

Elektrotehnički fakultet Sarajevo

Razvoj programskih rješenja

Izvještaj o projektu

Baza podataka za naučne radove - ENIGMA

Student: Selma Čelosmanović

Predmetni nastavnik: Doc. dr Vedran Ljubović

Sadržaj

1 Opis aplikacije.....	3
2 Osnovne ideje pri implementaciji.....	4
3 Implementacija	5
3.1 Osnovni koncepti	5
3.2 Koncepti objektno orijentiranog programiranja	5
3.3. Baza podataka	5
3.4 Enumi	5
3.5 Grafički interfejs.....	5
3.6 Vlastiti tipovi izuzetaka	6
3.7 Datoteke.....	6
3.8 Funkcionalno programiranje	6
3.9 Niti.....	6
3.10 Izvještaji.....	6
3.11 Maven	6
3.12 Junit testovi i TestFX	7
3.13 Lokalizacija i internacionalizacija	7
4 Prilike za unaprjeđivanje	8
5 Dodatak	9

1 Opis aplikacije

Aplikacija koja će biti opisana u nastavku predstavlja simulaciju softvera za uređivanje baze podataka naučnih radova. S obzirom da se radi o projektu nižeg nivoa, napravljena aplikacija je dosta ograničena, ali sadrži sve osnovne funkcionalnosti baze podataka za naučne radove. Aplikacija je nazvana ENIGMA jer, kao što sama riječ znači, predstavlja nešto, ne svima lahko razumljivo, tj. predstavlja skupinu naučnih radova koji koriste uglavnom ljudi višeg obrazovnog nivoa. Također, riječ ENIGMA je lahko pamtljiva i zvuči primamljivo.

Aplikaciju ENIGMA koriste dvije vrste korisnika – administrator i korisnici koji kreiraju svoj nalog. Lozinka za pristup administratorskom dijelu će biti navedena na kraju ovog izvještaja. Nakon što se administrator prijavi u aplikaciju sa ispravnom šifrom, daje mu se mogućnost dodavanja novih naučnih radova, kao i, u sekciji pregleda svih radova, uređivanja postojećih radova. Pored toga, data je mogućnost brisanja određenog naučnog rada ukoliko postoji potreba za tim. Postoji i opcija štampanja spiska svih trenutnih naučnih radova ukoliko se ti spiskovi žele koristiti kao uvid u raznovrsnost i količinu radova s vremenom. Administratori imaju dužnost da se pobrinu o ispravnom unošenju teksta naučnog rada, kao i informacija o istom. S druge strane, na glavnoj formi kreiran je i odjeljak za kreiranje novog naloga, kao i prijava u već postojeći nalog. Nakon ispravno unešenog imena, prezimena, maila, stepena obrazovanja i lozinke, kreira se korisnikov nalog kojem se pristupa unošenjem ispravnog maila i šifre. Prijavom u prethodno kreirani nalog, otvara se prozor na kojem korisnik ima uvid u sve prisutne radove u bazi. Omogućeno je filtriranje naučnih radova po određenim parametrima. Dakle, korisniku je dat izbor da odredi koji je najstariji datum izdanja svih naučnih radova (ukoliko ne želi vidjeti zastarjele radove), te da odredi želi li prikaz naučnih radova iz određene kategorije, određenog tipa ili određenog autora. Ukoliko želi, može filtrirati po svim parametrima, a može i odrediti samo par njih koji mu odgovaraju. Kada se odluči koji rad želi pogledati, potrebno je da ga selektuje u tabeli, te da klikne na dugme „Pogledaj“, čime će se otvoriti pdf sa izabranim naučnim radom. Kreirani pdf je moguće sačuvati na desktop, štampati, slati kolegama i sl. Na kraju, kreirana je opcija uređivanja trenutnog naloga, gdje se korisniku daje mogućnost da promijeni trenutnu šifru, mail, te da obriše svoj korisnički nalog ako želi.

2 Osnovne ideje pri implementaciji

Pri implementaciji navedene aplikacije, korištena je SQLite baza podataka jer je zadovoljavajuća za projekat ovih razmjera. Osnovna ideja bila je da se razviju osnovne funkcionalnosti za rad sa bazom podataka. Svi naučni radovi i korisnički nalozi se spremaju u istoj, tako da ostaju sačuvani pri ponovnom pokretanju aplikacije. Baza podataka je kreirana koristeći DB Browser for SQLite.

Izgled aplikacije je obogaćen bojama, tako da je kao pozadina postavljena druga boja osim uobičajene bijele. Iako kreiran sa nekoliko boja, izgled je konzistentan kroz cijelu aplikaciju, te je vođeno računa da se dobro uklapa. Jezik aplikacije je, po defaultu engleski, međutim postoji opcija promjene jezika već odmah na početnom ekranu.

Za spašavanje teksta naučnih radova korištene su datoteke koje se smještaju u poseban folder u projektu.

Izveštaj spiska svih radova je kreiran koristeći Jaspersoft studio.

UML class diagram je generisan koristeći IntelliJ okruženje, te je smješten u resources folder u projektu.

3 Implementacija

U nastavku ovog odjeljka slijede opisi izabranih dijelova implementacije kako bi se objasnili načini na koje je korišteno stečeno znanje na predmetu Razvoj programskih rješenja.

3.1 Osnovni koncepti

U projektu sve klase koriste JavaBean specifikaciju, što znači da sve klase imaju navedene privatne attribute, konstruktore, te gettere i settere za njih. Zastupljena je upotreba osnovnih Java kolekcija (pretežno `ArrayList()`) i streamova. Streamovi se koriste pri filtriranju naučnih radova u dijelu kojem pristupa korisnik svojim nalogom, dok su kolekcije korištene pri prikupljanju podataka iz baze i sl.

3.2 Koncepti objektno orijentiranog programiranja

Osnovni koncepti OOP-a (nasljeđivanje, polimorfizam i enkapsulacija) iskorišteni su kako bi se prikazala veza između klase `User`, `Admin` i `Author`. Sve navedene klase su naslijeđene iz klase `Person`.

3.3 Baza podataka

Kako je već navedeno, u projektu je korištena SQLite baza podataka. Baza podataka sadrži dvije tabele – tabelu `user` i tabelu `scientific_paper`. U tabelu `user` se smještaju podaci o svim trenutno postojećim korisničkim nalogima. Tabela `scientific_paper` sadrži sve prisutne naučne radove spremljene od strane administratora.

Osnovne metode za rad sa bazom nalaze se u singleton klasi `PapersUsersDAO`. Zbog jednostavnosti projekta, nije bilo potrebe za dijeljenjem na više DAO klasa. U ovoj klasi se uspostavlja konekcija sa bazom, te se vrši spašavanje novih naučnih radova i korisnika, a i prikupljanje postojećih podataka iz baze. Također, baza se generiše pozivom metode `regenerateBase()` koja kupi podatke iz skripte generisane navedenim DB Browser-om.

3.4 Enumi

U projektu je iskorišten pobrojani tip podataka `Enum` (`PaperType`) kako bi se predstavio tip naučnog rada. S obzirom da su tipovi naučnih radova fiksni i ne mogu se proširivati, implementacija pomoću enuma je bila dobra solucija. Pored toga, u enum `PaperType` dodani su konstruktor i getter koji daju ljepši prikaz naziva tipova u aplikaciji.

3.5 Grafički interfejs

Prilikom implementiranja grafičkog interfejsa korišteno je mnogo GUI elemenata kao što su: `AnchorPane`, `GridPane`, `TableView`, `ChoiceBox`, `Button`, `DatePicker`, `TextField`, `Label`, `PasswordField`, i sl. Funkcije navedenih elemenata nadgledaju klase kontroleri.

Pri validaciji podataka, korišteni su listeneri, te pasivni indikatori. Listener se koristi za prijavu administratora pri upisu šifre dok su pasivni indikatori pogrešnih podataka prikazani

crvenim obrubom oko polja i/ili crvenim porukama upozorenja. Tačnost podataka označena je zelenom bojom. Obrubi oko polja kreirani su pomoću .css fajla.

Kreirana je posebna sekcija za filtriranje naučnih radova pri čemu se filtrirani naučni radovi prikazuju u TableView.

Primijenjeni su i Gestalt principi – principi dobrog dizajna korisničkog interfejsa. Svi međusobno bliski elementi izgledaju kao logička cjelina. Boje pozadine i teksta su uklopljene kako bi korisniku privukle pažnju.

3.6 Vlastiti tipovi izuzetaka

Kreirana je klasa za vlastiti tip izuzetka pod imenom „NoPaperSelectedException“. Ukoliko se želi izvršiti neka akcija nad naučnim radom kao što je izmjena, brisanje ili otvaranje naučnog rada, a da pri tome nije izabran naučni rad nad kojim se akcija želi vršiti, dolazi do bacanja izuzetka NoPaperSelectedException. Navedeni izuzetak je naslijeđen iz klase Exception, te je time označen kao checked tip izuzetka koji je potrebno uhvatiti. Metoda kojom je definisan ovaj izuzetak otvara dijaloški prozor upozorenja u kojem upozorava korisnika da nije izabrao nijedan naučni rad te da je nemoguće izvršiti željenu akciju.

3.7 Datoteke

Kako je već navedeno, svi naučni radovi se smještaju u tekstualne datoteke koje se nalaze u posebnom folderu u resources dijelu projekta. Pozivom metode readFile i writeFile se upisuje ili čita naučni rad. Nazivi datoteka su isti kao i naslovi naučnih radova kako bi se olakšalo povezivanje sa korisničkim interfejsom. Datoteke se sastoje od naslova rada te teksta naučnog rada (svi ostali podaci se drže u bazi podataka).

3.8 Funkcionalno programiranje

Pri kreiranju korisničkog interfejsa korištene su lambda funkcije kako bi se onemogućio odabir budućih datuma na DatePickeru za „datum izdanja naučnog rada“. Također, lambda funkcije su primijenjene i u pisanju listenera.

3.9 Niti

Klikom na dugme „Pogledaj“ u korisničkom nalogu, s obzirom na potrebno vrijeme generisanja pdf fajla, otvara se prozor za učitavanje. Prozor se prikazuje 2 sekunde, dok se u to vrijeme pokupi tekst naučnog rada iz datoteke, te se generiše pdf. Nakon gašenja prozora za učitavanje, otvara se izabrani naučni rad.

3.10 Izvještaji

Izvještaj je kreiran koristeći Jaspersoft Studio alat za kreiranje izvještaja. U administratorskom dijelu aplikacije, klikom na dugme „Štampa spiska svih radova“ poziva se metoda showReport klase PrintReport nakon čega se otvara Jasper izvještaj sa spiskom svih naučnih radova.

3.11 Maven

U projekat su dodane sljedeće maven koordinate koje su olakšale implementaciju određenih metoda.

- `com.github.davidmoten:word-wrap:0.1.6` – iskorišten za uredno generisanje stringova koji se upisuju u datoteku
- `net.sf.jasperreports:jasperreports:6.11.0` – iskorišteno za kreiranje Jasper izvještaja
- `org.apache.pdfbox:pdfbox:2.0.4` – iskorišteno za generisanje pdf fajlova
- `org.mindrot:jbcrypt:0.3m` – koristi se pri heširanju korisnikove lozinke kako ne bi bila vidljiva administratoru ili osobi koja pristupa bazi

3.12 Junit testovi i TestFX

U test folderu označenom kao „Test sources root“ uvedeni su testovi koji testiraju pojedinačne cjeline programskog koda. Tako su za ovaj projekat za klase koje nisu povezane sa grafičkim interfejsom dodani Junit testovi, dok su za controller klase dodani TestFX testovi. U kreiranim testovima se testiraju postojeće metode, kao i ispravnost rada korisničkog interfejsa.

3.13 Lokalizacija i internacionalizacija

Kako je već spomenuto, pri pokretanju projekat je defaultno na engleskom jeziku. Međutim, dodana je opcija promjene jezika u korisničkom interfejsu klikom na naziv jezika. Omogućeno je korištenje aplikacije i na bosanskom jeziku.

Internacionalizacija i lokalizacija realizovana je u Resource Bundle folderu pod nazivom „translation“ u kojem se nalaze fajlovi sa prevodima svih tekstova koji se nalaze u projektu.

4 Prilike za unaprjeđivanje

Svjesna sam da ovaj projekat možda i ne nudi toliko mogućnosti i različitih opcija. S obzirom da mnogo stvari izlazi van opsega ovog kursa, mnoge opcije stoga i nisu implementirane.

Da bi ova aplikacija zaista postala upotrebljiva, potrebno bi bilo koristiti drugu bazu koja bi sadržavala mnogo više podataka nego što sada sadrži. Dalje, trebala bi se dodati mogućnost računanja statistike najposjećenijih radova, kao i davanja mogućnosti korisniku za spašavanje svojih „omiljenih“ radova. Također, bilo bi potrebno omogućiti i da korisnici podnesu svoje naučne radove koje bi nakon toga išle na pregled, te bi bilo odlučeno da li ih prikazati javnosti ili ne. Postoji još mnogo zanimljivih stvari koje bi se mogle dodati, međutim implementacija istih je veoma kompleksna i iziskuje mnogo više vremena nego što je bilo na raspolaganju.

5 Dodatak

Lozinka za pristup administratorskom nalogu -> administrator