|  |  |  |
| --- | --- | --- |
| ZnakUniverziteta | Univerzitet u Novom Sadu  Tehnički fakultet »MihajloPupin«  Zrenjanin | LOGOpupin |

***Školska 2023/24 godina***

***Nastavni predmet: Softversko inzenjerstvo 2***

**SEMINARSKI RAD**

Tema:

Evidencija realizovane zanatske proizvodnje

|  |  |
| --- | --- |
| Predmetni nastavnik: | Student: |
| Prof. dr Ljubica Kazi | Petar Banović IT45/19 |

Zrenjanin,  2024. godina

Contents

[1. ANALIZA POSLOVNOG SISTEMA, SPECIFIKACIJA ZAHTEVA I PLANIRANJE RAZVOJA 5](#_Toc145003044)

[1.1. Analiza poslovnog sistema 5](#_Toc145003045)

[1.1.1. Svrha postojanja sistema i kategorizacija poslova 5](#_Toc145003046)

[1.1.2. Osnovni objekti obrade i opis toka poslovnog procesa 5](#_Toc145003047)

[1.1.3. Organizaciona struktura sistema i pripadnost drugim organizacionim strukturama 5](#_Toc145003048)

[1.1.4. Radna mesta i opis zaduženja 6](#_Toc145003049)

[1.1.5. Analiza dokumentacije (obrazaca) i arhiva 6](#_Toc145003050)

[1.1.6. Poslovna pravila 8](#_Toc145003051)

[1.2. Specifikacija zahteva korisnika 9](#_Toc145003052)

[1.2.1. Profili korisnika i očekivane funkcionalne mogućnosti softvera 9](#_Toc145003053)

[1.2.2. Spisak potrebnih ulaznih ekranskih formi 9](#_Toc145003054)

[1.2.3. Spisak poslovnih pravila, automatizama i ograničenja 9](#_Toc145003055)

[1.2.4. Spisak potrebnih tabelarnih prikaza sa kriterijumima filtriranja 9](#_Toc145003056)

[1.2.5. Spisak potrebnih izveštaja 9](#_Toc145003057)

[1.2.6. Opis toka primene elemenata sistema (softverskih funkcija) u okviru poslovnog procesa 9](#_Toc145003058)

[1.2.7. Potrebne karakteristike korisničkog interfejsa i ostali nefunkcionalni zahtevi 10](#_Toc145003059)

[1.2.8. Potrebna podrška odlučivanju – statistički izveštaji 10](#_Toc145003060)

[1.2.9. Spisak potrebnih eksporta podataka sa formatima datoteka 10](#_Toc145003061)

[1.2.10. Potrebna hardverska i mrežna oprema 10](#_Toc145003062)

[1.2.11. Bezbedonosni aspekti i kreiranje rezervnih kopija podataka 10](#_Toc145003063)

[1.2.12. Uticaj uvođenja novog sistema na kadrove, organizaciju rada i strateške ciljeve organizacije 10](#_Toc145003064)

[1.3. Elementi projektnog planiranja 11](#_Toc145003065)

[1.3.1. SCOPE – Ciljevi, očekivani rezultati, obuhvat i granice projekta 11](#_Toc145003066)

[1.3.2. STAKEHOLDERS, REQUIREMENTS – Potencijalni profili korisnika i funkcionalne karakteristike planiranog rešenja 11](#_Toc145003067)

[1.3.3. PROCUREMENT – potreban materijal, alati i ostala nabavka 11](#_Toc145003068)

[1.3.4. SCHEDULE - Vremenski plan aktivnosti u realizaciji projekta i ključne kontrolne tačke 11](#_Toc145003069)

[1.3.5. HUMAN RESOURCES – Učesnici projekta (realizator poslovne analize, dizajna, implementacije, testiranja, učesnik iz organizacionog sistema) 12](#_Toc145003070)

[1.3.6. COST - Troškovi realizacije projekta (na osnovu nabavke i procene utrošenih sati rada) 12](#_Toc145003071)

[1.3.7. QUALITY - Merne karakteristike uspeha projekta 12](#_Toc145003072)

[1.3.8. INTEGRATION – Integracija planiranog rešenja u radno okruženje 13](#_Toc145003073)

[1.3.9. COMMUNICATIONS - organizacioni i tehnički preduslovi uspešne komunikacije i realizacije projekta 13](#_Toc145003074)

[1.3.10. RISK – Ograničenja projekta, rizici realizacije projekta, rizici uvođenja rešenja 13](#_Toc145003075)

[2. POSLOVNA SISTEMSKA ANALIZA I DIZAJN SOFTVERSKOG RЕŠENJA 14](#_Toc145003076)

[2.1. Model poslovnih procesa 14](#_Toc145003077)

[2.1.1. Algoritamski model poslovnih procesa 14](#_Toc145003078)

[2.1.2. Spisak skladišta podataka 15](#_Toc145003079)

[2.1.3. Analiza jednog poslovnog dokumenta 15](#_Toc145003080)

[2.2. Dizajn softverskih funkcija 17](#_Toc145003081)

[2.2.1. Tabela preslikavanja primitivnih procesa u softverske funkcije 17](#_Toc145003082)

[2.2.2. USE CASE dijagram softverskih funkcija 19](#_Toc145003083)

[2.2.3. Specifikacija slučaja korišćenja za unos i tabelarni prikaz podataka 19](#_Toc145003084)

[2.3. Dizajn modela podataka 23](#_Toc145003085)

[2.3.1. Konceptualni model podataka 23](#_Toc145003086)

[2.3.2. Fizički (relacioni) model podataka 24](#_Toc145003087)

[2.4. Test-bazirana specifikacija zahteva korisnika 24](#_Toc145003088)

[2.4.1. Test-bazirana specifikacija za ekransku formu unosa 24](#_Toc145003089)

[2.4.2. Test-bazirana specifikacija za ekransku formu tabelarnog prikaza sa filterom 27](#_Toc145003090)

[2.5. Plan realizacije elemenata kvaliteta softvera 29](#_Toc145003091)

[2.5.1. Funkcionalni aspekt kvaliteta softvera 29](#_Toc145003092)

[2.5.1.1. Planirane funkcionalne mogućnosti softvera i profili korisnika 29](#_Toc145003093)

[2.5.1.2. Plan realizacije osnovne validacije podataka 29](#_Toc145003094)

[2.5.1.3. Plan realizacije podrške poslovnoj logici 29](#_Toc145003095)

[2.5.1.4. Plan podrške personalizaciji aplikacije i bezbednosti korišćenja 29](#_Toc145003096)

[2.5.2. Strukturni aspekt kvaliteta softvera 30](#_Toc145003097)

[2.5.2.1. Plan arhitekture softvera 30](#_Toc145003098)

[2.5.2.2. Plan realizacije clean code pristupa 31](#_Toc145003099)

[3. IMPLEMENTACIJA, TESTIRANJE I DOKUMENTOVANJE REALIZOVANOG SOFTVERA 32](#_Toc145003100)

[3.1. Dokumentovanje implementacije softverskog rešenja 32](#_Toc145003101)

[3.1.1. Kratak opis primenjenih tehnologija, programskih jezika i razvojnih alata 32](#_Toc145003102)

[3.1.2. Modeli opisa implementacije rešenja 32](#_Toc145003103)

[3.1.2.1. Dijagram komponenti (prikaz arhitekture softvera – višeslojna, MVC, mikroservisi…) 32](#_Toc145003104)

[3.1.2.2. Djagram klasa 33](#_Toc145003105)

[3.1.3. Tabelarni prikaz slojeva i podslojeva aplikacije sa ilustracijom konkretnim fajlovima u rešenju 33](#_Toc145003106)

[3.1.5. Delovi koda sa objašnjenjima po slojevima 36](#_Toc145003108)

[3.1.5.1. Sloj za rad sa podacima (tehnološke klase, klase podataka, tj. repository) 36](#_Toc145003109)

[3.1.5.2. Sloj servisa i sloj poslovne logike sa opisom poslovnog pravila, kodom klase poslovne logike, strukturom i sadržajem datoteka u kojima su poslovna ograničenja 56](#_Toc145003110)

[3.1.5.3. Sloj prezentacione logike - sa klasama koje odgovaraju ekranskim formama 56](#_Toc145003111)

[3.1.5.3.1. Klase za pripremu prikaza podataka - klase sa metodama za preuzimanje i formatiranje podataka iz baze podataka kroz poziv klasa sloja za rad sa podacima 56](#_Toc145003112)

[3.1.5.3.2. Klase za preuzimanje podataka – klase za preuzimanje komandi i podataka sa korisničkog interfejsa i osnovne validacije 66](#_Toc145003113)

[3.1.5.4. Korisnički interfejs – koristi klase prezentacione logike 73](#_Toc145003114)

[3.1.6. Delovi koda i opis implementacije podrške personalizaciji aplikacije i bezbednosti korišćenja (prijava korisnika, rad sa sesijama, razmena podataka putem post metode) 87](#_Toc145003115)

[3.1.7. Delovi koda i opis implementacije osnovnih principa OOP (enkapsulacija, nasleđivanje, polimorfizam) i SOLID principa (single responsibility, interface segregation, dependency inversion sa dependency injection tehnikom) 89](#_Toc145003116)

[3.1.8. Delovi koda objašnjenja realizacije clean code pristupa (čitljivost koda kroz nazive polja, procedura, klasa, komentare (uključujući CRC card komentar o svakoj klasi), konvencije oblika koda i dr) 96](#_Toc145003117)

[3.2. Korisničko i tehničko uputstvo 99](#_Toc145003118)

[3.2.1. Korisničko uputstvo (ekrani i kratak opis načina korišćenja) 99](#_Toc145003119)

[3.2.2. Tehničko uputstvo (uputstvo za instalaciju, početno konfigurisanje parametara rada) 103](#_Toc145003120)

[3.3. Izveštaj o testiranju softvera 103](#_Toc145003122)

[3.3.1. Test slučajevi ispravnog rada korisnika i prikaz rezultata testiranja (ekrana aplikacije i ponašanja) 103](#_Toc145003123)

[3.3.3. Zaključak o testiranju - u odnosu na test-baziranu specifikaciju ponašanja aplikacije 105](#_Toc145003124)

[ZAKLJUČAK 106](#_Toc145003125)

[LITERATURA 107](#_Toc145003126)

1. ANALIZA POSLOVNOG SISTEMA, SPECIFIKACIJA ZAHTEVA I PLANIRANJE RAZVOJA
   1. Analiza poslovnog sistema
      1. Svrha postojanja sistema i kategorizacija poslova(osnovne, pomoćne i upravljačke delatnosti)

Svrha postojanja poslovnog sistema: ostvarivanje prava proizvođača u evidentiranju svoje proizvodnje, proveravati da li je kvota ispunjena na mesečnom nivou u zavisnosti od samog proizvoda i ukoliko je potrebno brisanje evidencije;

**Osnovne delatnosti**:Dodavanje nove evidencije, proveravanje stanja kvote na mesečnom nivou u zavisnosti od samog proizvoda, brisanje evidencije, dodavanje novih proizvođača.

**Pomoćne delatnosti**: Izdavanje izvoda transakcija, uvid u trenutno stanje evidencija.

Upravljačke delatnosti: Omogućavanje bržeg pristupa podacima kako bi se znalo da li će se kvota ispuniti I dobiti bonus na kraju meseca.

* + 1. Osnovni objekti obrade i opis toka poslovnog procesa(životni ciklus obrade za svaki objekat, opis posla)

Osnovni objekat obrade predstavlja proizvođač, koji unosi novu evidenciju i proverava za određeni proizvod da li je kvota realizovana na mesečnom nivou. Tok poslovnog procesa prilikom dodavanja, provere ili brisanja evidencije:

* Dodavanje nove evidencije: lice koje želi da podnese zahtev za dodavanje nove evidencije mora prethodno biti registrovan kao proizvođač. Nakon toga ukoliko je registrovan moguće je dodati novu evidenciju u suprotnom nije moguće dalje nastaviti sa akcijom.

NAPOMENA:Prilikom otvaranja naloga proizvođača, moraju se poneti osnovni dokumenti radi verifikacije.

* Provera: U ovoj akciji proverava se da li je ispunjena mesečna norma za proizvodnju određenog proizvoda. Ovo se sve radi izvlačenjem vrednosti iz xml-a.
* Brisanje: Ukoliko lice više ne želi da proizvodi dati proizvod, moguće je obrisati evidenciju, jer praćenje tog proizvoda više nije neophodno.
  + 1. Organizaciona struktura sistema i pripadnost drugim organizacionim strukturama

Kao što možemo videti sa slike br. 1 u nastavku, na čelu organizacije se nalazi preduzeće, odnosno svaka firma je samostalna i zadužena za svoje usluge. Na sledećem nivou se nalaze se mašine kojima se radi na realizaciji proizvodnje i “motori” firme u vidu proizvodnje, marketinga, istraživačkog razvoja kao i računovodstvo i kadrovi.

A diagram of a company structure

Description automatically generated

*Slika 1 – Organizaciona šema privrede zanatstva*

* + 1. Radna mesta i opis zaduženja

Građansko lice – osoba koja želi da podnese zahtev za pristup proizvođačima;

Proizvođač – osoba koja otvara evidenciju;

* + 1. Analiza dokumentacije (obrazaca) i arhiva(knjiga evidencije, arhive dokumenata)

Ključna dokumentacija potrebna za dodavanje, proveru ili brisanje postojeće evidencije:

* Zahtev za pristup registru privrednih subjekata
* Lična karta

A form with text and numbers

Description automatically generated with medium confidence

*Slika 2 – Primer obrasca za prijavu u registar privrednih subjekata*

* + 1. Poslovna pravila

Poslovno pravilo u ovom seminarskom radu bazirano je na proveri stanja mesečnog kontigenta proizvoda određenog tipa, koje je bitno da svaki proizvođač ostvari kako bi se smatralo da je ispunio mesečnu normu. Ove vrednosti se povlače iz xml-ova, jer je za svakog preduzetnija drugačije kao i za svaki proizvod.

* + 1. Zakonska regulativa, standardi i interni propisi organizacionog sistema

Za sve stavke u vezi proizvodnje, odgovorno je ministartsvo privrede: https://privreda.gov.rs

* 1. Specifikacija zahteva korisnika
     1. Profili korisnika i očekivane funkcionalne mogućnosti softvera
* Korisnik – tabelarni prikaz otvorenih evidencijama sa osnovnim podacima proizvođača, unos podataka o novim evidencijama (lični podaci, priložena dokumenta), tabelarni prikaz kreiranih evidencija sa mogućnošću filtriranja na osnovu mesta proizvodnje, štampanje potvrde o trenutnom stanju svih evidencija, unos podataka potrebnih za prijavu preko interneta.
  + 1. Spisak potrebnih ulaznih ekranskih formi

Unos podataka za evidencije realizovane proizvodnje preko interneta;

Unos podataka o proizvođaču koji želi da kreira evidenciju;

* + 1. Spisak poslovnih pravila, automatizama i ograničenja

OGRANIČENJE

Na osnovu proizvedenog broja, gleda se da li je ispunjena norma

AUTOMATIZAM

Ako je datum do kojeg se gleda norma u zaostatku od 30%, alarmirati u tabelarnom prikazu evidencija ŽUTOM BOJOM sva lica kojima norma nije ispunjena, a ako je u intervalu od 50% alarmirati CRVENOM BOJOM.

* + 1. Spisak potrebnih tabelarnih prikaza sa kriterijumima filtriranja
* Tabelarni pregled evidencija realizovane proizvodne sa mogućnošću filtera prema mestu proizvodnje;
* Tabelarni prikaz proizvođača koji su kreirali evidenciju;
  + 1. Spisak potrebnih izveštaja
* Zahtev za kreiranje evidencije;
  + 1. Opis toka primene elemenata sistema (softverskih funkcija) u okviru poslovnog procesa

Lice koje želi da kreira evidenciju prvobitno mora biti registrovan u rigistru privrednih subjekata. Nakon toga, kad dobije parametre za pristup sajtu, preko **ekranske forme za prijavljivanje zaposlenih**, unosi svoje lične podatke potrebne za prijavu, odabira opciju otvaranje dodavanja nove evidencije realizovane proizvodnje, popunjava i tada snima nove podatke.

Nakov ovoga izvršava se provera svih bitnih parametra, i ukoliko je sve u skladu sa poslovnim pravilima, dodaje se nova evidencija realizovane proizvodnje.

* + 1. Potrebne karakteristike korisničkog interfejsa i ostali nefunkcionalni zahtevi

Aplikacija treba da je realizovana primenom ASP.NET ILI PHP zbog postojeće hosting podrške.

* + 1. Potrebna podrška odlučivanju – statistički izveštaji

Statistički izveštaji o evidencijama realizovane proizvodnje se nalazi unutar ministarstava privrede.

* + 1. Spisak potrebnih eksporta podataka sa formatima datoteka

Eksport podataka o evidenciji realizovane proizvodnje;

Eksport podataka o proizvođačima i podnetim zahtevima;

* + 1. Potrebna hardverska i mrežna oprema

Za realizaciju ovog softverskog projekta potreban je 1 laptop računar.

Nakon realizacije, softver se postavlja na postojeći server, tako da nema potrebe za dodatnim nabavkama.

* + 1. Bezbedonosni aspekti i kreiranje rezervnih kopija podataka
* Bezbednost softverskog rešenja ogleda se u primeni prijave korisnika, razmeni podataka putem sesije i post metode, proveri autentifikovanog pristupa stranicama, kriptovanje šifre za pristup.
* Kreiranje rezervnih kopija podataka realizuje se periodično od strane sistem administratora, koji kopira sadržaj celog servera na eksterni hard disk.
  + 1. Uticaj uvođenja novog sistema na kadrove, organizaciju rada i strateške ciljeve organizacije

Uvođenje novog sistema olakšava rad zaposlenih jer jednim klikom mogu da vide sve evidencije realizovane proizvodnje.

* 1. Elementi projektnog planiranja
     1. SCOPE – Ciljevi, očekivani rezultati, obuhvat i granice projekta
* Ciljevi realizacije projekta razvoja novog softvera su unapređenje i ubrzanje procesa za kreiranje nove evidencije realizovane proizvodnje.
* Očekivani rezultat – veb aplikacija koja treba da podrži osnovne evidencije o proizvođačima i evidencije realizovane proizvodnje, i automatsko signaliziranje u vezi potreba pojačavanje rada kako bi se ispunila norma.
* Biće realizovana – osnovna evidencija realizovane proizvodnje i proizvodjača– unos, tabelarni prikaz prijava za evidencije realizovane proizvodnje sa filterom prema datumu proizvodnje, štampa spiska evidencija realizovane proizvodnje.
  + 1. STAKEHOLDERS, REQUIREMENTS – Potencijalni profili korisnika i funkcionalne karakteristike planiranog rešenja

Za celoviti softver zainteresovane strane i njihovi interesi su:

* Privreda
* Agencija za privredne registre

Za ovaj softver profili korisnika su:

* Zaposleni – unos podataka i uvid o njima;
* Proizvođač – osoba koja unosi podatke o evidencijama
  + 1. PROCUREMENT – potreban materijal, alati i ostala nabavka

Za realizaciju ovog softvera potrebno je:

* Alat – razvojno okruţenje Visual Studio Code
* Alat – sistem za upravljanje bazom podataka MariaDB MySQL
* Alat – tekst editor Microsoft Word 2016

Za snimanje rešenja – 1 CD ili USB memorija;

* + 1. SCHEDULE - Vremenski plan aktivnosti u realizaciji projekta i ključne kontrolne tačke

1. Analiza poslovnog domena, specifikacija zahteva i elementi projektnog planiranja – rok do 05.09.2023.

2. Poslovna sistemska analiza, dizajn softvera – rok do 04.03.2024.

3. Implementacija – rok do 04.03.2024.

4. Testiranje – rok do 04.03.2024.

5. Dokumentovanje – rok do 04.03.2024.

6. Finaliziranje projekta – rok do 04.03.2024..

* + 1. HUMAN RESOURCES – Učesnici projekta (realizator poslovne analize, dizajna, implementacije, testiranja, učesnik iz organizacionog sistema)

U realizaciji svih aktivnosti učestvuje Prof. Dr Ljubica Kazi kao predmetni nastavnik i mentor studentu Petar Banoviću.

* + 1. COST - Troškovi realizacije projekta (na osnovu nabavke i procene utrošenih sati rada)

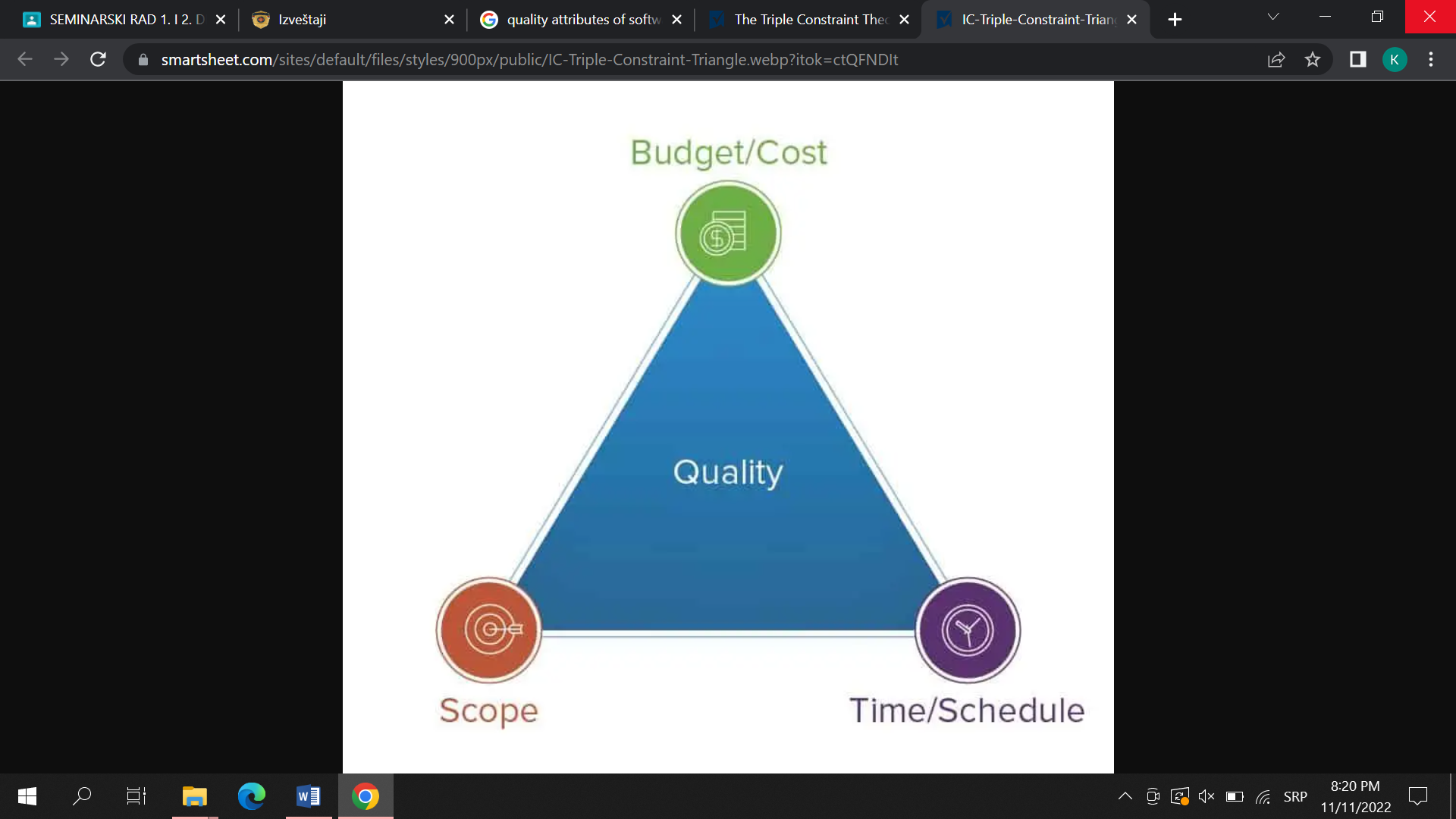
Troškovi:

* Nabavka alata – besplatno
* Nabavka 1 CD – 36 RSD
* Nabavka 1 USB Flash memorije od 32GB – 1199 RSD
* Radni sati – procena da je potrebno okvirno 54 dana po 8 h dnevno = 432 sata.
  + 1. QUALITY - Merne karakteristike uspeha projekta

Uspeh projekta – obuhvat (ako je sve realizovano), troškovi (ako troškovi nisu veći od planiranog),

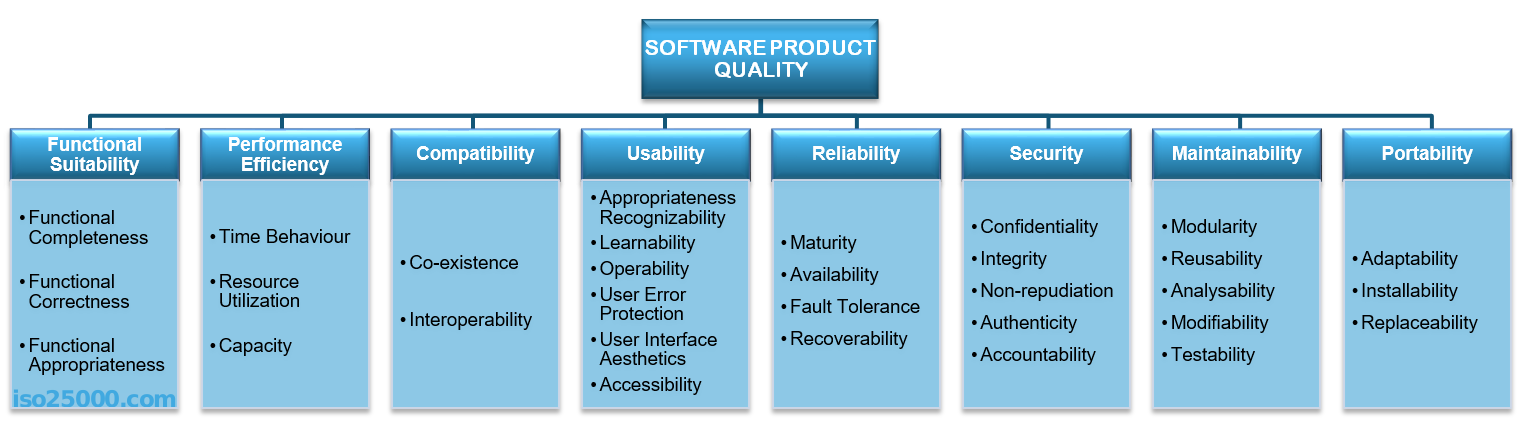
kvalitet (ako realizovano rešenje ima odgovarajuće karakteristike kvaliteta), vreme (ako je urađeno

u okviru planiranog vremena).



*Slika 3 – Tri osnovna atributa kvaliteta softvera*

Karakteristike kvaliteta softvera prema ISO standardima su:



*Slika 4 – Šematski prikaz karakteristika kvaliteta softvera prema standardu ISO 25010*

* + 1. INTEGRATION – Integracija planiranog rešenja u radno okruženje

Da bi se rešenje moglo integrisati na postojeću hosting platformu, mora biti realizovano u okviru

ASP.NET ili PHP tehnologije.

* + 1. COMMUNICATIONS - organizacioni i tehnički preduslovi uspešne komunikacije i realizacije projekta

U okviru realizacije komunikacija se realizuje sa administrativnim radnikom službe koji treba da daju svoje sugestije u vezi preciznijih zahteva za karakteristikama

softvera i ocene softver nakon izrade. Komunikacija se realizuje putem emaila ili putem Google meet-a, i uz nekoliko sastanaka uživo.

* + 1. RISK – Ograničenja projekta, rizici realizacije projekta, rizici uvođenja rešenja

Rizici realizacije projekta:

* Kratak rok izrade
* Raspoloživost administrativnog radnika kadrovske službe za saradnju zbog velikog radnog opterećenja osnovnim obavezama.
* Snalaženje (odnosno nesnalaženje) zaposlenih u upravi u korišćenju novog softvera.

1. POSLOVNA SISTEMSKA ANALIZA I DIZAJN SOFTVERSKOG RЕŠENJA
   1. Model poslovnih procesa
      1. Algoritamski model poslovnih procesa

Modelom poslovnih procesa predstavljamo suštinske poslovne procese organizacionog sistema (na osnovu opisa toka poslovnog procesa), nezavisno od trenutne ili buduće implementacije. Model poslovnih procesa je osnova dizajna softverskog rešenja i za jedan model poslovnih procesa može se uraditi više različitih predloga dizajna.

A diagram of a flowchart

Description automatically generated

*Slika 5 – Business Process Model tipa Analysis za proces skladišenja evidencije realizovane proizvodnje*

*Izvor: Uradio student – autor seminarskog rada*

* + 1. Spisak skladišta podataka

U OKVIRU POSLOVNOG PROCESA KREIRANJE NOVE EVIDENCIJE REALIZOVANE PROIZVODNJE

Zapisnik ličnih dokumenata;

Zapisnik o prijavama;

* + 1. Analiza jednog poslovnog dokumenta

Primer dokumentacije: U okviru veb stranice Agencija za privredne registre dat je spisak potrebne dokumentacije za kreiranje naloga za registar subjektivnih proizvodjača.

https://www.apr.gov.rs/registri/privredna-drustva/uputstva/opsta-uputstva.2029.html

A close-up of a form

Description automatically generated

*Slika 6 – Izgled veb stranice Agencije za privredne registre sa prikazanim spiskom potrebnih dokumenata*

*Izvor:* https://www.apr.gov.rs/upload/Portals/0/privredna%20drustva/2023/JRPPS/JRPPS\_\_\_Predstavnistvo\_stranog\_privrednog\_drustva\_2023.pdf

* 1. Dizajn softverskih funkcija
     1. Tabela preslikavanja primitivnih procesa u softverske funkcije

Poslednja kolona ove tabele je tip aplikacije i nema potrebe da se svuda isto postavlja - web.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Primitivni  poslovni  proces | Radno  mesto | Softverska funkcija  1.prioriteta | Actor softverske funkcije 1.prioriteta | Softverska funkcija  2.prioriteta | Actor  softverske  funkcije 2.  prioriteta | Softverska  funkcija  preduslov | Actor  softverske  funkcije preduslova |
| Uvid u spisak evidencije realizovane proizvodnje | Službenik | Tabelarni prikaz evidencije realizovane proizvodnje sa filterom prema mestu proizvodnje | Službenik | Štampa spiska evidencije realizovane proizvodnje | Službenik | Unos novih evidencija realizovane proizvodnje | Službenik |
| Dodavanje novih evidencija realizovane proizvodnje | Službenik | Unos novih evidencija realizovane proizvodnje | Službenik | Tabelarni prikaz evidencija sa filterom prema mestu proizvodnje  Štampa spiska evidencije realizovane proizvodnje | Službenik | Uvid u kalendar rada i satnicu | Službenik |
| Podnošenje zahteva za evidencije realizovane proizvodnje | Službenik | Unos podataka o proizvođačima koje se prijavljuje za evidenciju realizovane proizvodnje | Službenik | Tabelarni prikaz podnetih zahteva sa filterom prema mestu proizvodnje  Štampa spiska zahteva | Službenik | Uvid u spisak potrebnih dokumenata | Službenik |
| Uvid u listu podnetih zahteva sa mogućnošću brisanja i izmene | Službenik | Tabelarni prikaz podnetih zahteva sa filterom prema mestu proizvodnje | Službenik | Štampa spiska zahteva | Službenik | Unos podataka o proizvođačima koji podnose zahtev za dodavanje nove evidencije | Službenik |
| Štampanje spiska zahteva | Službenik | Štampa spiska zahteva za dodavanje | Službenik | Slanje spiska u vidu emaila | Službenik | Unos podataka o proizođaču koji se prijavljuje za evidenciju | Službenik |
| Prijavljivanje | Službenik | Unos korisničkog imena i šifre radi prijave | Službenik | Tabelarni prikaz podnetih zahteva sa filterom prema mestu proizvodnje | Službenik | Unos podataka o proizvođaču koji se prijavljuje za evidencija realizovane proizvodnje Unos novih evidencija realizovane proizvodnje | Službenik |
| Odabir evidencije realizovane proizvodnje i prijava | Službenik | Unos ličnih podataka | Službenik | Tabelarni prikaz evidencije realizovane proizvodnje sa filterom prema mestu proizvodnje | Službenik | Unos korisničkog imena i šifre radi prijave | Službenik |
| Uvid u listu zahteva | Službenik | Tabelarni prikaz podnetih zahteva sa filterom prema mestu proizvodnje | Službenik | Štampa spiska zahteva | Službenik | Unos podataka o proizvodjačau koji se prijavljuje za evidenciju realizovane proizvodnje | Službenik |
| Pretraživanje zahteva | Službenik | Tabelarni prikaz podnetih zahteva sa filterom prema mestu proizvodnje | Službenik | Štampa spiska zahteva | Službenik | Unos podataka o proizvođaču koji se prijavljuje za evidenciju realizovane proizvodnje | Službenik |

* + 1. USE CASE dijagram softverskih funkcija

Na narednom dijagramu su prikazane samo ključne softverske funkcije iz prethodno date tabele preslikavanja primitivnih poslovnih procesa u skupove softverskih funkcija. Izostavljeni su tabelarni prikazi sa filterom i eksportom u PDF, kao i softverske funkcije koje se odnose na razne nivoe odlučivanja. Naravno, gde je napisan unos, CRUD operacije se podrazumevaju.



*Slika 7 – Use case dijagram*

* + 1. Specifikacija slučaja korišćenja za unos i tabelarni prikaz podataka

Za softversku funkciju unosa podataka o evidencijama realizovane proizvodnje iz tabelarnog prikaza:

A screenshot of a computer

Description automatically generated

*Slika 8 – Preconditions*

A screenshot of a computer

Description automatically generated

*Slika 9 – Action steps*

A screenshot of a computer

Description automatically generated

*Slika 10 – Extension points*

A screenshot of a computer

Description automatically generated

*Slika 11 – Exceptions*

A screenshot of a computer

Description automatically generated

*Slika 12 – Post Conditions*

* 1. Dizajn modela podataka
     1. Konceptualni model podataka



*Slika 13 – Conceptual model*

* + 1. Fizički (relacioni) model podataka



*Slika 14 – Phisical model*

* 1. Test-bazirana specifikacija zahteva korisnika
     1. Test-bazirana specifikacija za ekransku formu unosa

OPIS STRUKTURE EKRANSKE FORME ZA UNOS:

Ekranska forma unosa odnosi se na unos lica koje kreira novu evidenciju. Podaci koji se upisuju su IdProizvodnje, NazivProizvodnje, MestoProizvodnje, BrojProizvodnihObjekata, IdProizvodjaca.

Test bazirana specifikacija definiše ponašanje ekranske forme u određenim situacijama:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ŠTA SE TESTIRA** | | **OBJAŠNJENJE USLOVA** | **ISPRAVNO / NEISPRAVNO** | **ULAZNI PODACI** | **OČEKIVANO PONAŠANJE** |
| **OSNOVNE**  **VALIDACIJE** | |  |  |  |  |
| Da li je popunjeno not null | – | S obzirom da su sva polja not null, moraju biti popunjena. | Ispravno | Popunjena su sva  polja | Sistem vraća poruku uspeha snimanja podataka. |
| neispravno | Neko od polja je ostalo prazno.  (testirati za svako polje, da u svakom test slučaju po 1 polje bude prazno). | U ovom test slucaju kada neko polje ostasne prazno, sistem vraca focus na to polje I menja boju pozadine tog polja za unos (text boxa). |
| Da li  odgovarajudi karakteri | su | Ispravni  karakteri kod imena,  prezimena su slova, pri čemu je prvo slovo veliko, ostala su mala. Za datum postoji kalendarska kontrola, tako da je ovde nemoguć neispravan unos u smislu karaktera. | Ispravno | Uneti sui me, prezime I ime roditelja sa malim slovima, a samo prvo je veliko. | Sistem vraća poruku uspeha snimanja podataka. |
|  |  | neispravno | Uneti sui me, prezime sa:  1. test slučaj -  brojevima I slovima, 2. test slučaj - sa slovima, gde su sva slova mala,  3. test slučaj - slovima gde su sva slova velika. | Sistem vraća poruku o greški. |
| Da li su odgovarajući tipovi podataka | | Ispravni tipovi podataka jesu oni tipovi korišteni u izradi SQL Skripta i za svaku promenljivu tip podataka u programu mora da odgovara tipu u SQL skriptu.  Prilikom unosa neophodno je da uneti podaci takođe odgovaraju zadatom tipu podataka. | Ispravno | Uneti su podaci odgovarajućeg tipa.  Npr: ukoliko je tip podataka zahtevan od strane forme za unos ID-a tipa char, unosi se 10 karaktera. | Sistem vraća poruku uspeha snimanja podataka. |
|  | |  | neispravno | Za isti primer, neispravno bi bilo da se u okviru vremena unose slova ili znakovi. | Sistem vraća poruku o greški |
| Da li je odgovarajuća  dužina unetog podatka | | Ispravna dužina zapisa odgovara unapred određenoj dužini svakog podatka. | Ispravno | Unet je ID od tačno 10 karaktera koliko je predviđeno u okviru izrade forme za unos. | Sistem vraća poruku uspeha snimanja podataka. |
| neispravno | Uneto je manje ili više od 10 karaktera za ID. | Sistem vraća poruku o greški. |
| Da li je jedinstveni  zapis po primarnom ključu | | Ne mogu postojati dva uneta podatka sa istim primarnim ključem. | Ispravno | Unet je jedinstven zapis po kriterijumu da kombinacija ključnih polja nije bila već uneta. | Sistem vraća poruku uspeha  snimanja podataka. |
| neispravno | Unet je zapis već postojećeg ID-a. | Sistem vraća poruku o greški. |
| Da li se kombinacija neključnih obeležja ponavlja | | Neključna obeležja se mogu ponavljati, sve dok je primarni ključ drugačiji. Na primer dve osobe mogu imati isto ime i prezime, ali moraju imati različite jedinstvene matične brojeve. | Ispravno | Unet je jedinstven zapis po kriterijumu da kombinacija neključnih polja nije bila već uneta.  Unet je zapis sa istim neključnim poljima ali primarni ključevi se razlikuju. | Sistem vraća poruku uspeha snimanja podataka. |
|  | | neispravno | Unet je zapis sa istim neključnim poljima, a takođe i sa već postojećim primarnim ključem. | Sistem vraća poruku o greški. |
| Da li su povezani podaci različitih  tabela | | Padajuće liste imaju kao  inicijalnu vrednost  “izaberite…”, ali se smatra ispravnim unosom kada se izabere konkretna vrednost. | Ispravno | U svim padajućim listama su odabrani podaci i ni na jednom mestu nije ostalo nepopunjeno, tj neodabrano. | Sistem vraća poruku uspeha snimanja podataka. |
|  | neispravno | U nekoj padajućoj listi ostalo je  “izaberite…” tj. nije izabran konkretan podatak.  (testirati za svako polje koje ima padajuću listu, da u svakom test slučaju po 1 polje bude sa neizabranom vrednošću). | Sistem vraća focus na padajuću listu u kojoj korisnik nije izabrao vrednost i menja pozadinsku boju padajuće liste. |
| **POSLOVNE**  **VALIDACIJE** | |  |  |  |  |
| Realizovane u poslovnoj logici, a odnose se na realizaciju poslovnih pravila AKO uslov ONDA automatska  akcija ili zabrana | | Poslovna logika unutar forme za unos na osnovu broja evidencije realizacije proizvodnje za datog proizvodjaca | Ispravno |  | Sistem vraća poruku uspeha snimanja podataka. |
|  | | neispravno | Neispravno ponašanje forme za unos je onemogućeno putem ograničenja. |  |

* + 1. Test-bazirana specifikacija za ekransku formu tabelarnog prikaza sa filterom

OPIS STRUKTURE EKRANSKE FORME ZA TABELARNI PRIKAZ SA FILTEROM:

Ekranska forma prikazuje sve unete evidencije, ali i nazive odgovarajućih podataka iz pomoćnih tabela. Filtriranje je po mestu proizvodnje (podaci iz pomoćne tabele, ali koriste se za filtriranje tabelarnog prikaza koji daje kompletne podatke osnovne i pomoćnih tabela).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ŠTA SE TESTIRA** | **OBJAŠNJENJE USLOVA** | **ISPRAVNO /**  **NEISPRAVNO** | **ULAZNI PODACI** | **OČEKIVANO PONAŠANJE** |
| **OSNOVNE**  **VALIDACIJE** |  |  |  |  |
| Da li je popunjeno  – not null | Polje za filter popunjava se minimum jednim karakterom koji je sadržan u podatku koji se traži. | Ispravno | Unet je jedan ili više karaktera koji su sadržani u podatku koji se traži. | Sistem prikazuje tabelu sa podacima koji sadrže unete karaktere. |
| neispravno | Polje za filter je ostavljeno nepopunjeno. | Sistem vraća korisnika na polje za unos filtera. |
| Da li su  odgovarajući karakteri | Ukoliko se filtriraju podaci koji sadrže mala i velika slova, u polju za unos filtera moraju biti uneta mala i velika slova na mestu na kom se nalaze u podacima koji se traže. | Ispravno | Uneti su podaci sa odgovarajućim karakterima.  Npr: prezime je ispisano velikim početnim slovom. | Sistem prikazuje tabelu sa podacima koji sadrže unete karaktere. |
| neispravno | Uneto je prezime sa malim početnim slovom. | Sistem vraća korisnika na polje za unos filtera. |
| Da li su odgovarajući tipovi podataka | Ukoliko se filtriraju podaci koji su predodređeni da budu uneti zadatim tipom podataka, u polju za unos filtera moraju biti uneti podaci koji su u skladu sa tipom podataka koji se traže. | Ispravno | U polje za unos filtera unesti su odgovarajući podaci.  Npr: ID je unet tipom podataka number. | Sistem prikazuje tabelu sa podacima koji sadrže unete karaktere. |
| neispravno | U polje tipa number su uneta slova. | Sistem vraća korisnika na polje za unos filtera. |
| Da li je  odgovarajuća  dužina  unetog  podatka | Ukoliko se filtriraju podaci koji imaju određenu dužinu, u polju za unos filtera moraju biti uneti podaci iste ili manje dužine od onih podataka u kojima su sadržani. | Ispravno | U polje za unos filtera unesti su podaci odgovarajuće dužine.  Npr: ID je unet dužinom od 10 ili manje karaktera, ali tačnim redosledom. | Sistem prikazuje tabelu sa podacima koji sadrže unete karaktere. |
| neispravno | Uneto je više od 10 znakova. | Sistem vraća korisnika na polje za unos filtera. |
| Da li su povezani podaci različitih tabela | S obzirom da se podaci za filter biraju iz padajuće liste, bitno je da se odabere  konkretna vrednost | Ispravno | Test slučaj 1 – izabrana je vrednost iz pomoćne tabele za koju ne postoji podatak u glavnoj  tabeli | Sistem prikazuje praznu tabelu I poruku o tome da za izabranu vrednost ne postoje zapisi koji bi odgovarali tom filteru. |
|  |  | Ispravno | Test slučaj 2 – izabrana je vrednost iz pomoćne tabele za koju postoji podatak u glavnoj tabeli | Sistem prikazuje filtrirane podatke u odnosu na izabranu vrednost. |
|  |  | neispravno | Nije izabrana vrednost iz padajuće tabele, već je ostalo “izaberite” | Sistem prikazuje poruku da nije izabrana vrednost filtera I vraća focus na padajuću listu za izbor filtera. |

* 1. Plan realizacije elemenata kvaliteta softvera
     1. Funkcionalni aspekt kvaliteta softvera
        1. Planirane funkcionalne mogućnosti softvera i profili korisnika
* PROFIL Administrator – unos podataka o korisnicima softvera;
* PROFIL Zaposleni – unos podataka o evidencijama realizovane proizvodnje i proizvođačima;
  + - 1. Plan realizacije osnovne validacije podataka

Na ekranskoj formi za unos proveravaće se ispravnost unetih podataka, i to:

Svi podaci za unos su obavezni: IdProizvodnje, NazivProizvodnje, MestoProizvodnje, BrojProizvodnihObjekata, Proizvodjac

Na ekranskoj formi za tabelarni prikaz lica vršiće se filtriranje podataka prema mestu proizvodnje, kao i prikaz podataka o svim proizvođačima.

* + - 1. Plan realizacije podrške poslovnoj logici

U okviru ekranske forme za unos realizuje se provera poslovnog pravila tipa ograničenja koje glasi:

Na osnovu izveštaja o proizvođaču, provera se da li je mesečna norma za dati proizvod ispunjena i vraća se rezultat.

3. Ograničenja sistematizacije se čitaju iz veb servisa od strane klase Sistematizacija. Ukupan broj proizvoda na mesečnom nivou se razlikuje se od proizvođača do proizvođača kao I tip samog proizvoda, a ti podaci se povlače iz xml fajla.

* + - 1. Plan podrške personalizaciji aplikacije i bezbednosti korišćenja

Personalizacija - Planirano je da se za svakog korisnika odredi profil korisnika (administrator ili radnik kadrovske službe). Svaki profil korisnika ima svoje menije.

Bezbednost – podaci korisnika se čuvaju u sesiji. Na svakoj stranici se proverava da li se pristupa putem prijavljenog korisnika, kroz proveru da li je sesija aktivna i da li ima vrednost. Podaci između formi se razmenjuju putem post metode ili putem sesije.

* + 1. Strukturni aspekt kvaliteta softvera
       1. Plan arhitekture softvera

|  |  |  |
| --- | --- | --- |
| SLOJ | PODSLOJ | TEHNOLOŠKA IMPLEMENTACIJA |
| PREZENTACIONI SLOJ | Korisnički interfejs | PHP, Multi page, HTML, CSS |
|  | Prezentaciona logika | Class Library, DLL, C# |
| SLOJ SERVISA | Veb servis | SOAP, ASMX, C# |
| SLOJ POSLOVNE LOGIKE | Biblioteka klasa poslovne logike koje kroz metode  realizuju poslovna pravila | Class Library, DLL, C#  (za potrebe implementacije svojih metoda poziva servis, klase za rad sa podacima iz baze podataka) |
| SLOJ ZA RAD SA PODACIMA | Klase podataka – izvedene klase u odnosu na tehnološke klase, definišu semantiku kroz konkretne CRUD operacije u metodama I upite, obraćaju se metodama baznih klasa  radi konkretne realizacije | Class Library, DLL, C#   * nasleđuju baznu klasu tabela I prosleđuju upit * realizuju CRUD operacije kroz svoje metode, imaju iste nazive kao tabele iz baze podataka * Za svaku tabelu u bazi podataka postoje 3 klase – pojedinac (get-set), lista pojedinaca, DB klasa (repository, realizuje CRUD operacije |
|  | Tehnološke klase – enkapsuliraju konkretnu tehnologiju konekcije I izvršavanja upita nad konkretnim DBMS, ponašaju se kao bazne klase | Class Library, DLL, C# |
| EKSTERNO | Relaciona baza podataka |  |

Planirana je višeslojna arhitektura softvera sa sledećim slojevima i podslojevima.

UML dijagram komponenti prikazan je na sledećoj slici:

A screenshot of a computer

Description automatically generated

*Slika 15 – Component diagram*

* + - 1. Plan realizacije clean code pristupa

U okviru plana realizacije clean code pridržavaćemo se nekih osnovnih smernica:

1. Postavljanje komentara – ispod naziva klase (CRC kartica – namena, sa kojim klasama sarađuje), ispod zaglavlja metode (namena, objašnjenja parametara)
2. Nazivi – camelCasing - atributi klase, parametri metoda, lokalne promenljive metoda, PascalCasing – naziv klase, naziv metoda, property
3. Primena “this.” u okviru tela metoda, kada se obraćamo atributima ili metodama te klase.
4. Obrisati neiskorišćene promenljive.
5. Koristiti eksplicitni type casting.
6. Na konstruktoru incijalizovati sve attribute.
7. Sve metode dobijaju sve potrebne podatke kroz parametre ili rade nad sopstvenim atributima klase.
8. Ako metoda vraća više vrednosti (izračunatu vrednost, status uspeha itd), koristiti out kod definisanja parametara metode.
9. Try-catch kod osetljivog koda koji radi sa konektovanjem na bazu podataka i izvršavanjem upita.
10. Zagrada za početak bloka “{“ počinje u sledećem redu u odnosu na uslov if, odnosno kao for izraz. If (uslov) { for () {
11. Svaki if treba da ima else (ne podrazumevati ništa).
12. IMPLEMENTACIJA, TESTIRANJE I DOKUMENTOVANJE REALIZOVANOG SOFTVERA
    1. Dokumentovanje implementacije softverskog rešenja
       1. Kratak opis primenjenih tehnologija, programskih jezika i razvojnih alata

Za izradu seminarskog rada korišćen je SQL upitni jezik u okviru Microsoft SQL Management Studio R8 integrisanog okruženja, programski jezik C# i .NET Framework u okviru razvojnog okruženja Visual Studio 2019.

SQL ili Structured Query Language je standardizovani upitni jezik koji se koristi za upravljanje relacionim bazama podataka i manipulaciju njihovim podacima. Koristi se za modifikovanje struktura tabele i indeksa, dodavanje, brisanje i aţuriranje podataka, preuzimanje podskupova podataka iz baze podataka za obradu transakcija i drugo. Upiti i druge SQL operacije imaju oblik komandi napisanih u vidu naredbi, pri ĉemu najĉešće korišćene ukljuĉuju create, select, add, insert, update, delete i alter.

* + 1. Modeli opisa implementacije rešenja
       1. Dijagram komponenti (prikaz arhitekture softvera – višeslojna, MVC, mikroservisi…)

Diagram

Description automatically generated

* + - 1. Djagram klasa



*Slika 16 – Class diagram*

* + 1. Tabelarni prikaz slojeva i podslojeva aplikacije sa ilustracijom konkretnim fajlovima u rešenju

|  |  |  |
| --- | --- | --- |
| SLOJ | PODSLOJ | TEHNOLOŠKA IMPLEMENTACIJA |
| PREZENTACIONI SLOJ | Korisnički interfejs | index.php  unos.php  unosSP.php  Welcome.php  prijava.php  EvidencijaRealizovaneProizvodnjeLista.php  EvidencijaRealizovaneProizvodnjeParametarskaStampa.php  EvidencijaRealizovaneProizvodnjeStampa.php |
|  | Prezentaciona logika | EvidencijaRealizovaneProizvodnjeIzmeni.php  EvidencijaRealizovaneProizvodnjeObrisi.php  evidencijarealizovaneproizvodnjesnimi.php  evidencijarealizovaneproizvodnjesnimiSP.php |
| SLOJ SERVISA | Veb servis |  |
| SLOJ POSLOVNE LOGIKE | Biblioteka klasa poslovne logike koje kroz metode  realizuju poslovna pravila | ProizvođačPL.php  (nije iskorišteno u projektu) |
| SLOJ ZA RAD SA PODACIMA | Klase podataka – izvedene klase u odnosu na tehnološke klase, definišu semantiku kroz konkretne CRUD operacije u metodama I upite, obraćaju se metodama baznih klasa  radi konkretne realizacije | BaznaKonekcija.php  BaznaTabela.php  BaznaTransakcija.php  DBKorisnik.php  DBEvidencijaRealizovaneProizvodnje.php  DBEvidencijaRealizovaneProizvodnjeV.php  DBEvidencijaRealizovaneProizvodnjeSP.php  DBProizvodjac.php |
|  | Tehnološke klase – enkapsuliraju konkretnu tehnologiju konekcije I izvršavanja upita nad konkretnim DBMS, ponašaju se kao bazne klase |
| EKSTERNO | Relaciona baza podataka | Proizvodnja\_Evidencija\_SI2\_2023 |

Procedura i pogledi

CREATE DATABASE `PrivredaZanatsvo` CHARACTER SET utf8 COLLATE utf8\_general\_ci;

create table `PrivredaZanatsvo`.`EvidencijaRealizovaneProizvodnje`

(

IdProizvodnje int NOT NULL PRIMARY KEY,

NazivProizvodnje varchar(100) not null,

MestoProizvodnje varchar(50) not null,

BrojProizvodnihObjekata varchar(100) null,

IdProizvodjaca int not null

);

create table `PrivredaZanatsvo`.`Proizvodjac`

(

Id int NOT NULL PRIMARY KEY,

Prezime varchar(50) not null,

Ime varchar(50) not null,

MestoPrebivalista varchar(50) not null,

UkupanBrojEvidencija int not null

);

alter table `PrivredaZanatsvo`.`EvidencijaRealizovaneProizvodnje` add constraint FK\_IMA foreign key (IdProizvodjaca) references `PrivredaZanatsvo`.`Proizvodjac`(Id) on delete restrict on update cascade;

create table `PrivredaZanatsvo`.`KORISNIK`

(

IDKORISNIKA int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

PREZIME varchar(50) not null,

IME varchar(40) not null,

EMAIL varchar(60) not null,

KORISNICKOIME varchar(30) not null,

SIFRA varchar(30) not null,

URLSLike varchar(250) null,

statusucesca varchar(30) not null

);

INSERT INTO `PrivredaZanatsvo`.`KORISNIK` (PREZIME, IME, EMAIL, KORISNICKOIME, SIFRA, URLSLIKE, STATUSUCESCA) VALUES ('Бановић', 'Петар', 'banovicp@gmail.com', 'banovicp', 'bp', 'admin.jpg', 'admin');

INSERT INTO `PrivredaZanatsvo`.`Proizvodjac` (Id, Prezime, Ime, MestoPrebivalista, UkupanBrojEvidencija) VALUES ('0', 'Петровић','Петар','Зрењанин', 0);

INSERT INTO `PrivredaZanatsvo`.`Proizvodjac` (Id, Prezime, Ime, MestoPrebivalista, UkupanBrojEvidencija) VALUES ('1', 'Марковић','Марко','Нови Сад', 0);

INSERT INTO `PrivredaZanatsvo`.`Proizvodjac` (Id, Prezime, Ime, MestoPrebivalista, UkupanBrojEvidencija) VALUES ('2', 'Илић','Илија','Београд', 0);

|  |
| --- |
| RESENJE - POGLEDI  ------------------------------------------  CREATE VIEW `PrivredaZanatsvo`.`SviPodacioEvidencijiRealizovaneProizvodnje` AS select `EvidencijaRealizovaneProizvodnje`.`IdProizvodnje`, `EvidencijaRealizovaneProizvodnje`.`NazivProizvodnje`, `EvidencijaRealizovaneProizvodnje`.`MestoProizvodnje`, `EvidencijaRealizovaneProizvodnje`.`BrojProizvodnihObjekata`, `Proizvodjac`.`Prezime` as `PrezimeProizvodjaca` from `PrivredaZanatsvo`.`EvidencijaRealizovaneProizvodnje` INNER JOIN `PrivredaZanatsvo`.`Proizvodjac` ON `PrivredaZanatsvo`.`EvidencijaRealizovaneProizvodnje`.`IdProizvodjaca`=`PrivredaZanatsvo`.`Proizvodjac`.`Id`;  RESENJE ZA STORED PROCEDURU- insert into  -----------------------  USE `PrivredaZanatsvo`;  DROP procedure IF EXISTS `DodajEvidenciju`;  DELIMITER $$  USE `PrivredaZanatsvo`$$  CREATE PROCEDURE `DodajEvidenciju` (IN IdProizvodnjeParametar int, IN NazivProizvodnjeParametar varchar(100), IN MestoProizvodnjeParametar varchar(100), IN BrojProizvodnihObjekataParametar int, IN IdProizvodjacaParametar int)  BEGIN  INSERT INTO `EvidencijaRealizovaneProizvodnje` (`IdProizvodnje`, `NazivProizvodnje`, `MestoProizvodnje`, `BrojProizvodnihObjekata`, `IdProizvodjaca`) VALUES (IdProizvodnjeParametar, NazivProizvodnjeParametar, MestoProizvodnjeParametar, BrojProizvodnihObjekataParametar, IdProizvodjacaParametar);  END  $$  DELIMITER ; |

1. * 1. Delovi koda sa objašnjenjima po slojevima
        1. Sloj za rad sa podacima (tehnološke klase, klase podataka, tj. repository)

BaznaKonekcija.php

|  |
| --- |
| <?php  class Konekcija{  // ATRIBUTI  public $konekcijaMYSQL;   //$db\_handle, konekcija na DBMS;  public $konekcijaDB;      //$db\_selected, konekcija na konkretnu bazu;  public $KompletanNazivBazePodataka; // prefiks zbog hostinga i naziv baze  public $VerzijaMYSQLNaredbi; //mysql ili mysqli  //  private $PutanjaNazivFajlaXMLParametriKonekcije;  // parametri konekcije  private $host;  private $korisnik;  private $sifra;  private $prefiks\_baze\_podataka;  private $naziv\_baze\_podataka ;  // METODE  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  private function UcitajVerzijuMYSQLNaredbi()  {      $VerzijaPHP = phpversion();      if ($VerzijaPHP<'7.0.0') // $VerzijaPHP>='5.5.0'      {          $this->VerzijaMYSQLNaredbi="mysql";      }      else      {          $this->VerzijaMYSQLNaredbi="mysqli";      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  private function UcitajParametreKonekcije($PutanjaNazivFajlaXMLParametriKonekcije)  {  // preuzimanje naziva baze podataka iz parametra ucitava se iz XML  //NE VAZI: $xml=simplexml\_load\_file("../parametri/konektovanje.xml")  //NE VAZI: $xml=simplexml\_load\_file("parametri/konektovanje.xml")  // novo, jer su parametri konekcije u istom folderu kao i klasa:  $xml=simplexml\_load\_file($PutanjaNazivFajlaXMLParametriKonekcije) or die("Greska: Ne postoji fajl BaznaParametriKonekcije.xml");  // ocitavanje elemenata XML fajla u promenljive  $this->host=$xml->host;  $this->korisnik=$xml->korisnik;  $this->sifra=$xml->sifra;  // PRILAGODJAVANJE HOSTINGU SISTEMA, PREFIKS BAZE PODATAKA SADRZI PODATKE SA LOGOVANJA NA HOSTING SERVER  $this->prefiks\_baze\_podataka = $xml->prefiks\_baze\_podataka;  $this->naziv\_baze\_podataka = $xml->naziv\_baze\_podataka;  $this->KompletanNazivBazePodataka =$this->prefiks\_baze\_podataka.$this->naziv\_baze\_podataka;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // ------- konstruktor  public function \_\_construct($NovaPutanjaNazivFajlaXMLParametriKonekcije)  {      $this->PutanjaNazivFajlaXMLParametriKonekcije=$NovaPutanjaNazivFajlaXMLParametriKonekcije;      //      $this->UcitajVerzijuMYSQLNaredbi();      $this->UcitajParametreKonekcije($NovaPutanjaNazivFajlaXMLParametriKonekcije);  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function connect()  {  if ($this->VerzijaMYSQLNaredbi=="mysqli")      {          $this->konekcijaDB = mysqli\_connect($this->host, $this->korisnik, $this->sifra, $this->KompletanNazivBazePodataka);      }      else // mysql      {          // ostvarivanje konekcije ka DBMS-u MYSQL          $this->konekcijaMYSQL = mysql\_connect($this->host, $this->korisnik, $this->sifra);            // ostvarivanje konekcije ka bazi podataka          $this->konekcijaDB = mysql\_select\_db($this->KompletanNazivBazePodataka, $this->konekcijaMYSQL);      }    if ($this->konekcijaDB)      {         // dodatak da moze da radi sa UTF8         if ($this->VerzijaMYSQLNaredbi=="mysqli")          {             mysqli\_set\_charset($this->konekcijaDB,"utf8");          }          else //mysql          {              mysql\_query('SET NAMES "utf8"',$this->konekcijaMYSQL);          } // zatvaranje if      } // ako je uspostavljena konekcija $this->konekcijaDB    } // zatvaranje procedure  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function disconnect(){      if ($this->VerzijaMYSQLNaredbi=="mysqli")      {          mysqli\_close($this->konekcijaDB);      }      else // mysql      {          mysql\_close($this->konekcijaMYSQL);      }  }    }  // zatvaranje klase      ?> |

BaznaTabela.php

|  |
| --- |
| <?php  class Tabela{  // atributi  public $OtvorenaKonekcija;  public $NazivBazePodataka;  public $NazivTabele;  public $TipMYSQL;  //  public $Kolekcija;  public $BrojZapisa;  public $PrviRedZapisa;  public $ListaZapisa;  // = array();  // metode  // ------- konstruktor  public function \_\_construct($NovaOtvorenaKonekcija, $NoviNazivTabele){  // podrazumevamo da je otvorena konekcija, a zatvara se spolja      $this->OtvorenaKonekcija = $NovaOtvorenaKonekcija;      $this->NazivBazePodataka = $NovaOtvorenaKonekcija->KompletanNazivBazePodataka;      $this->NazivTabele = $NoviNazivTabele;      $this->TipMYSQL = $NovaOtvorenaKonekcija->VerzijaMYSQLNaredbi;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajSve($KriterijumSortiranja)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        $SQL = "select \* from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` ORDER BY ".$KriterijumSortiranja;        if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);        }      else // mysql      {          $this->Kolekcija = mysql\_query($SQL);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajSvePoUpitu($Upit)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $Upit);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);        }      else // mysql      {          $this->Kolekcija = mysql\_query($Upit);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajSvaPoljaFiltrirano($KriterijumFiltriranja, $KriterijumSortiranja)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        $SQL = "select \* from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja." ORDER BY ".$KriterijumSortiranja;        if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);      }      else // mysql      {          $this->Kolekcija = mysql\_query($SQL);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajPoljaFiltrirano($Polja, $KriterijumFiltriranja, $KriterijumSortiranja)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        $SQL = "select ".$Polja." from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja." ORDER BY ".$KriterijumSortiranja;      if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);      }      else // mysql      {          $this->Kolekcija = mysql\_query($SQL);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function DajVrednostJednogPoljaPrvogZapisa ($NazivTrazenogPolja, $KriterijumFiltriranja, $KriterijumSortiranja)  {      $SQL = "select ".$NazivTrazenogPolja." from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja." ORDER BY ".$KriterijumSortiranja;        if ($this->TipMYSQL=="mysqli")      {          $Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $row=mysqli\_fetch\_array($Kolekcija,MYSQLI\_NUM);  // fetch row          $Vrednost=$row [0];      }      else // mysql      {          $Kolekcija = mysql\_query($SQL);          $Vrednost=mysql\_result($Kolekcija,0,$NazivTrazenogPolja);      }      return $Vrednost;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function PrebaciKolekcijuUListu($Kolekcija) //kolekcija je result  {      $ListaZapisa = array();      if ($this->TipMYSQL=="mysqli")      {          while($RedZapisa = mysqli\_fetch\_array($Kolekcija,MYSQLI\_NUM))              {                  $this->ListaZapisa[] = $RedZapisa;              }      }      else // mysql      {          while($RedZapisa = mysql\_fetch\_array($Kolekcija,MYSQLI\_NUM))              {                  $this->ListaZapisa[] = $RedZapisa;              }      }        return $ListaZapisa;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function DajVrednostPoRednomBrojuZapisaPoRBPolja ($Kolekcija, $RBZapisa, $RBPolja)  {      if ($this->TipMYSQL=="mysqli")      {          $ListaZapisa = array();          $ListaZapisa= $this->PrebaciKolekcijuUListu($Kolekcija);          $RedZapisa=$this->ListaZapisa[$RBZapisa];          $Vrednost=$RedZapisa [$RBPolja];      }      else // mysql      {          $Vrednost=mysql\_result($Kolekcija,$RBZapisa, $RBPolja);   //$NazivPolja);      }      return $Vrednost;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function PostojiZapis($KriterijumFiltriranja)   // da li postoji, da li je jedinstven itd.  {      // ne puni kolekciju atribut, vec samo okalnu promenljivu da bi vratio da li postoji zapis        $SQL = "SELECT \* FROM `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja;   //PRIMER: KORISNICKOIME='".$korisnickoime."' AND SIFRA='".$sifra."'";      if ($this->TipMYSQL=="mysqli")      {          $KolekcijaLokalna = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $BrojZapisaLokalna = mysqli\_num\_rows($KolekcijaLokalna);        }      else // mysql      {          $KolekcijaLokalna = mysql\_query($SQL);          $BrojZapisaLokalna = mysql\_num\_rows($KolekcijaLokalna);      }      if ($BrojZapisaLokalna>0)      {          $postoji=true;      }      else      {          $postoji=false;      }      return $postoji;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function IzvrsiAktivanSQLUpit($AktivanSQLUpit) {      //PRIMER: $AktivanSQLUpit = "INSERT INTO `".$this->bazapodataka."`.`KORISNIK` (IME, PREZIME, KORISNICKOIME, SIFRA, EMAIL, URLSlike, statusucesca, DatumRodjenja) VALUES ('$Ime', '$Prezime', '$KorisnickoIme', '$Sifra', '$Email', '', '$Status', '$DatumRodjenja')";      // ---------------      if ($this->TipMYSQL=="mysqli")      {          $retval = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $AktivanSQLUpit);          $Greska = mysqli\_error($this->OtvorenaKonekcija->konekcijaDB);      }      else // mysql      {          $retval = mysql\_query( $AktivanSQLUpit, $this->OtvorenaKonekcija->konekcijaMYSQL);          $Greska = mysql\_error();      }      return $Greska;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  }  ?> |

BaznaTransakcija.php

|  |
| --- |
| <?php  class Transakcija{  // atributi  private $OtvorenaKonekcija;  private $VerzijaMySQLNaredbi;  // metode  // ------- konstruktor  public function \_\_construct($NovaOtvorenaKonekcija)  {      $this->OtvorenaKonekcija=$NovaOtvorenaKonekcija;      $this->VerzijaMySQLNaredbi=$NovaOtvorenaKonekcija->VerzijaMYSQLNaredbi;  }  public function DajVerzijuMySQL()  {      return $this->VerzijaMySQLNaredbi;  }  public function ZapocniTransakciju()  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"SET AUTOCOMMIT=0");          mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"START TRANSACTION");      }      else      {          mysql\_query("SET AUTOCOMMIT=0");          mysql\_query("START TRANSACTION");      }    } // zatvaranje procedure  public function ProveriGresku()  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          $greska= mysqli\_error($this->OtvorenaKonekcija->konekcijaDB);        }      else      {          $greska= mysql\_error();        }        return $greska;    } // zatvaranje procedure  public function PonistiTransakciju()  // samo ako poslednje izvrsavanje aktivnog SQL upita ima gresku  //onda radi rollback  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"ROLLBACK");      }      else      {          mysql\_query("ROLLBACK");      }    } // zatvaranje procedure  public function ZavrsiTransakciju($UtvrdjenaGreska)  // samo ako poslednje izvrsavanje aktivnog SQL upita ima gresku  //onda radi rollback  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          if (empty($UtvrdjenaGreska))              {                  mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"COMMIT");              }              else              {                  mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"ROLLBACK");              }      }      else      {          if (empty($UtvrdjenaGreska))              {              mysql\_query("COMMIT");              }          else              {              mysql\_query("ROLLBACK");              }      }    } // zatvaranje procedure    }  // zatvaranje klase      ?> |

DBEvidencijaRealizovaneProizvodnje.php

|  |
| --- |
| <?php  class DBEvidencijaRealizovaneProizvodnje extends Tabela  {  // ATRIBUTI  private $bazapodataka;  private $UspehKonekcijeNaDBMS;  //  public $IdProizvodnje;  public $NazivProizvodnje;  public $MestoProizvodnje;  public $BrojProizvodnihObjekata;  public $IdProizvodjaca;  // METODE  // konstruktor  public function DajKolekcijuSvihEvidencijaProizvodnje()  {  $SQL = "select \* from `EvidencijaRealizovaneProizvodnje` ORDER BY MestoProizvodnje ASC";  $this->UcitajSvePoUpitu($SQL); // puni atribut bazne klase Kolekcija  return $this->Kolekcija; // uzima iz baznek klase vrednost atributa  }  public function UcitajEvidencijuPoIdProizvodnje($IdProizvodnjeParametar)  {  $SQL = "select \* from `EvidencijaRealizovaneProizvodnje` where `IdProizvodnje`='".$IdProizvodnjeParametar."'";  $this->UcitajSvePoUpitu($SQL); // puni atribut bazne klase Kolekcija  // raspolazemo sa:  // $Kolekcija;  //  $BrojZapisa;  }  public function DodajNovuEvidenciju()  {      $SQL = "INSERT INTO `EvidencijaRealizovaneProizvodnje`(IdProizvodnje, NazivProizvodnje, MestoProizvodnje, BrojProizvodnihObjekata, IdProizvodjaca) VALUES ('$this->IdProizvodnje', '$this->NazivProizvodnje', '$this->MestoProizvodnje', '$this->BrojProizvodnihObjekata', '$this->IdProizvodjaca')";      $greska=$this->IzvrsiAktivanSQLUpit($SQL);        return $greska;  }    public function ObrisiEvidenciju($IdProizvodnjeZaBrisanje)  {      $SQL = "DELETE FROM `EvidencijaRealizovaneProizvodnje` WHERE IdProizvodnje='".$IdProizvodnjeZaBrisanje."'";      $greska=$this->IzvrsiAktivanSQLUpit($SQL);        return $greska;  }  // TO DO  public function IzmeniEvidenciju($StariIdProizvodnje, $idProizvodnje, $nazivProizvodnje, $mestoProizvodnje, $brojProizvodnihObjekata, $idProizvodjaca)  {      $SQL = "UPDATE `EvidencijaRealizovaneProizvodnje` SET IdProizvodnje='".$idProizvodnje."', NazivProizvodnje='".$nazivProizvodnje."',MestoProizvodnje='".$mestoProizvodnje."', BrojProizvodnihObjekata='".$brojProizvodnihObjekata."', IdProizvodjaca='".$idProizvodjaca."' WHERE IdProizvodnje='".$StariIdProizvodnje."'";      $greska=$this->IzvrsiAktivanSQLUpit($SQL);        return $greska;  }  // ostale metode    }  ?> |

DBEvidencijaRealizovaneProizvodnjeSP.php

|  |
| --- |
| <?php  class DBEvidencijaRealizovaneProizvodnje extends Tabela  // rad sa stored procedurom za snimanje novog studenta  {  // ATRIBUTI  private $bazapodataka;  private $UspehKonekcijeNaDBMS;  //  public $IdProizvodnje;  public $NazivProizvodnje;  public $MestoProizvodnje;  public $BrojProizvodnihObjekata;  public $IdProizvodjaca;  // METODE  // konstruktor  public function DodajEvidenciju()  {              $GreskarezultatPar1 = $this->IzvrsiAktivanSQLUpit ("SET @IdProizvodnjeParametar='".$this->IdProizvodnje."'");            $GreskarezultatPar2 = $this->IzvrsiAktivanSQLUpit ("SET @NazivProizvodnjeParametar='".$this->NazivProizvodnje."'");            $GreskarezultatPar3 = $this->IzvrsiAktivanSQLUpit ("SET @MestoProizvodnjeParametar='".$this->MestoProizvodnje."'");            $GreskarezultatPar4 =  $this->IzvrsiAktivanSQLUpit ("SET @BrojProizvodnihObjekataParametar='".$this->BrojProizvodnihObjekata."'");            $GreskarezultatPar5 = $this->IzvrsiAktivanSQLUpit (  "SET @IdProizvodjacaParametar='".$this->IdProizvodjaca."'");            $GreskarezultatCall = $this->IzvrsiAktivanSQLUpit ( "CALL `DodajEvidenciju`(@IdProizvodnjeParametar, @NazivProizvodnjeParametar, @MestoProizvodnjeParametar, @BrojProizvodnihObjekataParametar,@IdProizvodjacaParametar);");          $greska=$GreskarezultatPar1.$GreskarezultatPar2.$GreskarezultatPar3.$GreskarezultatPar4.$GreskarezultatPar5.$GreskarezultatCall;      return $greska;  }  }  ?> |

DBEvidencijaRealizovaneProizvodnjeV.php

|  |
| --- |
| <?php  class DBEvidencijaRealizovaneProizvodnje extends Tabela  // rad sa pogledom  {  // METODE  // konstruktor  public function DajSvePodatkeOEvidencijiProizvodnje($filterParametar)  {      if (isset($filterParametar))      {          // nad pogledom se moze dodati filter, jer se pogled koristi kao da je tabela          $upit="select \* from `".$this->NazivBazePodataka."`.`SviPodacioEvidencijiRealizovaneProizvodnje` where `MestoProizvodnje`='".$filterParametar."'";      }      else      {          $upit="select \* from `".$this->NazivBazePodataka."`.`SviPodacioEvidencijiRealizovaneProizvodnje`";      }      $this->UcitajSvePoUpitu($upit);      // sada raspolazemo sa:      //$this->Kolekcija      //$this->BrojZapisa  }  }  ?> |

DBKorisnik.php

|  |
| --- |
| <?php  class DBKorisnik extends Tabela{  // ATRIBUTI  public $IDKorisnika; // auto increment u bazi podataka  public $Prezime;  public $Ime;  public $Email;  public $KorisnickoIme;  public $Sifra;  public $Stari\_IDKorisnika; // potrebno zbog izmene  // metode  // ------- konstruktor - uzima se iz klase roditelja - Tabela  // ------- preostale metode  public function UcitajSveKorisnike()  {          $SQL = "select \* from korisnik";          $this->UcitajSvePoUpitu($SQL);  } // kraj metode  public function DaLiPostojiKorisnik($loginusername,$loginpassword)  {      $postoji="";      $SQLKorisnik = "SELECT \* FROM `".$this->OtvorenaKonekcija->KompletanNazivBazePodataka."`.`korisnik` WHERE KORISNICKOIME='".$loginusername."' AND SIFRA='".$loginpassword."'";      $this->UcitajSvePoUpitu($SQLKorisnik);      // raspolazemo sa kolekcijom i brojem zapisa nakon ucitaj sve po upitu        // NEPOTREBNO - $this->PrebaciKolekcijuUListu($this->Kolekcija);      if ($this->BrojZapisa>0)      {          $postoji="DA";      }      else      {          $postoji="NE";      }      return $postoji;  }  public function DajImePrijavljenogKorisnika($loginusername,$loginpassword)  {      $korisnik="";      $SQLKorisnik = "SELECT \* FROM `".$this->OtvorenaKonekcija->KompletanNazivBazePodataka."`.`KORISNIK` WHERE KORISNICKOIME='".$loginusername."' AND SIFRA='".$loginpassword."'";      $this->UcitajSvePoUpitu($SQLKorisnik);      $this->PrebaciKolekcijuUListu($this->Kolekcija);      if ($this->BrojZapisa>0)      {          // postoji zapis          foreach ($this->ListaZapisa as $VrednostCvoraListe)          {              $ime=$VrednostCvoraListe[2];            }      }      else      {          $ime='NEPOZNAT KORISNIK';      }      return $ime;  }  public function DajPrezimePrijavljenogKorisnika($loginusername,$loginpassword)  {      $korisnik="";      $SQLKorisnik = "SELECT \* FROM `".$this->OtvorenaKonekcija->KompletanNazivBazePodataka."`.`KORISNIK` WHERE KORISNICKOIME='".$loginusername."' AND SIFRA='".$loginpassword."'";      $this->UcitajSvePoUpitu($SQLKorisnik);      $this->PrebaciKolekcijuUListu($this->Kolekcija);      if ($this->BrojZapisa>0)      {          // postoji zapis          foreach ($this->ListaZapisa as $VrednostCvoraListe)          {              $prez=$VrednostCvoraListe[1];            }      }      else      {          $prez='NEPOZNAT KORISNIK';      }      return $prez;  }  public function DajImePrezimePrijavljenogKorisnika($loginusername,$loginpassword)  {      $korisnik="";      $SQLKorisnik = "SELECT \* FROM `".$this->OtvorenaKonekcija->KompletanNazivBazePodataka."`.`KORISNIK` WHERE KORISNICKOIME='".$loginusername."' AND SIFRA='".$loginpassword."'";      $this->UcitajSvePoUpitu($SQLKorisnik);      $this->PrebaciKolekcijuUListu($this->Kolekcija);      if ($this->BrojZapisa>0)      {          // postoji zapis          foreach ($this->ListaZapisa as $VrednostCvoraListe)          {              $prez=$VrednostCvoraListe[1];              $ime=$VrednostCvoraListe[2];              $korisnik=$prez.' '.$ime;          }      }      else      {          $korisnik='NEPOZNAT KORISNIK';      }      return $korisnik;  }  public function DajIDPrijavljenogKorisnika($loginusername,$loginpassword)  {      $id=0;      $SQLKorisnik = "SELECT \* FROM `".$this->OtvorenaKonekcija->KompletanNazivBazePodataka."`.`KORISNIK` WHERE KORISNICKOIME='".$loginusername."' AND SIFRA='".$loginpassword."'";      $this->UcitajSvePoUpitu($SQLKorisnik);      $this->PrebaciKolekcijuUListu($this->Kolekcija);      if ($this->BrojZapisa>0)      {          // postoji zapis          foreach ($this->ListaZapisa as $VrednostCvoraListe)          {              $id=$VrednostCvoraListe[0];          }      }      // else - ostaje 0      return $id;  }  public function SnimiNovo()  {      $AktivanSQLUpit = "";      $this->IzvrsiAktivanSQLUpit($AktivanSQLUpit);  }  // brisanje  public function Obrisi()  {      $AktivanSQLUpit = "DELETE from ";      $this->IzvrsiAktivanSQLUpit($AktivanSQLUpit);  }  public function ObrisiSve()  {      $AktivanSQLUpit = "DELETE from ";      $this->IzvrsiAktivanSQLUpit($AktivanSQLUpit);  }  public function IzmeniVrednostPolja()  {      // transformisemo datum u formu pogodnu za insert into      //  $DatumskaVrednost=date\_create($this->Datum\_PoslednjePromene);      //  $DatumUnosa=date\_format($DatumskaVrednost,"Y-m-d");      // konacan upit      $AktivanSQLUpit = "UPDATE  SET " ;      $this->IzvrsiAktivanSQLUpit($AktivanSQLUpit);  } // kraj metode  } // kraj klase  ?> |

DBProizvodjac.php

|  |
| --- |
| <?php  class DBProizvodjac extends Tabela  {  // ATRIBUTI  private $bazapodataka;  private $UspehKonekcijeNaDBMS;  //  public $Id;  public $Prezime;  public $Ime;  public $MestoPrebivalista;  public $UkupanBrojEvidencija;  // METODE  // konstruktor  public function UcitajKolekcijuSvihProizvodjaca()  {  $SQL = "select \* from `Proizvodjac` ORDER BY Prezime ASC";  $this->UcitajSvePoUpitu($SQL); // puni atribut bazne klase Kolekcija  //return $this->Kolekcija; // uzima iz baznek klase vrednost atributa  }  // inkrement evidencija za proizvodjaca koji se nalazi u bazi  public function InkrementirajBrojEvidencijaProizvodnje($IDProizvodjaca)  {        $KriterijumFiltriranja="Id='".$IDProizvodjaca."'";      $StaraVrednostUkBrEvidencija=$this->DajVrednostJednogPoljaPrvogZapisa ('UkupanBrojEvidencija', $KriterijumFiltriranja, 'UkupanBrojEvidencija');        // izracunavanje nove vrednosti      $NovaVrednostUkBrEvidencija=$StaraVrednostUkBrEvidencija + 1;        // izvrsavanje izmene      $SQL = "UPDATE `".$this->NazivBazePodataka."`.`Proizvodjac` SET UkupanBrojEvidencija=".$NovaVrednostUkBrEvidencija." WHERE Id='".$IDProizvodjaca."'";      $greska= $this->IzvrsiAktivanSQLUpit($SQL);      return $greska;        }  }  ?> |

* + - 1. Sloj servisa i sloj poslovne logike sa opisom poslovnog pravila, kodom klase poslovne logike, strukturom i sadržajem datoteka u kojima su poslovna ograničenja

BaznaParametriKonekcije.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <konekcija>  <host>localhost</host>  <korisnik>root</korisnik>  <sifra></sifra>  <prefiks\_baze\_podataka></prefiks\_baze\_podataka>  <naziv\_baze\_podataka>PrivredaZanatsvo</naziv\_baze\_podataka>  </konekcija> |

* + - 1. Sloj prezentacione logike - sa klasama koje odgovaraju ekranskim formama
         1. Klase za pripremu prikaza podataka - klase sa metodama za preuzimanje i formatiranje podataka iz baze podataka kroz poziv klasa sloja za rad sa podacima

EvidencijaRealizovaneProizvodnjeIzmeniForm.php

|  |
| --- |
| <?php  // OVO JE SUSTINSKO ODJAVLJIVANJE KORISNIKA         session\_start();           // citanje vrednosti iz sesije         $korisnik=$\_SESSION["korisnik"];          // ako nije prijavljen korisnik, vraca ga na pocetnu stranicu                  if (!isset($korisnik))                  {                      header ('Location:index.php');                  }  // REALIZACIJA CITANJA hidden polja za filter radi pristupa, cita sa StudentiLista  $StariIdProizvonjeZaIzmenu=$\_POST['IdProizvodnje'];  // KONEKTOVANJE NA BAZU      require "klase/BaznaKonekcija.php";      $KonekcijaObject = new Konekcija("klase/BaznaParametriKonekcije.xml");      $KonekcijaObject->connect();      $db\_handle = $KonekcijaObject->konekcijaMYSQL;      $bazapodataka=$KonekcijaObject->KompletanNazivBazePodataka;      $UspehKonekcijeNaBazu=$KonekcijaObject->konekcijaDB;        require "klase/BaznaTabela.php";        // IZDVAJANJE PODATAKA KORISTECI KLASU SMER      require "klase/DBProizvodjac.php";      $ProizvodjacObject = new DBProizvodjac($KonekcijaObject, "Proizvodjac");      $ProizvodjacObject->UcitajKolekcijuSvihProizvodjaca();      $KolekcijaZapisa= $ProizvodjacObject->Kolekcija;      $UkupanBrojZapisa= $ProizvodjacObject->BrojZapisa;      // PREUZIMANJE STARIH VREDNOSTI ZA IZABRANOG STUDENTA      require "klase/DBEvidencijaRealizovaneProizvodnje.php";      $EvidencijaObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject, 'EvidencijaRealizacijeProizvodnje');      $EvidencijaObject->UcitajEvidencijuPoIdProizvodnje($StariIdProizvonjeZaIzmenu);      $KolekcijaZapisaEvidencija= $EvidencijaObject->Kolekcija;      $UkupanBrojZapisaEvidencija = $EvidencijaObject->BrojZapisa;        if ($UkupanBrojZapisaEvidencija>0)      {          $row=0;  // prvi i jedini red ima taj id          $StariIdProizvodnje = $EvidencijaObject->DajVrednostPoRednomBrojuZapisaPoRBPolja ($KolekcijaZapisaEvidencija, $row, 0);          $StariNazivProizvodnje =$EvidencijaObject->DajVrednostPoRednomBrojuZapisaPoRBPolja ($KolekcijaZapisaEvidencija, $row, 1);//mysql\_result($result,$row,"REGISTARSKIBROJ");          $StaroMestoProizvodnje=$EvidencijaObject->DajVrednostPoRednomBrojuZapisaPoRBPolja ($KolekcijaZapisaEvidencija, $row, 2);          $StariBrojProizvodnihObjekata=$EvidencijaObject->DajVrednostPoRednomBrojuZapisaPoRBPolja ($KolekcijaZapisaEvidencija, $row, 3);          $StariIdProizvodjaca=$EvidencijaObject->DajVrednostPoRednomBrojuZapisaPoRBPolja ($KolekcijaZapisaEvidencija, $row, 4);        }  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција реализоване проиводње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  <script src="JavaScript/provera.js"> </script>  </head>  <body>  <!-----VELIKA TABELA KOJA SADRZI SVE---->  <!-----10% SADRZAJ 10%---->  <table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" style="border-spacing: 0;">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljewelcome.php';?>  <!-------------------------- DONJI DEO  ------->  <tr style="padding:0px;">  <!-----LEVO PRAZNINA---->  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle" style="width:80%; padding:0" >  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:15%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevoadmin.php';?>  </td>  <td style="width:1%;">  </td>  <td style="width:80%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnoEvidencijaRealizovaneProizvodnjeIzmeniForm.php';?>  </td>  <td style="width:1%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <!-----DESNO PRAZNINA---->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr style="padding:0px;">  <td style="width:10%;"></td>  <td align="center" valign="middle"></td>  <td style="width:10%;"></td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

EvidencijaRealizovaneProizvodnjeIzmeni.php

|  |
| --- |
| <?php            session\_start();         // citanje vrednosti iz sesije - da bismo uvek proverili da li je to prijavljeni korisnik         // citanje vrednosti iz sesije - da bismo uvek proverili da li je to prijavljeni korisnik         $korisnik=$\_SESSION["korisnik"];          // ako nije prijavljen korisnik, vraca ga na pocetnu stranicu                  if (!isset($korisnik))                  {                      header ('Location:index.php');                  }              // -------------------------------------          // UPLOAD FAJLA SLIKE         // preuzimanje vrednosti sa forme         $idProizvodnje=$\_POST['idProizvodnje'];         $StariIdProizvodnje=$\_POST['StariIdProizvodnje'];         $nazivProizvodnje=$\_POST['nazivProizvodnje'];         $mestoProizvodnje=$\_POST['mestoProizvodnje'];         $brojProizvodnihObjekata=$\_POST['brojProizvodnihObjekata'];         if (isset($\_POST['idProizvodjaca']))         {          $idProizvodjaca=$\_POST['idProizvodjaca'];         }         else // ako nije nista promenjeno         {          $StariIdProizvodjaca=$\_POST['idProizvodjaca'];          $idProizvodjaca=$StariIdProizvodjaca;         }           // koristimo klasu za poziv procedure za konekciju          require "klase/BaznaKonekcija.php";          require "klase/BaznaTabela.php";          $KonekcijaObject = new Konekcija('klase/BaznaParametriKonekcije.xml');          $KonekcijaObject->connect();          if ($KonekcijaObject->konekcijaDB) // uspesno realizovana konekcija ka DBMS i bazi podataka          {              require "klase/DBEvidencijaRealizovaneProizvodnje.php";              $EvidencijaObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject, 'EvidencijaRealizovaneProizvodnje');              $greska=$EvidencijaObject->IzmeniEvidenciju($StariIdProizvodnje, $idProizvodnje, $nazivProizvodnje, $mestoProizvodnje, $brojProizvodnihObjekata, $idProizvodjaca);          }          else          {              echo "Nije uspostavljena konekcija ka bazi podataka!";          }        $KonekcijaObject->disconnect();        // prikaz uspeha aktivnosti      //echo "Ukupno procesirano $retval zapisa";      //echo "Greska $greska";      header ('Location:EvidencijaRealizovaneProizvodnjeLista.php');            ?> |

EvidencijaRealizovaneProizvodnjeObrisi.php

|  |
| --- |
| <?php            session\_start();         // citanje vrednosti iz sesije - da bismo uvek proverili da li je to prijavljeni korisnik         $korisnik=$\_SESSION["korisnik"];          // ako nije prijavljen korisnik, vraca ga na pocetnu stranicu                  if (!isset($korisnik))                  {                      header ('Location:index.php');                  }           // preuzimanje vrednosti sa forme         $IdZaBrisanje=$\_POST['RedniBrojPrijave'];          // koristimo klasu za poziv procedure za konekciju      require "klase/BaznaKonekcija.php";      require "klase/BaznaTabela.php";      $KonekcijaObject = new Konekcija('klase/BaznaParametriKonekcije.xml');      $KonekcijaObject->connect();      if ($KonekcijaObject->konekcijaDB) // uspesno realizovana konekcija ka DBMS i bazi podataka      {          require "klase/DBEvidencijaRealizovaneProizvodnje.php";          $EvidencijaObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject, 'EvidencijaRealizovaneProizvodnje');          $greska=$EvidencijaObject->ObrisiEvidenciju($IdProizvodnjeZaBrisanje);      }        $KonekcijaObject->disconnect();        // prikaz uspeha aktivnosti      //echo "Ukupno procesirano $retval zapisa";      //echo "Greska $greska";      header ('Location:EvidencijaRealizovaneProizvodnjeLista.php');            ?> |

evidencijarealizovaneproizvodnjesnimi.php

|  |
| --- |
| <?php            //session\_start();         // citanje vrednosti iz sesije - da bismo uvek proverili da li je to prijavljeni korisnik         //$idkorisnika=$\_SESSION["idkorisnika"];              // -------------------------------------          // UPLOAD FAJLA SLIKE           // preuzimanje vrednosti sa forme         $IdProizvodnje=$\_POST['idProizvodnje'];         $NazivProizvodnje=$\_POST['nazivProizvodnje'];         $MestoProizvodnje=$\_POST['mestoProizvodnje'];         $BrojProizvodnihObjekata=$\_POST['brojProizvodnihObjekata'];         $IdProizvodjaca =$\_POST['idProizvodjaca'];           //KONEKCIJA KA SERVERU    // koristimo klasu za poziv procedure za konekciju      require "klase/BaznaKonekcija.php";      require "klase/BaznaTabela.php";      $KonekcijaObject = new Konekcija('klase/BaznaParametriKonekcije.xml');      $KonekcijaObject->connect();      if ($KonekcijaObject->konekcijaDB) // uspesno realizovana konekcija ka DBMS i bazi podataka      {          //echo "USPESNA KONEKCIJA";          require "klase/BaznaTransakcija.php";          $TransakcijaObject = new Transakcija($KonekcijaObject);          $TransakcijaObject->ZapocniTransakciju();            require "klase/DBEvidencijaRealizovaneProizvodnje.php";          $EvidencijaObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject, 'EvidencijaRealizovaneProizvodnje');          $EvidencijaObject->IdProizvodnje=$IdProizvodnje;          $EvidencijaObject->NazivProizvodnje=$NazivProizvodnje;          $EvidencijaObject->MestoProizvodnje=$MestoProizvodnje;          $EvidencijaObject->BrojProizvodnihObjekata=$BrojProizvodnihObjekata;          $EvidencijaObject->IdProizvodjaca=$IdProizvodjaca;          $greska1=$EvidencijaObject->DodajNovuEvidenciju();            // inkrement broja studenata kroz klasu DBSmer          require "klase/DBProizvodjac.php";          $EObject = new DBProizvodjac($KonekcijaObject, 'Proizvodjac');          $greska2=$EObject->InkrementirajBrojEvidencijaProizvodnje($IdProizvodjaca);            // zatvaranje transakcije          //$UtvrdjenaGreska=$greska1 or $greska2;          $UtvrdjenaGreska=$greska1.$greska2;          $TransakcijaObject->ZavrsiTransakciju($UtvrdjenaGreska);            } // od if db selected        // ZATVARANJE KONEKCIJE KA DBMS        $KonekcijaObject->disconnect();        // prikaz uspeha aktivnosti        if ($UtvrdjenaGreska) {      echo "Greska $UtvrdjenaGreska";       }       else       {          //echo "Snimljeno!";          header ('Location:EvidencijaRealizovaneProizvodnjeLista.php');       }            ?> |

evidencijarealizovaneproizvodnjesnimiSP.php

|  |
| --- |
| <?php            //session\_start();         // citanje vrednosti iz sesije - da bismo uvek proverili da li je to prijavljeni korisnik         //$idkorisnika=$\_SESSION["idkorisnika"];              // -------------------------------------          // UPLOAD FAJLA SLIKE           // preuzimanje vrednosti sa forme         $IdProizvodnje=$\_POST['idProizvodnje'];         $NazivProizvodnje=$\_POST['nazivProizvodnje'];         $MestoProizvodnje=$\_POST['mestoProizvodnje'];         $BrojProizvodnihObjekata=$\_POST['brojProizvodnihObjekata'];         $IdProizvodjaca =$\_POST['idProizvodjaca'];           //KONEKCIJA KA SERVERU    // koristimo klasu za poziv procedure za konekciju      require "klase/BaznaKonekcija.php";      require "klase/BaznaTabela.php";      $KonekcijaObject = new Konekcija('klase/BaznaParametriKonekcije.xml');      $KonekcijaObject->connect();      if ($KonekcijaObject->konekcijaDB) // uspesno realizovana konekcija ka DBMS i bazi podataka      {          //echo "USPESNA KONEKCIJA";          require "klase/BaznaTransakcija.php";          $TransakcijaObject = new Transakcija($KonekcijaObject);          $TransakcijaObject->ZapocniTransakciju();            require "klase/DBEvidencijaRealizovaneProizvodnjeSP.php";          $EvidencijaObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject, 'EvidencijaRealizovaneProizvodnje');          $EvidencijaObject->IdProizvodnje=$IdProizvodnje;          $EvidencijaObject->NazivProizvodnje=$NazivProizvodnje;          $EvidencijaObject->MestoProizvodnje=$MestoProizvodnje;          $EvidencijaObject->BrojProizvodnihObjekata=$BrojProizvodnihObjekata;          $EvidencijaObject->IdProizvodjaca=$IdProizvodjaca;          $greska1=$EvidencijaObject->DodajEvidenciju();            // inkrement broja studenata kroz klasu DBSmer          require "klase/DBProizvodjac.php";          $ProizvodjacObject = new DBProizvodjac($KonekcijaObject, 'Proizvodjac');          $greska2=$ProizvodjacObject->InkrementirajBrojEvidencijaProizvodnje($IdProizvodjaca);            // zatvaranje transakcije          //$UtvrdjenaGreska=$greska1 or $greska2;          $UtvrdjenaGreska=$greska1.$greska2;          $TransakcijaObject->ZavrsiTransakciju($UtvrdjenaGreska);            } // od if db selected        // ZATVARANJE KONEKCIJE KA DBMS        $KonekcijaObject->disconnect();        // prikaz uspeha aktivnosti        if ($UtvrdjenaGreska) {      echo "Greska $UtvrdjenaGreska";       }       else       {          //echo "Snimljeno!";          header ('Location:EvidencijaRealizovaneProizvodnjeLista.php');       }            ?> |

* + - * 1. Klase za preuzimanje podataka – klase za preuzimanje komandi i podataka sa korisničkog interfejsa i osnovne validacije

unos.php

|  |
| --- |
| <?php  // OVO JE SUSTINSKO ODJAVLJIVANJE KORISNIKA         session\_start();           // citanje vrednosti iz sesije         $korisnik=$\_SESSION["korisnik"];          // ako nije prijavljen korisnik, vraca ga na pocetnu stranicu                  if (!isset($korisnik))                  {                      header ('Location:index.php');                  }  // REALIZACIJA CITANJA SVIH I FILTRIRANIH PODATAKA  // KONEKTOVANJE NA BAZU      require "klase/BaznaKonekcija.php";      $KonekcijaObject = new Konekcija("klase/BaznaParametriKonekcije.xml");      $KonekcijaObject->connect();      $db\_handle = $KonekcijaObject->konekcijaMYSQL;      $bazapodataka=$KonekcijaObject->KompletanNazivBazePodataka;      $UspehKonekcijeNaBazu=$KonekcijaObject->konekcijaDB;        //echo "KONEKCIJA-BAZA:".$bazapodataka;        // IZDVAJANJE PODATAKA KORISTECI KLASU SMER      require "klase/BaznaTabela.php";      require "klase/DBProizvodjac.php";      $EvidencijaObject = new DBProizvodjac($KonekcijaObject, "Proizvodjac");      $EvidencijaObject->UcitajKolekcijuSvihProizvodjaca();      $KolekcijaZapisa= $EvidencijaObject->Kolekcija;      $UkupanBrojZapisa= $EvidencijaObject->BrojZapisa;  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција реализоване производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  <script src="JavaScript/provera.js"> </script>  </head>  <body>  <!-----VELIKA TABELA KOJA SADRZI SVE---->  <!-----10% SADRZAJ 10%---->  <table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" style="border-spacing: 0;">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljewelcome.php';?>  <!-------------------------- DONJI DEO  ------->  <tr style="padding:0px;">  <!-----LEVO PRAZNINA---->  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle" style="width:80%; padding:0" >  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:17%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevoadmin.php';?>  </td>  <td style="width:1%;">  </td>  <td style="width:80%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnounos.php';?>  </td>  <td style="width:1%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <!-----DESNO PRAZNINA---->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr style="padding:0px;">  <td style="width:10%;"></td>  <td align="center" valign="middle"></td>  <td style="width:10%;"></td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

unosSP.php

|  |
| --- |
| <?php  // OVO JE SUSTINSKO ODJAVLJIVANJE KORISNIKA         session\_start();           // citanje vrednosti iz sesije         $korisnik=$\_SESSION["korisnik"];          // ako nije prijavljen korisnik, vraca ga na pocetnu stranicu                  if (!isset($korisnik))                  {                      header ('Location:index.php');                  }  // REALIZACIJA CITANJA SVIH I FILTRIRANIH PODATAKA  // KONEKTOVANJE NA BAZU      require "klase/BaznaKonekcija.php";      $KonekcijaObject = new Konekcija("klase/BaznaParametriKonekcije.xml");      $KonekcijaObject->connect();      $db\_handle = $KonekcijaObject->konekcijaMYSQL;      $bazapodataka=$KonekcijaObject->KompletanNazivBazePodataka;      $UspehKonekcijeNaBazu=$KonekcijaObject->konekcijaDB;        //echo "KONEKCIJA-BAZA:".$bazapodataka;        // IZDVAJANJE PODATAKA KORISTECI KLASU SMER      require "klase/BaznaTabela.php";      require "klase/DBProizvodjac.php";      $EvidencijaObject = new DBProizvodjac($KonekcijaObject, "Proizvodjac");      $EvidencijaObject->UcitajKolekcijuSvihProizvodjaca();      $KolekcijaZapisa= $EvidencijaObject->Kolekcija;      $UkupanBrojZapisa= $EvidencijaObject->BrojZapisa;  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција реализоване производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  <script src="JavaScript/provera.js"> </script>  </head>  <body>  <!-----VELIKA TABELA KOJA SADRZI SVE---->  <!-----10% SADRZAJ 10%---->  <table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" style="border-spacing: 0;">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljewelcome.php';?>  <!-------------------------- DONJI DEO  ------->  <tr style="padding:0px;">  <!-----LEVO PRAZNINA---->  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle" style="width:80%; padding:0" >  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:17%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevoadmin.php';?>  </td>  <td style="width:1%;">  </td>  <td style="width:80%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnounosSP.php';?>  </td>  <td style="width:1%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <!-----DESNO PRAZNINA---->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr style="padding:0px;">  <td style="width:10%;"></td>  <td align="center" valign="middle"></td>  <td style="width:10%;"></td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

prijava.php

|  |
| --- |
| <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција реализоване производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  </head>  <body>  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljeprijava.php';?>  <!-------------------------- DONJI DEO  ------->  <tr>  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle">  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:15%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevofinal.php';?>  </td>  <td style="width:2%;">  </td>  <td style="padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnoprijava.php';?>  </td>  <td style="width:2%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr>  <td style="width:10%;">  </td>  <td align="center" valign="middle" >  </td>  <td style="width:10%;">  </td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

* + - 1. Korisnički interfejs – koristi klase prezentacione logike

EvidencijaRealizovaneProizvodnjeLista.php

|  |
| --- |
| <?php         session\_start();           // citanje vrednosti iz sesije         $korisnik=$\_SESSION["korisnik"];          // ako nije prijavljen korisnik, vraca ga na pocetnu stranicu                  if (!isset($korisnik))                  {                      header ('Location:index.php');                  }  // REALIZACIJA CITANJA SVIH I FILTRIRANIH PODATAKA  //KONEKCIJA KA SERVERU    // koristimo klasu za poziv procedure za konekciju      require "klase/BaznaKonekcija.php";      require "klase/BaznaTabela.php";      $KonekcijaObject = new Konekcija('klase/BaznaParametriKonekcije.xml');      $KonekcijaObject->connect();      if ($KonekcijaObject->konekcijaDB) // uspesno realizovana konekcija ka DBMS i bazi podataka      {          //echo "Успешна конекција!";          require "klase/DBEvidencijaRealizovaneProizvodnjeV.php";          $EvidencijaViewObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject,"EvidencijaRealizovaneProizvodnje");          if (isset($\_GET['filtriraj']))              {                  // prikaz filtriranih podataka primenom pogleda nad kojim je dodat filter                  $filter=$\_GET['filter'];                  $EvidencijaViewObject->DajSvePodatkeOEvidencijiProizvodnje($filter);              }              else              {                  // prikaz svih podataka primenom pogleda koji je u bazi podataka                  $filter=null;                  $EvidencijaViewObject->DajSvePodatkeOEvidencijiProizvodnje($filter);                  // sada raspolazemo sa:                  //$StudentViewObject->Kolekcija                  //$StudentViewObject->BrojZapisa              }        }      else      {          echo "Неуспешна конекција!";      }  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евинденција реализоване производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  </head>  <body>  <!-----VELIKA TABELA KOJA SADRZI SVE---->  <!-----10% SADRZAJ 10%---->  <table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" style="border-spacing: 0;">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljewelcome.php';?>  <!-------------------------- DONJI DEO  ------->  <tr style="padding:0px;">  <!-----LEVO PRAZNINA---->  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle" style="width:80%; padding:0" >  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:17%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevoadmin.php';?>  </td>  <td style="width:1%;">  </td>  <td style="width:80%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnoEvidencijaRealizovaneProizvodnjeLista.php';?>  </td>  <td style="width:1%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <!-----DESNO PRAZNINA---->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr style="padding:0px;">  <td style="width:10%;"></td>  <td align="center" valign="middle"></td>  <td style="width:10%;"></td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

EvidencijaRealizovaneProizvodnjeParametarskaStampa.php

|  |
| --- |
| <?php  // OVO JE SUSTINSKO ODJAVLJIVANJE KORISNIKA  session\_start();  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција реализоване производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  </head>  <body>  <!-----VELIKA TABELA KOJA SADRZI SVE---->  <!-----10% SADRZAJ 10%---->  <table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" style="border-spacing: 0;">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljewelcome.php';?>  <!-------------------------- DONJI DEO  ------->  <tr style="padding:0px;">  <!-----LEVO PRAZNINA---->  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle" style="width:80%; padding:0" >  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:17%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevoadmin.php';?>  </td>  <td style="width:1%;">  </td>  <td style="width:80%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnoParametarskaStampa.php';?>  </td>  <td style="width:1%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <!-----DESNO PRAZNINA---->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr style="padding:0px;">  <td style="width:10%;"></td>  <td align="center" valign="middle"></td>  <td style="width:10%;"></td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

EvidencijaRealizovaneProizvodnjeStampa.php

|  |
| --- |
| <?php  // OVO JE SUSTINSKO ODJAVLJIVANJE KORISNIKA  session\_start();  // remove all session variables  session\_unset();  // destroy the session  session\_destroy();  //KONEKCIJA KA SERVERU    // koristimo klasu za poziv procedure za konekciju  require "klase/BaznaKonekcija.php";  require "klase/BaznaTabela.php";  $KonekcijaObject = new Konekcija('klase/BaznaParametriKonekcije.xml');  $KonekcijaObject->connect();  if ($KonekcijaObject->konekcijaDB) // uspesno realizovana konekcija ka DBMS i bazi podataka  {          //echo "Успешна конекција!";          require "klase/DBEvidencijaRealizovaneProizvodnjeV.php";          $EvidencijaViewObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject,"EvidencijaRealizovaneProizvodnje");          if (isset($\_GET['filtriraj']))              {                  // prikaz filtriranih podataka primenom pogleda nad kojim je dodat filter                  $filter=$\_GET['filter'];                  $EvidencijaViewObject->DajSvePodatkeOEvidencijiProizvodnje($filter);              }              else              {                  // prikaz svih podataka primenom pogleda koji je u bazi podataka                  $filter=null;                  $EvidencijaViewObject->DajSvePodatkeOEvidencijiProizvodnje($filter);                  // sada raspolazemo sa:                  //$StudentViewObject->Kolekcija                  //$StudentViewObject->BrojZapisa              }  }  else  {      echo "Неуспешна конекција!";  }  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција реализоване производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  </head>  <body>  <!-----VELIKA TABELA KOJA SADRZI SVE---->  <!-----10% SADRZAJ 10%---->  <table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" style="border-spacing: 0;">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljestampa.php';?>  <!-------------------------- DONJI DEO  ------->  <tr style="padding:0px;">  <!-----LEVO PRAZNINA---->  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle" style="width:80%; padding:0" >  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#FFFFFF">  <tr>  <td style="width:1%;">  </td>  <td style="width:80%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnostampa.php';?>  </td>  <td style="width:1%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <!-----DESNO PRAZNINA---->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr style="padding:0px;">  <td style="width:10%;"></td>  <td align="center" valign="middle"></td>  <td style="width:10%;"></td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footerstampa.php';?>  </table>  </body>  </html> |

index.php

|  |
| --- |
| <?php  // OVO JE SUSTINSKO ODJAVLJIVANJE KORISNIKA  session\_start();  // remove all session variables  session\_unset();  // destroy the session  session\_destroy();  // REALIZACIJA CITANJA SVIH I FILTRIRANIH PODATAKA  //KONEKCIJA KA SERVERU    // koristimo klasu za poziv procedure za konekciju      require "klase/BaznaKonekcija.php";      require "klase/BaznaTabela.php";      $KonekcijaObject = new Konekcija('klase/BaznaParametriKonekcije.xml');      $KonekcijaObject->connect();      if ($KonekcijaObject->konekcijaDB) // uspesno realizovana konekcija ka DBMS i bazi podataka      {          //echo "Успешна конекција!";              require "klase/DBEvidencijaRealizovaneProizvodnjeV.php";          $EvidencijaViewObject = new DBEvidencijaRealizovaneProizvodnje($KonekcijaObject,"EvidencijaRealizovaneProizvodnje");          if (isset($\_GET['filtriraj']))              {                  // prikaz filtriranih podataka primenom pogleda nad kojim je dodat filter                  $filter=$\_GET['filter'];                  $EvidencijaViewObject->DajSvePodatkeOEvidencijiProizvodnje($filter);              }              else              {                  // prikaz svih podataka primenom pogleda koji je u bazi podataka                  $filter=null;                  $EvidencijaViewObject->DajSvePodatkeOEvidencijiProizvodnje($filter);                  // sada raspolazemo sa:                  //$StudentViewObject->Kolekcija                  //$StudentViewObject->BrojZapisa              }        }      else      {          echo "Неуспешна конекција!";      }  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција реализоване производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  </head>  <body>  <!-----VELIKA TABELA KOJA SADRZI SVE---->  <!-----10% SADRZAJ 10%---->  <table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" style="border-spacing: 0;">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljeindex.php';?>  <!-------------------------- DONJI DEO  ------->  <tr style="padding:0px;">  <!-----LEVO PRAZNINA---->  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle" style="width:80%; padding:0" >  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:15%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevofinal.php';?>  </td>  <td style="width:1%;">  </td>  <td style="width:80%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnopocetna.php';?>  </td>  <td style="width:1%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <!-----DESNO PRAZNINA---->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr style="padding:0px;">  <td style="width:10%;"></td>  <td align="center" valign="middle"></td>  <td style="width:10%;"></td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

Welcome.php

|  |
| --- |
| <?php         session\_start();           // citanje vrednosti iz sesije         $korisnik=$\_SESSION["korisnik"];          // ako nije prijavljen korisnik, vraca ga na pocetnu stranicu                  if (!isset($korisnik))                  {                      header ('Location:index.php');                  }  ?>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">  <meta charset="UTF-8">  <head>  <title>Евиденција Реализације Производње</title>  <meta charset="UTF-8">  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">  </head>  <body>  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0">  <!-------------------------- ZAGLAVLJE ------->  <?php include 'delovi/zaglavljewelcome.php';?>  <!-------------------------- DONJI DEO  ------->  <tr>  <td style="width:10%;">  </td>  <!------------------------------------------------------------------------------------------->  <!---------------------- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ---------------------->  <td align="center" valign="middle">  <table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0" bgcolor="#003366">  <tr>  <td style="width:1%;">  </td>  <td style="width:20%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <?php include 'delovi/menilevoadmin.php';?>  </td>  <td style="width:2%;">  </td>  <td style="padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">  <!------- GLAVNI SADRZAJ desno ----------->  <?php include 'delovi/desnowelcome.php';?>  </td>  <td style="width:2%;">  </td>  </tr>  </table>  </td>  <!---------------------- SADRZAJ zavrsava ovde ---------------------->  <td style="width:10%;">  </td>  </tr>  <!---------------------- DONJI DEO zavrsava ovde ---------------------->  <tr>  <td style="width:10%;">  </td>  <td align="center" valign="middle" >  </td>  <td style="width:10%;">  </td>  </tr>  <!--- DONJI DEO sa donjom ivicom zavrsava ovde  ------->  <!-- footer panel starts here -->  <?php include 'delovi/footer.php';?>  </table>  </body>  </html> |

* + 1. Delovi koda i opis implementacije podrške personalizaciji aplikacije i bezbednosti korišćenja (prijava korisnika, rad sa sesijama, razmena podataka putem post metode)

prijavaprovera.php

|  |
| --- |
| <?php  session\_start();         $loginUserName=$\_POST['korisnickoIme'];         $loginPassword=$\_POST['sifra'];      // zato sto se prilikom require uradi copy paste u ovaj fajl,  // onda se putanja do parametra gleda u odnosu na lokaciju ovog fajla  require 'klase/BaznaKonekcija.php';  require 'klase/BaznaTabela.php';  require 'klase/DBKorisnik.php';  $korisnik='NEPOZNAT KORISNIK';  $objKonekcija = new Konekcija('klase/BaznaParametriKonekcije.xml');  $objKonekcija->connect();  if ($objKonekcija->konekcijaDB)      {          $objKorisnik = new DBKorisnik($objKonekcija, 'korisnik');          $postojiKorisnik=$objKorisnik->DaLiPostojiKorisnik($loginUserName,$loginPassword);          if ($postojiKorisnik=="DA")          {              // rad sa sesijama              $\_SESSION["prez"] = $objKorisnik->DajPrezimePrijavljenogKorisnika($loginUserName,$loginPassword);              $\_SESSION["ime"] = $objKorisnik->DajImePrijavljenogKorisnika($loginUserName,$loginPassword);              $\_SESSION["idkorisnika"] = $objKorisnik->DajIDPrijavljenogKorisnika($loginUserName,$loginPassword);              $\_SESSION["korisnik"] = $objKorisnik->DajImePrezimePrijavljenogKorisnika($loginUserName,$loginPassword);              // ucitavanje pocetne personalizovane stranice              header ('Location:Welcome.php');          }          else          {              // neuspeh izaziva ponovo ucitavanje stranice za prijavu              header ('Location:prijava.php');            }      }      else      {          echo "Neuspeh konekcije na bazu podataka!";        }    ?> |

* + 1. Delovi koda i opis implementacije osnovnih principa OOP (enkapsulacija, nasleđivanje, polimorfizam) i SOLID principa (single responsibility, interface segregation, dependency inversion sa dependency injection tehnikom)

DBEvidencijaRealizovaneProizvodnje nasleđuje klasu BaznaTabela i koristi njene metode (izmenjene)

|  |
| --- |
| <?php  class Tabela{  // atributi  public $OtvorenaKonekcija;  public $NazivBazePodataka;  public $NazivTabele;  public $TipMYSQL;  //  public $Kolekcija;  public $BrojZapisa;  public $PrviRedZapisa;  public $ListaZapisa;  // = array();  // metode  // ------- konstruktor  public function \_\_construct($NovaOtvorenaKonekcija, $NoviNazivTabele){  // podrazumevamo da je otvorena konekcija, a zatvara se spolja      $this->OtvorenaKonekcija = $NovaOtvorenaKonekcija;      $this->NazivBazePodataka = $NovaOtvorenaKonekcija->KompletanNazivBazePodataka;      $this->NazivTabele = $NoviNazivTabele;      $this->TipMYSQL = $NovaOtvorenaKonekcija->VerzijaMYSQLNaredbi;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajSve($KriterijumSortiranja)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        $SQL = "select \* from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` ORDER BY ".$KriterijumSortiranja;        if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);        }      else // mysql      {          $this->Kolekcija = mysql\_query($SQL);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajSvePoUpitu($Upit)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $Upit);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);        }      else // mysql      {          $this->Kolekcija = mysql\_query($Upit);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajSvaPoljaFiltrirano($KriterijumFiltriranja, $KriterijumSortiranja)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        $SQL = "select \* from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja." ORDER BY ".$KriterijumSortiranja;        if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);      }      else // mysql      {          $this->Kolekcija = mysql\_query($SQL);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function UcitajPoljaFiltrirano($Polja, $KriterijumFiltriranja, $KriterijumSortiranja)  {      // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa        $SQL = "select ".$Polja." from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja." ORDER BY ".$KriterijumSortiranja;      if ($this->TipMYSQL=="mysqli")      {          $this->Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $this->BrojZapisa = mysqli\_num\_rows($this->Kolekcija);      }      else // mysql      {          $this->Kolekcija = mysql\_query($SQL);          $this->BrojZapisa = mysql\_num\_rows($this->Kolekcija);      }  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function DajVrednostJednogPoljaPrvogZapisa ($NazivTrazenogPolja, $KriterijumFiltriranja, $KriterijumSortiranja)  {      $SQL = "select ".$NazivTrazenogPolja." from `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja." ORDER BY ".$KriterijumSortiranja;        if ($this->TipMYSQL=="mysqli")      {          $Kolekcija = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $row=mysqli\_fetch\_array($Kolekcija,MYSQLI\_NUM);  // fetch row          $Vrednost=$row [0];      }      else // mysql      {          $Kolekcija = mysql\_query($SQL);          $Vrednost=mysql\_result($Kolekcija,0,$NazivTrazenogPolja);      }      return $Vrednost;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function PrebaciKolekcijuUListu($Kolekcija) //kolekcija je result  {      $ListaZapisa = array();      if ($this->TipMYSQL=="mysqli")      {          while($RedZapisa = mysqli\_fetch\_array($Kolekcija,MYSQLI\_NUM))              {                  $this->ListaZapisa[] = $RedZapisa;              }      }      else // mysql      {          while($RedZapisa = mysql\_fetch\_array($Kolekcija,MYSQLI\_NUM))              {                  $this->ListaZapisa[] = $RedZapisa;              }      }        return $ListaZapisa;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function DajVrednostPoRednomBrojuZapisaPoRBPolja ($Kolekcija, $RBZapisa, $RBPolja)  {      if ($this->TipMYSQL=="mysqli")      {          $ListaZapisa = array();          $ListaZapisa= $this->PrebaciKolekcijuUListu($Kolekcija);          $RedZapisa=$this->ListaZapisa[$RBZapisa];          $Vrednost=$RedZapisa [$RBPolja];      }      else // mysql      {          $Vrednost=mysql\_result($Kolekcija,$RBZapisa, $RBPolja);   //$NazivPolja);      }      return $Vrednost;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function PostojiZapis($KriterijumFiltriranja)   // da li postoji, da li je jedinstven itd.  {      // ne puni kolekciju atribut, vec samo okalnu promenljivu da bi vratio da li postoji zapis        $SQL = "SELECT \* FROM `".$this->NazivBazePodataka."`.`".$this->NazivTabele."` WHERE ".$KriterijumFiltriranja;   //PRIMER: KORISNICKOIME='".$korisnickoime."' AND SIFRA='".$sifra."'";      if ($this->TipMYSQL=="mysqli")      {          $KolekcijaLokalna = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $SQL);          $BrojZapisaLokalna = mysqli\_num\_rows($KolekcijaLokalna);        }      else // mysql      {          $KolekcijaLokalna = mysql\_query($SQL);          $BrojZapisaLokalna = mysql\_num\_rows($KolekcijaLokalna);      }      if ($BrojZapisaLokalna>0)      {          $postoji=true;      }      else      {          $postoji=false;      }      return $postoji;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  public function IzvrsiAktivanSQLUpit($AktivanSQLUpit) {      //PRIMER: $AktivanSQLUpit = "INSERT INTO `".$this->bazapodataka."`.`KORISNIK` (IME, PREZIME, KORISNICKOIME, SIFRA, EMAIL, URLSlike, statusucesca, DatumRodjenja) VALUES ('$Ime', '$Prezime', '$KorisnickoIme', '$Sifra', '$Email', '', '$Status', '$DatumRodjenja')";      // ---------------      if ($this->TipMYSQL=="mysqli")      {          $retval = mysqli\_query($this->OtvorenaKonekcija->konekcijaDB, $AktivanSQLUpit);          $Greska = mysqli\_error($this->OtvorenaKonekcija->konekcijaDB);      }      else // mysql      {          $retval = mysql\_query( $AktivanSQLUpit, $this->OtvorenaKonekcija->konekcijaMYSQL);          $Greska = mysql\_error();      }      return $Greska;  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  }  ?> |

|  |
| --- |
| <?php  class DBEvidencijaRealizovaneProizvodnje extends Tabela  {  // ATRIBUTI  private $bazapodataka;  private $UspehKonekcijeNaDBMS;  //  public $IdProizvodnje;  public $NazivProizvodnje;  public $MestoProizvodnje;  public $BrojProizvodnihObjekata;  public $IdProizvodjaca;  // METODE  // konstruktor  public function DajKolekcijuSvihEvidencijaProizvodnje()  {  $SQL = "select \* from `EvidencijaRealizovaneProizvodnje` ORDER BY MestoProizvodnje ASC";  $this->UcitajSvePoUpitu($SQL); // puni atribut bazne klase Kolekcija  return $this->Kolekcija; // uzima iz baznek klase vrednost atributa  }  public function UcitajEvidencijuPoIdProizvodnje($IdProizvodnjeParametar)  {  $SQL = "select \* from `EvidencijaRealizovaneProizvodnje` where `IdProizvodnje`='".$IdProizvodnjeParametar."'";  $this->UcitajSvePoUpitu($SQL); // puni atribut bazne klase Kolekcija  // raspolazemo sa:  // $Kolekcija;  //  $BrojZapisa;  }  public function DodajNovuEvidenciju()  {      $SQL = "INSERT INTO `EvidencijaRealizovaneProizvodnje`(IdProizvodnje, NazivProizvodnje, MestoProizvodnje, BrojProizvodnihObjekata, IdProizvodjaca) VALUES ('$this->IdProizvodnje', '$this->NazivProizvodnje', '$this->MestoProizvodnje', '$this->BrojProizvodnihObjekata', '$this->IdProizvodjaca')";      $greska=$this->IzvrsiAktivanSQLUpit($SQL);        return $greska;  }    public function ObrisiEvidenciju($IdProizvodnjeZaBrisanje)  {      $SQL = "DELETE FROM `EvidencijaRealizovaneProizvodnje` WHERE IdProizvodnje='".$IdProizvodnjeZaBrisanje."'";      $greska=$this->IzvrsiAktivanSQLUpit($SQL);        return $greska;  }  // TO DO  public function IzmeniEvidenciju($StariIdProizvodnje, $idProizvodnje, $nazivProizvodnje, $mestoProizvodnje, $brojProizvodnihObjekata, $idProizvodjaca)  {      $SQL = "UPDATE `EvidencijaRealizovaneProizvodnje` SET IdProizvodnje='".$idProizvodnje."', NazivProizvodnje='".$nazivProizvodnje."',MestoProizvodnje='".$mestoProizvodnje."', BrojProizvodnihObjekata='".$brojProizvodnihObjekata."', IdProizvodjaca='".$idProizvodjaca."' WHERE IdProizvodnje='".$StariIdProizvodnje."'";      $greska=$this->IzvrsiAktivanSQLUpit($SQL);        return $greska;  }  // ostale metode    }  ?> |

* + 1. Delovi koda objašnjenja realizacije clean code pristupa (čitljivost koda kroz nazive polja, procedura, klasa, komentare (uključujući CRC card komentar o svakoj klasi), konvencije oblika koda i dr)

Sve metode i objekti su napisani na srpskom jeziku, camelCasing-om.

BaznaTransakcija.php

|  |
| --- |
| <?php  class Transakcija{  // atributi  private $OtvorenaKonekcija;  private $VerzijaMySQLNaredbi;  // metode  // ------- konstruktor  public function \_\_construct($NovaOtvorenaKonekcija)  {      $this->OtvorenaKonekcija=$NovaOtvorenaKonekcija;      $this->VerzijaMySQLNaredbi=$NovaOtvorenaKonekcija->VerzijaMYSQLNaredbi;  }  public function DajVerzijuMySQL()  {      return $this->VerzijaMySQLNaredbi;  }  public function ZapocniTransakciju()  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"SET AUTOCOMMIT=0");          mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"START TRANSACTION");      }      else      {          mysql\_query("SET AUTOCOMMIT=0");          mysql\_query("START TRANSACTION");      }    } // zatvaranje procedure  public function ProveriGresku()  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          $greska= mysqli\_error($this->OtvorenaKonekcija->konekcijaDB);        }      else      {          $greska= mysql\_error();        }        return $greska;    } // zatvaranje procedure  public function PonistiTransakciju()  // samo ako poslednje izvrsavanje aktivnog SQL upita ima gresku  //onda radi rollback  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"ROLLBACK");      }      else      {          mysql\_query("ROLLBACK");      }    } // zatvaranje procedure  public function ZavrsiTransakciju($UtvrdjenaGreska)  // samo ako poslednje izvrsavanje aktivnog SQL upita ima gresku  //onda radi rollback  {      if ($this->VerzijaMySQLNaredbi=="mysqli")      {          if (empty($UtvrdjenaGreska))              {                  mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"COMMIT");              }              else              {                  mysqli\_query($this->OtvorenaKonekcija->konekcijaDB,"ROLLBACK");              }      }      else      {          if (empty($UtvrdjenaGreska))              {              mysql\_query("COMMIT");              }          else              {              mysql\_query("ROLLBACK");              }      }    } // zatvaranje procedure    }  // zatvaranje klase      ?> |

1. Korisničko i tehničko uputstvo
   * 1. Korisničko uputstvo (ekrani i kratak opis načina korišćenja)

Prilikom pokretanja aplikacije otvara se stranica index.php. Ova stranica je za neregistrovanog korisnika, ima mogućnost pristupa stranici za prijavu, kao i spisak evidencija realizovane proizvodnje sa mogućnošću filtriranja prema mestu proizvodnje.

A screenshot of a website

Description automatically generated

Prilikom klika na “Пријава“, otvara se stranica prijava.php, na kojoj korisnik ima mogućnost prijave pomoću unošenja korisničkog imena i šifre, kao i mogućnost povratka na naslovnu stranu.

A screenshot of a computer

Description automatically generated

Nakon uspešne prijave otvara se stranica Welcome.php, sa leve strane se nalazi ime korisnika, a sa desne opcija za odjavu, a ispod je poruka dobrodošlice, a levo je meni koji ima samo admin. Na meniju admina postoje opcije za dodavanje, dodavanje sa stored procedurom, listu i ažuriranje, štampu, parametarsku štampu, spisak evidencija realizovane proizvodnje i odabir evidencija.

A blue and white rectangular object with a red object in the middle

Description automatically generated with medium confidence

Prilikom odabira opcije za dodavanje, otvara se stranica unos.php, na kojoj ostaje meni za admina, mogućnost odjave i forma za unos nove evidencije. Na formi se nalaze text box-ovi za unos id-a, naziv proizvodnje, mesto proizvodnje, broj proizvodnih objekata, zatim ima combobox za odabir proizvodjača.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Nakon snimanja podataka, otvara se stranica EvidencijaRealizovaneProizvodnjeLista.php, i na spisku evidencija se vidi id, naziv, mesto, broj i prezime proizvodjača

A screenshot of a computer

Description automatically generated

Nakon odabira opcije za izmenu, otvara se stranica EvidencijaRealizovaneProizvodnjeIzmeniForm.php na kojoj postoji mogućnost izmene samo onoga što se nalazi na spisku, bez mogućnosti izmene Id-a.

A screenshot of a computer

Description automatically generated

Odabirom opcije za brisanje stavke spiska, otvara se prozor za potvrđivanje brisanja.

A screenshot of a computer

Description automatically generated

Odabirom opcije za štampu, otvara se stranica EvidencijaRealizovaneProizvodnjeStampa.php.

A screenshot of a computer

Description automatically generated

Prilikom odabira opcije za parametarsku štampu, otvara se stranica EvidencijaRealizovaneProizvodnjeParametarskaStampa.php na kojoj postoji opcija odabira samo jedne stavke spiska preko mesta proizvodnje.

A screenshot of a computer

Description automatically generated

Nakon unošenja datuma, i klika na dugme „Štampaj“, otvara se stranica StampaPodatakaOEvidencijiRealizovaneProizvodnje.php.

A white background with blue and red text

Description automatically generated

1. Tehničko uputstvo (uputstvo za instalaciju, početno konfigurisanje parametara rada)

Kako bi informacioni sistem privrede radio, neophodno je postojanje baze podataka, sa odgovarajućim tabelema i uskladištenim procedurama odgovarajućih naziva.

* 2. Izveštaj o testiranju softvera

1. Test slučajevi ispravnog rada korisnika i prikaz rezultata testiranja (ekrana aplikacije i ponašanja)

Prilikom uspešnog snimanja podataka sa stranice za unos nove evidencije realizovane proizvodnje, u bazi podataka se u tabeli EvidenijaRealizovaneProizvodnje prikazuju snimljeni podaci, a u tabeli Zahtev se UkupanBrojEvidencija inkrementira za 1 na osnovu odabranog evidencija .

A screenshot of a computer

Description automatically generated

U nastavku će biti prikazana tabela proizvodjač.

A screenshot of a computer

Description automatically generated

Prilikom prijave korisnika, aplikacija povlači podatke iz tabele Korisnik, u koju je unapred uneto dva korisnika. Ukoliko su prilikom prijave uneti tačno korisničko ime i šifra, otvara se stranica Welcome.php.

A screenshot of a computer

Description automatically generated

3.3.3. Zaključak o testiranju - u odnosu na test-baziranu specifikaciju ponašanja aplikacije

Testiranje ponašanja aplikacije se slaže sa test-baziranom specifikacijom. Prilikom unosa, ukoliko nisu popunjena sva polja softver prikazuje poruku i ne vrši upis podataka u bazu. Ukoliko su sva polja popunjena ispravnim vrednostima, softver vrši snimanje zapisa u bazu, u suprotnom prikazuje poruku o pogrešnom unosu i ne vrši snimanje.

Na osnovu sprovedenog testiranja izvodi se zaključak da je aplikacija u skladu sa test-

baziranom sprecifikacijom.

ZAKLJUČAK

Cilj ovog seminarskog rada jeste da predstavi vrste softverske dokumentacije i dijagrama, njihov značaj i ulogu u procesu razvoja softvera, kao i primenu pojedinih vrsta na konkretnom primeru razvoja višeslojne veb aplikacije.

Predstavljeni su UML dijagrami, njihova primena, vrste i uloga u procesu razvoja softvera, kao i postojeća rešenja za potrebe ručnog, ili automatskog modelovanja postupkom reverznog inženjeringa.

Na kraju rada, prikazan je proces razvoja višeslojne veb aplikacije za evidenciju realizovane proizvodnje, sa odgovarajućom softverskom dokumentacijom i dijagramima, čija je uloga doprinos u razumevanju softverskog sistema i njegovog razvoja.

LITERATURA

1. Skripta za pripremu teorijskog dela ispita – Prof.dr Ljubica Kazi, 2022.
2. Ministarstvo priverede - <https://privreda.gov.rs>
3. Agencija za privredne registre - https://www.apr.gov.rs/registri/privredna-drustva/uputstva/opsta-uputstva.2029.html