

Elektrotehnički fakultet Sarajevo

Izvještaj o projektu  
E-Index

Student: Emina Hadžić

Broj indeksa: 17909

Predmetni nastavnik: Doc. dr Vedran Ljubović

## Contents

1 Opis aplikacije.....	3
2 Osnovne ideje pri implementaciji.....	4
3 Implementacija.....	5
3.1 Osnovni koncepti.....	5
3.2 Baza podataka .....	5
3.3 Koncepti objektno orijentiranog programiranja .....	5
3.4 Grafički interfejs .....	5
3.5 Datoteke .....	6
3.6 Enumi i interfejsi.....	6
3.7 Lokalizacija.....	6
3.8 Izuzeci .....	6
3.9 Izvještaji .....	6
4 Prilike za unaprjeđivanje .....	<b>Error! Bookmark not defined.</b>
5 Dodatak .....	7

# 1 Opis aplikacije

Aplikacija koja se opisuje u nastavku izvještaja predstavlja ograničenu simulaciju aplikacije za elektronski index. Osnovni cilj je razviti platformu sa fundamentalnim funkcionalnostima kao što su logovanje admina, profesora i studenta, zatim omogućavanje pregleda i modifikacije pripadajućih informacija. U razradi i specificiranju ideje pomogli su sami zahtjevi korisnika. Prema tome, aplikacija ima sljedeće osobine:

- Svim korisnicima je omogućeno logovanje na sistem i logout sa sistema, kao i izbor jednog od ponuđenih jezika
- Profesoru je omogućena modifikacija postojećih ocjena, kao i dodavanje nove ukoliko student nije ocijenjen
- Profesoru je omogućen pregled studenata na predmetima koje isti predaje, zajedno sa njihovim ocjenama radi lakše integracije sa mogućim nadogradnjama ili drugim aplikacijama
- Profesoru ima mogućnost eksporta podataka o predmetima koje predaje, studentima koji ih slušaju kao i njihovim ocjenama
- Admin ima mogućnost dodavanja i brisanja korisnika iz sistema, kao i pregled postojećih korisnika sa njihovim korisničkim imenima, imenima i prezimenima
- Student ima mogućnost pregleda svojih ocjena po predmetu
- Studentu je omogućeno da generiše izvještaj o predmetima po semestrima, kao i export svojih ocjena u txt datoteku radi lakše integracije sa mogućim nadogradnjama ili drugim aplikacijama

## 2 Osnovne ideje pri implementaciji

U implementaciji je korištena SQLite baza podataka, jer zadovoljava potrebe projekta ovih razmjera. Projekat je, sam po sebi, skalabilan, te je moguće lako promijeniti tip baze. Funkcionalnosti za rad nad bazom podataka se nalaze u klasi EIndexDAO koja ujedno predstavlja Data Access Object i Data Access Layer. Pomoću navedene klase, dio za rad sa bazom je ograđen od same aplikacije. U navedenoj klasi nalaze se metode koje učitavaju, upisuju, obrađuju ili modifikuju podatke iz baze i nude ih na korištenje aplikaciji. Time je postignuto odvajanje baze podataka od poslovne logike.

Izgled aplikacije je rađen koristeći JavaFX. Izgled je jednostavan i konzistentan, te je njegov cilj demonstracija funkcionalnosti aplikacije. Grafički korisnički interfejs je pravljen slijedeći principe dobrog dizajna te su ispunjeni Gestaltt principi.

Radi uspostavljanja veze između prikaza (View) i samih podataka, napravljene su klase koje ispunjavaju Java Bean specifikaciju. Uz ovaj rad priložene su slike koje predstavljaju dijagram klasa kao i ERD.

Izvještaji su kreirani pomoću JasperSoft alata.

## 3 Implementacija

U nastavku slijede opisi izabranih dijelova implementacije kako bi se objasnila zastupljenost obuhvaćenog gradiva na predmetu Razvoj programskih rješenja

### 3.1 Osnovni koncepti

U projektu se koriste model klase koje prate JavaBeans specifikaciju. Zastupljena je i upotreba osnovnih Java kolekcija, klasičnih nizova i kolekcije `ArrayList()` koje su bile dovoljne za potrebe ovog projekta. Također, pažnja je posvećena i imenovanju varijabli i pisanju komentara kako bi se pospješilo razumijevanje koda.

### 3.2 Baza podataka

Kao što je već navedeno, korištena je SQLite baza podataka. Za potrebe projekta kreirano je 8 tabela. Uz ovaj rad nalazi se ER dijagram (u vidu slike) generisan od strane Intelij okturženja za bolju vizualizaciju. Tabele Course, Grade, Person, Login, sqlite\_master, Professor, Student i Admin su dovoljno jasne da ne zahtijevaju nikakvo objašnjavanje, osim da se navede da su Student, Admin i Professor specifikacija tabele Person.

### 3.3 Koncepti objektno orijentiranog programiranja

Koncepti OOP-a (nasljeđivanje, polimorizam i enkapsulacija) vide se direktno kroz model klase i to kada Admin, Student i Professor nasljeđuju Person, te prilikom enkapsulacije polja u svim klasama kako bi se odvojio javni od privatnog dijela klasa. Segregacija dužnosti klasa se ogleda kroz klasu Validacija interfejs.

### 3.4 Grafički interfejs

Prilikom implementiranja grafičkog interfejsa korišteni su razni GUI elementi, kao što su:

Button, Label, TextField, PasswordField, ListView, TableView itd.

Funkcije istih nadgledaju kontroleri.

Uzeti su u obzir koncepti dobrog dizajna korisničkog interfejsa (Gestalt principi) gdje su svi logički vezani elementi grupisani i na ekranu. Za prikaz validnih/nevalidnih polja pozivamo se na .css fle i u kontroleru se ovisno o validnosti postavlja odgovarajuća boja polja.

### 3.5 Datoteke

Prilikom iskorištavanja datoteka vidjela sam kod Studenta i Profesora, te je dodana mogućnost da Student i Profesor exportuju izvještaj u txt datoteku i to klikom na „file“ zatim „save“.

### 3.6 Enumi i interfejsi

U projektu se koristi jedan interfejs Validacija koji je iskorišten za validacije osnovnih podataka o korisnicima kao što su : email, name, jmbg, username...

Enum je iskorišten kako bi se čuvali tipovi mogućih osoba koje koriste aplikaciju: Student, Profesor i Admin, te za tipove ocjena.

### 3.7 Lokalizacija

Kako je ranije navedeno, korisnik može izabrati između dva jezika u izgledu aplikacije, a to su bosanski i engleski.

### 3.8 Izuzeci

Vlastitu klasu izuzetaka IllegalSemesterException sam iskoristila da naznačim da semestar ne može biti negativan broj.

### 3.9 Izvještaji

Izvještaji su implementirani korištenjem Jaspersoft Studio alata za kreiranje izvještaja. U aplikaciji se mogu vidjeti klikom na file zatim print.

## 5 Dodatak

### STUDENT

username: neko

password: neko

### PROFESOR

username: profesor

password: test

### ADMINISTRATOR

username: emina

password: emina