

**UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET**

**PRIMENA JASPER – ALATA U KREIRANJU
WEB APLIKACIJA ZA PRIKAZ PODATAKA U
RAZLIČITIM FORMATIMA**

-master rad-

Jelena V. Timčenko

Beograd, 2010

SADRŽAJ

SPISAK SLIKA.....	2
SPISAK TABELA I BLOKOVA.....	3
REČNIK TERMINA.....	4
PREDGOVOR	6
1. UVOD	7
2. ALATI ZA PREZENTACIJU I REPREZENTACIJU PODATAKA	8
2.1 PREGLED ALATA ZA GENERISANJE IZVEŠTAJA	8
2.2 JASPERREPORTS	9
KARAKTERISTIKE JASPERREPORTS ALATA	9
Izbor formata.....	9
Izgled izveštaja.....	9
Internacionalizacija	10
Neograničena veličina izveštaja.....	11
Lako ugradljiv u druge aplikacije.....	11
Primena savremenih tehnologija	11
2.3 JASPER REPORTS ALAT I VEZA SA BAZAMA PODATAKA	11
Arhitektura	12
Povezivanje alata sa bazom podataka	13
Kreiranje izveštaja.....	16
3. OPIS KONKRETNE APLIKACIJE I NJENE STRUKTURE	23
4. PRIMENA JASPERREPORTS APLIKACIJE	36
5. ZAKLJUČAK	39
LITERATURA.....	40

SPISAK SLIKA

Slika 2-3-1 Prikaz arhitekture *JasperReport* alata

Slika 2-3-2 Opcija za kreiranje veze sa odgovarajućom bazom podataka

Slika 2-3-3 Prozor aplikacije koji omogućava izbor tipa konekcije sa bazom podataka

Slika 2-3-4 Prozor koji omogućava unos osnovnih podataka vezanih za bazu podataka

Slika 2-3-5 Lista generisanih konekcija na kreirane baze podataka

Slika 2-3-6 Prikaz *Report Wizard* prozora *JasperReports* alata

Slika 2-3-7 Prikaz liste konekcija na baze koje su kreirane u *JasperReports* alatu

Slika 2-3-8 Unos lozinke kojom se omogućava pristup bazi podataka

Slika 2-3-9 Spisak tabela u odabranoj bazi podataka

Slika 2-3-10 Postupak odabira neophodnih polja tabele za upit u bazu podataka

Slika 2-3-11 Odabir polja tabele koja će ući u generisanje izveštaja

Slika 2-3-12 generisanog upita u bazu podataka koji je kreiran na osnovu izabranih polja

Slika 2-3-13 Postupak grupisanja podataka u skladu sa nekim definisanim pravilom

Slika 2-3-14 Primer celina jednog izveštaja

Slika 2-3-15 Paleta karakteristika i elemenata izveštaja

Slika 3-1-1 Primer spiska klijenata

Slika 3-1-2 Primer pregleda podataka klijenta

Slika 3-1-3 Prikaz kreiranog izveštaja za registraciju klijenata

Slika 3-1-4 Primer forme izveštaja za registraciju u **.doc** formatu

Slika 3-1-5 Izveštaj sa podacima o klijentu i njegovom detetu u **.xls** formatu

Slika 4-1 Prikaz prozora aplikacije za inicijalno logovanje na sistem

Slika 4-2 Prikaz prozora koji obaveštava da je unešena pogrešna lozinka

Slika 4-3 Osnovni meni aplikacije

Slika 4-4 Informacija o identitetu trenutno ulogovanog korisnika sistema

SPISAK TABELA I BLOKOVA

Tabela 2-1 Osnovne osobine *JasperReports* alata

Blok 3-1 Osnovni *get*-eri i *set*-eri koji su vezani za odgovarajućeg klijenta

Blok 3-2 Primer generisanja veze između parametara u izveštaju i Java koda klijenta

Blok 3-3 Primer prepoznavanje formata izveštaja

REČNIK TERMINA

CSV (*Comma-Separated Values*) – fajl se koristi za digitalno skladištenje podataka strukturiranih u tabeli sastavljenoj od lista. Svaki podatak u grupi je povezan sa ostalim podacima iz grupe odvojeni zarezima. Svaka linija u CSV fajlu odgovara jednom redu u tabeli. U okviru linije, polja su odvojena zarezima, svako polje pripada jedoju kolini tabele.

DocBook – Je semantički jezik za označavanje i koristi se uglavnom za tehničku dokumentaciju. Inicijalno je stvoren za rad i pisanje tehničke dokumentacije vezane za kompjuterski hardver ali može biti korišćen za bilo koju vrstu dokumentacije.

EJB (*Enterprise Java Beans*) – je deo arhitekture aplikacije koja obuhvata poslovnu logiku aplikacije i putem koje se ta logika kontroliše kroz kompletну aplikaciju.

EJBQL – ili EJB-QL je portabilan jezik baze podataka za definisanje upita za *Enterprise Java Beans* koji može biti pokrenut na bilo kojoj bazi podataka. Koristi se u J2EE aplikacijama. U poređenju sa SQL-om je kompleksniji ali je i manjih mogućnosti.

J2EE – je veoma rasporostranjena platforma za serversko programiranje u Java programskom jeziku. Java2EE platforma (*Java Enterprise Edition*) se razlikuje od Java SE (*Java Standard Edition*) platforme u tome što su dodate biblioteke koje daju veće mogućnosti kreiranja serverskih aplikacija koje su tolerantne na greške, koje je moguće distribuirati, kod kojih su proces rada sa podacima i bazama podataka i sama aplikacija tretirani kao dva procesa i koja je u najvećoj meri zasnovana na većem broju modula koji istovremeno rade na aplikacionom serveru.

JDBC (*Java Database Connectivity*) – je API za Java programski jezik koji definiše kako klijent može pristupiti bazi podataka. Daje metode za generisanje upita i ažuriranje podataka u bazi. Fokus je dat na rad sa relacionim bazama podataka.

LaTeX2e () – Je jezik za označavanje i pripremu sistema za rad sa TeX programom.

LDAP (*Lightweight Directory Access Protocol*) – je protokol korišćen u aplikacijama za postavljanje upita ili izmenu *directory servisa* koji rade preko TCP/IP.

ODT (*The OpenDocument Text file*) – je format dokumenata baziran na XML-u za reprezentovanje elektronske dokumentacije kao što su tabele, grafikoni, prezentacije ili obrada dokumenata.

OLAP (*OnLine Analytical Processing*) – je pristup za nalazenje brzih odgovora na komplikovane multidimenzionalne analitičke upite. OLAP je deo šire kategorije u okviru biznis inteligencije i ekstrahovanja obrazaca iz podataka. Tipične aplikacije koje koriste OLAP su u okviru izveštavanja o prodaji, prognoza, finansijskih izveštaja i sličnih oblasti.

POJO (*Plain Old Java Object*) – Ime označava da je objekat koji se koristi običan Java objekat a ne specijalni a posebno ne EJB objekat.

RDBMS (*Relational DataBase Management System*) – je sistem za upravljanjem dazom podataka koja je bazirana relacionom modelu. Većina danas korišćenih baza podataka, što komercijalnih što *open-source*, se zasniva na relacionom modelu. Ukratko označava sistem za upravljanjem bazom podataka gde su podaci baze skladišteni u vidu tabela a veza podataka u tabeli je takođe čuvana u formi tabele.

RTF (*Rich Text Format*) – format dokumenta razvijen od strane Microsoft kompanije za razmenu podataka medju različitim platformama.

SSO (*Single Sign-On*) – je sistem zaštite pristupa većeg broja srodnih ali nezavisnih softverskih sistema. Sa ovim načinom pristupa korisnik se loguje na sistem – aplikaciju samo jednom, čime nadalje dobija pristup svim sistemima medjusobno povezanim bez potrebe da se loguje na svaki zasebno. *Single Sign-Off* je obrnuti pristup gde jedna *logout* akcija prekida pristupe svim vezanim softverskim sistemima jedne aplikacije.

UTF-8 (*8-bit UCS/Unicode Transformation Format*) – je Unicode kodiranje sa promenljivom dužinom karaktera na osnovu kojeg je moguće predstaviti bilo koji karakter u Unicode standardu. Postao je standard za aplikacije kao što su E-mail, Web strane i drugo.

PREDGOVOR

U ovom radu je predstavljen alat *JasperReports*. Razvijen je u Java objektno orijentisanom programskom jeziku. *JasperReports* je fleksibilan i moćan, *open source* alat za generisanje izveštaja, koji omogućava prikaz izveštaja na ekranu korisnika, šalje izveštaj na štampu ili vrši eksportovanje podataka u neki od formata PDF, HTML, XLS, RTF, ODT, CSV, TXT i XML.

Razlog zbog kojeg sam se opredelila da pišem rad na ovu temu, jeste upravo to što se u današnje vreme ne može zamisliti dobra prezentacija i aplikacija, a da pritom ne sadrži neku vrstu izveštaja i prikaza podataka u nekom od opšteprihvaćenih formata

Rad se sastoji iz šest celina – poglavlja.

U prvom poglavlju je predstavljen prikaz opštih osobina alata za formiranje izveštaja koji se mogu koristiti u izradi aplikacija. Biće prikazan rad i komunikacija alata sa bazama podataka i način korišćenja podataka iz baza za generisanje izveštaja.

U drugom poglavlju je dat opis konkretnog alata za prezentovanje podataka – *JasperReports*, grafički interfejs koji ovaj alat koristi - *iReport*, kao i veza *JasperReports* i MySQL baze podataka prilikom generisanja izveštaja.

U trećem poglavlju sledi opis konkretnе aplikacije koja koristi *JasperReports* alat, prolazak kroz strukturu aplikacije i njenih delova, pri čemu je prikazana upotreba alata za generisanje izveštaja.

U četvrtom poglavlju su prikazani delovi koda i prikaz grafičkog korisničkog interfejsa aplikacije u određenim trenucima tokom rada i primene *JasperReports* alata (*screenshots*) kojima je predstavljena primena *JasperReports* alata za potrebe generisanja izveštaja u .doc, .pdf. i .xls formatima.

Peto poglavlje obuhvata zaključak nakon čega sledi spisak literature uz pomoć kojje je ovaj rad sačinjen.

1. UVOD

Jedna od namena softverske industrije je i da automatizuje rutinske aktivnosti i da sumira podatke u vidu specifičnih izveštaja. To je razlog što veliki broj korisnika shvata softver kao sredstvo pomoću kojeg će veliki broj svojih podataka koncentrisati na jedno mesto i prikazati na lep način u vidu grafikona, tabela i slično.

Na tržištu postoji veliki broj alata za generisanje izveštaja u različitim formatima, u zavisnosti od potreba konkretnе aplikacije i korisnika. Najčešći formati za exportovanje podataka su: PDF, HTML, Microsoft Excel, RTF, ODT, CSV, XML kao i grafički izveštaji u vidu Chart-ova, Pie, 3D Pie, XY i Bar prikaza generisanih uz pomoć *JfreeChart* biblioteke, ili na neki drugi način [1].

Neki od poznatijih alata za kreiranje izveštaja su: Jasper Reports, ReportMill, RAQSoft, Eclipse BIRT Project i Agata Report.

Svaki od ovih alata ima neke svoje osobine koje će kasnije biti izložene. Neki od njih su namenjeni za Java web aplikacije a neki za .Net aplikacije. Komercijalna rešenja kao sto je Crystal Reports moguće je koristiti kako u Java tako i u .Net aplikacijama, što je svakako njegova prednost.

U ovoj tezi su detaljno opisane karakteristike *JasperReports* alata, kao i njegovo korišćenje na konkretnom primeru. *JasperReports* omogućava kreiranje dinamičkih izveštaja uključujući komunikaciju i razmenu podataka putem JDBC (*Java Database Connectivity*). Osim prikaza u formatima kao što su : HTML, Text, PDF, MS Excel, RTF, ODT, *Comma-Separated Values*, XML ili slika, *JasperReports* alat omogućava prikaz i u vidu grafikona (Graph, Pie, Chart, 3D Pie, XY line ...).

Izgled izveštaja zavisi isključivo od samog korisnika jer je fleksibilnost alata velika.

2. ALATI ZA PREZENTACIJU I REPREZENTACIJU PODATAKA

2.1 PREGLED ALATA ZA GENERISANJE IZVEŠTAJA

Neki od najčešćih alata korišćenih u kreiranju izveštaja su [2] :

BIRT (*Business Intelligence Reporting Tool*) – je alat čiji je rad zasnovan na Eclipse okruženju. To je *open source* sistem za generisanje izveštaja za Web aplikacije i posebno za aplikacije zasnovane na Java i J2EE tehnologiji. BIRT ima dve važne komponente: Interfejs za kreiranje izveštaja u Eclipse okruženju i *runtime* komponentu koja može biti postavljena na odgovarajući server na kojem se aplikacija pokreće. Ovaj alat nudi *chart engine*, odnosno generator grafikona koji se može integrisati u okviru aplikacije [3].

jCharts – je alat koji je u potpunosti razvijen u Java programskom jeziku i omogućava generisanje izlaznih informacija u različitim formatima. Od samog početka, na razvoju ovog alata su najviše radili brojni volonteri kojima je bio potreban alat za prikaz grafikona putem Servlet, JSP i Swing aplikacija.

DataVision – je *open source* alat za generisanje izveštaja. Izveštaji mogu biti kreirani prevlačenjem mišem iz menija u oblast rada (*drag-and-drop*) u okviru grafičkog interfejsa. Generisanje i prikaz izveštaja se obavlja pokretanjem aplikacije, a izveštaji mogu biti generisani u formatima kao što su HTML, XML, PDF, LaTeX2e, DocBook. Dokumenti generisani putem LaTex2e ili DocBook aplikacija mogu biti pretvoreni u PDF, Text, HTML, PostScript i druge.

FreeReportBuilder – je softver za brzo kreiranje jednostavnih izveštaja. Neke od osobina ovog alata su:

- **FreeReportBuilder** može da radi sa bilo kojom bazom koja ima JDBC driver.
- Uključuje **FreeQuerzBuilder** za kreiranje SQL upita bez eksplicitnog pisanja upita.
- Kreira grafikone iz izveštaja jer sadrži **JfreeChart**.
- JDBC administracija, *drag-and-drop* pristup komponentama izveštaja
- Dodavanje slika u izveštaje, postojanje gornjeg i donjeg zaglavlja (*Header* i *Footer*) segmenata
- Kreiranje ugnježđenih izveštaja

2.2 JASPERREPORTS

JasperReports alat je, za razliku nekih drugih alata, zasnovan na Java jeziku i nema svoju sopstvenu sintaksu, što ga čini jednostavnijim za rad i korišćenje u Java Web aplikacijama.

KARAKTERISTIKE JASPERREPORTS ALATA

IZBOR FORMATA

JasperReports je fleksibilan i moćan, *open source* alat za generisanje izveštaja, koji omogućava prikaz izveštaja na ekranu korisnika, šalje izveštaj na štampu ili vrši eksportovanje podataka u neki od formata PDF, HTML, XLS, RTF, ODT, CSV, TXT i XML.

Jedan od važnijih ciljeva ovog alata je da pomogne i isporuči stranično orijentisane (*page oriented*) izveštaje spremne za rad i štampu.

IZGLED IZVEŠTAJA

Izveštaji koji se kreiraju su sačinjeni od skupa nezavisno definisanih i formatiranih sekcija:

- *Title* – naslov izveštaja
- *Page Header* – zaglavlje izveštaja koje će se ponavljati na početku svake strane
- *Column Header* – zaglavlje kolone izveštaja
- *Group Header* – zaglavlje grupe podataka u izveštaju
- *Detail* – niz podataka i detalji samog izveštaja
- *Group Footer* – donje zaglavlje grupe podataka u izveštaju
- *Column Footer* – donje zaglavlje kolone izveštaja
- *Page Footer* - donje zaglavlje strane izveštaja ponavlja se na dnu svake strane
- *Summary* – Rezime podataka

Izveštaj može sadržati bilo koji broj ugnježđenih grupa podataka. Pri svakom prelazu na sledeću grupu, *JasperReports* automatski dodaje grupna zaglavља (*Group Footer* i *Group Header*) za sledeću grupu i tako omogućava viši nivo grupisanja podataka.

Sekcije mogu sadržati elemente kao što su linije, pravougaonici, slike, statički tekstualni podaci, dinamički tekstualni podaci, polja, izvorne i obrađene podatke. Svi elementi izveštaja mogu precizno biti pozicionirani, a njihova veličina može biti definisana čime se postiže kvalitetan i prgledan izgled štampanog izveštaja. Osim ovoga mogu se kreirati i višeredni izveštaji.

JasperReports sa lakoćom dostavlja i najkompleksnije izveštaje jer podržava podizveštaje. Neograničen broj izveštaja može biti upotrebljen kao podizveštaj i bilo koji broj podizveštaja može biti ugnježđen unutar jednog izveštaja. S druge strane, podizveštaji mogu sadržati druge podizveštaje do kojeg god želimo nivoa ugnježđivanja na taj način možemo kreirati proizvoljno kompleksne izveštaje.

Veliki broj korisnika, zbog lakšeg razumevanja, daje prednost grafički prikazanim podacima. *JasperReports* omogućava grafički prikaz podataka i podržava veliki broj tipova grafikona.

Izveštaji mogu sadržati vezu ka drugoj Web stranici, nekom fajlu ili nekom drugom sadržaju u formi tzv. hiperlinkova (*hyperlinks*), koji omogućavaju korisnicima kreiranje OLAP (*OnLine Analytical Processing*) stilizovanih i personalizovanih izveštaja. Time se korisnicima omogućava dinamičko kretanje kroz generisani izveštaj i njegove određene delove u skladu sa određenim potrebama i željama. Ovim se može postići i generisanje tabela sa odvojenim sekcijama, pri čemu je moguće jednim klikom doći do željenog dela ili sekcije izveštaja.

INTERNACIONALIZACIJA

JasperReports je potpuno internacionalno zasnovan zahvaljujući povezanosti sa Java platformom [2].

Na osnovu *Unicode (UTF-8)* su u okviru *JasperReports* podržani svi svetski jezici, a takođe su podržani drugi jezički kodovi kao što je **ISO-8859-1** (Latin 1) za Zapadno Evropske jezike. Osim toga, omogućena je podrška prevoda delova aplikacije (*lokalizacije*) kroz izvorne fajlove čime izveštaj može biti prikazan na različitim jezicima i formatiran za različite dijalekte.

NEOGRANIČENA VELIČINA IZVEŠTAJA

Veličina generisanih izveštaja je ograničena jedino dostpnim memorijskim prostorom na hard disku, što omogućava kreiranje veoma složenih i zahtevnih izveštaja uz obradu velikog broja podataka.

LAKO UGRADLJIV U DRUGE APLIKACIJE

S obzirom na to da je *JasperReports* alat specifičan vid Java biblioteke, izveštaji generisani pomoću ovog alata mogu jednostavno da se ugrađuju u druge Java aplikacije.

Na taj način, *JasperReports* može da predstavlja deo neke druge Java aplikacije. *JasperReports* ima ugrađen i Java Swing modul za prikaz izveštaja i njihovo kreiranje.

PRIMENA SAVREMENIH TEHNOLOGIJA

Razvoj *JasperReports* alata je započet tokom 2001. godine kao *open source* projekat i kao odgovor na sve veću potrebu mnogobrojnih preduzeća u svetu za alatom koji može da obezbedi specifično generisanje izveštaja. Brzi razvitak i popularnost *JasperReports* projekta omogućen je sve većim brojem zainteresovanih strana, kao i pojačanom aktivnošću korisnika foruma posvećenih alatima za kreiranje izveštaja.

JasperReports alat je zasnovan na savremenim Web standardima: na programskom jeziku Java i jeziku za opis podataka, XML.

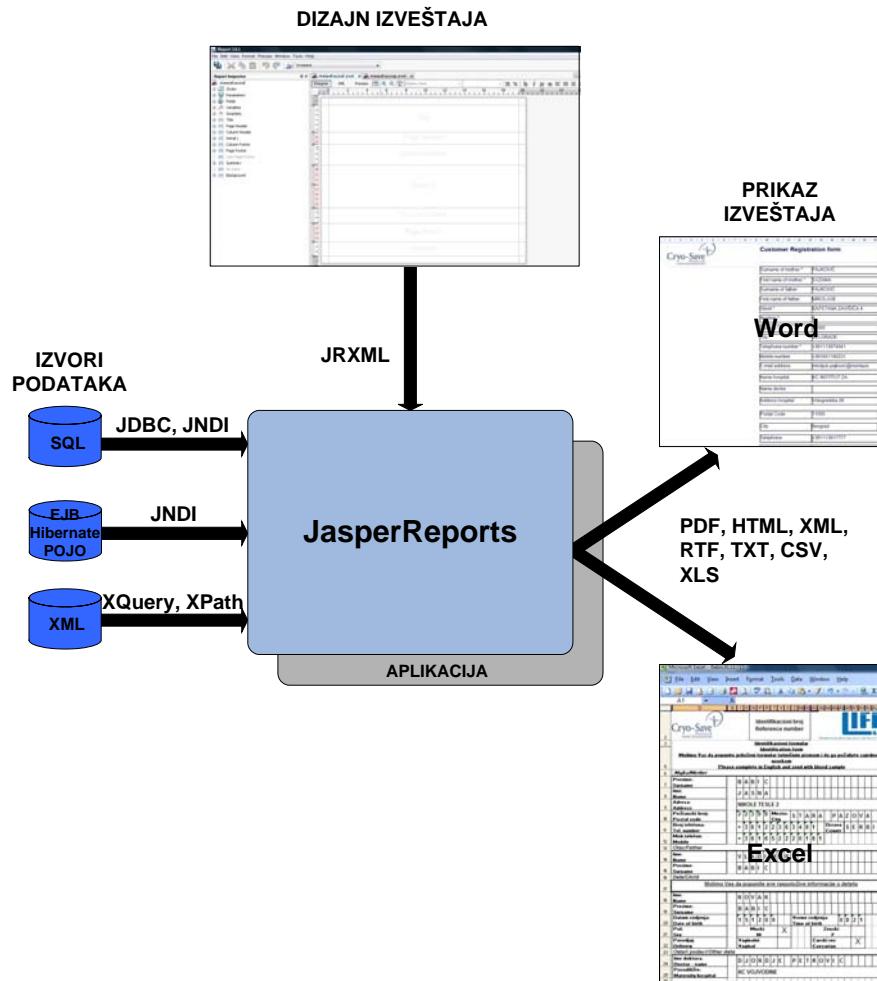
Definicije u izveštajima su materijalizovane u vidu posebnog XML formata, JRXML (*JasperReports JRXML Report Definition*).

Podaci, kojima se popunjava izveštaj, mogu biti unešeni bilo kao ulazni parametri od strane osnovne aplikacije ili iz izvora podataka definisanih u *JasperReports*.

2.3 JASPER REPORTS ALAT I VEZA SA BAZAMA PODATAKA

JasperReports uključuje prikaz podataka iz različitih izvora kao što su JDBC, Relational Databases (RDBMS), JavaBeans (EJB, Hibernate), Plain Old Java Objects (POJO), i XML izvor podataka.

Arhitektura *JasperReports* alata data je na sledećoj slici.



Slika 2-3-1 Prikaz arhitekture *JasperReport* alata

ARHITEKTURA

Kao što je prikazano na slici 2-3-1, *JasperReports* arhitektura je zasnovana na deklarativnim XML fajlovima koji po konvenciji imaju ekstenziju JRXML, a koja sadrži karakteristike izveštaja. Postoji mnogo alata za dizajniranje izveštaja i generisanje JRXML fajlova na lak način. Dve često korišćena alata su: *iReport* i *JasperAssistant*. Izveštaj je popunjeno podacima dobijenim iz baze podataka, XML fajlova, Java kolekcija, CSV fajlova i drugih izvora [1].

JasperReports komunicira sa tim izvorima podataka, a osim toga omogućava povezivanje velikog broja izvora podataka u jednu celinu i vrši manipulaciju podacima koji su dobijeni iz mnoštva kombinacija tog povezivanja.

Ova komunikacija se ostvaruje putem JDBC, JNDI, Xquery, EJBQL, Hibernate ili Oracle PL/SQL. Takodje, korisnik može definisati sopstveni izvor podataka kroz klase i preneti ih direktno *jasper engine* modulu. Nakon definisanja izgleda izveštaja i podataka koje će taj izveštaj koristiti u .jrxml fajlu, *jasper engine* dalje funkcioniše samostalno. Kompajliranjem se kreira fajl koji je ispunjen rezultatima dobijenim iz definisanog izvora podataka i zatim generiše izveštaj u izabranom formatu (PDF, Excel, HTML, XML, RTF, TXT i drugim.)

U tabeli 2-3 je prikazan pregled osnovnih osobina *JasperReport* alata, među kojima su minimalni zahtevi vezani za sistem na kojem će se obaviti instalacija, kompatibilnosti, formata u kojima se mogu izveštaji generisati kao i integracija sa drugim tehnologijama.

Tabela 2-3 Osnovne osobine Jasper Report alata

Licenca	GPL 2
Tehnologija	J2EE, iReport, Liferay
Nivo jednostavnosti upotrebe	Visok – Najbolji korisnički interfejs u grupi alata za generisanje izveštaja
Formati za prikaz izveštaja	PDF, HTML, XLS, CSV, RTF, TXT, XML, Flash
Dizajn izveštaja	Visual iReport dizajner
Pretraga izveštaja	Preko internet sajta, dostavljanjem preko e-maila ili pretragom zakazanih izveštaja.
Zakazivanje redovnih akcija aplikacije (<i>Schedulability</i>)	Visok nivo mogućnosti zakazivanja željenih akcija vremenski određenih kroz korisnički interfejs.
Instalacija/ Konfiguracija/Zahtevani prostor na racunaru	Laka / Laka / 500MB
Integracija	SOAP API, bilo koji J2EE server, bilo koji JDBC drajver

POVEZIVANJE ALATA SA BAZOM PODATAKA

Da bi se *JasperReports* povezao sa odgovarajućom bazom podataka i generisao izveštaj koji sadrži zahtevane podatke, potrebno je kreirati konekciju sa već napravljenom bazom podataka, što je najlakše uraditi kroz već ugrađeni *Report Wizard* [4],[5].

Da bi bilo koji izveštaj mogao da bude kreiran, potrebno je prethodno definisati konekciju sa odgovarajućom bazom podataka iz koje ćemo čitati podatke i popunjavati izveštaj.

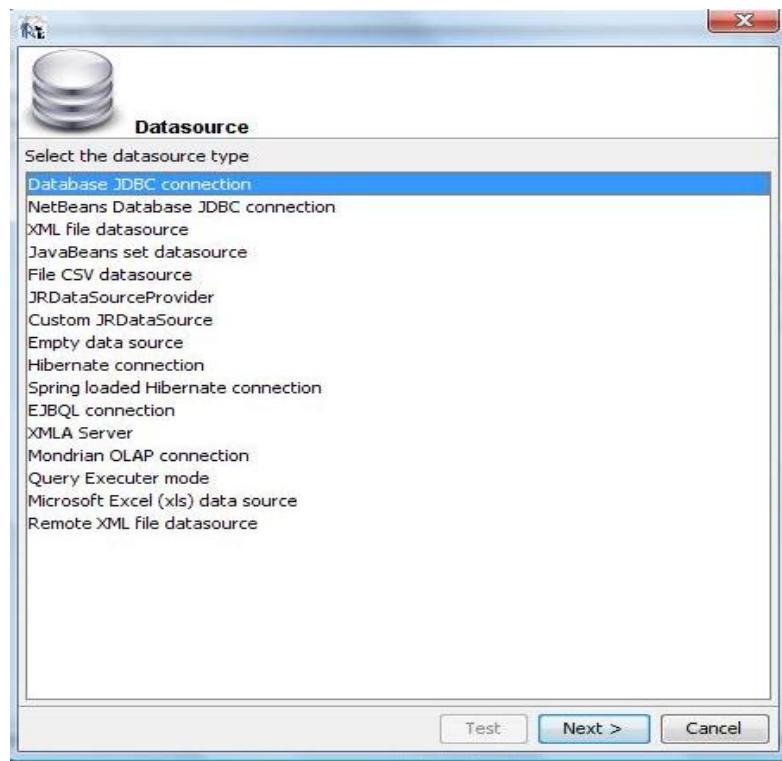
Na slikama 2-3-2, 2-3-3 i 2-3-4 je prikazan postupak kreiranja konekcije sa bazom podataka koja je prethodno gnenerisana u okviru sistemu.

Na naslovnoj strani *JasperReports* alata je potrebno odabrati opciju (*DBConnection*), a zatim odabrati opciju kreiranja nove konekcije (Slika 2-3-2).



Slika 2-3-2 Opcija za kreiranje veze sa odgovarajućom bazom podataka

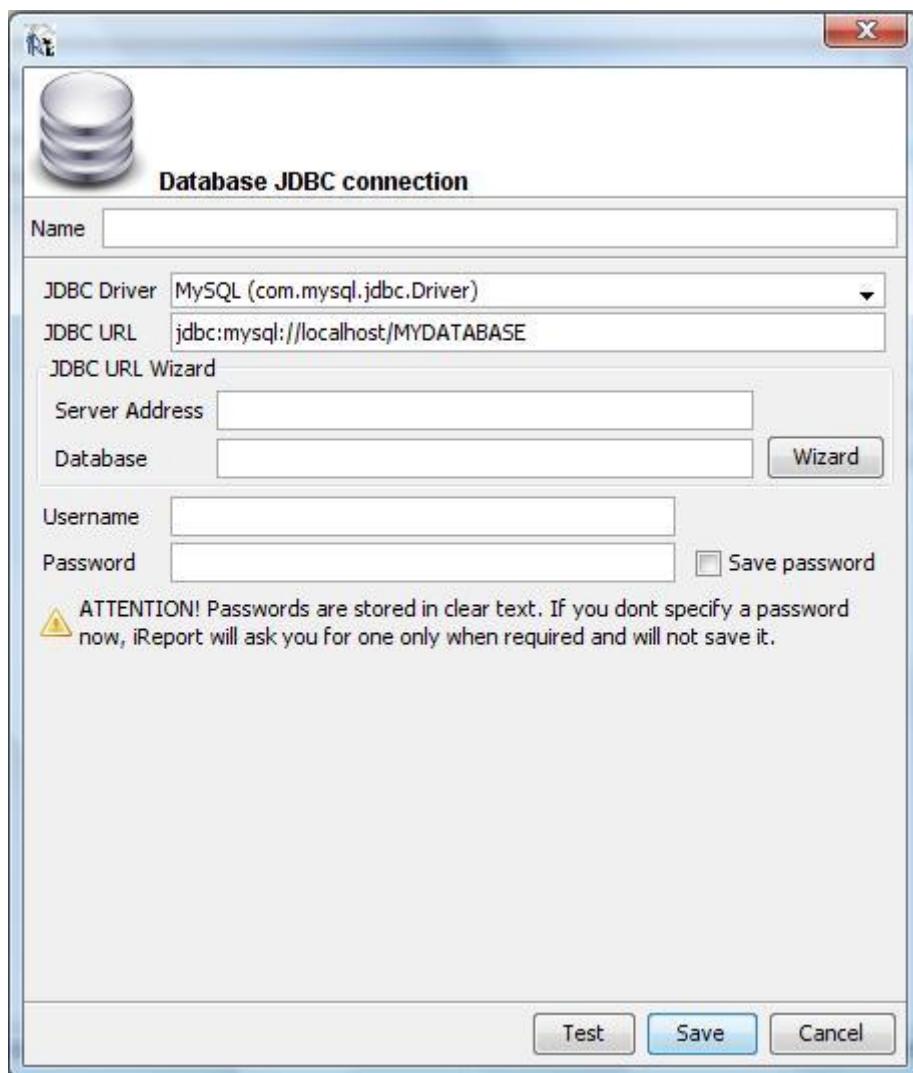
Nakon toga se otvara prozor aplikacije koji omogućava odabir tipa konekcije sa bazom podataka za koju se kreira konekcija:



Slika 2-3-3 Prozor aplikacije koji omogućava izbor tipa konekcije sa bazom podataka

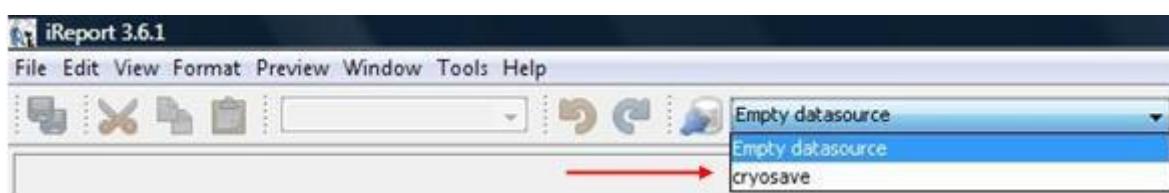
Uobičajeno je da izbor bude JDBC tip konekcije, ali je moguće odabrati i neke od tipova iz liste ukoliko je baza kreirana na drugi način (Slika 2-3-3).

Slika 2-3-4 prikazuje prozor za unos osnovnih podataka baze, kao što su naziv konekcije, putanja do baze na sistemu na na koju ćemo se povezivati (*JDBC URL*), korisničko ime i lozinka za pristup bazi.



Slika 2-3-4 Prozor koji omogućava unos osnovnih podataka vezanih za bazu podataka

Nakon osnovnih podešavanja, izborom opcije *Test* se proverava da li je sistem uspeo da se poveže sa bazom podataka. Po dobijanju potvrđnog odgovora, ova podešavanja se čuvaju za kasnija korišenja (Slika 2-3-5). Rezultat ove procedure je da se u istom trenutku, u padajućoj listi baze podataka, nalazi i novouspostavljena konekcija (a to je, na primer, *Cryosave*).



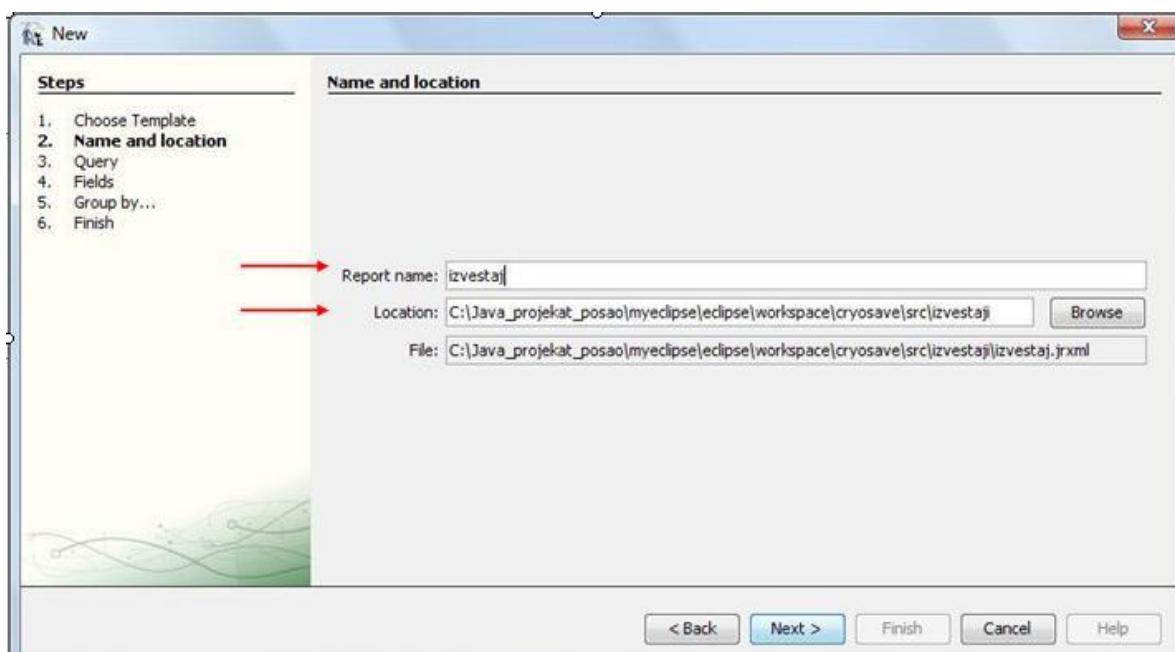
Slika 2-3-5 Lista generisanih konekcija na kreirane baze podataka

KREIRANJE IZVEŠTAJA

Nakon što je uspešno kreirana konekciju na željenu bazu podatka, sledeći korak je kreiranje konkretnog izveštaja. Najlakši način za to je korišćenje *Report Wizard* dela aplikacije koji kroz nekoliko osnovnih koraka omogućava generisanje osnovnog izgleda izveštaja.

Na slici 2-3-6 je prikazan interfejs *Report Wizard* dela aplikacije. Prvi korak je izbor naziva izveštaja i lokacije na hard disku gde će izveštaj biti sačuvan na računaru.

Važno je da ta lokacija odgovara lokaciji na kojoj je i aplikacija koja ga poziva, dakle odgovarajuće „radno okruženje“, *workspace*.

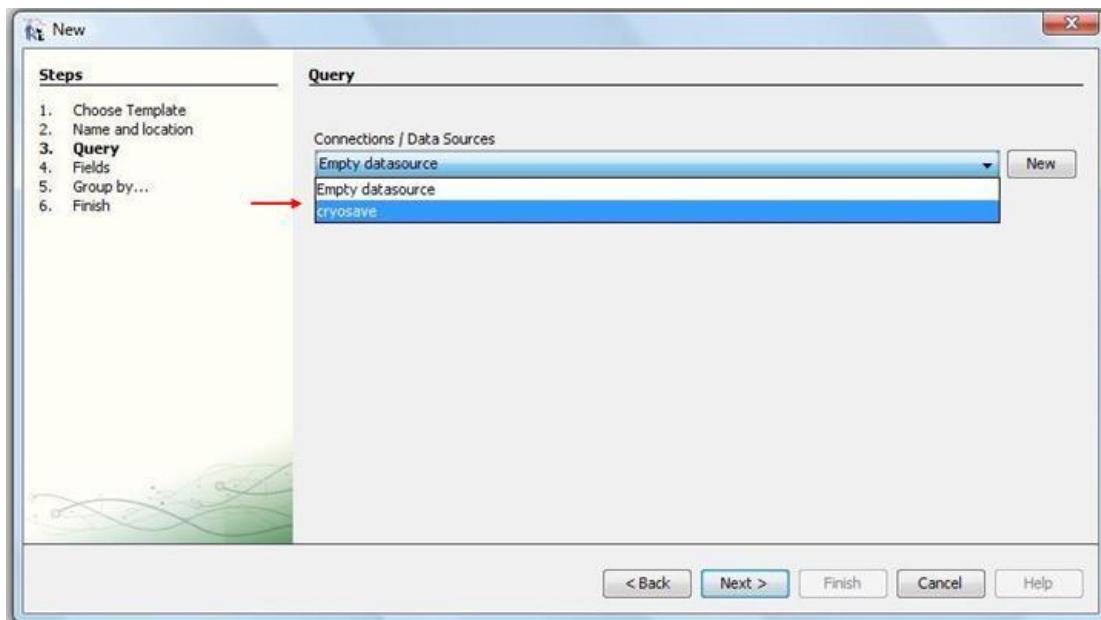


Slika 2-3-6 Prikaz Report Wizard prozora JasperReports alata

Nakon izbora naziva izveštaja, sledi odabir baze podataka iz koje će se preuzimati podaci za generisanje izveštaja.

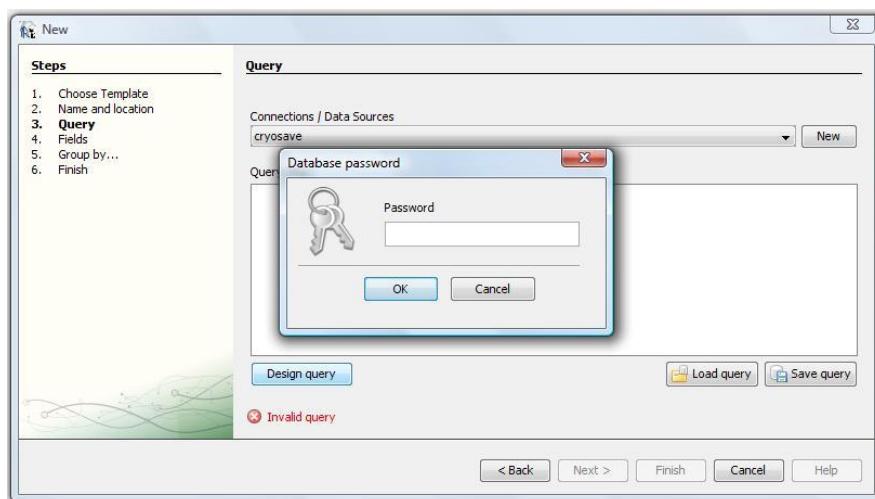
Potrebno je da baza podataka prethodno bude kreirana, popunjena i da postoji veza (*connection*) na nju u okviru *JasperReports* aplikacije. Način da se to postigne je opisan u prethodno datom tekstu.

Slika 2-3-7 daje prikaz prozora u okviru *Report Wizard* dela aplikacije kojim je omogućen opisani postupak izbora baze.



Slika 2-3-7 Prikaz liste konekcija na baze koje su kreirane u *JasperReports* alatu

Nakon odabira baze iz koje će se čitati podaci, u zavisnosti od toga da li je upisana lozinka za odabranu bazu podataka ili ne, pojaviće se prozor za njen unos (Slika 2-3-8).



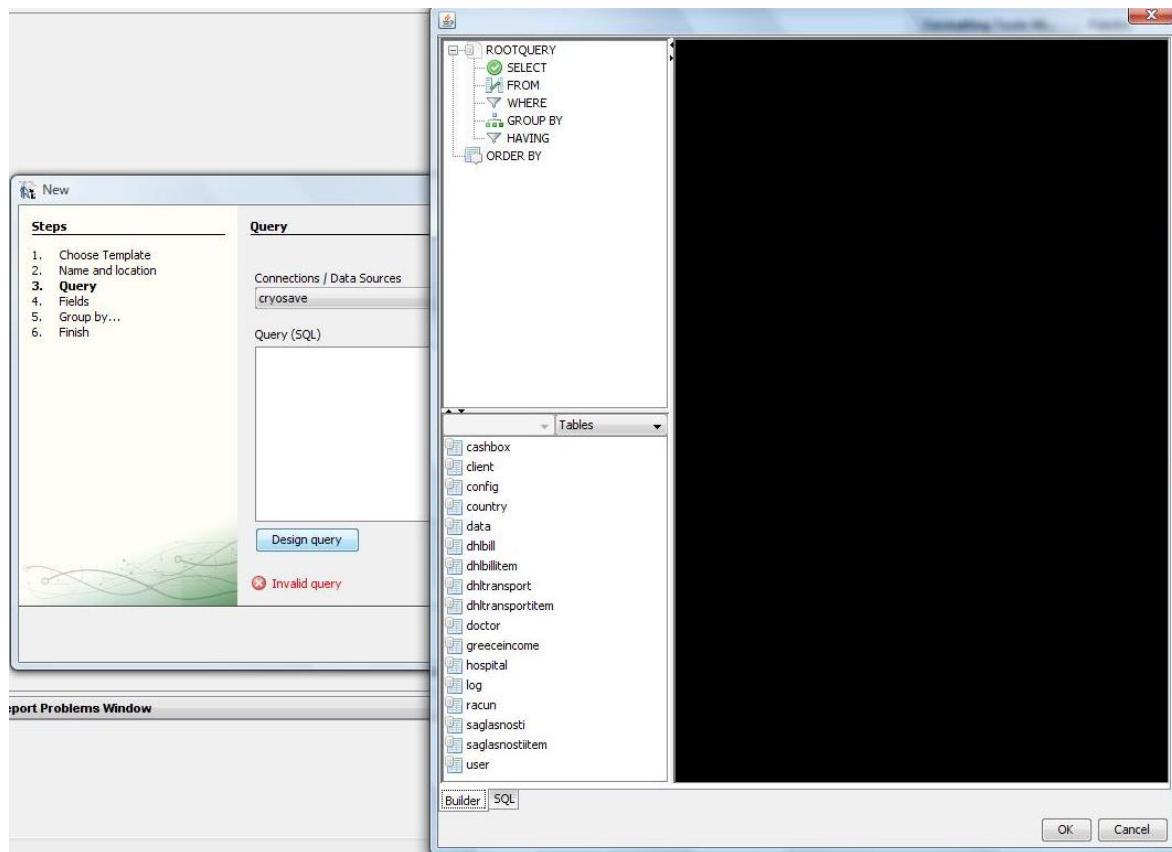
Slika 2-3-8 Unos lozinke kojom se omogućava pristup bazi podataka

Time je omogućeno uspostavljanje konekcije i prikazivanje tabela koje postoje u bazi. Odmah nakon unosa lozinke, generisaće se prozor aplikacije kojim će biti ponuđen i kreiran upit bazi radi selekcije željenih tabela koje u njoj postoje a na osnovu kojih će biti generisan izveštaj.

Ukoliko postoji neki prethodno generisani upit u bazu podataka, moguće ga je izabrati izborom opcije *Load Query*. Novi upit se može generisati izborom opcije *Design Query*,

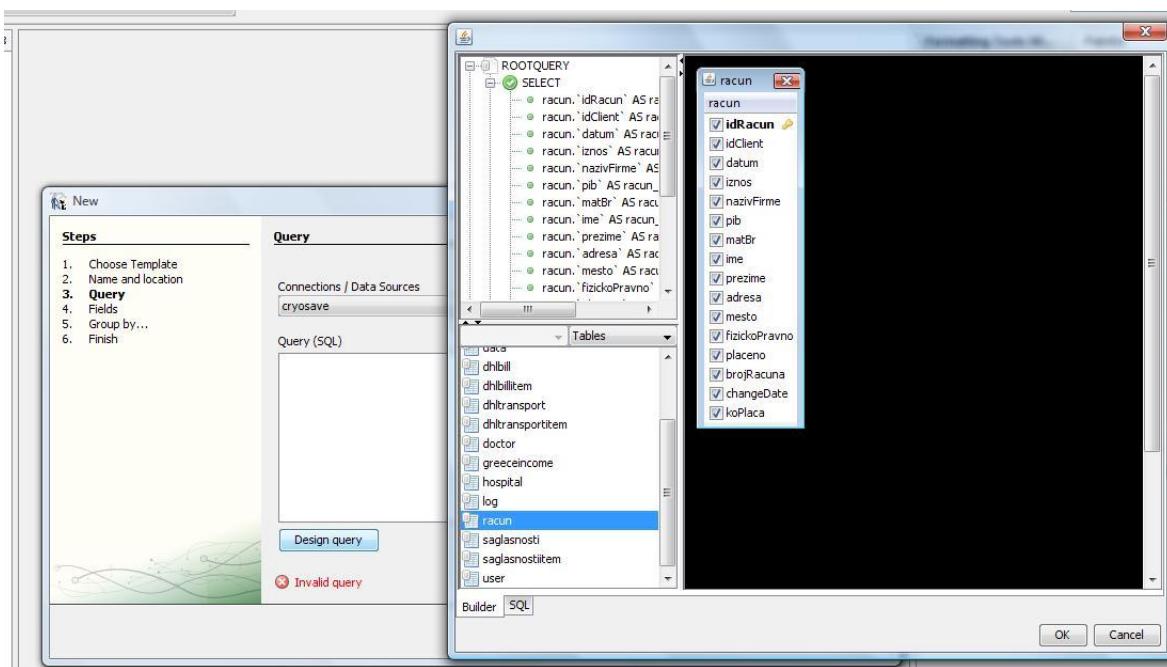
čime se omogućava da u posebnom prozoru aplikacije budu izlistane postojeće tabele baze i sva polja tih tabela. Iz generisane liste je moguće izabrati ona polja koja je potrebno da učestvuju u generisanju izveštaja.

Na levoj strani prozora su prikazane sve tabele koje formiraju odabranu bazu podataka. Uz pomoć miša je moguće (drag-and-drop) prevući tabele na desnu stranu prozora (Slika 2-3-9).



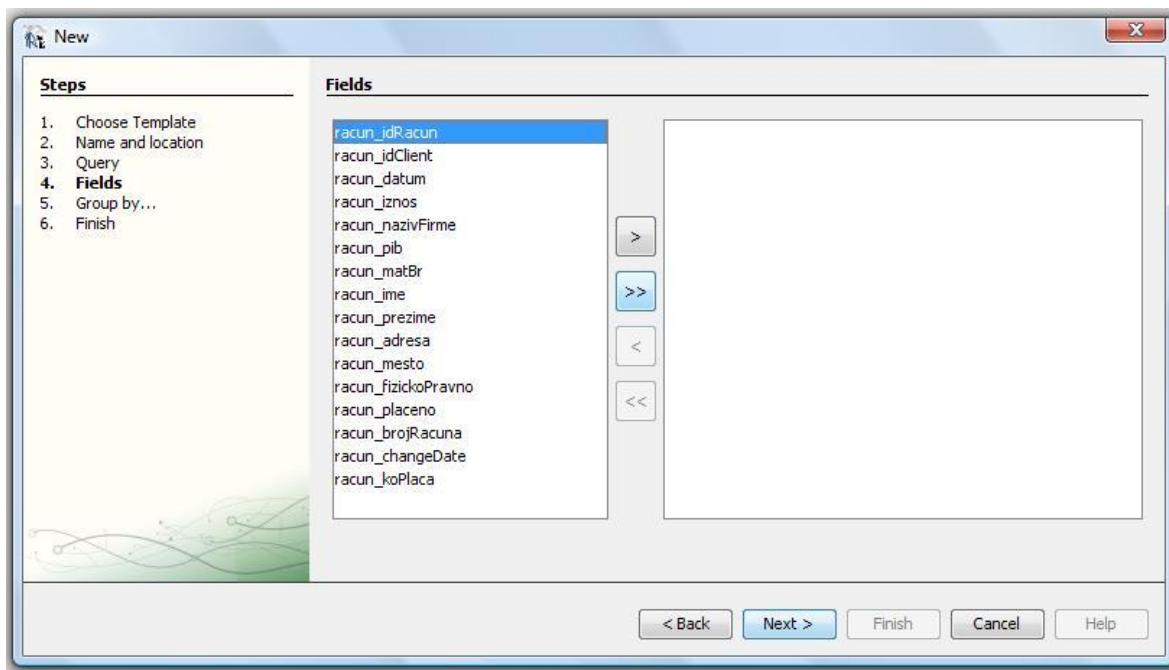
Slika 2-3-9 Spisak tabela u odabranoj bazi podataka

Nakon toga se vrši izbor polja iz odabranih tabela koja će učestvovati u generisanju izveštaja (Slika 2-3-10). Ukoliko se desi da se neki od podataka zaboravi u ovom trenutku, uvek se može ponovo doći do ovog dela ekrana i dodati željeni podatak u generisani izveštaj.



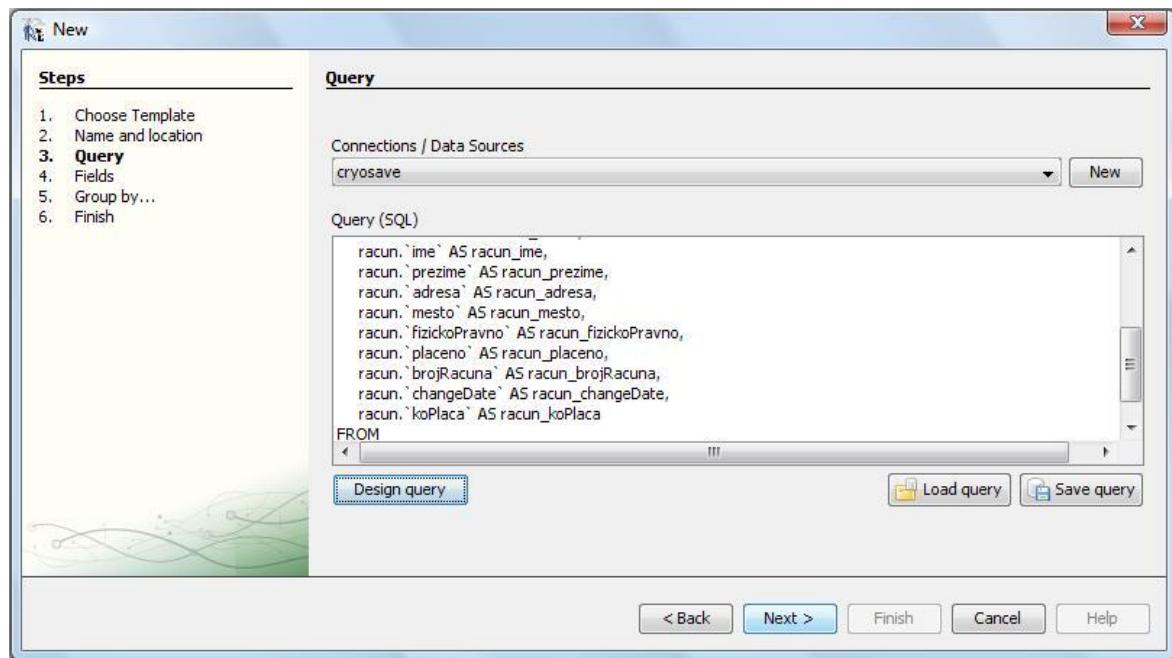
Slika 2-3-10 Postupak odabira neophodnih polja tabele za upit u bazu podataka

Pošto se izaberu željena polja iz tabele, prikazaće se prozor aplikacije koji omogućava izbor polja koja će biti pozivana u izveštaju (slika 2-3-11). Nakon toga moguće je odabrati da li će se vršiti neko grupisanje podataka u izveštaju i koji podaci će učestvovati u grupisanju.



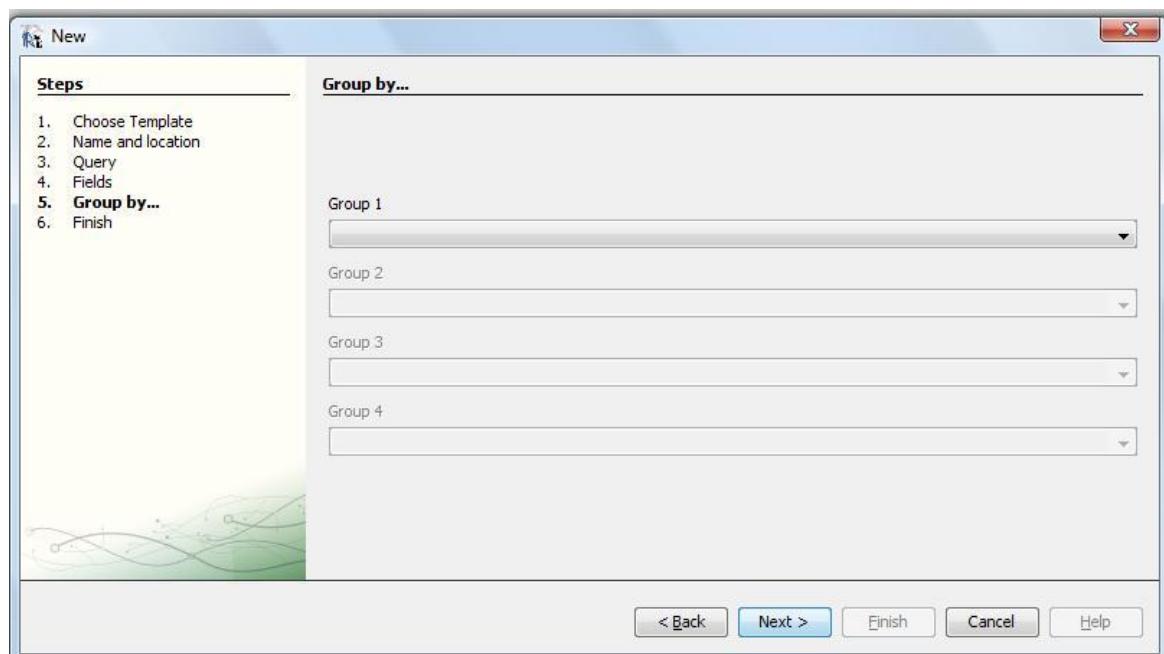
Slika 2-3-11 Odabir polja tabele koja će ući u generisanje izveštaja

Na osnovu svih odabralih opcija i izršenih izbora, *Report Wizard* će da generiše upit u bazi podataka (Slika 2-3-12).



Slika 2-3-12 Primer generisanog upita u bazu podataka koji je kreiran na osnovu izabranih polja

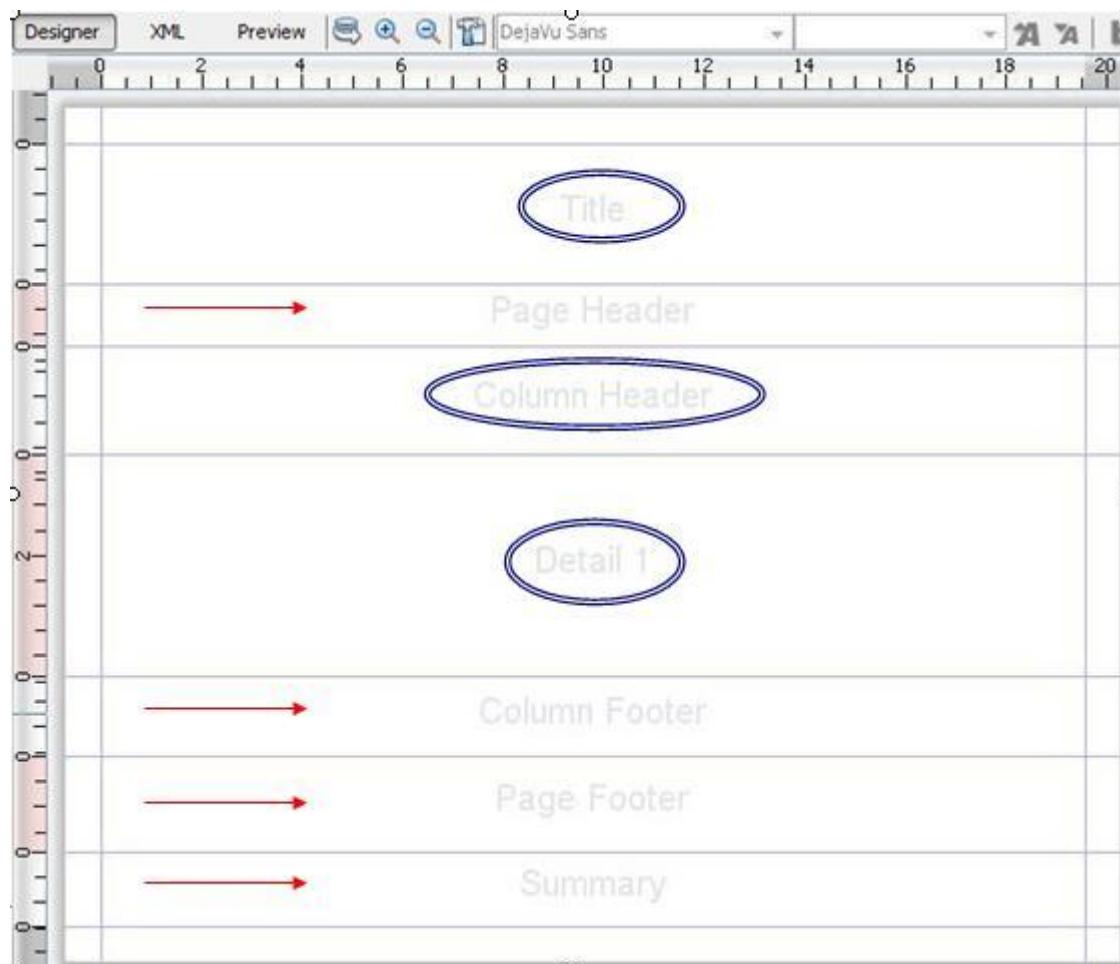
U krajnjem koraku (Slika 2-3-13) je potrebno odrediti da li je potrebno grupisanje podataka:



Slika 2-3-13 Postupak grupisanja podataka u skladu sa nekim definisanim pravilom

Nakon pravilnog izvršenja opisanih faza kreiranja izveštaja, neophodno je proveriti osnovni izgled generisanog izveštaja, a zatim i zatvaranje *Report Wizard* aplikacije. U izveštaju će, na početku, biti prikazana prazna strana podeljena u nekoliko celina.

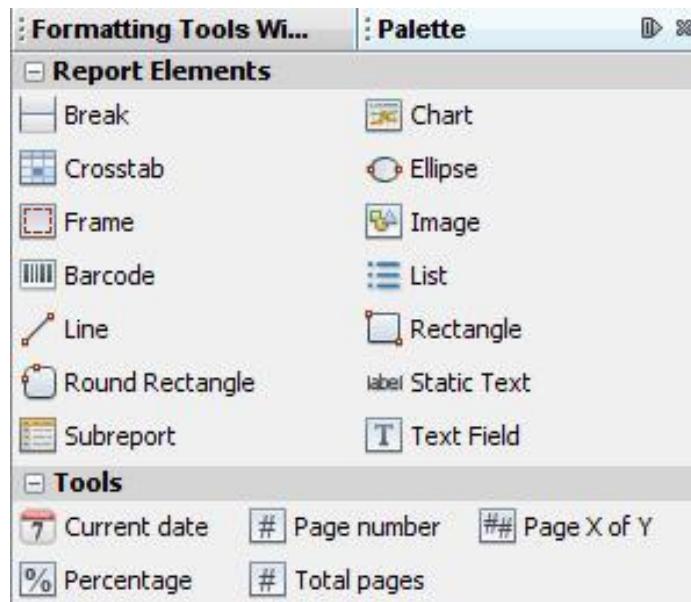
Najvažnije celine izveštaja su Naslov (*Title*), Zaglavje kolone (*Column header*) i Detalji (*Detail*), i prikazani su na slici 2-3-14.



Slika 2-3-14 Primer celina jednog izveštaja

Nakon ovih postupaka obezbeđeni su neophodni uslovi za početak kreiranja izgleda izveštaja. U svaku od celina izveštaja moguće je postaviti sliku, labelu, tekstualno polje, grafikon ili neki od ponuđenih elemenata izveštaja (slika 2-3-15).

Takođe, moguće je postaviti neko od obeležja koje karakteriše stranicu izveštaja, kao što je broj strane, ukupan broj strana izveštaja, datum i drugo.



Slika 2-3-15 Paleta karakteristika i elemenata izveštaja

Svaku od željenih karakteristika i elemenata izveštaja je moguće postaviti u odgovarajući odeljak prevlačenjem na prozor izveštaja.

Na konkretnom primeru je predstavljen postupak primene elemenata ove palete, kao i postupak unošenja odgovarajućih vrednosti tekstualnih polja u okviru izveštaja.

3. OPIS KONKRETNE APLIKACIJE I NJENE STRUKTURE

Aplikacija na kojoj će biti predstavljena primena *JasperReports* alata, kreirana je tako da koristi obimnu bazu podataka iz koje čita podatke i popunjava izveštaje. Korišćena je aplikacija koja ima primenu u radu sa matičnim ćelijama i njihovim čuvanjem u odgovarajućim bankama matičnih ćelija u svetu. Glavni deo aplikacije je namenjen generisanju izveštaja u različitim formatima. Aplikacija se zasniva na brojnim tehnologijama: Java 2EE, Struts, Ajax, Dojo, *Jasper Reports*, JSP, i MySQL baza podataka.

S obzirom da je u pitanju aplikacija rađena za potrebe medicinske ustanove, njena struktura je prilagođena potrebama klijenata i karakteristikama banke ćelija. Iz tih razloga se baza podataka sastoji iz velikog broja tabela vezanih za svakodnevno poslovanje.

U ovom radu izdvojene su neke od važnijih tabela podataka koje su korišćene u daljoj obradi teme. Tabele koje su od značaja su:

- **Client** – sadrži sve relevantne podatke o klijentu i detetu koji se registruje za davanje matičnih ćelija u banku
- **Doctor** – podaci o lekaru, odgovornom licu koje je uzelo uzorak matičnih ćelija deteta pri rođenju
- **Hospital** – podaci vezani za bolnicu u kojoj je dete rođeno

Izveštaji, koji se generišu u aplikaciji i podaci koji se koriste, su kreirani u skladu sa potrebama klijenata. Potreba za njihovim postojanjem je želja korisnika aplikacije da svoj svakodnevni posao u velikoj meri automatizuje i da podatke, koje jednom unese u toku rada, ne unosi ponovo, već da se ti podaci prenose kroz aplikaciju i njene različite prozore.

Dakle, ako korisnik želi da registruje klijenta, on će te podatke uneti samo prilikom te registracije. U svakoj daljoj akciji vezanoj za konkretnog klijenta, ti podaci će biti automatski dostupni i upisani u odgovarajuća polja kao i u generisanom izveštaju (Slika 3-1-1).

Aplikacija uključuje obavezno logovanje konkretnog korisnika na sistem, korisnici imaju različita prava pristupa i mogućnost korišćenja aplikacije, tj. imaju određene uloge i samim tim određena prava. Nakon logovanja na sistem, prikazuje se tabelarno predstavljen spisak klijenata. Tabela je editabilna (izmenljiva) i sadrži kolone podataka za svakog klijenta. Najvažniji podaci koji se čuvaju o klijentima su:

- Ime majke
- Prezime majke

- Ime oca
- Prezime oca
- Adresa stanovanja
- Broj stana
- Grad prebivališta
- Poštanski broj
- Broj/brojevi telefona
- Bolnica u kojoj se prati trunoća
- Ime doktora koji prati trudnoću, itd.

Izmeni	Prezime majke	Bar kod	Prezime majke	Ime majke	Prezime oca	Ime oca	Termin	Datum porođaja	Bolnica	Doktor na porodaju	Zaduženje	Adresa	Broj stana	Grad	Poštanski
<input type="checkbox"/>	381/09/01717	Nedović Preradović	Branka	Preradović	Vladan	25.12.09		KC Srbija - institut za ginekologiju i akušerstvo Beograd		0	Jurija Gagarina	204/48	Belgrade	11070	
<input checked="" type="checkbox"/>	381/09/01715	Babić	Jasna	Babić	Vladimir	10.12.09		KC Vojvodina		0	Nikola Tesla	2	Stara Pazova	22300	
<input type="checkbox"/>	381/09/01714	Brodić-Miličević	Rahila	Miličević	Ljubiša	16.12.09		KC Vladičin Han		0	Džona Kenedija	34	Belgrade	11000	
<input type="checkbox"/>	381/09/01713	Džudžević	Silvana	Džudžević	Saša	23.12.09		Gak Narodni front		0	Bulevar oslobođenja	86/III/33	Beograd	11000	
<input type="checkbox"/>	381/09/01712	Šijan-Dorđević	Anita	Dorđević	Vladimir	10.02.10		Gak Narodni front		0	Ace Joksimovića	59/7	Beograd	11000	
<input type="checkbox"/>	381/09/01711	Ivanović	Marijana	Ivanović	Milan	14.12.09		Gak Narodni front		0	Kosmajskog odreda	111g	Vlasika Mladenovac	11408	
<input type="checkbox"/>	381/09/01710	Kostić	Lada	Kostić	Miroslav	08.01.10		KBC Zvezdara		0	Kralja Petra	10/49	Beograd	11000	
<input type="checkbox"/>	381/09/01709	Ivković	Milena	Milenković	Milan	12.12.09				0	Njegoševa	4	Pančevo	28000	
<input type="checkbox"/>	381/09/01708	Stojiljković	Marijana	Stojiljković	Dragan	04.01.10		KBC Dr. Dragiša Mišović		0	Vladimira Nazora	20	Bor	18210	

Slika 3-1-1 Primer spiska klijenata

Da bi se pristupilo podacima određenog klijenta, u tabeli je neophodno u okviru kolone *Izmeni* za razmatranog klijenta odabratи opciju editovanja (ikonica kojom se obeležava ova opcija je:).

Time se otvara posebna Web stranica koja daje detaljan prikaz svih podataka unešenih za klijenta. Na toj stranici postoji mogućnost da se unesu podaci kao što je *Ime deteta*, *Prezime deteta*, *Način porođaja*, *Pol deteta*, *Datum i vreme porođaja*, *Broj uzoraka matičnih ćelija* (koji su uzeti i koji će se skladištiti), *Jedinstveni broj klijenta* (bar kod) i drugo (Slika 3-1-2). Sve ove podatke je moguće kasnije menjati u skladu sa novim informacijama koje se vremenom dobijaju.

Pri prvom registrovanju i unošenju podataka za novog klijenta, polja u okviru ovog prozora aplikacije će biti prazna. S obzirom na to da tada još uvek nisu poznati podaci o detetu, odgovornom lekaru koji vodi trudnoću i bolnici u kojoj je planirano da se obavi

porođaj, korisnik aplikacije će moći da popuni samo neka od postojećih polja. Ostali podaci će sukcesivno biti unošeni tokom trudnoće i nakon porođaja.

PODACI O KLIJENTU		PODACI O DETETU										
Prezime majke	Pajković *	Ime deteta	Lana									
Ime majke	Suzana *	Prezime deteta	Pajković									
Prezime oca	Pajković *	Pol deteta	Zensko ▾									
Ime oca	Miroslav *	Tip porodjaja	Carski rez ▾									
Adresa stanovanja	Kapetana Zavišića *	Datum porodjaja	29.12.09									
Broj stana	4 *	Vreme porodjaja	8 : 40 ▾									
Postanski broj	11000 *	Predat sertifikat	<input type="checkbox"/>									
Grad	Belgrade *	<input type="text"/> Novi fajl <input type="button" value="Browse..."/> <input type="button" value="Snimi"/> <table border="1"> <thead> <tr> <th>Naziv</th> <th></th> <th>Brisi</th> </tr> </thead> <tbody> <tr> <td>1824.pdf</td> <td></td> <td></td> </tr> <tr> <td>DHL291209.jpg</td> <td></td> <td></td> </tr> </tbody> </table>		Naziv		Brisi	1824.pdf			DHL291209.jpg		
Naziv		Brisi										
1824.pdf												
DHL291209.jpg												
Država	Serbia ▾											
Telefon	+381113974941											
Mobilni telefon	+381641190231											
Mobilni telefon 2	+38163200972											
E-mail	miroljub.pajkovic@mon											
Bar kod	381/09/01824											
Doktor (trudnoca)	Štefana Vidaković											
Doktor (porodjaj)	KC Institut za ginekologiju											
Bolnica												
Napomene												
Broj uzoraka	1											
Očekivan datum porodjaja	29.12.09 *											
Zaduzenje	0											
Grčka racun												
Iznos uplate	0											
Broj racuna												
Broj iz ministarstva												
<input type="button" value="Upisi"/> <input type="button" value="Kraj"/>												

Slika 3-1-2 Primer pregleda podataka klijenta

Osim ovih podataka, za svakog klijenta se generiše skup fajlova kojim se dodatno dokumentuje sve što je vezano za određenog klijenta. U pitanju su sledeći dokumenti:

- Dokument koji sadrži vrednost bar koda, odnosno jedinstveni broj koji klijent dobija pri registraciji. Dokument je u **.pdf** formatu.
- Dokument koji se kreira pri prvoj registraciji u **.doc** formatu, a dobija se konverzijom podataka klijenta izvedenog u vidu izveštaja.
- Fajl u **.xls** formatu koji se kreira prilikom dobijanja krajnjih podataka o detetu.
- Sertifikat koji sadrži kôd mesta na kojem je uskladišten uzorak matičnih ćelija deteta u okviru svetske banke matičnih ćelija u Belgiji.
- Dokumenta vezana za plaćanje troškova.

- Dokument vezan za transport uzoraka u banku matičnih celija.

Neke do ovih dokumenata dobijamo direktno koristeći *Jasper Reports* alat i konvertovanjem podataka u izveštaj u određenom formatu.

Da bi bilo moguće da se Java kôd poveže sa odgovarajućim podacima predstavljenim u izveštaju, potrebno je da se kreira posebna klasa objekata koja sadrži odgovarajuće funkcije, odnosno *get*-ere (funkcije za čitanje vrednosti polja) i *set*-ere (funkcije za postavljanje vrednosti polja) za sva polja koja treba da se nalaze u izveštaju.

U okviru Bloka 3-1 je predstavljen primer jedne funkcije koja zadovoljava opisane uslove.

```
package beans.izvestaji;

import java.io.Serializable;
import java.sql.Time;
import java.text.SimpleDateFormat;
import java.util.Date;

public class ClientUnosBean implements Serializable {

    private String client_surnameMother;
    private String client_nameMother;
    private String client_surnameFather;
    private String client_nameFather;
    private String client_clientAddress;
    private String client_flatNumber;
    private int client_clientZipCode;
    private String client_clientCity;
    private String client_clientPhone;
    private String client_clientMobilePhone;
    private String client_clientEmail;
    private String hospital_name;
    private String doctor_name;
    private String doctor_surname;
    private String hospital_address;
    private String hospital_zipCode;
    private String hospital_city;
    private String hospital_tel;
    private String hospital_email;
    private Date client_expectedBirthDate;
    private String client_barCode;
    private String client_name1;
    private String client_street1;
    private String client_number1;
    private int client_zipCode1;
    private String client_city1;
    private String client_country1;

    public String getClient_city1() {
        return client_city1;
    }
}
```

```
public void setClient_city1(String client_city1) {
    this.client_city1 = client_city1;
}

public String getClient_country1() {
    return client_country1;
}

public void setClient_country1(String client_country1) {
    this.client_country1 = client_country1;
}

public String getClient_name1() {
    return client_name1;
}

public void setClient_name1(String client_name1) {
    this.client_name1 = client_name1;
}

public String getClient_number1() {
    return client_number1;
}

public void setClient_number1(String client_number1) {
    this.client_number1 = client_number1;
}

public String getClient_street1() {
    return client_street1;
}

public void setClient_street1(String client_street1) {
    this.client_street1 = client_street1;
}

public String getClient_zipCode1() {
    return "" + client_zipCode1;
}

public void setClient_zipCode1(int client_zipCode1) {
    this.client_zipCode1 = client_zipCode1;
}

public String getClient_surnameMother() {
    return client_surnameMother;
}

public void setClient_surnameMother(String client_surnameMother) {
    this.client_surnameMother = client_surnameMother;
    if(this.client_surnameMother != null)
        this.client_surnameMother =
this.client_surnameMother.toUpperCase();
}

public String getClient_expectedBirthDate() {
```

```
SimpleDateFormat dateFormat = new
SimpleDateFormat("dd.MM.yy");
    if(client_expectedBirthDate != null)
        return dateFormat.format(client_expectedBirthDate);
    else return "";
}

public void setClient_expectedBirthDate(Date
client_expectedBirthDate) {
    this.client_expectedBirthDate = client_expectedBirthDate;
}

public String getClient_clientAddress() {
    return client_clientAddress + " " + (client_flatNumber ==
null ? "" : String.valueOf(client_flatNumber));
}

public void setClient_clientAddress(String client_clientAddress) {
    this.client_clientAddress = client_clientAddress;
    if(this.client_clientAddress != null)
        this.client_clientAddress =
this.client_clientAddress.toUpperCase();
}

public String getClient_clientCity() {
    return client_clientCity;
}

public void setClient_clientCity(String client_clientCity) {
    this.client_clientCity = client_clientCity;
    if(this.client_clientCity != null)
        this.client_clientCity =
this.client_clientCity.toUpperCase();
}

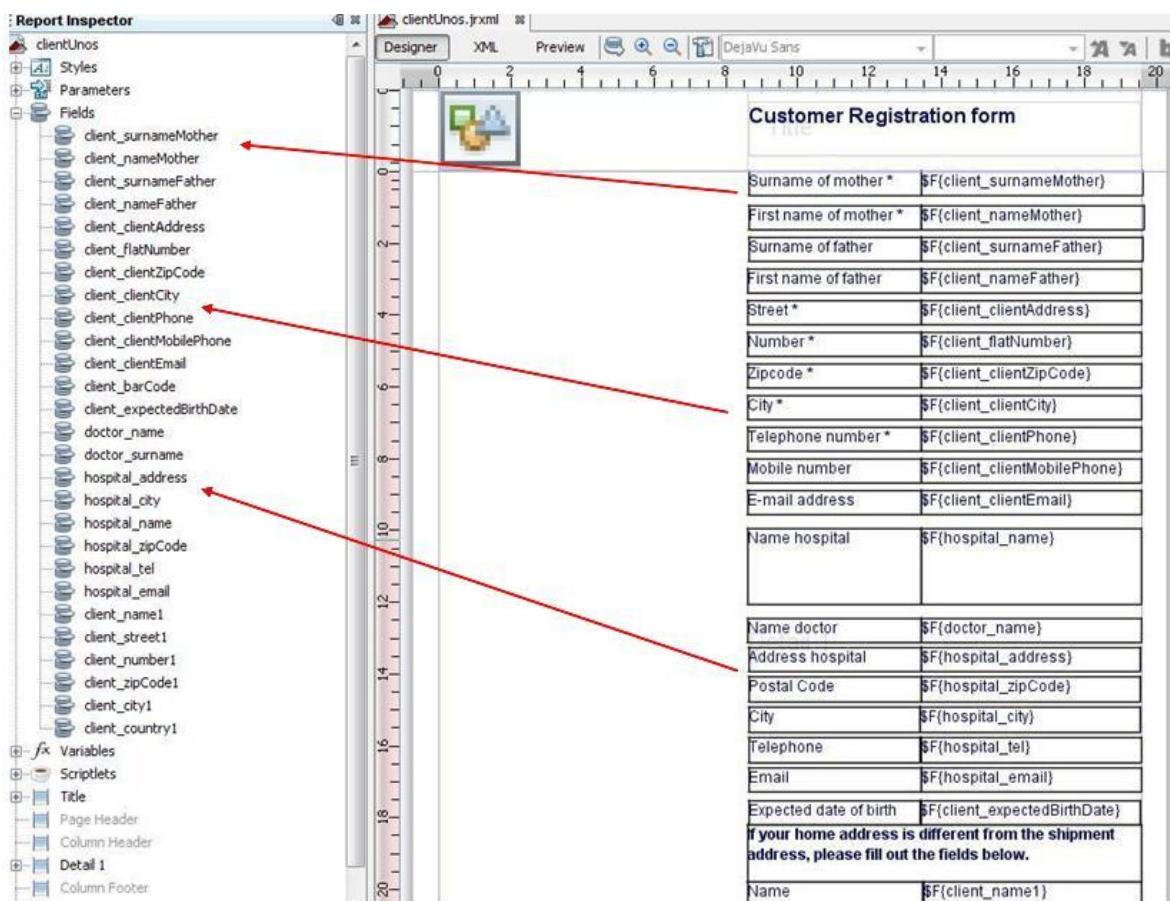
public String getClient_clientMobilePhone() {
    return client_clientMobilePhone;
}

public void setClient_clientMobilePhone(String
client_clientMobilePhone) {
    this.client_clientMobilePhone = client_clientMobilePhone;
    if(this.client_clientMobilePhone != null)
        this.client_clientMobilePhone =
this.client_clientMobilePhone.toUpperCase();
}

....
```

Blok 3-1 Osnovni *get*-eri i *set*-eri koji su vezani za odgovarajućeg klijenta

Analogno ovim parametrima, slični su generisani i postoje u okviru *JapserReport* alata, čime je postignuta veza između Java koda i alata za generisanje izveštaja. Na osnovu uspostavljene veze, JasperReports alat će omogućiti pravilno čitanje i tumačenje podataka predstavljenih kodom aplikacije (Slika 3-1-3).



Slika 3-1-3 Prikaz kreiranog izveštaja za registraciju klijenata

Blok 3-2 daje pregled funkcija čija je uloga da za zadate parametre, koji su obeleženi istim imenima kao i u okviru izveštaja, čita vrednosti iz baze podataka i puni vektor vrednostima koje se prikazuju u izveštaju.

```

public static Vector getClientUnos(UserBean korisnik, int sifra)
throws DatabaseException {

    Connection con = null;
    PreparedStatement st = null;
    ResultSet rs = null;

    try {
        con = DBUtils.getConnection();
        st = con.prepareStatement("SELECT "
                + "client.`surnameMother` AS "
        client_surnameMother, "
                + "client.`nameMother` AS client_nameMother, "
                + "client.`surnameFather` AS "
        client_surnameFather, "

```

```

+ "client.`nameFather` AS client_nameFather, "
+ "client.`clientAddress` AS
client_clientAddress, "
+ "client.`flatNumber` AS client_flatNumber, "
+ "client.`clientZipCode` AS
client_clientZipCode, "
+ "client.`clientCity` AS client_clientCity, "
+ "client.`clientPhone` AS client_clientPhone, "
+ "client.`clientMobilePhone` AS
client_clientMobilePhone, "
+ "client.`clientEmail` AS client_clientEmail, "
+ "client.`barCode` AS client_barCode, "
+ "client.`nameChild` AS client_nameChild, "
+ "client.`surnameChild` AS client_surnameChild,
"
+ "client.`childGender` AS client_childGender, "
+ "client.`birthTime` AS client_birthTime, "
+ "client.`birthType` AS client_birthType, "
+ "client.`birthDate` AS client_birthDate, "
+ "client.`idDoctorBirth` AS
client_idDoctorBirth, "
+ "client.`idDoctorPregnacy` AS
client_idDoctorPregnacy, "
+ "client.`idHospital` AS client_idHospital, "
+ "client.`notes` AS client_notes, "
+ "client.`sampleNumber` AS client_sampleNumber,
"
+ "client.`expectedBirthDate` AS
client_expectedBirthDate, "
+ "client.`idGreeceIncome` AS
client_idGreeceIncome, "
+ "client.`chargeValue` AS client_chargeValue, "
+ "client.`payValue` AS client_payValue, "
+ "client.`payDate` AS client_payDate, "
+ "client.`firmCountDate` AS
client_firmCountDate, "
+ "client.`firmCountNumber` AS
client_firmCountNumber, "
+ "client.`name1` AS client_name1, "
+ "client.`street1` AS client_street1, "
+ "client.`number1` AS client_number1, "
+ "client.`zipCode1` AS client_zipCode1, "
+ "client.`city1` AS client_city1, "
+ "client.`country1` AS client_country1, "
+ "doctor.`name` AS doctor_name, "
+ "doctor.`surname` AS doctor_surname, "
+ "doctor.`idHospital` AS doctor_idHospital, "
+ "hospital.`address` AS hospital_address, "
+ "hospital.`city` AS hospital_city, "
+ "hospital.`name` AS hospital_name, "
+ "hospital.`zipCode` AS hospital_zipCode, "
+ "hospital.`tel` AS hospital_tel, "
+ "hospital.`email` AS hospital_email, "
+ "hospital.`city` AS hospital_city "
+ "FROM "
+ "`client` client LEFT JOIN `doctor` doctor ON
doctor.`idDoctor` = client.`idDoctorPregnacy` "
+ "LEFT JOIN `hospital` hospital ON
client.`idHospital` = hospital.`idHospital` "

```

```
        + "where idClient = ?" );
st.setInt(1, sifra);
rs = st.executeQuery();

if (rs.next()) {
    ClientUnosBean bean = new
ClientUnosBean();

    bean.setClient_surnameMother(rs.getString("client_surnameMother"));
;

    bean.setClient_nameMother(rs.getString("client_nameMother"));
;

    bean.setClient_surnameFather(rs.getString("client_surnameFather"));
;

    bean.setClient_nameFather(rs.getString("client_nameFather"));
;

    bean.setClient_clientAddress(rs.getString("client_clientAddress"));
;

    bean.setClient_flatNumber(rs.getString("client_flatNumber"));
;

    bean.setClient_clientZipCode(rs.getInt("client_clientZipCode"));
;

    bean.setClient_clientCity(rs.getString("client_clientCity"));
;

    bean.setClient_clientPhone(rs.getString("client_clientPhone"));
;

    bean.setClient_clientMobilePhone(rs.getString("client_clientMobile
Phone"));

    bean.setClient_clientEmail(rs.getString("client_clientEmail"));
;

    bean.setHospital_name(rs.getString("hospital_name"));
;

    bean.setDoctor_name(rs.getString("doctor_name"));
;

    bean.setDoctor_surname(rs.getString("doctor_surname"));
;

    bean.setHospital_address(rs.getString("hospital_address"));
;

    bean.setHospital_zipCode(rs.getString("hospital_zipCode"));
;

    bean.setHospital_city(rs.getString("hospital_city"));
;

    bean.setHospital_tel(rs.getString("hospital_tel"));
;

    bean.setHospital_email(rs.getString("hospital_email"));
;

    bean.setClient_expectedBirthDate(rs.getDate("client_expectedBirthD
ate"));
;

    bean.setClient_barCode(rs.getString("client_barCode"));
;

    bean.setClient_name1(rs.getString("client_name1"));
;
```

```

        bean.setClient_street1(rs.getString("client_street1"));

        bean.setClient_number1(rs.getString("client_number1"));

        bean.setClient_zipCode1(rs.getInt("client_zipCode1"));

        bean.setClient_city1(rs.getString("client_city1"));

        bean.setClient_country1(rs.getString("client_country1"));

        Vector v = new Vector();
        v.addElement(bean);
        return v;
    } else {
        throw new DatabaseException("Nepoznata sifra
klijenta");
    }
} catch (SQLException e) {
    throw new DatabaseException(e.getMessage());
}
}
}

```

Blok 3-2 Primer generisanja veze između parametara u izveštaju i Java koda klijenta

U procesu generisanja izveštaja je važno određivanje formata izveštaja. Format izveštaja se određuje pomoću posebne klase, Action klase u okviru Java koda. Naime, u osnovi ovakvog generisanja izveštaja je *Struts* tehnologija koja je zasnovana na primeni Form i Action klase za čitanje zahteva i zatim za slanje odgovora na zahtev (*Request*) od strane klijenta. Blok 3-3 daje prikaz opisanog dela koda :

```

.....
if(format.equalsIgnoreCase("pdf"))
    bytes = JasperExportManager.exportReportToPdf(jp);
else if(format.equalsIgnoreCase("xls")){
JRXlsExporter xlsExporter = new JRXlsExporter();
ByteArrayOutputStream bo = new ByteArrayOutputStream();
xlsExporter.setParameter(JRExporterParameter.JASPER_PRINT, jp);
xlsExporter.setParameter(JRExporterParameter.OUTPUT_STREAM, bo);
xlsExporter.setParameter(JRXlsExporterParameter.IGNORE_PAGE_MARGINS,
    new Boolean(true));
xlsExporter.setParameter(JRXlsExporterParameter.IS_ONE_PAGE_PER_SHEET,
    Boolean.TRUE);
xlsExporter.setParameter(JRXlsExporterParameter.IS_REMOVE_EMPTY_SPACE_BETWEEN_ROWS,
    Boolean.TRUE);
xlsExporter.setParameter(JRXlsExporterParameter.IS_WHITE_PAGE_BACKGROUND
    ,
    Boolean.FALSE);
xlsExporter.setParameter(
JRXlsExporterParameter.IS_REMOVE_EMPTY_SPACE_BETWEEN_COLUMNS, Boolean.TRUE);
}

```

```
        xlsExporter.exportReport( );

        bytes=bo.toByteArray();
    }
else if(format.equalsIgnoreCase( "doc" )){
    JRRtfExporter rtfExporter = new JRRtfExporter();
    //JRDocxExporter rtfExporter = new JRDocxExporter();
    ByteArrayOutputStream bo = new
ByteArrayOutputStream();
    rtfExporter.setParameter(JRExporterParameter.JASPER_PRINT , jp);

    rtfExporter.setParameter(JRExporterParameter.OUTPUT_STREAM,bo);
    rtfExporter.exportReport();
    bytes=bo.toByteArray();

}
}
catch(DatabaseException e)
{
    session.setAttribute("poruka","Greska pri citanju
podataka iz baze: " + e.getMessage());
    return mapping.findForward("poruka");

}
catch(JRException e)
{
    session.setAttribute("poruka","Greska pri kreiranju
izvestaja: " + e.getMessage());
    return mapping.findForward("poruka");
}

session.setAttribute("file" , bytes);
session.setAttribute("format" , format);

return mapping.findForward("sendDoc");

}
```

Blok 3-3 – Primer prepoznavanje formata izveštaja

Ovim su opisani najvažniji koraci u procesu kreiranja jednog izveštaja. U nastavku su na slikama 3-1-4 i 3-1-5 prikazani primeri konkretnih izveštaja dobijenih u **.doc** i **.xls** formatima koji sadrže različite tipove podataka o klijentima.

3. Opis konkretne aplikacije i njene strukture

The screenshot shows a Microsoft Word document window with a toolbar at the top. The main content area contains a registration form titled "Customer Registration form". The form includes fields for personal information, address details, and hospital contact information. At the bottom, it displays an ID number.

Cryo-Save
Balcanica Ltd.
A member of the Life Sciences Group

Customer Registration form

Surname of mother *	BABIĆ
First name of mother *	JASNA
Surname of father	BABIĆ
First name of father	VLADIMIR
Street *	NIKOLE TESLE 2
Number *	2
Zipcode *	22300
City *	STARNA PAZOVA
Telephone number *	+38122363481
Mobile number	+381652220181
E-mail address	vladshop@yahoo.com
Name hospital	KC VOJVODINE
Name doctor	
Address hospital	Hajduk Veljkova 1
Postal Code	21000
City	Novi Sad
Telephone	+381214899222
Email	
Expected date of birth	10.12.09
Marked with * mandatory field	

ID Number: 381/09/01715

Slika 3-1-4 Primer forme izveštaja za registraciju u .doc formatu

Microsoft Excel - BabicXLS1715													
File Edit View Insert Format Tools Data Window Help 													
A1	fx												
	G	I	K	N	P	R	T	V	X	Z	ABADAGAIKANACAFATAVAXAZBE		
2				Identifikacioni broj			Reference number						
3	<u>Identifikacioni formular</u> <u>Identification form</u> Molimo Vas da popunite priloženi formular latiničnim pismom i da ga pošaljete zajedno sa uzorkom <u>Please complete in English and send with blood sample</u>												
4	Majka/Mother												
5	Prezime: Surname	B	A	B	I	C							
6	Ime: Name	J	A	S	N	A							
7	Adresa: Address	NIKOLO TESLE 2											
8	Poštanski broj: Postal code	2	2	3	0	0	Mesto: City	S	T	A	R		
9	Broj telefona: Tel. number	+381	2	2	3	6	3	4	8	1	Drzava Country		
10	Mob.telefon: Mobile	+381	6	5	2	2	2	0	1	8	1		
11	Otac/Father	V L A D I M I R											
12	Ime: Name	B	A	B	I	C							
13	Prezime: Surname												
14	Dete/Child	Molimo Vas da popunite sve raspoložive informacije o detetu											
15	Ime: Name	N	O	V	A	K							
16	Prezime: Surname	B	A	B	I	C							
17	Datum rođenja: Date of birth	1	5	1	2	0	9	Vreme rođenja: Time of birth	0	8	2		
18	Pol: Sex	Muski M				X		Zenski F					
19	Porodjaj: Delivery	Vaginalni Vaginal						Carski rez Caesarian	X				
20	Ostali podaci/Other data												
21	Ime doktora: Doctor - name	D	J	O	R	D	J	E	P	E	T		
22	Porodilište: Maternity hospital	KC VOJVODINE										C	
23													
24													
25													
26													

Slika 3-1-5 Izveštaj sa podacima o klijentu i njegovom detetu u .xls formatu

Konkretno, u XLS izveštaju postignuto je da se za neke vrednosti polja svaka ćelija popuni jednim karakterom, što je značajno kompleksnije nego da su ti karakeri bili u jednoj ćeliji tabele. Sličan postupak ovome je i kreiranje izveštaja u .pdf formatu.

4. PRIMENA JASPERREPORTS APLIKACIJE

U osnovne funkcije koje se primenjuju nad aplikacijom, osim onih predstavljenih u poglavlju 3 ovog rada, ubrajaju se i postavljene pretrage na odgovarajuće sekcije aplikacije.

Prva strana, koja se pojavljuje pri pokretanju aplikacije, je strana za registraciju korisnika (*login* strana), gde je potrebno da korisnik unese svoje korisničko ime, koje je prethodno dobio od administratora, kao i odgovarajuću lozinku, na osnovu kojih sistem izvršava autentifikaciju korisnika (Slika 4-1). Ukoliko je unešeno nepostojeće korisničko ime ili pogrešna lozinka, sistem neće ulogovati korisnika i obavestiće korisnika da je došlo do greške (Slika 4-2) i da je potrebno da ponovo pokuša da unese svoje podatke za logovanje.



Slika 4-1 Prikaz prozora aplikacije za inicijalno logovanje na sistem

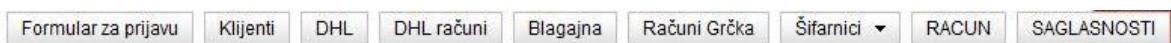


Slika 4-2 Prikaz prozora koji obaveštava da je unešena pogrešna lozinka

Nakon uspešnog logovanja, prikazuje nam se naslovna strana Klijenti. Na naslovnoj strani postoji nekoliko pretraga koje ubrzavaju rad korisnika. Pretraga se može obavljati u odnosu na:

1. Prezime majke,
2. Prezime oca,
3. Bar kôd,
4. Lekara koji je obavio porodaj i
5. Bolnicu.

Na toj strani se još nalazi i opcija za kreiranje novog klijenta, pri čemu se otvara prozor za upis najosnovnijih podataka klijenta, a naziva se „*Formular za prijavu*“.



Slika 4-3 Osnovni meni aplikacije

Na vrhu stranice se nalazi meni sa opcijama koje omogućavaju prolazak kroz aplikaciju i njene različite sekcije. Neke od opcija su:

- **DHL** – opcija neophodna za unos spiska klijenata čiji uzorci će biti transportovani jednom DHL pošiljkom.

- **DHL računi** – Opcija koja daje pregled i spisak računa vezanih za sve transporte obavljene DHL-om.
- **Šifarnici** – opcija sadrži podmeni koji se sastoji od sekcija:
 1. **Doktori** – Spisak svih doktora čije usluge su koristili klijenti
 2. **Bolnice** – Spisak svih bolnica gde su se trudnice porađale
 3. **Korisnici** – Spisak korisnika aplikacije
- **RAČUN** – Sekcija gde se unose podaci vezani za dugovanja klijenata i generišu izveštaji u vidu računa koji se šalje klijentu putem e-mail-a ili se predaje na ruke, a koji sadrži sumu potrebnu da se uplati za usluge firme.
- **SAGLASNOSTI** – Sekcija gde se beleži spisak klijenata za koje je firma dobila saglasnost od strane ministarstva zdravlja da može vršiti prikupljanje uzoraka i slanje istih u banku matičnih celija. Saglasnost se dobija od ministarstva zdravlja za svakog klijenta ponaosob.

Osim ovoga, prilikom rada na aplikaciji je prisutna i informacija o tome ko je trenutno ulogovan na aplikaciju (Slika 4-4).



Slika 4-4 Informacija o identitetu trenutno ulogovanog korisnika sistema

5. ZAKLJUČAK

Najznačajnije osobine JasperReports alata koje treba spomenuti, a koje ga čine jednim od vodećih alata koji se koriste za generisanje izveštaja su:

- Prikaz tabela i grafikona
- Izlazni formati izveštaja PDF, XML, HTML, CSV, XLS, RTF, TXT
- Pod-izveštaji za rad sa kompleksnim dizajnom izveštaja
- Integrisana mogućnost prikaza bar kodova u okviru izveštaja
- Rotacija teksta u izveštaju
- Veliki izbor stilova koji određuju izgled izveštaja
- Neograničena veličina izveštaja
- Mogućnost korišćenja više izvora podataka u okviru jednog izveštaja.
- Omogućen prevod delova aplikacije na skup stranih jezika (*Localization*)

JasperReports alat je lak za instalaciju i upotrebu. U okviru instalacije nalaze se potrebne Java biblioteke funkcija za generisanje, pregled i štampu izveštaja.

Kod JasperReports alata je otvoren (*open-source*) i dakle dostupan svima putem Interneta. Osim toga, na glavnoj Internet strani JasperReports alata (*sajt*) nalazi se i vrlo sadržajna dokumentacija uz pomoć koje korisnici mogu vrlo brzo da se upoznaju sa alatom [6].

JasperReports može koristiti bilo koji izvor podataka. Čak je moguće da u okviru jednog izveštaja koristimo više različitih izvora podataka za generisanje izveštaja.

Zbog svih gore navedenih osobina, *JasperReports* je jedan od alata koji se najčešće koristi u radu sa web aplikacijama i za generisanje izveštaja.

LITERATURA

- [1] H. Sadik 2008, <http://java.dzone.com/articles/java-reporting-part-2>
- [2] <http://jasperforge.org/uploads/publish/jasperreportswebsite/trunk/jasperreports-ds.pdf>
- [3] <http://java-source.net/open-source/charting-and-reporting>
- [4] O. Topsakal 2006, <http://www.cise.ufl.edu/~otopsaka/CIS4301/ReportDemo/>
- [5] D. Vohra 2006, http://www.theregister.co.uk/2006/10/24/jasperreports_tutorial/
- [6] B. Gentile 2009, <http://jasperforge.org/plugins/wpress/?s=JasperReports>