

KORIŠTENJE ON-LINE BAZA PODATAKA KROZ JAVA DESKTOP APLIKACIJE USE OF ON-LINE DATABASES IN JAVA DESKTOP APPLICATIONS

Edita Zec, Željko Gavrić

REZIME: Napredak informaciono-komunikacijskih tehnologija ima za posljedicu da se sve više koriste napredni računari koji imaju napredne računarske i komunikacijske sposobnosti. *Sistemi baza podataka nalaze se svuda, a oni ne samo da čuvaju podatke, već također obezbjeđuju sredstva za pristup, ažuriranje, rukovanje i analizu podataka.* U ovom radu su predstavljeni sistemi baza podataka, sistemi za upravljanje bazama podataka, osnovni pojmovi vezani za Java objektno-orijentisano programiranje, te razvoj Java aplikacija, isključivo onih koje koriste on-line baze podataka.

KLJUČNE RIJEČI: Baza podataka, sistem za upravljanje bazom podataka, Java, objektno-orijentisano programiranje.

ABSTRACT: The advancement of information and communication technologies has resulted in the increasing use of advanced computers with advanced computing and communication skills. Database systems are everywhere, and they are not only used to store data, but also provide resources for accessing, updating, handling and analyzing data. This paper presents database systems, database management systems, basic concepts related to Java object-oriented programming, and development of Java applications, exclusively for those using on-line databases.

KEY WORDS: Database, database management system, Java, object-oriented programming.

1. UVOD

Baza podataka je strukturno povezan skup podataka. Glavni način organizacije podataka u svim sistemima su baze podataka. Ako se na primjer kupovina obavlja putem internet trgovine, informacije o transakcijama se čuvaju u bazi podataka kojom upravlja kompanija. Na univerzitetima akademski podaci se čuvaju u bazi podataka kojom upravlja univerzitet. Baza podataka može biti bilo šta, od obične liste za kupovinu, preko popisa slika umjetničke galerije, pa sve do velike količine podataka u korporacijskim mrežama. Za unos, pregled, izmjenu ili brisanje podataka u bazama podataka potreban je sistem za upravljanje bazama podataka. Jedan od sistema za upravljanje bazama podataka koji je veoma često korištene je MySQL.[1]

Java je računarski programski jezik koji je razvila firma Sun Microsystems. Programski jezik se koristi za izdavanje instrukcija računaru da obavi konkretne poslove. Java programski jezik je nastao 1995. godine, i od tada je stekao veliku popularnost. Koristi se za razvoj raznovrsnih aplikacija. Postoje jednostavni tekstualno-zasnovani programi koji se nazivaju konzolne aplikacijama. Takve aplikacije podržavaju samo tekstualni unos i ispis na ekranu. Takođe, mogu se praviti aplikacije sa grafičkim korisničkim interfejsom (engl. *GraphicalUserInterface – GUI*). Ove aplikacije raspolazu sa menijima, paletama alatki, dugmadima, trakama za pomeranje sadržaja, te drugim kontrolama.

2. BAZE PODATAKA

Od samog početka korištenja računara, obrada različitih vrsta podataka bila je jedan od osnovnih zadataka. Podaci i informacije su postali pokretačka snaga modernog poslovanja u cijelom svijetu. Kada želimo da imamo kvalitetne informacije o svim segmentima našeg poslovnog ili privatnog života, najbolje je da na određen način organizujemo sve podatke od važnosti u određene strukture. Ovo je posebno bitno kada je u kratkom roku potrebno donijeti kvalitetnu ili sudbonosnu odluku. Podaci se smiještaju u tabele, koje imaju koncizno definisana zaglavlja.

Uglavnom se u bazi podataka nalazi više tabela koje obuhvataju sve segmente interesovanja. Svi ti segmenti se nerijetko zbog svoje prirode moraju organizovati u posebne tabele, a te tabele se mogu povezivati preko određenih zajedničkih elemenata. Skup više tabela koje služe jednom zajedničkom cilju, skupa sa njihovim veznim elementima naziva se *bazom podataka*.^[3] Njihova zajednička svrha je dobijanje veoma brze i uspješne informacije o svim događajima koji se dešavaju unutar jedne cjeline. Kada kucamo nešto u tekstualnom procesoru, ili vršimo neke tabelarne proračune u pogramu za tabelarnu obradu, onda imamo dodira sa bazom podataka. To je u stvari pitanje organizacije naših podataka. Ako kreiramo datoteke u tekstualnom procesoru i smještamo ih po određenim direktorijima, na neki način organizujemo bazu podataka.

2.1. Definicija baze podataka

Jednostavno rečeno, baza podataka je softverska konstrukcija namjenjena za skladištenje, analizu i pretraživanje grupe srodnih i povezanih podataka, kao što su podaci o kupcima, pacijentima, telefonskim brojevima i sl.

Baza podataka se sastoji od jedne ili više (dvodimenzionalnih) tabela koje međusobno mogu biti povezane. Svaka tabela čuva istovrsne podatke (npr. podatke o nekoj osobi, predmetu i sl.). Svaki red u tabeli predstavlja jedan *slog* u tabeli (najmanja grupa podataka u bazi koja u potpunosti opisuje neki od koncepta koje baza modelira), dok svaka kolona predstavlja jedno od *polja* unutar tog sloga. Dakle, slog može biti grupa podataka koja opisuje npr. neku osobu, a polja unutar tog sloga mogu sadržavati ime, prezime, adresu stanovanja ili datum rođenja te osobe. Slog se u literaturi još ponegdje naziva i *entitet*, a polje se naziva *atribut*. Svaki slog tabele se može jedinstveno identificirati putem jedne ili kombinacijom vrijednosti nekog polja tog sloga. To polje ili kombinaciju polja tada nazivamo *primarni dio* ili *peimarni ključ*. Tako neku osobu može jedinstveno identifikovati njen matični broj ili kombinacija vrijednosti polja imena i prezimena. U jednoj tabeli može postojati više polja ili kombinacija polja koji mogu biti kao primarni ključ.

Pored toga što primarni ključ ima ulogu jedinstvenog identifikovanja sloga on ima važnu ulogu i u povezivanju tabela. Ukoliko tabela na primjer predstavlja listu pisaca, te pored te tabele postoji i tabela lista knjiga, te je potrebno ove dvije tabele povezati kako bi smo znali koji je pisac napisao koju knjigu. Ako u slog knjige ubacimo polje koje sadrži vrijednost primarnog ključa pisca, ove dvije tabele su povezane. Novo polje (koje isključivo služi za povezivanje dvije tabele) u tabeli se zove *strani ključ* ili *spoljni ključ*. Ovakav način povezivanja nazivamo relacioni model baza podataka.

2.2. ORGANIZACIJA PODATAKA

Organizacija podataka je od velikog značaja kada potrebno raditi sa bazom podataka. Jedan od ključnih aspekata dobrog kreiranja baza podataka jeste kako će podaci biti organizovani u bazi podataka. Da bi postigli dobro kreiranu bazu podataka, podatke bi trebalo organizovati tako da su lako dostupni i da omogućavaju lako održavanje baze podataka.

Treba odrediti koji će podaci ulaziti u bazu podataka, zatim koji će se podaci smjestiti u određene tabele među kojima će biti uspostavljen odnos, te kakav je odnos među tim podacima. Potrebno je smanjiti mogućnost da se isti podatak zapisuje više puta (redundacija), jer višestrukim zapisivanjem nastaju problemi očuvanja stvarne, jedinstvene vrijednosti svih podataka pri ažuriranju. To utiče i na pouzdanost informacija koje se dobijaju iz tih podataka. Potrebno je upravljati smještanjem podataka i očuvanja tih podataka od namjernih i nenamjernih uništenja, tj. da ne dođe do gubitka integriteta podataka. Neke podatke treba zaštititi od toga da ih neovlašteni korisnici ne mijenjaju, što se zove tajnost ili privatnost podataka.

2.3. RELACIONI MODEL BAZA PODATAKA

Relaciona baza podataka se sastoji od serije dvodimenzionalnih tabela [2]. Termin „relaciona baza podataka“ dolazi od činjenice da ona koristi relaciju (odnos) umjesto datoteke. *Relacija* je tabela sastavljena od slogova. Unutar jedne tabele može postojati samo jedna vrsta slogova ili entiteta. Relacione tabele pokazuju logičke, a ne fizičke odnose, a zanemaruje se redoslijed podataka, odnosno slogova, uključenih u relaciju. Relacioni model odvaja bazu podataka od operativnog sistema, kao i od aplikacije. Kada se da zahtjev za informacijama, sistem napravi tabelu koja sadrži te informacije.

Standardni programski jezik za izražavanje pristupa podacima i manipulaciju sa tabelama u relacionoj bazi podataka se naziva SQL (eng. *Structured Query Language*). Softverski program, koji se zove prirodni jezik (eng. *natural language*) i koji dozvoljava upite u ograničenoj formi prirodnog jezika, analizira korisnikov upit, prevodi ga u upit na SQL programskom jeziku, prenosi SQL zahtjev DBMS-u (eng. *Data Base Menagment System*) i daje na ekranu rezultujuće podatke korisniku. Relacioni model je smišljen početkom osamdesetih godina od strane Ted Cotta, uposlenika IBM korporacije i trenutno je najraširenija paradigma za razvoj podataka.

3. SISTEMI ZA UPRAVLJANJE BAZAMA PODATAKA

U slučaju kada baza postane toliko komplikovana da više nismo u stanju da jednostavno kontrolišemo razvoj podataka, potrebno je preći na viši nivo organizacije podataka i tj. koristiti sistem za upravljanje bazom podataka. Postoji više sistema za rad sa bazama podataka kao što su: MySQL, DBSM, ACCESS, MICROSOFT SQL, ORACLE i dr.

MySQL je sistem za upravljanje bazama podataka. Riječ SQL u MySQL znači “Structured Query Language” (strukturni jezik za pretraživanje) – najčešći standardizovan jezik za pristupanje bazama podataka. MySQL se cijenom i prihvatljivim performansama nametnuo kao odlično rješenje za male i srednje veb sajtove, bez obzira rade li oni na Linux ili Windows operativnom sistemu. Ukoliko imamo sajt koji će istovremeno mijenjati veliki broj korisnika, MySQL se ponovo nameće kao bolje rješenje. Glavni nedostatak mu je to što ne podržava tzv. *stored procedure*, što može nedostajati prilikom razvoja velikih projekata. *Stored procedures* ili pohranjene procedure pomažu u smanjenju mrežnog prometa smještanjem programske logike na poslužitelja. Pohranjene procedure se mogu koristiti ako je potrebno smjestiti programersku logiku unutar grupe SQL izraza. Pored toga, MySQL ne podržava ni podupite (eng. *subqueries*) što je prije svega praktičan problem obzirom da se većina takvih upita može prepisati na drugačiji način. Podrška za spoljne ključeve postoji na transparentnom nivou, odnosno samo da bi se zadovoljila sintaksna kompatibilnost sa drugim SQL implementacijama, pa možemo reći da spoljni ključevi u MySQL-u zapravo ni ne postoje.

MySQL je proizvod koji nudi besplatan izvorni kod (eng. *open-source*), što znači da je dozvoljeno svakome da ga koristi i prilagođava za svoje potrebe. Svakome je omogućeno preuzimanje MySQL-a sa interneta i korištenje bez obaveze plaćanja. MySQL spada pod GLP (eng. *GNU General Public License*). Takođe postoji MySQL Enterprise verzija programa koja nije besplatna i koja nudi dodatne opcije naprednim korisnicima.

4.1. Sistemi za upravljanje bazama podataka kao cloud servisi

Većina sistema za upravljanje bazama podataka su jednostavno softverki paketi pomoću kojih korisnici kreiraju, održavaju i koriste bazu podataka. Međutim, od uvođenja *cloud* računarstva, sistemi za upravljanje bazama podataka su prešli u sasvim novu vrstu usluge sa mnogim prednostima. Tradicionalni sistemi za upravljanje bazama podataka jednostavno nisu podešeni ili opremljeni da bi se bavili zahtjevima *cloud* računarstva. Ovakvi sistemi su izuzetno skalabilni, što znači da su u stanju da obrađuju količine podataka koje bi iscrpile tradicionalne sisteme za upravljanje bazama podataka. Također, ovi sistemi su projektovani kao prilagodljiva usluga dostupna na *cloud* infrastrukturi. Budući da online sistemi za upravljanje bazama podataka ispunjavaju određene standarde po pitanju skalabilnosti, pouzdanosti i sigurnosti, njihova implementacija u kratkoročnim projektima i malim aplikacijama u odjeljenjima predstavlja idealno rješenje. Na ovaj način značajno je smanjenje troškova u odnosu na implementaciju unutar IT odjela.[9]

4. PROGRAMSKI JEZIK JAVA

Pojam *programiranje* znači kreirati (ili razvijati) softver, koji se vrlo često naziva *program*. Jednostavno rečeno, softver sadrži instrukcije koje računaru (ili drugom kompjuterizovanom uređaju) govore šta treba da radi. Softver je svuda oko nas, od personalnih računara, do određenih uloga u pokretanju aviona, automobila, mobilnih telefona, pa čak i tostera. Na personalnom računaru koristimo softver da pišemo tekstualne dokumente. Veb pretraživače koristimo za pretraživanje na internetu, a e-mail programe za slanje i primanje elektronske pošte. Svi ovi programi su primjeri softvera. Programeri pišu softver uz pomoć moćnih alata poznatih kao *programski jezici*. [5]

Postoji veliki broj različitih programskih jezika, a neki od njih su stari nekoliko desetina godina. Jedan od najpopularnijih programskih jezika današnjice je upravo *Java*. Java je jednostavan, objektno orijentisan, siguran, prenosiv, dinamičan programski jezik visokih performansi. Postao je veoma popularan zbog toga što se programi pisani u Javi mogu pokretati u Veb pretraživačima. Danas se ne primjenjuje samo u veb programiranju, već i za razvoj aplikacija na serverima, personalnim računarima, te mobilnim uređajima. Mnoge kompanije koriste Javu za kreiranje distribuiranih aplikacija kojima pristupaju klijenti i partneri putem interneta. Takođe, softveri za Android mobilne telefone se najčešće pišu pomoću Jave.

Za *Javu* danas možemo reći da je:

- specifikacija programskog jezika i standardni zbir klasa;
- implementacija navedenog programskog jezika i njegovih pratećih datoteka (eng. *libraries*) u okolini prevođenja i izvođenja (eng. *compile and run time environment*) za izradu i izvršavanje aplikacija;
- implementacija navedenog programskog jezika kao podskup ugrađenog koda u HTML stranicama (eng. *applet*);
- implementacija navedenog programskog jezika kao datak animaciji i interakciji kod 3D objekata i scena.

Možemo tvrditi da svaku od datih stavki prezentuje drugačija implementacija Jave. Svaka od njih ima svojih prednosti, kao i ograničenja. Važno je razumjeti, iako Java programski jezik podržava neke vrste manifestacija, to ne znači da će pojedina manifestacija biti dopuštena, ili čak moguća u svim implementacijama. Gledano s praktične strane, performansa i sigurnost nam ograničavaju šta danas možemo napraviti unutar Java programskog okruženja.

4.1. Java kao jezik za pisanje aplikacija

Mnogo ljudi misli da Javu koristimo samo kao dodatak vlastitim Veb HTML stranicama tako da rade interesantne efekte sa slikama ili računaju koliko ste dana ili minuta stari. Međutim, Javu možemo koristiti isto kao C++ ili C#, za pisanje samostalnih aplikacija. Postupak je sljedeći:

- prevodimo Java izvorni kod u tzv. *bajtni kod*;
- izvršavamo bajtni kod interpretiranjem unutar Java Virtualne Mašine (eng. *Java Virtual Machine*).

Po strukturi implementacije programskog jezika vidi se da je Java negdje na prelazu između do sad kristalno jasnih definicija pojmova *kompajler* i *interpreter*. Ona spada u obje klasifikacije dok zapravo nije potpuno nijedna od njih.

Sam bajtni kod je dosta manji od ekvivalentnog izvršnog koda C programskog jezika, ali brzina njegovog interpretiranja daleko zaostaje naspram brzine izvršavanja ekvivalentnih programa pisanih u C-u. Alternativno rješenje je u korištenju tzv. *just-in-time* interpretiranja, gdje JVM prebacuje bajtni kod u *native kod* prije samog izvršavanja. To za manje aplikacije pruža razumnu performansu uz zadržanu prenosivost izvršnog koda.

Međutim, čak i toliko naglašavana prenosivost možda nije zasnovana na čvrstim temeljima. Veće aplikacije će vjerovatno, zbog ograničenja nametnutih standardnim Javaokruženjem, biti prisiljene koristiti vlastite interfejsa u vidu Java izvornog interfejsa (eng. *Java Native Interface*) čijim se funkcijama proširuju mogućnosti Jave. U tom slučaju, mada je osnovna aplikacija prenosiva, morat će se prilagođavati od operativnog sistema do različitih arhitektura računara i datoteke pisane u C/C++-u, bez kojih osnovna aplikacija neće moći raditi.

Java je prvi široko rasprostranjeni programski jezik koji na de facto standardan način podržava

programiranje grafičkog korisničkog interfejsa. [6]

4.2. Osnovni pojmovi objektno-orijentisanog programiranja

Kada se koristi programski jezik koji nije objektno orijentisan, rješenja se izražavaju u terminima brojeva i karaktera. U objektno-orijentisanim programskim tehnikama, program se piše koristeći objekte i klase. Objekat može biti kako fizički tako i logički entitet, dok je klasa samo logički entitet. Objekat je integralna cjelina podataka i procedura za rad sa njima. Objektno orijentisano programiranje je programska paradigma zasnovana na skupu objekata koji djeluju međusobno. Glavne obrade zasnivaju se na rukovanju objektima. Objekat ima tri karakteristike [5]:

- stanje: predstavlja podatke (vrijednosti) objekta;
- ponašanje: predstavlja ponašanje (funktionalnost) objekata;
- identitet: identitet objekta se tipično implementira preko jedinstvenog ID. Vrijednost ID nije vidljiva vanjskom korisniku, ali se ona koristi interno od strane JVM (Java Virtual Machine) da identifikuje svaki objekt na jedinstven način.

Klasa je opis (specifikacija) jedne vrste objekata. To je grupa objekata koji imaju zajedničke osobine, pa se može reći i da je klasa šablon ili nacrt po kome se objekti kreiraju. Nazivi Java klase počinju velikim slovom. Definicija svake Java klase nalazi se u posebnoj datoteci koja ima isto ime kao i klasa, te ima ekstenziju *.java*. Klasa u Javi može sadržati:

- podatak-član,
- metodu,
- konstruktor,
- blok,
- klasu i interfejs.

Instanca klase je konkretan objekat klase. Instancne promenljive klase su podaci koji opisuju osobine objekata te klase. Varijabla koja je kreirana unutar klase ali izvan metoda, se naziva instancna varijabla. Instancna varijabla ne dobija memoriju u vremenu kompajliranja. Ona dobija memoriju u vremenu izvršavanja /RUNTIME/ kada je objekt (instancja) kreiran. To je razlog zbog kojeg se naziva instancna varijabla.

Promjenljive mogu biti primitivnog tipa ili objekti neke druge klase, na primjer:

- za automobil: boja, proizvođač, model;
- za pravougaonik: dužine stranica (a, b);
- za krug: koordinate centra (x, y) i poluprečnik.

5. KORIŠTENJE ONLINE BAZE PODATAKA U JAVA APLIKACIJI

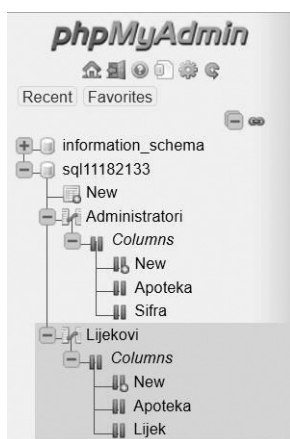
Ovo poglavlje prikazuje praktičan primjer korištenja baze podataka na internetu, kroz Java desktop aplikaciju. Bazi podataka se pristupa koristeći Java kod. Razvojno okruženje korišteno za izradu ovog primjera je NetBeans.

5.1. Struktura baze podataka i JDBC

Java koristi nešto što se zove JDBC ili Java veza sa bazom podataka (eng. *Java Database Connectivity*) za spajanje sa bazama podataka. JDBC dozvoljava spajanje sa širokim spektrom baza podataka, uključujući Oracle, MySQL i druge.[7]

U ovom primjeru korištena je MySQL baza podataka koja se nalazi na internetu, i to na: <https://www.freemysqlhosting.net/>. Prednost korištenja online baze podataka je ta što se toj bazi može pristupiti sa različitih računara, u bilo koje doba.

Baza iz ovog primjera se sastoji iz dvije jednostavne tabele, što se može vidjeti na slici 1.



Slika 1 – Prikaz tabela iz baze podataka

Tabela *Administratori* sadrži spisak administratorskih korisničkih imena i lozinki, potrebne za administratorski pristup bazi podataka. Tabela je prikazana na slici 2.

Tabela *Lijekovi* sadrži podatke koje Administratori unose u bazu podataka, a koje su potrebne korisnicima. U ovom

primjeru, to su podaci o dostupnosti lijekova u apotekama na području jednog grada. Tabela je prikazana na slici 3.

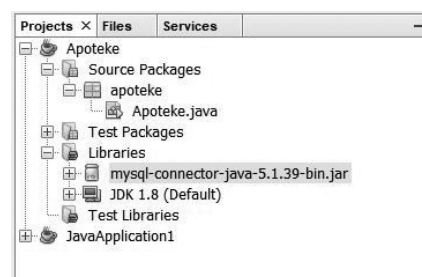
+ Options	
Apoteka	Sifra
Moja apoteka	123456
Betty apoteka	222222
Gradska apoteka	111111
Nova apoteka	123456

Slika 2 – Tabela Administratori

+ Options	
Lijek	Apoteka
	Moja apoteka
Acidi Borici	Betty apoteka
Amoxicilin	Gradska apoteka
Brufen	Betty apoteka
Brufen	Gradska apoteka
Brufen	Nova apoteka
Byol	Moja apoteka
Caffetin	Gradska apoteka
Linex	Betty apoteka
Loperamid	Moja apoteka
Nalgesin	Betty apoteka
Paracetamol	Betty apoteka
Vitamin C	Betty apoteka

Slika 3 – Tabela Lijekovi

Verzija JDBC-a korištena u ovom primjeru, iz razvojnog okruženja NetBeans IDE 8.2, prikazana je na slici 4.



Slika 4 – Verzija JDBC-a

Sljedeći kod je neophodan za ostvarivanje konekcije za spajanje sa bazom podataka:

```
String host = "jdbc:mysql://sql11.freemysqlhosting.net:3306/sql11182133";
String UName = "sql11182133";
String UPass = "FACuPXy3wf";
Connection con = DriverManager.getConnection(host, UName, UPass);
```

5.2. SQL naredbe

Skraćenica SQL (eng. *Structured Query Language*) označava programski jezik namijenjen za upravljanje podacima u relacionim sistemima za upravljanje bazama podataka. Pomoću ovog alata mogu se odabirati, unositi, brisati i ažurirati podaci, kreirati i brisati tabele, i još mnogo toga.

SQL koristi jednostavne ključne riječi za obavljanje funkcija. Na primjer, za prikaz svih podataka iz tabele, koriste se ključne riječi SELECT i FROM, uz simbol *:

```
SELECT * FROM ime_tabele
```

Korištenjem sljedećeg koda u datom primjeru, realizuje se funkcija prikaza podataka iz tabele *Lijekovi*:

```
Statement zahtjevZaRez = con.createStatement();
String zahtjev = "SELECT * FROM Lijekovi";
ResultSet rezultat = zahtjevZaRez.executeQuery(zahtjev);
```

Za upisivanje u bazu podataka koriste se ključne riječi INSERT INTO, s tim što se redosljed unošenja vrijednosti mora poklapati sa redosljedom kolona u tabeli. Sintaksa INSERT naredbe je sljedeća:

```
INSERT INTO ime_tabele
VALUES (vrijednost1, vrijednost2, vrijednost3, ...);
```

U primjeru je korišten sljedeći kod za unos novih podataka u tabelu *Lijekovi*:

```
String unos = "INSERT INTO `Lijekovi`(`Lijek`, `Apoteka`)"+
VALUES (?,?,)";
PreparedStatement preparedStmt = con.prepareStatement(unos);
preparedStmt.setString (1, noviUnos);
preparedStmt.setString (2, aktivnaApoteka);
preparedStmt.execute();
con.close();
```

6.3. WHERE klauza

WHERE klauza se koristi za filtriranje spremljenih podataka. Uglavnom se koristi za izdvajanje samo onih unosa koji ispunjavaju određeni uslov. Može se kombinovati uz SELECT, UPDATE, DELETE i druge SQL naredbe.

Sintaksa korištenja WHERE klauze u kombinaciji sa SELECT naredbom je sljedeća:

```
SELECT kolona1, kolona2, ...
FROM ime_tabele
WHERE uslov;
```

U programu je korištena DELETE naredba u kombinaciji sa WHERE klauzom, što je prikazano u sljedećem kodu:

```
Statement zahtjevZaRez = con.createStatement();
String sql = "DELETE FROM `Lijekovi` WHERE `Lijek` = '" + obrisiLijek + "' AND `Apoteka` = '" + aktivnaAp + "'";
zahtjevZaRez.executeUpdate(sql);
```

6.4. Grafičko okruženje i funkcionalnosti aplikacije

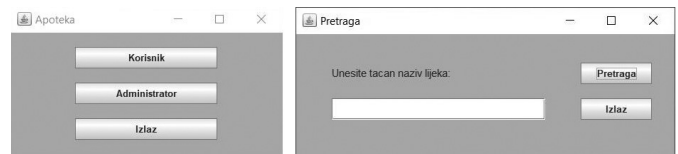
U narednim koracima prikazana je Java desktop aplikacija, čiji je zadatak, kao što je već navedeno, pretraživanje dostupnosti lijekova u apotekama na području jednog grada. Aplikaciju mogu koristiti korisnici, koji mogu samo vršiti čitanje iz baze podataka, i administratori koji vrše pretragu, upisivanje i brisanje iz online baze podataka.

1. Glavni meni prilikom pokretanja aplikacije vidi se na sljedećoj slici:



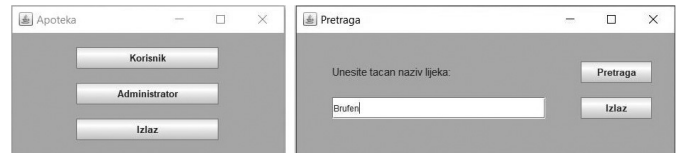
Slika 5 – Glavni meni

2. Odabirom opcije *Korisnik*, pokreće se korisnička verzija programa:



Slika 6 – Korisnička verzija programa

3. Pretraga baze podataka od strane korisnika pokreće se klikom na dugme *Pretraga*, kojim se vrši pretraživanje unesenih podataka u tabeli *Lijekovi* u online bazi podataka.



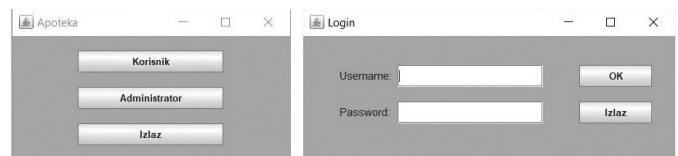
Slika 7 – Pretraga

4. Rezultat pretrage za korisnika prikazan je na sljedećoj slici. Za povratak na glavni meni, dovoljno je odabrati opciju *Izlaz*.



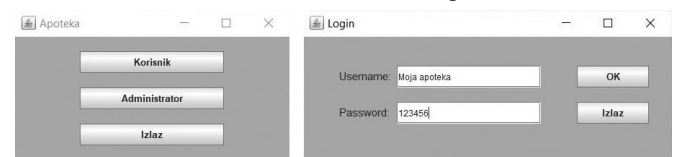
Slika 8 – Rezultat pretrage

5. Odabirom opcije *Administrator* sa glavnog menija, otvara se sljedeći prozor:



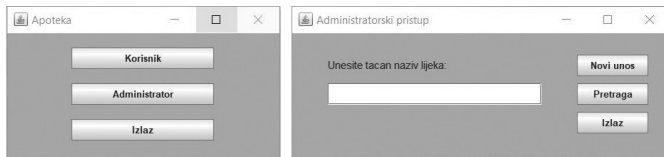
Slika 9 – Administrator

6. Nakon unošenja korisničkog imena i lozinke, klikom na dugme *OK* vrši se provjera tačnosti unesenih podataka u tabeli *Administratori* u online bazi podataka.



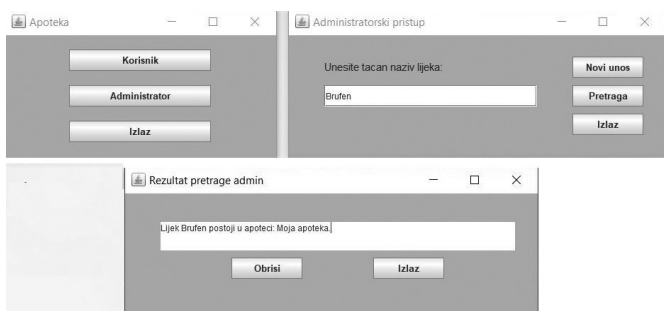
Slika 10 – Administrator Login

7. Ukoliko su podaci o administratoru bili tačni, otvara se sljedeći prozor za administratore, sa opcijama novog unosa u bazu i pretraživanja baze.



Slika 11 – Administratorski pristup

8. Klikom na dugme *Novi unos*, uneseni naziv lijeka se prosljeđuje na upis u online bazu podataka, zajedno sa nazivom trenutno aktivne apoteke. Klikom na dugme *Pretraga*, otvara se sljedeći prozor sa rezultatom pretraživanja baze.



Slika 12 – Novi unos

9. Ukoliko je pronađen rezultat pretrage, postoji opcija brisanja datog unosa. Klikom na dugme *Obrisi* administrator vrši brisanje iz online baze podataka. Povratak na meni za pretragu vrši se klikom na dugme *Izlaz*.

7. ZAKLJUČAK

Razvojem informacionih sistema i težnjom istih kao distribuiranom pristupu od strane korisnika došlo je do potrebe da se kompletan sistem, ili neki njegov dio smjesti na lokaciju koja je dostupna sa bilo kog mjesta u svijetu. U ovom radu je opisan primjer gdje se sama aplikacija izvršava lokalno, dok se baza podataka nalazi na udaljenoj lokaciji. Ovakve baze podataka nazivaju se on-line baze podataka.

U zavisnosti od zahtjeva koje treba ispuniti prilikom izrade aplikacije, odlučuje se da li koristiti klasičnu ili on-line bazu podataka. On-line baza podataka podrazumijeva bazu podataka koja je klijentima dostupna na zahtjev putem Interneta, a smještena je na serverima hostujućeg provajdera. Ovakva baza

podataka može biti tradicionalna baza podataka poput MySQL ili SQL Server baze podataka koja je usvojena za on-line upotrebu. On-line baze podataka mogu ponuditi značajne prednosti u odnosu na njihove tradicionalne ekvivalente. Te prednosti uključuju povećanu mogućnost pristupa, minimalno ulaganje i održavanje hardvera, te potencijalno bolje performanse[8]. Pristup on-line bazi podataka moguć je u bilo koje vrijeme, sa bilo kog mjesta. Jedini uslov jeste da postoji Internet veza.

8. LITERATURA

- [1] Uvod u MySQL. Dostupno na: <http://www.vitez-studios.com/web-development/hrvatski-tutorijali/uvod-u-mysql.html> [pristupljeno 25.11.2018]
- [2] Bjeladinović, S., Marjanović, Z. (2016). Primena virtuelizacije u cilju integracije relacionih i NoSQL baza podataka, na primeru Oracle-a. vol. 15, br. 57, str. 4-10, InfoM, Fakultet organizacionih nauka Beograd.
- [3] Latinović, B. (2007). *Informacione tehnologije*, Banja Luka: Panevropski univerzitet APEIRON.
- [4] Liang, Y.D. (2011). *Introduction to Java programming*, Comprehensive version. New Jersey: Pearson Education.
- [5] Java tutorijali OOP. Dostupno na: <http://www.znanje.org/knjige/computer/Java/ib01/300Java/31000120.htm> [pristupljeno 27. 11.2018]
- [6] Univerzitet u Kragujevcu, Prirodno-matematički fakultet, Institut za matematiku i informatiku. Dostupno na: <https://imi.pmf.kg.ac.rs/moodle/> [pristupljeno 27. 11.2018]
- [7] http://www.homeandlearn.co.uk/java/sql_commands.html [pristupljeno 28.11.2018]
- [8] https://www.webopedia.com/TERM/C/cloud_database.html [pristupljeno 28.11.2018]
- [9] Gelogo, Y.E., Lee, S. (2012). Database Management System as a Cloud Service. *International Journal of Future Generation Communication and Networking*. Vol. 5, Br. 2, str. 72-74.



Edita Zec, asistent, Slobomir P Univerzitet, Doboj Bijeljina, BIH
Kontakt: m.edita@hotmail.com
Oblast interesovanja: Programiranje, Baze podataka, Internet tehnologije



Mr. Željko Gavrić, istraživač saradnik, Fakultet organizacionih nauka, Univerzitet u Beogradu
Kontakt: gavric.zeljko@yahoo.com
Oblast interesovanja: Programiranje, HCI, Internet tehnologije

CIP – Каталогизacija u publikaciji Narodna biblioteka Srbije, Beograd 659.25

INFO M : časopis za informacione tehnologije i multimedijalne sisteme = journal of Information technology and multimedia systems / glavni i odgovorni urednik Miroslav Minović. - [Štampano izd.]. - God. 1, br. 1 (2002)- . - Beograd : Fakultet organizacionih nauka, 2002- (Novi Sad : Sajnos). - 30 cm Dostupno i na: <http://www.infom.org.rs>. - Tromesečno. - Tekst ćir. i lat. - Je nastavak: Info Science = ISSN 1450-6254. - Drugo izdanje na drugom medijumu: ISSN 1451-4397 = Info M (Štampano izd.)

COBISS.SR-ID 105690636