA KORIŠTENJEM KRIPTOGRAFSKIH ALGORITAMA I FUNKCIJA MASKIRANJA

5. DATA PROTECTION USING CRYPTOGRAPHIC ALGORITHMS AND MASKING FUNCTIONS

Kriptografija je korištena kako bi se prilikom unosa donorovih podataka šifrirao njegov OIB.

ime	prezime	oib	email	telefon
Ivan	Horvat	0x00B4B3CCB17B274CBA3CD2A680EE7CEC02000000F...	iXXX@XXXX.com	xxx221
Marija	Winson	0x00B4B3CCB17B274CBA3CD2A680EE7CEC02000000A...	mXXX@XXXX.com	xxx263
Erika	Tomić	0x00B4B3CCB17B274CBA3CD2A680EE7CEC020000005...	eXXX@XXXX.com	xxx200

Slika 3 Rezultat SELECT upita nad tablicom donor

Figure 3 Resultset of the SELECT query executed on the donor table

Prilikom šifriranja korišten je asimetrični kriptografski algoritam AES-256, a sam rezultat (šifrirani OIB) možemo vidjeti na slici (Slika 3).

Na istom primjeru (Slika 3) vidimo da su telefonski broj i adresa e-pošte prikazani u maskiranom obliku, što je rezultat korištenja SQL Server funkcija za maskiranje *email* i *partial*. [9] Na ovaj način osigurali smo tajnost podataka u bazi, i njihovu dostupnost samo onim korisnicima s odgovarajućim pravima i dozvolama u bazi podataka.

6. KOMUNIKACIJA IZMEĐU BAZE PODATKA I APLIKACIJE

6. COMMUNICATION BETWEEN DATABASE AND APPLICATION

Klijent aplikacija pomoću impersonalizacije autorizira se na bazu podataka. Korištenjem impersonalizacije sprječavamo slanje važnih podataka mrežom (npr. korisničko ime i lozinku stvarnog korisnika koji ima ovlasti nad bazom podataka), već impersonalizacijom takvog korisnika aplikacija dobiva sva potrebna prava i dozvole za rad s podacima. [10] U procesu impersonalizacije identitet stvarnog korisnika poznat je samo klijent aplikaciji te se u postupku autorizacije ne prenosi putem mreže. Nakon uspješne autorizacije klijent aplikacija omogućuje rad s podacima, tj. njihovo čitanje, unos, izmjenu i brisanje iz baze podataka.

Slika 4 prikazuje dijalog klijent aplikacije za unos podataka o novom donoru krvi. Nakon unosa i potvrde, podaci se spremaju u bazu podataka. Osim ove funkcionalnosti klijent aplikacija omogućuje i dohvat, ažuriranje i brisanje svih podataka koji se spremaju kao dio procesa dobrovoljnog darivanja krvi (Slika 1). Primjerice, kroz aplikaciju je omogućena obrada podataka o donoru, zaposleniku, donaciji, ustanovi, banci krvi i lokaciji.

Slika 4 Klijent aplikacija - unos podataka o novom donoru krvi

Figure 4 Client application - new donor data entry

Također, aplikacija omogućuje provjeru stanja u određenoj banci korištenjem prethodno implementirane uskladištene procedure (Primjer 1), slanje naloga prema banci krvi i slanje pošiljke prema ustanovi itd.

7. ZAKLJUČAK

7. CONCLUSION

U ovom radu opisan je razvoj dodanog modula za unaprjeđenje procesa dobrovoljnog darivanja krvi. Kao njegove glavne komponente istaknuli smo bazu podataka i klijent aplikaciju, tj. objekte baze podataka korištene za rad i manipulaciju nad podacima, sigurnosne komponente za autorizaciju i rad s korisnicima te sigurnost podataka korištenjem metoda šifriranja i maskiranja podataka. Konačno, prikazali smo kako se sadržaj baze podataka može uređivati novorazvijenom klijent aplikacijom.

Iz postojećeg sustava izdvojen je procesa dobrovoljnog darivanja krvi. Time je osiguran ispravan put krvi od donora do pacijenta u najjednostavnijem obliku. Ključna dobit koju donosi novi modul rasterećenje je medicinskog osoblja od poznavanja cijelog i kompleksnog sustava, te procesa s kojima nisu u neposrednoj interakciji. Potrebna količina interakcije korisnika sa sustavom smanjena je na minimum, čime je

sam proces postao jednostavniji i brži.

8. REFERENCE

8. REFERENCES

- [1.] »Microsoft,« Microsoft Corporation, [Mrežno]. Available: <https://www.microsoft.com/en-us/sql-server>. [Pokušaj pristupa 10. 07. 2020.].
- [2.] »Hemasoft We care,« Hemasoft, [Mrežno]. Available: <http://www.hemasoft.com/donorCenter.html>. [Pokušaj pristupa 10. 07. 2020.].
- [3.] Balija M., Očić T. i Herceg M., »Informatizacija transfuzijske djelatnosti RH - Izvješće za 2011.g,« Transfuziološki vjesnik (51), 2011.
- [4.] Balija M., Herceg M., Očić T., Kovač T. i Vuković Z., »Informatizacija transfuzijske službe RH,« Transfuziološki vjesnik (55), 2015.
- [5.] Šuman S., Jakupović A. i Liverić D., »Uvod u formalizaciju metoda modeliranja podataka,« Zbornik Veleučilišta u Rijeci (3), pp. 83-92, 2015.
- [6.] Massod-Al-Farooq B., SQL Server 2014 Development Essentials, Packt Publishing Ltd., 2014.

- [7.] Nielsen P., White M. i Parui U., »DDL Triggers,« u Microsoft SQL Server 2008 Bible, Wiley Publishing Inc., 2009.
- [8.] Rankins R., Bertucci P., Gallelli C. i Silverstein A., »Using DML Triggers,« u Microsoft SQL Server 2008 R2 Unleashed, Sams Publishing, 2010.
- [9.] Gorman K., Hirt A., Noderer D., Rowland-Jones J., Sirpal A., Rayan D. i Woody B., »Introducing Microsoft SQL Server 2019,« Packt Publishing, 2019.
- [10.] Bruchez R., »Using EXECUTE AS to change the user context,« u Microsoft SQL Server Security Cookbook, Packt Publishing Ltd., 2012.

AUTORI · AUTHORS



• Nikolina Jurak

Završila preddiplomski stručni studij Primjena informacijske tehnologiju u poslovanju na Fakultetu organizacije i informatike u Varaždinu. Trenutno studentica

specijalističkog diplomskog studija informatike na Tehničkom veleučilištu u Zagrebu. Kroz dosadašnje radno iskustvo sudjeluje na projektima koji uključuju korištenje Oracle baze podataka i Oracle Application Express (APEX) razvojne platforme. Slobodno vrijeme posvećuje istraživanju i nadopuni znanja iz područja baza podataka te učenju C# programskog jezika.

Korespondencija · Correspondence

nikolina.jurak@tvz.hr



• Željko Kovačević

Viši je predavač na Tehničkom veleučilištu u Zagrebu gdje sudjeluje u nastavi iz kolegija orijentiranih prema učenju programskih jezika i baza podataka. Diplomirao je na

Tehničkom veleučilištu u Zagrebu 2010.g, a zvanje višeg predavača stječe 2020.g. Autor i koautor je pet knjiga i desetak stručnih i znanstvenih radova iz područja programskih jezika i baza podataka, a 2015.g. dobiva posebno priznanje MVP (Most Valuable Professional) tvrtke Embarcadero za rad u RAD Studio alatima C++ Builder i Delphi. Trenutno pohađa doktorski studij Računarstva i Informatike u Mariboru, gdje radi na istraživanju iz područja domensko specifičnih programskih jezika.

Korespondencija · Correspondence

zeljko.kovacevic@tvz.hr