**UNIVERZITET U BIHAĆU**

**TEHNIČKI FAKULTET**

**BIHAĆ**

**PREDMET: Osnovne programskog jezika JAVA**

**TEMA: Laboratorija**

**PROFESOR/ICA:** dr. Jasna Hamzabegović **STUDENTICA:** Sumea Kličić

**ASISTENT/ICA:** Amel Džanić **BROJ INDEXA:** 1285

Bihać, 2025

**Sadržaj**

[**OPIS APLIKACIJE:** 1](#_Toc200999330)

[**1.ER MODEL** 2](#_Toc200999331)

[**2.PRIJAVA ADMINA** 3](#_Toc200999332)

[**3. GLAVNI PROZOR** 4](#_Toc200999333)

[**4.OPCIJE ZA ENTITET – PACIJENT** 5](#_Toc200999334)

[**4.1.Dodavanje novog pacijenta** 5](#_Toc200999335)

[**4.2. Pretraga pacijenta zajedno sa ostalim funkcionalnostima** 6](#_Toc200999336)

[**5.OPCIJE ZA ENTITET - NALAZ** 7](#_Toc200999337)

[**5.1.Dodavanje nalaza** 8](#_Toc200999338)

[**5.2.Pretraživanje nalaza** 9](#_Toc200999339)

[**6. OPCIJE ZA ENTITE – PRETRAGA** 11](#_Toc200999340)

[**6.1. Dodavanje nove pretrage** 11](#_Toc200999341)

[**6.2.Pretraga + Ažuriranje pretrage** 12](#_Toc200999342)

[**7.Nalaz-pretraga N:M veza** 13](#_Toc200999343)

[**7.1. Povezivanje nalaza sa pretragama** 13](#_Toc200999344)

[**7.2.Brisanje veza** 14](#_Toc200999345)

[**ZAKLJUČAK:** 15](#_Toc200999346)

# **OPIS APLIKACIJE:**

Aplikacija se bavi organizacijom pacijenata, nalaza i pretraga unutar jedne laboratorije. Pacijenti dolaze na preglede gdje mogu raditi nalaze. Jedan pacijent može odraditi više nalaza. Pošto nalaz ima datum omogućeno je da pratimo koliko često i kad je radio koji nalaz. Jedan nalaz pripada tačno jednom pacijentu pa tako tabela nalaz ima ključ od pacijenta kojem pripada. Ovo je tipičan primjer 1:N veze gdje pacijent učestvuje više puta u vezi, a nalaz samo jednom. Unutar jednog nalaza se može nalaziti više pretraga. Zbog preglednosti maksimalan broj pretraga za jedan nalaz jeste 3. Jedna pretrega nije jedinstvena samo za jedan nalaz tako da i ona može biti u više nalaza. Iz tog razloga ovo je N:M veza.

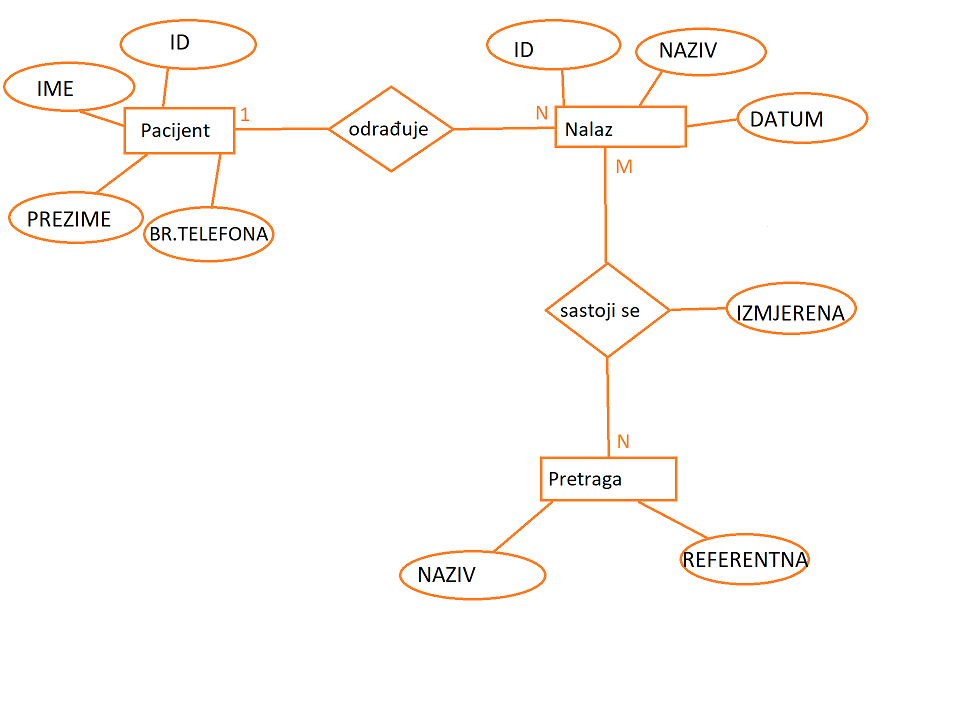
Prikupljamo sve podatke od korisnika te ih spremamo unutar jedne baze podataka. Baza je napravljena u SQLite te kroz cijelu aplikaciju koristimo SQL naredbe da bismo pristupili našoj bazi. Aplikacija nudi oko 15 opcija nad 3 entiteta sa kojim radim. Svaki entitet se može dodati, ažurirati, obrisati i pretražiti. Na ovaj način dajemo fleksibilnost korisniku i mogućnost obrade svih akcija nad svakim entitetom. Cilj je olakšati spremanje svih podataka u nekoj laboratoriji.

Ovakav vid aplikacije pomaže u svim firmama koje se bave obradom nekoliko stotina ili hiljada informacija. Svaka radnja urađena odmah biva spremljena te se akcija poduzima u konkretnoj bazi podataka.

Radimo sa tačnim unijetim podacima koji se već tamo nalaze što omogućava da pri svakoj pretrazi dođu samo oni podaci koji su opstali u bazi ili da dođu sa svoji novim promijenjenim vrijednostima.

# **1.ER MODEL**

Da bismo mogli napraviti bazu prvo nam treba ER model na kojem se vide sve veze te atributi svih entiteta koji će se nalaziti u bazi.

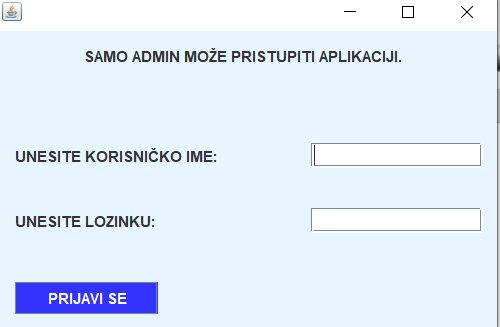


Slika 1 1: ER model

Ovdje vidimo tri entiteta te veze između njih. U situaciji kad imamo 1:N vezu tad ID kao strani ključ ide u onu tabelu koja samo jednom učestvuje u toj vezi. U slučaju moje 1:N veze nalaz učestvuje jednom tako da će u njegovoj tabeli pored njegovih atributa biti i ID pacijenta. U slučaju N:M veze kao što je druga tada se formira skroz nova tabela u kojoj će biti vezani ID obe tabele, te ID veze i još neki atribut ako je neophodno. U slučaju ove aplikacije potrebno je da tabela koja je nastala kao veza ima atribut izmjerena vrijednost. Ako bismo izmjerenu vrijednost pridružili pretrazi značilo bi da je ona tu i u tabeli što bi uništilo smisao. Cilj je da ta izmjerena vrijednost ima veze sa jednim nalazom koji pripada jednom pacijentu . Iz tog razloga postoji artibut IZMJERENA kod veze N:M. Ona simbolizira izmjerenu vrijednost za samo jednog pacijenta.

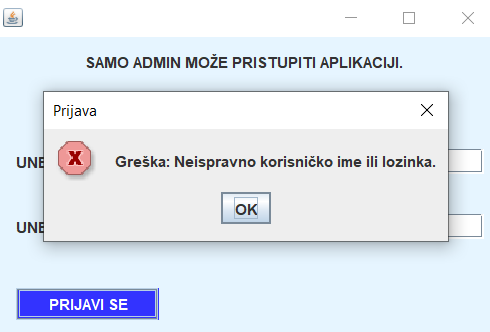
# **2.PRIJAVA ADMINA**

Prvi prozor koji se pojavi kad pokrenemo aplikaciju jeste prozor za prijavu. On od nas traži unos samo korisničkog imena i lozinke. U programu je već tačno definisano ime kao i lozinka. Osoba koja je admin zna te podatke za razliku od ostalih. Sa ovakvim pristupom se omogućuje pristup aplikaciji samo osobi koja za to i jeste zadužena.



Slika 2. 1: Prijava admina

U slučaju da se prilikom unosa korisničkih podataka pojavi greška, kao što je neispravno korisničko ime ili lozinka, u programu se koristi komponenta JOptionPane koja na ekranu ispisuje dijalog s porukom o grešci. Time korisnik odmah dobije vizualnu informaciju o problemu i može preduzeti potrebne korake za ispravak.



Slika 2. 2: Javljanje greške u prijavi

Ukoliko se greška pojavi korisnik zna koji je njegov idući korak i bolje razumije ponašanje našeg programa. Tako da se pri kreiranju aplikacije često koristi komponenta JOptionPane koja signalizira neku vrstu promjene, problema ili prijedloga šta korisnik da uradi

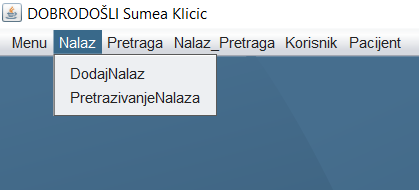
# **3. GLAVNI PROZOR**

Da bismo mogli sve opcije ponuditi korisniku prvo imam glavni prozor gdje imamo smještan meni za svaki entitet



Slika 3. 1: Prikaz glavnog prozora

Na ovaj prozor možemo doći tek nakon sto su korisničko ime i lozinka ispravni. Na samom vrhu prozora imamo moje ime te dobrodošlicu. Ime koje bi se gore nalazilo je ime admina naše aplikacije. Vidimo jednu traku sa više opcija. Svaka opcija ima svoje opcije, a to izgleda ovako:



Slika 3. 2: Prikaz opcija rada nad entitetom

Svaku od opcija možemo izabrati, a ona nudi ono što je i navedeno. Svaka od opcija je novi JInternalFrame Form, dok je glavni prozor JFrame, kao i prozor za prijavu.

# **4.OPCIJE ZA ENTITET – PACIJENT**

Postoje 4 implementirane opcije za rad nad tabelom sa pacijentima

1.Dodavanje novog pacijenta

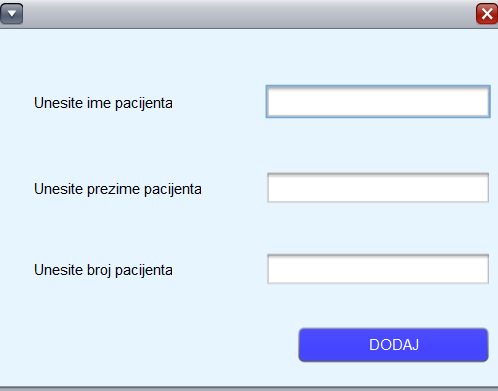
2.Azuriranje nekog podatka od već unijetog pacijenta

3. Pretraga svih nalaza za određenog pacijenta

4.Pretraga pacijenta

# **4.1.Dodavanje novog pacijenta**

Prozor koji se otvori nudi nam mogućnost da unesemo potrebne podatke za pacijenta. Pacijent ima svoju tabelu u bazi i u njoj ima ID, ime, prezime i broj telefona. ID nije nužno da se unese zato što je u tabeli automatski podešen da se povećava.

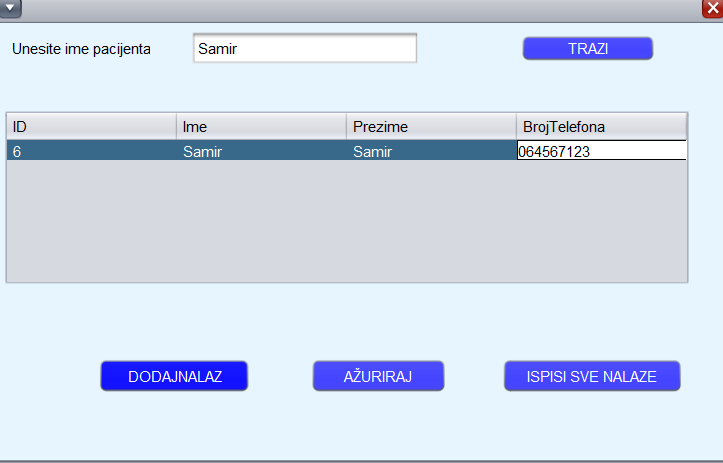


Slika 4. 1: Prozor za unos pacijenta

Imamo 3 komponente Label koje služe za pisanje teksta koji neće biti interaktivan, korisnik ga neće moći kliknuti već samo pročitati. Dalje imamo 3 JTextField-a. Oni su interaktivni i upravo tu korisnik unosi podatke koje želi kasnije vidjeti u bazu. Ispod se nalazi button sa natpisom dodaj. Iza svakog buttona krije se akcija koju on treba odraditi. U ovom slučaju potrebno je da pokupi sve vrijednosti koje je korisnik unio i da ih upiše u bazu. Za upisivanje se poziva metoda iz kontrolera, a u samom GUI takav kod ne pišemo već samu funkcionalnost. Ukoliko neka od tri nužna polja ostanu prazna pojavi se greška da ne možemo imati prazna polja.

# **4.2. Pretraga pacijenta zajedno sa ostalim funkcionalnostima**

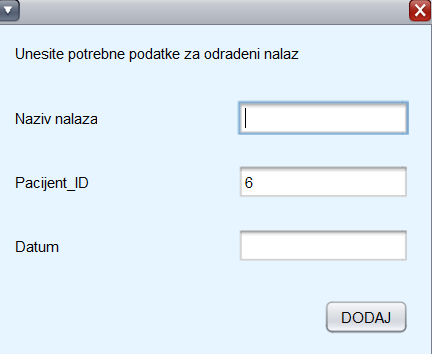
Da bismo odradili bilo koju akciju nad pacijentima prvo je potrebno pronaći tačno pacijenta nad kojim želimo odraditi npr. ažuriranje. Prozor koji imamo je idući:



Slika 4. 2: Pretraživanje pacijenta

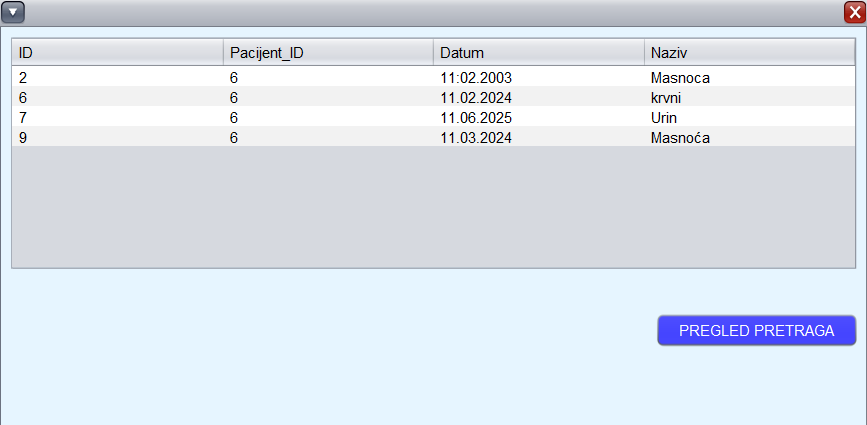
Potrebno je u tekstualno polje unijeti ime neko pacijenta kojeg želimo pretražiti. Nakon toga u tabeli se pojave svi pacijenti sa tim imenom. Možemo kliknuti na jedan od tri buttona i odraditi neku opciju od navedenih.

Ako želimo samo ažurirati dovoljno je da u polje unesemo novu vrijednost i klikom na button ažuriraj nova vrijednost će se spremiti u bazi podataka.



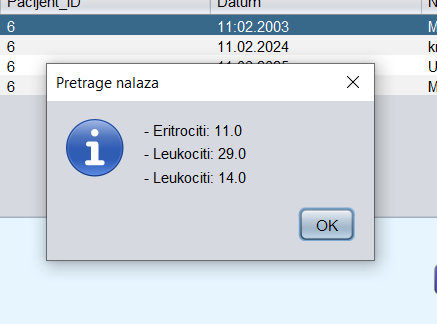
Slika 4. 3: Dodavanje nalaza iz prozora za pretragu pacijenta

Takođe možemo odmah za pacijenta unijeti nalaz klikom na button nalaz pri čemu se odmah polje za ID pacijenta popuni. Na ovaj način ne moramo razmišljati o tome koji ID ima koji pacijent već njegovim izborom u tabeli odmah se popuni njegov ID, a mi sami unosimo ostale potrebne informacije



Slika 4. 4: Pregled svih nalaza za određenog pacijenta

Ako izaberemo button za ispis svih nalaza odmah vidimo datume te nazive odrađenih nalaza. Ako želimo vidjeti koji su pretrage odrađene na nekom nalazu dovoljno je da kliknemo na button pregled pretraga i pojavi se ispisano sve koje su pretrage rađene i koje su izmjerene vrijednosti.

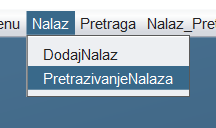


Slika 4. 5: Pregled pretraga vezanih za nalaz

Ovako odmah možemo vidjeti sve podatke za pacijenta uključujući sve nalaze i pretragu povezanu sa svakim nalazom.

# **5.OPCIJE ZA ENTITET - NALAZ**

Nalaz ima svoje atribute a to su: ID, datum, naziv. Takođe prima strani ključ, odnosno ID pacijenta. Nalaz ima više pretraga. Da bismo to mogli pratiti imamo tabelu u kojoj su sve pretrage od svakog odrađenog nalaza. Opcije koje nudi nalaz su sljedeće

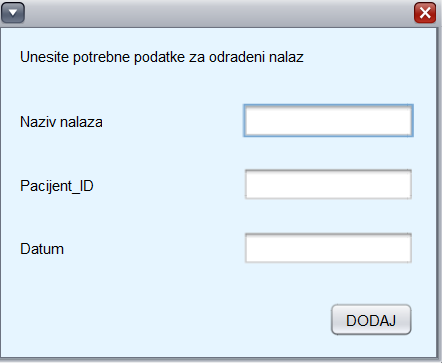


Slika 5. 1: Izbor za rad nad nalazima

Nakon što na traci izaberemo Nalaz ponudi nam se navedenih opcija sa slike. Detaljno ćemo prikazati svaku od opcija.

# **5.1.Dodavanje nalaza**

To podrazumijeva dodavanje novog nalaza u bazu podataka. Potrebno je unijeti podatke za novi nalaz, a prozor za to izgleda ovako:

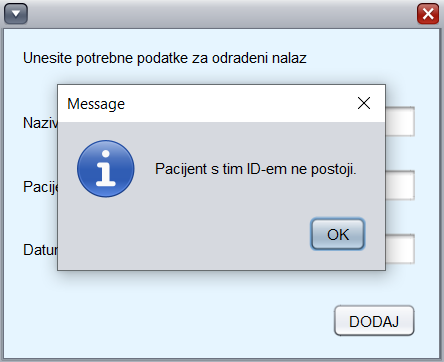


Slika 5. 2: Unos podatak u tabelu nalaz

Iako imamo ID on se automatski podešava tako da to korisnik ne treba unositi.

Potrebno je da unese naziv koji ne mora biti jedinstven zato što je ID primarni ključ, zatim se unosi datum kad je nalaz odrađen.

Pošto je ID pacijenta strani ključ potrebno je da unijeti ID odgovara ID kojeg imamo u tabeli pacijenata. Ako se ne unese odgovarajući ID pojavit će se naredna greška.



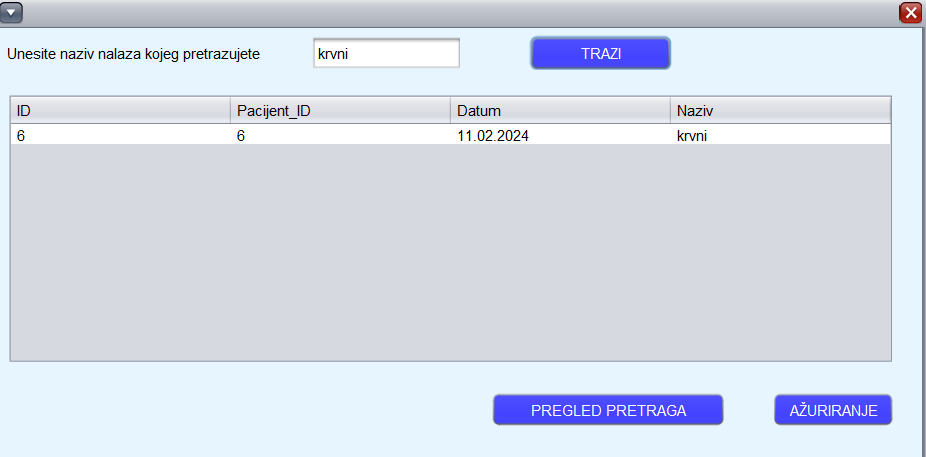
Slika 5. 3: Greška pri unosi nepostojećeg ID pacijenta

Potrebno je da postoji ovakav princip jer ne bi imalo smisla da imamo ID pacijenta u tabeli nalaza, a da taj pacijent zapravo ne postoji. Ispis greške je ovakav da bi korisnik u našem slučaju admin znao gdje je nastupio problem.

Ovaj prozor postoji i može se samostalno izabrati, ali takođe prozor se otvara i kod pretrage pacijenta ako je potrebno dodati novi nalaz

# **5.2.Pretraživanje nalaza**

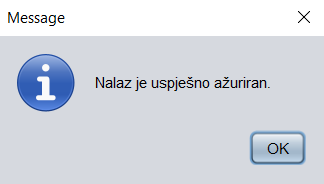
Pošto naziv nalaza nije jedinstven i možemo imati više puta isto ime tada je dobro da se nalaz može pretražiti po imenu. Ideja se da korisnik u tekstualno polje unese ime nalazate klikom na button da se ispišu podaci o svim nalazima koji nose to ime.



Slika 5. 4: Ispis svih nalaza istog imena

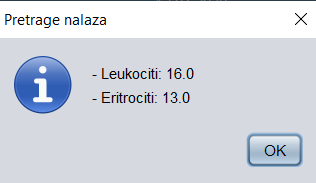
U tabeli se prikažu svi podaci koje imamo i u bazi podataka. Ovako korisnik ima jednostavan pregled nalaza i može jednostavno unositi nova i druga imena da provjeri podatke koje čuvamo.

Možemo kliknuti na neki red te možemo ažurirati neko polje nakon kojeg se pojavi poruka o tome.



Slika 5. 5: Potvrda o tome da je nalaz ažuriran

Za svaki nalaz možemo izabrati i opciju da se ispišu pretrage koje su odrađene za taj nalaz.



Slika 5. 6: Ispis pretraga za odraženi nalaz

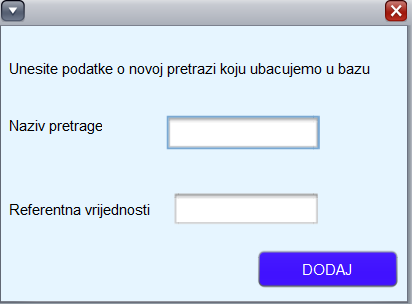
Pošto nalaz može imati do 3 pretrage i onda se ispišu sve koje su dodane i odrađene u tom nalazu.

# **6. OPCIJE ZA ENTITE – PRETRAGA**

Ideja je da se pretrage mogu nezavisno unositi. Recimo da laboratorija dobija nove mašine samim tim mogućnost da se one unesu u bazu. One imaju svoj naziv te referentnu vrijednost. Ta referentna vrijednost predstavlja onu granicu koja se u medicini postavila. Cilj je da kad korisnik odradi pretragu vidi da li je u granicama prihvatljivosti ili odstupa.

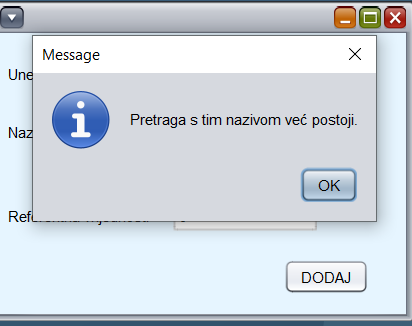
# **6.1. Dodavanje nove pretrage**

Upravo u sitaciji u kojoj sam navela potrebno je unijeti novu pretragu u bazu pa se ta opcija nudi korisniku baze podataka.



Slika 6. 1: Unos nove pretrage

U bazi naziv pretrage je jedinstveni ključ samim tim ne možemo dva puta unijeti isto ime. To je onemogućeno a korisnik vidi i grešku koju je napravio

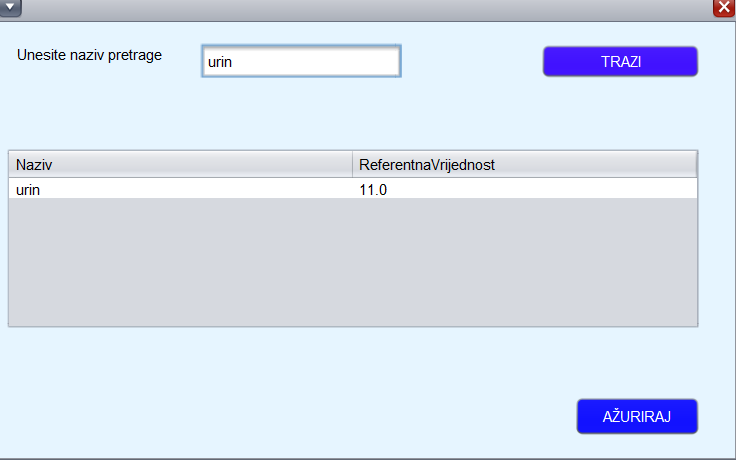


Slika 6. 2: Greška pri unosu dva puta istog naziva

Ovako osiguravamo da nalaz ostane jedinstveni ključ.

# **6.2.Pretraga + Ažuriranje pretrage**

Imamo prozor gdje možemo pretražiti sve pretrage pod odrađenim imenom. Taj prozor izgleda ovako:



Slika 6. 3: Pretraživanje pretrage

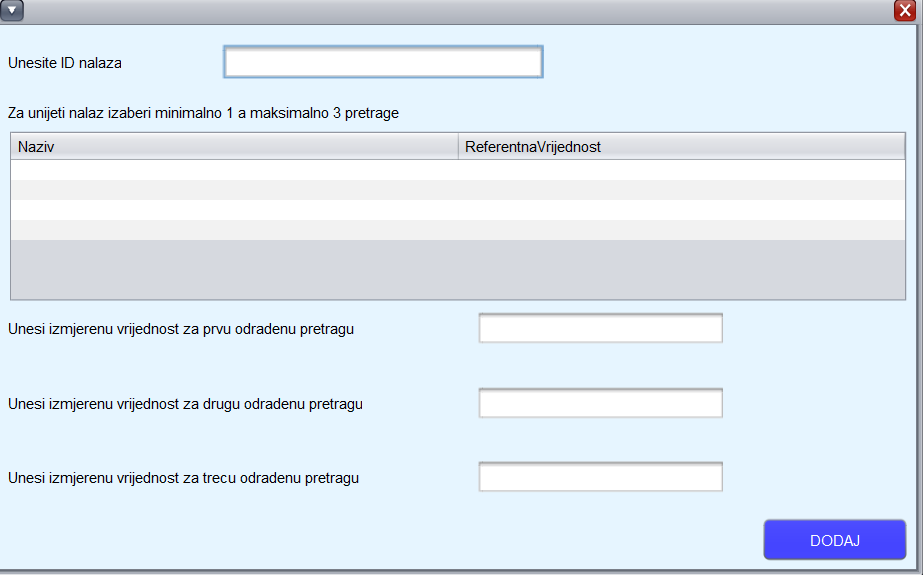
Možemo kliknuti na bilo koji red te u nekom polju promijeniti vrijednost i klikom na Ažuriraj ta vrijednost će da se prebaci i u bazu podataka.

# **7.Nalaz-pretraga N:M veza**

Nalaz i pretraga se u N:M vezi i zato postoji tabela u kojoj se nalazi taj odnos. Potrebno je odraditi vezanje ID nalaza s ID pretrage koji su strani ključevi u ovoj tabeli. Primarni ključ je ID veze. Tako da je potrebno da ID nalaza bude ID kojeg imamo u bazi. Ako unesemo nalaz odmah nakon toga možemo uvezati nalaz sa pretragom i taj ID se automatski doda u prozor

# **7.1. Povezivanje nalaza sa pretragama**

Dva su načina za pojavljivanje ovog prozora. Automatski nakon što korisnik unese novi nalaz u bazu otvori se prozor te mu nudi da poveže nalaz sa pretragama koje već imamo u bazi.



Slika 7. 1: Povezivanje pretraga sa nalazom

U gorenjem dijelu se unosi ID nalaza za kojeg se vezuju unijete pretrage. Nakon toga može se izabrati 1, 2 ili maksimalno 3 pretrage te u zavisnosti od broja izabranih pretraga iz tabele unose se izmjerene vrijednosti za odrađene pretrage. Nakon što se klikne button dodaj unosi se u bazu Nalaz sa svakom od odabranih pretraga i sa izmjerenim vrijednostima.

Ako unesemo prvo nalaz ovaj prozor se sam automatski pojavljuje tako da taj polje sa ID bude popunjeno sa nalazom koji je tad unijet.

Takođe na glavnom prozoru imamo izbornik gdje možemo naknadno izabrati uvezivanje. To može zatrebati u slučaju da smo nekom nalazu zaboravili pridružiti neku pretragu.

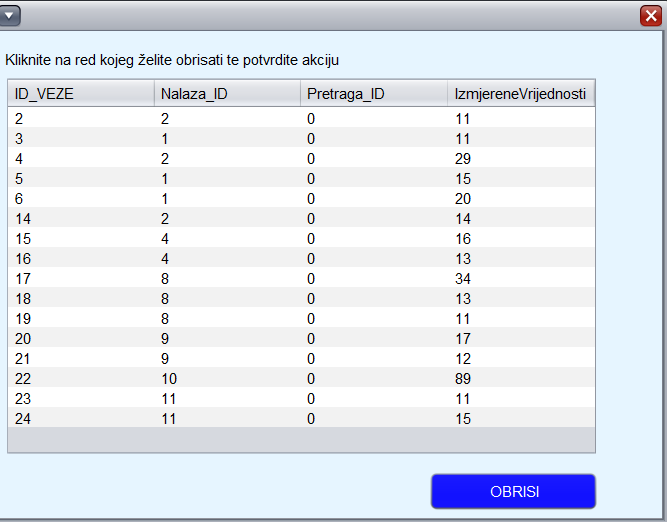
Na ovaj način jednostavno se dodaju podaci te korisnik ima uvid i to što je uradio.

# **7.2.Brisanje veza**

Može se desiti da neku pretragu slučajno dodamo određenom nalazu ili da je rezervisana pa otkazana. U tom slučaju potrebno je imati opciju za brisanje određene veze koju smo formirali.

Izborom brisanja veze pojavljuje nam se prozor u kome se odmah pojave svi redovi iz tabele Nalaz\_Pretraga. Ovako korisnik ima jasan uvid u sve veze koje su već formirane.

Da bi neku od postojećih veza obrisao dovoljno je da samo odabere taj red u tabeli te klikne obriši.



Slika 7. 2: Brisanje veze

Sve nam je ponuđeno dovoljno je samo da znamo šta nam nije potrebno. Klikom na obriši pojavljuje se pitanje jesmo li sigurni te se konačno briše red. Pošto brisanje ovog reda ne zavisi od ostalih tabela nije ptorebno da to brisanje bude vezano za brisanje iz neke druge tabele.

# **ZAKLJUČAK:**

Prikazom svih prozora moje aplikacije može se uočiti da je najbitnija stvar kod aplikacije da se ne gubi integritet i da se smanji redudansa (ponavljanje podataka). Svaka tabela je povezana preko kontrolera te se preko toga omogućuje dodavanje, brisanje, ažuriranje te pretraga. Koriste se jednostavni SQL upiti nad bazom da bismo sve postigli. U GUI dijelu samo pozivamo metode koje smo već kreirali te uz njih dodajem dio koda koji je potreban da bi akcija nad dugmetom bila željena.

Ovakvim pristupom imamo bolje organiziran kod u kojem ćemo kasnije moći lakše naći i riješiti problem ukoliko do njega dođe. Svaki entitet ima svoj posebni paket u kojem se nalazi svih njegovi prozori. Svi kontroleri za rad nad bazom su smješteni u drugi paket, dok imamo u defoltnom paketu informacije vezane za prijavu na aplikaciju.

Svaki prozor ima dovoljan broj elemenata za ispravno funkcionisanje, ali ne previše da zbuni korisnika. Kroz cijelu aplikaciju imamo Label komponentu koja upućuje na akciju koju korisnik treba učiniti. Druga najvažnija komponenta jeste polje za unos teksta jer upravo tako uzimamo podatke od korisnika za daljnji rad nad njima. A komponenta bez koje ništa ne bismo mogli jeste dugme. Klik na dugme signalizira početak analiziranja do tad unijetih ili čekiranih stvari. On omogućava da se prethodni dio akcija pregleda te da se postupi u skladu sa onim što je njegova namjena.

Prozori za pojedine entitete su iste boje, a cilj toga je samo da razlikujemo prozore vezane za entitete i pomoću različitih boja. Smatram da je to efektan način za razumijevanje nad kojim entitetom radimo određenu akciju.

Svaki odrađeni projekat se može unaprijediti i uvijek treba ostaviti prostora za napredak. Pa tako i ovaj projeat ima dosta mjesta za napredak npr uređivanje glavnog prozora i menia da se drugačije kreće. Stavljanje određenog template-a koji će biti praćen kroz sve prozore, te možda još nekoliko tabela gdje ćemo moći spremati radnike laboratorije isl.

Projekat me naučio da se bolje snalazim sa više klasa te da razumijem kako funkcionišu aplikacije koje svakodnevno koristimo. Imala sam priliku da sama kreiram jednu takvu aplikaciju te da na maksimalan način unaprijedim i optimizujem rad.