



Univerzitet u Novom Sadu
Tehnički fakultet »Mihajlo Pupin«
Zrenjanin



Diplomski Rad

PHP višeslojna web aplikacija za evidenciju projekata

Multi-layered PHP Web Application for Projects Records Support

Mentor:
Doc. dr Ljubica Kazi

Student:
Branislav Đekić IT 16/13

Zrenjanin, 2018.



Univerzitet u Novom Sadu
Tehnički fakultet »Mihajlo Pupin«
Zrenjanin



Diplomski

PHP višeslojna web aplikacija za evidenciju projekata

Multi-layered PHP Web Application for Projects Records Support

Mentor:
Doc. dr Ljubica Kazi

Student:
Branislav Đekić IT 16/13

Zrenjanin, 2018.

Sadržaj

1. Uvod.....	4
2. Teorijsko-tehnološke osnove.....	5
2.1. Web aplikacija	5
2.2. Metodologija razvoja softvera	7
2.3. Višeslojna arhitektura softvera	8
2.4. PHP programski jezik	11
2.5. MySQL baza podataka.....	12
2.6. Komunikacija između PHP-a i MySQL-a	13
3. Analiza semantičke oblasti	14
3.1.Pojam projekta	14
3.2. Upravljanje projektima	15
3.3. Standardna dokumentacija u upravljanju projektima u okviru PRINCE 2 metodologije	22
4. Postojeća Rešenja	26
5. Opis korišćenih softverskih alata u realizaciji primera.....	30
5.1.XAMPP.....	30
5.2. Notepad ++	31
6. Realizovani primer.....	32
6.1. Specifikacija zahteva	32
6.2. Opis posla	34
6.3. Model poslovnog procesa	35
6.4. Modeli opštег dizajna softvera	36
6.5. Konceptualni model podataka	38
6.6. Modeli implementiranog rešenja	39
6.6.1. Dijagram komponenti	39
6.6.2. Dijagram razmeštaja	40
6.7. Korisničko uputstvo.....	41

6.8. Opis implementacije	53
6.8.1. Tabela sa prikazom cele arhitekture po slojevima	53
6.8.2. Sloj podataka - SQL Skript i šema baze podataka	54
6.8.3. Klasa Konekcija i Parametri	60
6.8.4. Klase sloja podataka	62
6.8.5. Poslovna Pravila	65
6.8.6. Poslovni Objekti	67
6.8.7. Prezentaciona Logika.....	70
6.8.8.Korisnički Interfejs	72
7. Zaključak	83
8. Literatura.....	84

1. Uvod

Internet se smatra jednim od najbitnijih izuma 20. veka koji nam je omogućio komunikaciju na velike daljine, skoro u trenutku. Zbog tako brze komunikacije, omogućene su mnoge usluge preko interneta, ne samo slanje poruka, već poslovanje, deljenje podataka, arhiviranje kao i upravljanje firmama i u svrhe vlade. Informacije od korisnika se prenose putem interneta. Mnoge stranice koriste bazu podataka sa skladištenje podataka, o čemu se posebno treba obratiti pažnja. Uz povećanu popularnost korišćenja interneta za poslovanje i ličnu upotrebu, došlo je do pojave zloupotrebe ove platforme, tako da se mora posebno obratiti pažnja na bezbednost kada se priča o internetu (lični podaci, prevare itd). To je dovelo do razvoje enkripcija, firewall i raznih tehnika koja bi smanjila šansu ugrožavanja bezbednosti. Skoro sve web aplikacije koriste bazu podatka za skladištenje podatka i posebno se treba obratiti pažnja na rukovođenje njih. Postoje nekoliko pristupa distribucije baza podatka i kako su struktuirane. Standardi omogućuju da data web aplikacija bude podržana na mnogo platformi. Većina korisnika interneta danas uz računar imaju i smart telefon, tako da je postalo skoro obavezno da data web aplikacija bude podržana i na tim uređajima.

Web aplikacija opisana u ovom radu dizajnirana je da služi evidenciji i podršci upravljanju projektima na specifičan način – projektima koji se realizuju na fakultetu. Osnovna ideja za realizaciju ovakve web aplikacije realizovana je u formi prototipa PHP aplikacije u okviru nastavnog predmeta "Stručna praksa", gde je pod mentorstvom doc. dr Ljubice Kazi realizovano rešenje kao deo internog projekta Tehničkog fakulteta "Mihajlo Pupin" Zrenjanin, pod nazivom "Profesionalno usavršavanje studenata u okviru unapređenja informacionog sistema fakulteta". U ovom radu opisano je softversko rešenje koje, iako je rađeno primenom PHP programskog jezika, u potpunosti je drugačije strukturirano – realizovano je višeslojno rešenje uz značajno veći obuhvat softverskih funkcija. Svakako, rešenje koje je predstavljeno u ovom radu još uvek se može smatrati verzijom softvera koju je moguće nadograditi i unaprediti.

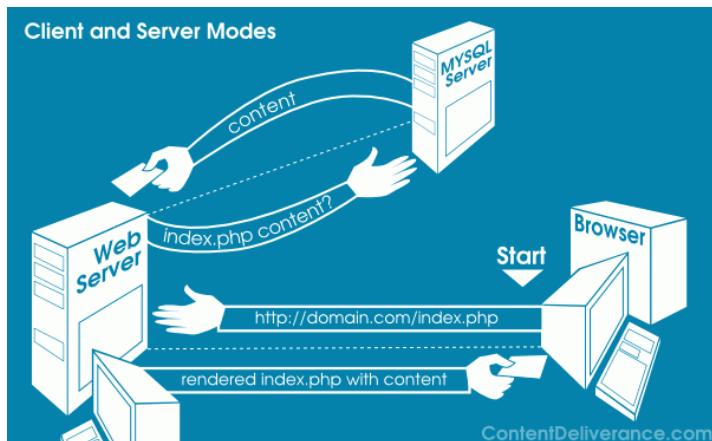
2. Teorijsko-tehnološke osnove

2.1. Web aplikacija

Web aplikacija predstavlja bilo koju aplikaciju koja koristi web pretraživač kao klijent. Složenost aplikacije može biti raznolika, od najjednostavnije aplikacije za poruke do web aplikacije za obradu teksta. [1]

Pod web aplikacijom se smatra kolekcija malih programa, HTML stranica, klase i drugih resursa koji pripadaju jednom domenu koji može da se nalazi na više lokacija i ima svoju putanju na web serveru. Web aplikacija postoji kao struktuirana hijararhija direktorijuma. [2]

Razlika između internet stranice i internet aplikacije jeste da je internet stranica statična dok je internet aplikacija interaktivna (dinamička). U suštini web aplikacija je web stranica sa kojom korisnik može da vrši interakciju. [4]



(Slika 1. - Vizualna ilustracija klijent-server arhitekture kod web aplikacija[5])

"Web aplikacije često koriste kombinaciju skripti koje se izvršavaju na serveru (ASP, PHP) i one koje se izvršavaju na klijentu (HTML, Javascript). Klijentska strana je zadužena za prezentaciju, dok je serverska strana za primanje i skladištenje podataka." [1]

Funkcionisanje klijent-server arhitekture se zasniva na vršenju transakcija ili razmeni podataka između klijenta i servera preko HTTP (Hyper Text Transfer Protocol) protokola. Klijent započinje razmenu tako što šalje zahtev serveru (piše serveru), a server koji je u stalnom stanju spremnosti, pruža uslugu klijentu tako što šalje podatke koje klijent zahteva. [5][6]

Kod web aplikacije, uloge ne moraju da budu iste, kod njih server može da se ponaša kao klijent. Primer toga bi bio da se podaci sa web aplikacije čuvaju na drugom serveru, koji služi kao baza podataka. U tom slučaju web server služi kao klijent serveru baze i šalje zahtev za podacima, koje baza šalje web serveru, web server te podatke obrađuje i napokon šalje stranicu sa podacima internet pretraživaču klijenta (Slika 1). [5]

Prednost web aplikacije jeste u tome što podaci mogu da budu pristupačni bilo gde. To znači da se dokumentu, koji je sačuvan od strane jednog klijenta na serveru, može pristupiti na drugom klijentu. Pošto se web aplikacija može prikazati na bilo kojoj platformi, izgled aplikacije zavisi od web pretraživača, a ne od operativnog sistema. [3]

Jedna od nedostataka web aplikacija jeste što nemaju kompletan pristup resursima klijenta, poput procesora i privremene ili trajne memorije. Zbog toga, zahtevnije aplikacije su prikladnije da budu u desktop formi u vidu programa koji se instalira na trajnoj memoriji računara, a ne u vidu web aplikacije. [3]

Drugi nedostatak je u tome da u slučaju gubitka konekcije ili prekida rada internet pretraživača, podaci mogu biti izgubljeni. Mnoge firme nude oba rešenja za njihove proizvode. [3]

"Bitna karakteristika bilo kakve internet stranice ili aplikacije jeste odziv (response time). To je brzina koja je potrebna klijentu da primi podatke koje je zatražio od servera. Aplikaciju treba testirati pod raznim uslovima, poput mnogo korisnika koji pristupaju stranici ili merenje koliko razmena je izvršeno sa korisnikom. Merenje se vrši od slanja zahteva od strane klijenta do poslednjeg bajta isporučeno klijentu." [7]

Na brzinu odziva utiče i fizička lokacija klijenta u odnosu na server, kao i količina i složenost podataka koju server treba da pošalje ili obradi. Ujedno, složenost same stranice utiče na brzinu učitavanja stranice u internet pretraživaču klijenta, tako da je poželjno da internet strana nema dosta složenih elemenata. [7]

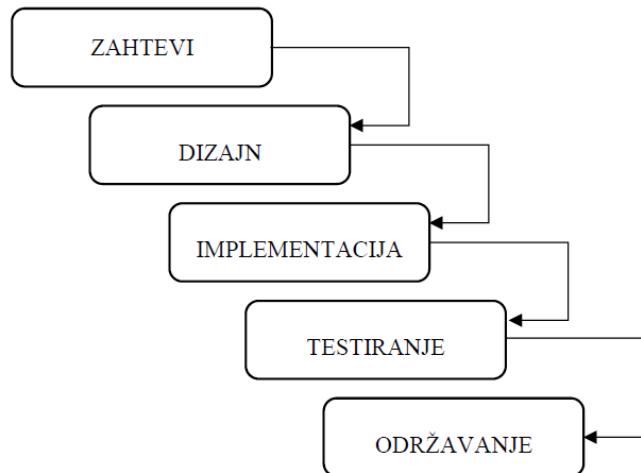
Dizajniranje arhitekture web aplikacije je identično izradi arhitekture bilo koje softverske aplikacije. Cilj jeste kreiranje izdržljivog skeleta aplikacije koji omogućava da aplikacija bude laka za održavanje, dobro optimizovana i da ima široku funkcionalnost. Za kreiranje arhitekture zaslužni su softverske arhitekte. [8]

2.2. Metodologija razvoja softvera

Potrebno je odlučiti za metodologiju razvoja softvera koja odgovara datom projektu kao i znanju i iskustvu osoba koji su uključeni u kreiranju datog projekta. Dva osnovna modela razvoja softvera su tradicionalna (sekvencialna) metodologija i agilna (iterativna) metodologija. [8]

Tradicionalna Metoda

Tradicionalna metodologija je najstarija, kao i najpoznatija metodologija. Drugi naziv za tradicionalnu metodologiju je model vodopada. Ovo je model u kom se razvoj softvera produžava iz jedne faze u drugu, pa ova metodologija ima još jedan naziv, a to je sekvencialna. Sa sledećom fazom se ne počinje dok prethodna nije uspešno završena (Slika 2). [8]

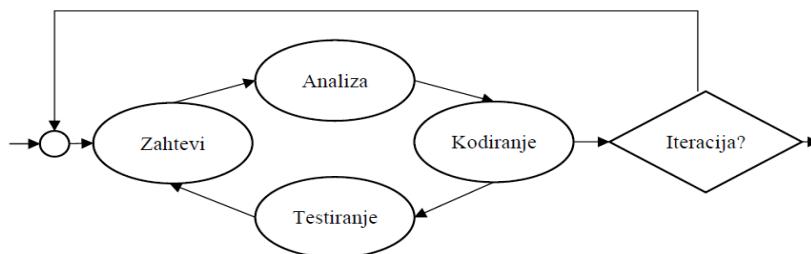


(Slika 2. - Ilustracija tradicionalnog modela[8])

Ovaj model je jednostavan i disciplinovan, ali primena je ograničena na jednostavnim projektima. Jedan od razloga jeste što u složenijim projektima nisu svi zahtevi dostupni arhitektama. Zbog toga nastale su razne varijacije ove metode. [8]

Agilna Metoda

Agilna metoda je razvijena kao alternativa za tradicionalnu metodu. Agilna metoda radi na principu više iteracija i naglašava postepen razvoj softvera. Projekat pri svakoj iteraciji prolazi kroz iste korake: analiza, kodiranje i testiranje (Slika 3). Pri svakoj iteraciji dobija se isporučiva, ali nekompletна verzija sistema, tako da je potrebno više iteracija da bi softver zadovoljio zahteve u vidu funkcionalnosti. [8]



(Slika 3.- Prikaz agilne metodologije [8])

Fokus agilne metodologije jesu osobe koje zajednički rade na projektu kao i na njihovoј komunikaciji. Informacije koje korisnik daje su vrednije od planiranja. Saradnja klijenta i programera je glavna prednost ove metodologije. To znači da je agilna metodologija spremna na brze promene. [8]

2.3. Višeslojna arhitektura softvera

Pošto su aplikacije u preduzećima postojale sve složenije i morale da služe hiljadama korisnika, tradicionalna dvoslojna klijent-server arhitektura je predstavljala problem prilikom proširivanja ili promena, jer je zahtevala znatno veću količinu resursa, ali i sama logika arhitekture nije dozvoljavala lako ažuriranje aplikacije. [8]

Zbog toga je nastala troslojna arhitektura, čiji su slojevi:

- Prezentacioni sloj - nalazi se na računaru klijenta
- Sloj poslovne logike - nalazi se na aplikativnom serveru
- Sloj pristupa podataka - nalazi se na serveru baze podataka

Prezentacioni sloj predstavlja vidljiv deo aplikacije. Aplikacija je beskorisna ako korisnik ne može sa njom da rukovodi ili ako se podaci ne mogu predstaviti korisniku. On se sastoji od dve komponente, korisničkog interfejsa i prezentacione logike. Korisnički interfejs prikazuje alate koje su dostupni korisniku. [8]

Pri interakciji sa elementima, akcija se prosleđuje prezentacionoj logici u kojoj se unete instrukcije ili podaci transformišu u podatke koje se mogu proslediti srednjem sloju. Ovaj sloj mora da zadovolji zahteve poput nezavisnosti od modela podataka i fizičkog korisničkog interfejsa kao i da bude pogodan za testiranje. [8]



(Slika 4. - ASP.NET arhitektura[8])

Sloj servisa se smatra delom poslovnog sloja. On služi kao granica između prezentacionog sloja i poslovne logike. U sloju servisa kod se poziva direktno iz korisničkog interfejsa, i prilikom toga se poziva metoda koja uzima podatke u zadatom formatu i vraća neke druge podatke natrag. Prenos kroz sloj se vrši pomoću objekata za prenos podataka (DTO - Data Transfer Objects). [8]

Poslovni sloj predstavlja jezgro aplikacije u kojoj se nalazi sva logika sistema kao i logika predviđena za interakciju sa drugim slojevima, pa se zato ovaj sloj i zove sloj poslovne logike. On je zastupljen kod složenih sistema, poput sistema u bankarstvu i drugim vidovima poslovanja. U ovom sloju treba posebno posvetiti pažnju na bezbednost koja je bazirana na privilegijama. Pristup poslovnim objektima je ograničen samo na autorizovane korisnike.

U vertikalnom razlaganju slojeva najčešće se nalaze četiri komponente:

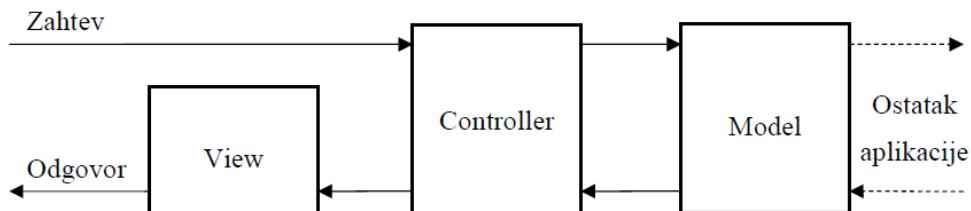
- Objektni model - modeluje učesnike u poslovnom procesu
- Poslovna pravila - predstavlja klijentske zahteve i politiku
- Servise - služi za implementaciju autonomne funkcionalnosti
- Proces rada - služi za definisanje načina prenosa podataka iz jednog modula ili sloja u drugi [8]

Sloj za pristup podacima (DAL - Data Access Layer) omogućava pristup podacima koji se nalaze u bazi podataka. On se sastoji od biblioteke koda koji je zadužen za primanje i samim tim zapisivanje podataka, kao i iščitavanje i slanje podataka slojevima iznad bez obzira na format podataka. [8]

Ovaj sloj mora da zadovolji trajno čuvanje, kao i dodavanje, čitanje, ažuriranje i brisanje (CRUD - Create Read Update Delete) podataka iz baze kao i da bude sposoban da vrši upravljanje transakcijama. [8]

MVC (Model-View-Controller) je veoma koristan patern za dizajn arhitekture, koji razdvaja aplikaciju u tri komponente:

- Model - predstavlja poslovne podatke koje će pogled da prikaže
- Controller - vrši interakciju sa modelom na osnovu zahteva koje prima preko browser-a
- View - pasivna komponenta koja služi za prikaz podataka koji su prethodno primljeni iz controller-a [8]



(Slika 5. - Prikaz MVC paterna [8])

Princip rada MVC se može opisati na sledeći način. Kada korisnik uputi zahtev, prvi koji prima i obrađuje jeste kontroler. Kontroler vrši interakciju sa modelom u zavisnosti od zahteva i dobavlja podatke. On šalje obrađene podatke koji trebaju da budu prikazani pogledu (Slika 5). [8]

2.4. PHP programski jezik

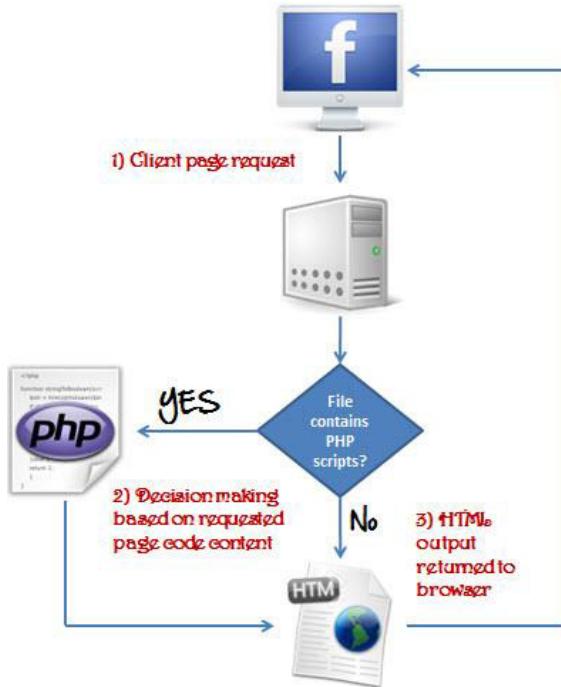
PHP (skraćeno od Hypertext Preprocessor) je često korišćeni jezik za skriptovanje opšte namene, koji je prilagođen upotrebi na internetu i koji se može ugraditi unutar HTML koda. PHP kod je ugrađen sa posebnim početnim i završnim tagom "<?php " i "?>" koji ugnezdi kod u HTML (Slika 6). [9]

```
<?php  
    echo 'Hello World';  
?>
```

(Slika 6.- Isečak primera PHP koda[46])

Jezik je po prirodi veoma sličan programskom jeziku poput C ili Perl i ima u cilju da omogući programerima da lakše dinamički generišu stranice. [10][12] Ono što razlikuje PHP od nekih drugih web skripti, poput JavaScript, jeste što se kod izvršava na serveru, pretvara u HTML i takav šalje korisniku, tako da primalac web stranice nezna šta se ustvari dešava u pozadini (Slika 7). Jezik je pristupačan novim korisnicima, jer je relativno jednostavan za razumevanje, ali i nudi mnoge napredne funkcije iskusnim programerima. [9]

Glavne prednosti ovog programskog jezika jeste što je prvo besplatan, što ga ujedno učinjava pristupačnim. Težina učenja je mala, kako zbog same prirode programskoj jezika tako i zbog veoma proširne dokumentacije i tutorijala. [11]



(Slika 7. - Algoritam obrade php koda , klijent - server [46])

Samim tim što se sam kod izvršava na serveru, klijent ne mora da ima PHP instaliran na računaru. To znači da se stranica može otvoriti na bilo kojoj platformi. Jezik je ujedno kompatabilan sa drugim sistemima, poput sistema za upravljanje bazom podataka. [11] PHP podržava i više poznate protokole poput POP3, IMAP i LDAP. [13]

2.5. MySQL baza podataka

MySQL je sistem za upravljanje relacionim bazama podataka koja je osnovana na SQL programskom jeziku. Ovaj sistem je moguće koristiti na mnogim platformama, poput Linux, Unix ili Windows. Iako ima široku upotrebu, MySQL se često koristi za internet funkcije. [14] MySQL je besplatan i otvoren je za izmene od strane korisnika. [16]

Platforma je dizajnirana za paralelno izvršavanje na serveru i za rad sa velikim opterećenjem. Sistem ujedno je struktuiran da bude pogodan za masovno ugrađivanje. MySQL je besplatan i otvoren svima za modifikaciju i unapređenje. Platforma je veoma brza i pouzdana i laka za korišćenje. MySQL je potpuno modularan i višeslojan. [15]

2.6. Komunikacija između PHP-a i MySQL-a

PHP predstavlja aplikacioni deo web aplikacije, dok MySQL predstavlja bazu podataka te web aplikacije. PHP se može koristiti kako za čist prikaz informacija , tako i za proveru informacija koje se tek unose. [16]

Glavna uloga PHP-a jeste sam prenos podataka sa forma u bazu podataka i vršenje komunikacije sa bazom. Baza ne mora da bude na istoj lokaciji kao i sam sajt, jer programski jezik je sposoban da šalje informacije preko interneta pomoću korisničkog imena i lozinke (eng. password). [16]

MySQL baza podataka prilikom komunikacije sa sajtom može da pošalje i poruke vezane za uspešnost izvršene date instrukcije, tako da programer može da bude informisan o mogućim problemima. Ta informacija može da bude integrisana u samu web aplikaciju u vidu poruke. [16]

3. Analiza semantičke oblasti

3.1.Pojam projekta

Definicija projekta data je u okviru PMBOK (Project Management Body of Knowledge). Projekat je privremeni poduhvati preduzet da bi se kreirao jedinstveni proizvod, usluga ili rezultat. Vremenska zavisnost projekta određena je time što projekat ima precizan početak i kraj. Kraj projekta je označen kada ciljevi projekta su dostignuti ili kada je projekat poništen, zato što ciljeve je nemoguće dostići. [20]

"Projekat može i terminisati i organizator ili klijent za koga je taj projekat namenjen. Svaki projekat stvara jedinstven rezultat, proizvod ili uslugu. Završni proizvod projekta može biti materijalan ili ne materijalan." [20]

Projekti se često koriste da direktno ili indirektno postignemo neki cilj ili strateški stan. Projekti često morao da budu opravdani i neki od razloga su:

- Zahtev tržišta (automobili koji manje troše, kao odgovor na veće cene goriva)
- Poslovna strategija ili potreba (firma za obuku radnika opravdava novi program da poveća njihov profit)
- Društvena potreba (projekat za razvoj navodnjavanja ili kanalizacije u delovima sveta koji su u razvoju)
- Zaštita okoline (potreba za tehnologijom koji manje šteti okolini)
- Tehnološki napredak (firma dizajnira novi računar koji ima manje dimenzije od prethodnog i ima bolje karakteristike) [20]

Pošto veliki zadatak je nemoguće uraditi odjednom, projekat omogućava da obiman posao bude razložen na manje poslove i dodeljen pojedincu ili grupama, prema njihovoј struci. Projekat treba da ima precizno definisan cilj, zahteve, promenjive i svrhu da bi bio pravilno definisan. [18] Time svim se bavi disciplina upravljanje projektima. [19]

3.2. Upravljanje projektima

"Upravljanje projektima je disciplina korišćenja već ustanovljenih principa, procedura i strategija da se upravlja projektom, od njegovog začeća do samog kraja. Ova disciplina nadgleda planiranje, organizovanje i implementaciju projekta." [17]

"Upravljanje projektima je primena znanja, veština, alata i tehnika da se projektuju aktivnosti i da se ispune zahtevi projekta. To se postiže kroz prikladnu primenu i integraciju 47 lokalno grupisani procesa upravljanje projekta, koji su kategorisani u pet Grupa Procesa." [20]

Oni su:

- Inicijacija
- Planiranje
- Izvršavanje
- Nadgledanje i Kontrola
- Zatvaranje [20]

Prilikom upravljanja projektima bitne su sledeće stvari:

- Identifikacija zahteva
- Pomiravanje potreba, briga i zahteva zainteresovanih strana prilikom planiranja i izvršavanja projekta
- Postavljanje, održavanje i vršenje komunikacije sa zainteresovanim stranama koje su aktivne, effektivne i sklone saradnji
- Balans ograničenja projekta poput obima, kvaliteta, budžeta, rizika, resursa i planova[20]

Postoji odnos između raznik ograničenja i faktora projekata. Menjanje jednog faktora često direktno utiče na druge faktore. Smanjenje krajnjeg roka završetka roka, za rezultat ima povećanje budžeta projekta zbog povećanja resursa potrebnih da bi se projekat završio ranije. Ako povećanje budžeta nije moguće, onda se obim projekata smanji. [20]

Zainteresovane strane mogu da imaju različita mišljenja koji faktori su bitni. Menjanje faktora nosi samo po sebi nove rizike. Tim projekta treba da bude sposoban da razreši situaciju, izbalansira zahteve da bi uspeli da izvrše uspešan projekat. [20]

Zbog same prirode razvoja plana za upravljanje projekata, i njegovoj sklonosti promenama, plan je dizajniran da bude menjan u više etapa (iteracija). Konstantno unapređenje plana omogućava daleko detaljniji opis projekta i mnogo preciznije procene. [20]

Oblasti znanja su grupa više procesa upravljanja projektima kojih ima 10 (Slika 8). U njih spada kolekcija koncepta, termina i aktivnosti koji čine profesionalno polje, polje upravljanje projekta, oblasti specijalizacije. Ovih deset oblasti se koriste na većini projekata. Te oblasti su:

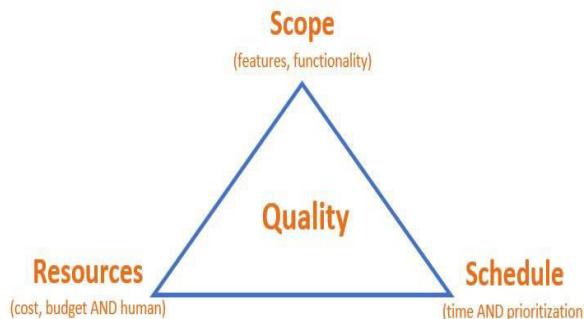
- Menadžment integracija projekta (Project Integration Management)
- Menadžment obima projekta (Project Scope Management)
- Menadžment vremena projekta (Project Time Management)
- Menadžment troškova projekta (Project Cost Management)
- Menadžment kvaliteta projekta (Project Quality Management)
- Menadžment ljudskih resursa (Project Human Resource Management)
- Menadžment komunikacija (Project Communications Management)
- Menadžment rizika projekta (Project Risk Management)
- Menadžment nabavka (Project Procurement Management)
- Menadžment upravljanja zainteresovanih strana (Project Stakeholder Management)[20]



(Slika 8. - 10 oblasti znanja [47])

Upravljanje poseduje više nivoa. Njihova svrha jeste da razdeli zadatke tako da ti poslovi mogu da budu završeni na efikasniji i brži način. Menadžeri su odgovorni za timove koji su pod njihovim nadgledanjem. To znači da oni snose punu odgovornost za bilo koju grešku ili ponašanje unutar njihovih timova. [22]

Menadžeri moraju da osiguraju kvalitet posla, planove i budžete članova tima. To postižu tako što dodeljuju određene poslove pravim ljudima. Time oni optimizuju kvalitet i pravilno iskoriste vreme za najbolje rezultate. Tri glavna aspekta čine takozvani "gvozdeni trougao" (eng. Iron Triangle) i ti aspekti su, vreme, budžet i obim (Slika 9). [22]



(Slika 9. - Gvozdeni trougao[48])

Vreme

Pošto svaki projekat mora da ima tačan redosled za svaki posao, menadžer mora da dizajnira redosled koji će doprineti da tim funkcioniše prema rokovima. Redosled uključuje faktore poput, vremena potrebnim za završetak projekta, rokovi za male poslove, poslovi za pojedinačne radnike. Cilj redosleda jeste da za što manje vreme dobijemo efikasniji proizvod. [22]

Budžet (Trošak)

Trošak znatno utiče na ceo projekat. Menadžer dodeljuje trošak za projekat tako što uzima sve faktore u obzir koje su vezane sa projektom. Cilj je balans između troška, proizvoda i kvaliteta. Menadžer mora da smanji trošak dok povećava proizvod i na taj način drži standard kvaliteta za proizvod ili uslugu. Faktori koji mogu da utiču na trošak su: cena opreme, cena plata ili troškovi za greške. Vreme ujedno je veliki uticaj na trošak, što duže projekat traje, duže treba plaćati radnicima plate. [22]

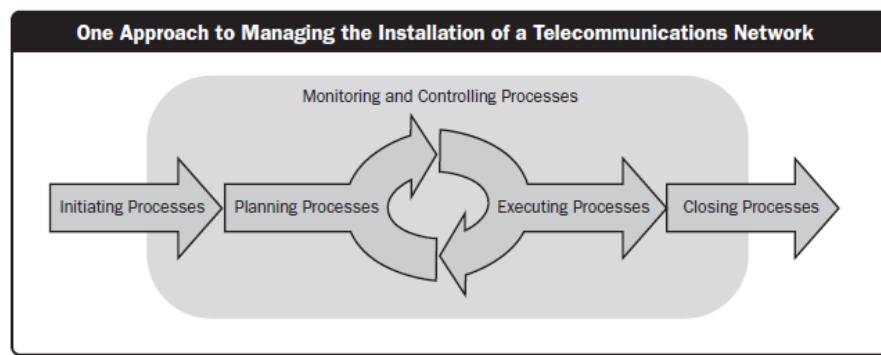
Obim

Pod obimom se smatra ciljevi (kratkotrajni i dugotrajni), moguće greške, plan implementacije i plan pregleda i korekcije. Ovo je najteži aspekt za menadžera, jer menadžer mora da uzme sva tri faktora u obzir i da osigura da tim razume veličinu projekta i šta se očekuje od njih. Ovaj aspekt je najčešći razlog neuspeha projekata, jer menadžeri često stave nerelističan cilj u odnosu na vreme i budžet ili članovi tima ne razumeju veličinu projekta. [22]

Glavni cilj svakog projekta jeste kvalitet. Sva tri prethodno spomenuta faktora su ograničenja koja utiču na taj cilj. [22] Ako projekat se fokusira na manji trošak i cenu, proizvod će biti lošeg kvaliteta. Ako se fokusira na mali trošak i kvalitet, onda će uticati na vreme potrebno za proizvodnju. Ako je fokus brzina i kvalitet, rezultat je visoka cena. [23]

Za uspešno izvršavanje projekta zaslužni su menadžeri za upravljanje projekata. [21] Menadžer projekta je osoba koju dodeljuje organizacija koja je stvorila projekat da vodi tim koji je zaslužan za uspešno izvršavanje zadatka. Oni imaju zadatak i odgovornost da brinu o potrebama tima i da osigura da se poslovi odigravaju po planu. Menadžer mora da ima sledeće kvalitete da bude smatrani sposobnim da rukovodi nekim projektom:

- Znanje (lično znanje i iskustvo)
- Performansa (sposobnost da primeni dato znanje)
- Lično ponašanje (karakteristike ličnosti i odnos sa članovima projekta) [20]



(Slika 10.- Primer projekta sa jednom fazom - instalacija telekomunikacija [20])

“Projekat je često potrebno podeliti i u zasebne faze (Slika 10). Faze projekta su kolekcije logički povezanih aktivnosti. Faze se često izvršavaju sekvencialno, ali nekad se mogu preklopiti da se izvršavaju istovremeno. Ovaj pristup podele po fazama, omogućava lakše upravljanje, planiranje i kontrola projekta. Broj faza ili sama potreba faza zavisi od veličine ili složenosti projekta. Bez obzira na broj faza projekta, sve faze su slične prema sledećim karakteristikama.” [20]

Prema [20], opšte karakteristike faza obuhvataju:

- Aktivnosti u okviru svake faze se razlikuju u odnosu na aktivnosti iz neke druge faze prema fokusu i organizaciji, lokaciji ili potrebnim veštinama za njigovo obavljanje.
- Ciljevi koji treba da se postignu u okviru faze su jedinstveni za svaku fazu, ali podležu kontroli koja je interna za tu fazu.
- Zatvaranje faze završava se određenim transferom rezultata rada nazvanim “fazna isporuka”. Ovaj trenutak se može nazvati “milestone” ili tačka u kojoj se vrši isporuka rezultata, ali i revizija procesa i problema u toku realizacije faze.

Kada projekti imaju više faza, one su uglavnom dizajnirane da budu sekventne. U slučaju sekventnih faza, faza počinje tek kada prethodna faza bude gotova. Kod faza koje se preklapaju, sledeća faza počinje pre završetka prethodne. One su rizičnije i zahtevaju dodatne resurse i komunikaciju sa prethodnom fazom (Slika 11). [20]



(Slika 11.- Primer projekta sa tri faze koje su sekvenčalne [20])

Projekat treba samo da obuhvati poslove vezane za dati projekat da budu izvršene pravilno. Upravljanje opsegom se bavi šta jeste i šta nije deo datog projekta. Ono ujedno vrši nadgledanje statusa projekta i kontrolu promena opsega projekta. [20]

Projekte je ujedno potrebno završiti u određenom vremenskom opsegu. Time se bavi disciplina menadžmenta vremena projekta. Procesi u toj disciplini su:

- Upravljanje planiranjem rasporedom radnih aktivnosti
- Definisanje aktivnosti
- Povezivanje aktivnosti
- Procena resursa potrebnih za aktivnosti
- Procena trajanja aktivnosti
- Razvoj rasporeda
- Kontrola rasporeda [20]

Kod nekih projekata, pogotovo kod manjih, definisanje i sekvensiranje aktivnosti i procene resursa i trajanja projekata su veoma povezanih, tako da se oni gledaju kao jedan proces koju izvršava jedna osoba u kratkom vremenskom periodu. [20]

Rizici

"Menadžment rizika projekta uključuje procese predviđene za sprovođenje planiranje menadžmenta rizika, identifikacija, analiza i kontrolu rizika na projektu. Ciljevi menadžmenta rizika jesu da povećaju učestalost pozitivnih događaja i da smanje učestalost i uticaj negativnih događaja u projektu." [20]

Procesi u menadžmentu rizika su:

- Menadžment planiranja rizika
- Identifikacija rizika
- Kvalitativna analiza rizika
- Kvantitativna analiza rizika
- Planski odgovor na rizik
- Kontrola rizika [20]

"Projektni rizik je neizvestan događaj ili okolnost (uslov) koji može da, ako pojavi, ima pozitivne ili negativne efekte na jedan ili više zadataka projekata, odnosno oblasti projekta poput opsega, redosleda, troškova ili kvaliteta. Rizik može imati jedan ili više uzroka, i kada se pojavio, može da ima jedan ili više posledica." [20]

Primer rizika bi bio da je potrebno da se nabavi neka dozovola za neki projekat. Rizik bi ovde bio slučaj da agenciji treba mnogo vremena da izda dozvolu. Zato je bitno planiranje odgovora u slučaju predvidljivih rizika. Rizike za koje nismo spremni moramo rešavati u toku samo projekta. [20]

Organizacije mogu da tolerišu određenu količinu rizika u zavisnosti od njihovog stava prima riziku. Na te stavove mogu uticati mnogi faktori, koji su klasifikovani u sledećoj podeli:

- Apetit rizika - koji stepen neizvesnosti je prihvatljiv, odnosno za koji stepen rizika je spreman da preuzme da bi dostigao željeni cilj.
- Tolerancija rizika - količina rizika koju organizacija ili pojedinac može da toleriše
- Granica rizika - odnosi se na količinu neizvesnosti ili nivo uticaja dati rizik ima na projekat. Ispod te granice, organizacija prihvata rizik, dok ako je rizik iznad definisane granice, organizacija ne toleriše rizik [20]

Pozitivni i negativni rizik se često drugačije mogu definisati kao prilike i pretnje (poteškoće). Ponekad, organizacije vrše pozitivan rizik ako je dobitak povoljan. Odnosi prema rizicima zavise od opažanja, tolerancije i drugih sklonosti, koje treba jasno definisati kad god je moguće. Da bi projekat bio uspešan, organizacije moraju da odgovaraju na rizike tokom izvršavanja celog projekta. [20]

3.3. Standardna dokumentacija u upravljanju projektima u okviru PRINCE 2 metodologije

Najpoznatija i najcenjenija metodologija za upravljanje projektima je Prince2 (Projects in Controlled Environments). Mnoge firme i organizacije iz raznih grana industrija koristi ovu metodologiju koja je fleksibilna i korisna pri organizovanju projekta bilo koje veličine ili složenosti. [24]

Projekti su detaljno već isplanirani još pre same implementacije projekta. [25] Metodologija je osnovana na sedam principa, tema i procesa i može se adaptirati za bilo koju situaciju ili zahtev. [24] Glavne prednosti ove metodologije je bolja kontrola resursa, bolji način kontrole rizika, lakše primećivanje problema, dobra komunikacija između tima i zainteresovanih strana kao i učenje na iskustvu. [26]

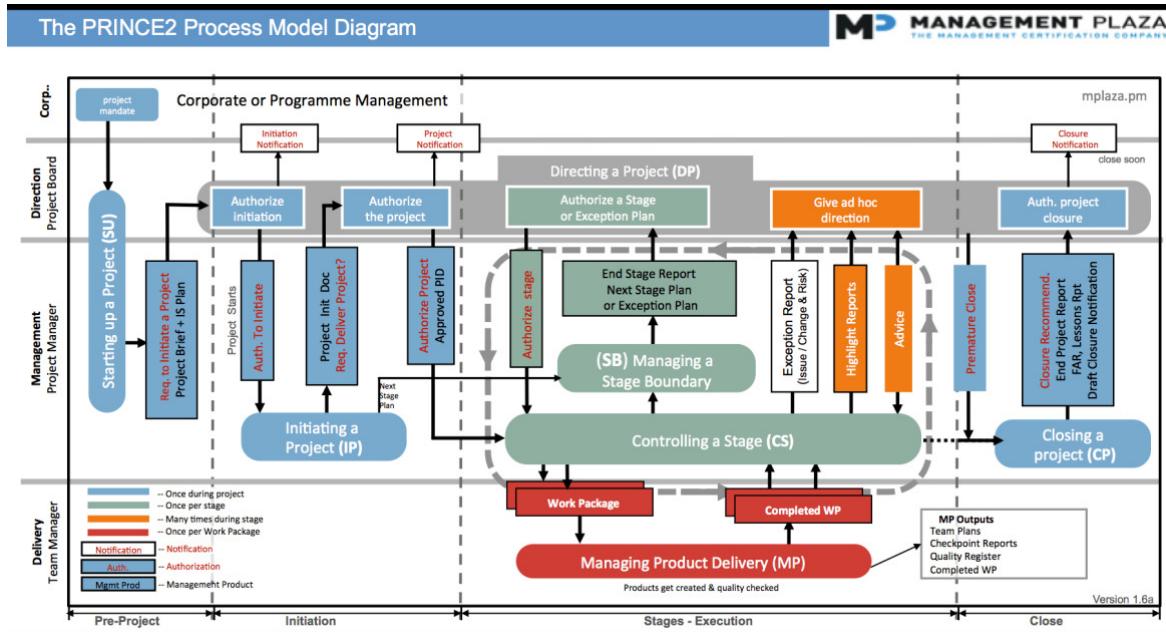
Principi u okviru PRINCE2 metodologije:

1. Mora da postoji razlog da projekat nastavi sa svojim postojanjem. Ako razloga nema, projekat treba zatvoriti
2. Timovi koji koriste metodologiju trebaju da uče na prethodnim iskustvima i da to znanje primene u budućim projektima
3. Timovi trebaju da imaju jasno definisanu strukturu i da uključe ljude koji su stručni u njihovoј struci
4. Projekti trebaju da budu podeljeni u već isplanirane etape koje treba kontrolisati i nadgledati
5. Članovima tima treba dati određeni deo ovlašćenja da mogu da rade unutar okruženja projekta
6. Projekat treba fokusirati na definiciju, isporuku i kvalitet proizvoda
7. Metodologija projekta treba da bude prilagođen prirodi okruženja, to se odnosi na veličinu, složenost, bitnost i rizike [24]

Teme u okviru PRINCE2 metodologije:

1. Potrebno je napraviti i održavati evidenciju o razlozima za stvaranje projekta
2. Definisati uloge pojedinca i odgovornosti za ceo tim
3. Definisati standarde kvaliteta i kako ih dostići
4. Definisati korake potrebne za razvoj plana i tehnika iz Prince2 metodologije koje treba koristiti
5. Identifikovati rizike i prilike koje mogu da imaju uticaj na projekat
6. Menadžer projekta treba da reaguje na promene i da vrši korekcije na projektu
7. Održivost projekta treba da utiče da li projekat treba nastaviti [24]

Grafikon na slici (Slika 12) prikazuje dijagram modela procesa Prince2 metodologije. Plavi pravougaonici predstavljaju procese koje treba izvršiti samo jedanput, zeleni pravougaonici predstavljaju procese koje treba izvršiti jedanput po etapi, narandžasti se mogu izvršavati više puta u etapi dok tamno crveni pravougaonici se mogu implementirati više puta unutar etape. [27]



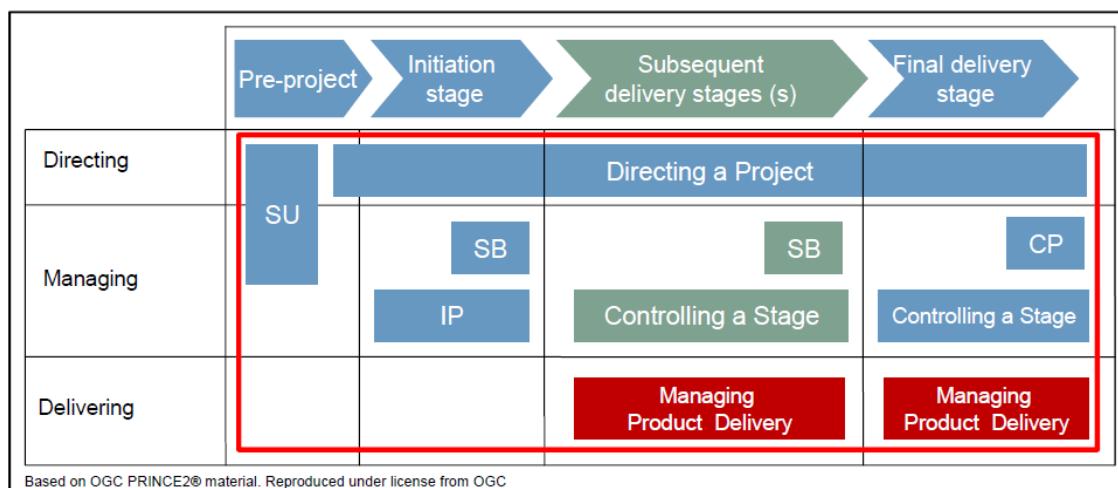
(Slika 12. - Dijagram Modela Procesa [49])

Na modelu procesa se vide četiri nivoa menadžmenta:

1. Nivo menadžmenta programa - za stvaranje mandata
2. Nivo usmerenja
3. Nivo menadžmenta projekta - oblast u kojoj se nalazi menadžer projekta, sadrži većinu aktivnosti vezanu za projekat, poput inicijalizacije i kontrole projekta
4. Nivo isporuke - nivo gde se stvaraju završni proizvodi projekta [27]

Prince2 metodologija poseduje sedam procesa koji su prikazani na slici (Slika 13), a oni su:

1. Započinjanje projekta - (SU) kratka faza koja sakuplja sve potrebne informacije za započinjanje projekta
2. Usmeravanje projekta - (DP) proces u kojоj glavni menadžer kontroliše projekat
3. Pokretanje projekta - (IP) ovaj proces ispituje razloge za postojanje projekta i stvara dokumentaciju za inicijalizaciju projekta, u kojoj se nalazi i plan samog projekta
4. Kontrola etapa - (CS) ovaj proces opisuje nadgledanje i kontrolu aktivnosti na dnevnom nivou od strane menadžera projekta
5. Upravljanje granica etapa - (SB) ovaj proces omogućuje kontrolisan način da se završi etapa i da se započne na planiranju sledeće
6. Upravljanje isporuke proizvoda - (MP) u ovom procesu nastaju proizvodi koji su napravili članovi tima
7. Zatvaranje projekta - (CP) ovo je poslednji proces i u njemu se potvrđuje isporuka proizvoda i menadžer projekta priprema projekat za zatvaranje [24][27]



(Slika 13. - Procesi u Prince2 [27])

U okviru PRINCE 2 metodologije upravljanja projektima, pojedini dokumenti PRINCE2 su dati kao obrasci/templejti i detaljnije opisani u [45]. U okviru PRINCE2 koriste se sledeći najvažniji opšti obrasci dokumenata (Prema verziji PRINCE2:2009 iz 2009. godine):

(Tabela 1. - Osnovni rezultati (proizvodi) primene PRINCE2: 2009 [45])

Prince2: 2009 Product	Old product or contributor
BASELINE	
Project Brief	Project Brief, Project Approach
Project Initiation Documentation	Project Initiation Document
Business Case	Business Case
Communication Management Strategy	Communications Plan
Quality Management Strategy	Project Quality Plan
Risk Management Strategy	None
Configuration Management Strategy	Configuration Management Plan
Project Product Description	Customer Quality Expectations, Acceptance Criteria
Work Package	Work Package
Benefits Review Plan	Post Project Review Plan
Plan (Project, Stage and Team)	Project, Stage, Exception Plan
Product Description	Product Description

(Tabela 2. - Zapisi i izveštaji prema metodologiji PRINCE2: 2009 [45])

Prince2: 2009 Product	Old product or contributor
RECORDS	
Daily Log	Daily Log
Lessons Log	Lessons Learned Log
Issue Register	Issue Log
Risk Register	Risk Log
Quality Register	Quality Log
Configuration Records	Configuration Item Records
REPORTS	
Highlight Report	Highlight Report
Checkpoint Report	Checkpoint Report
Exception Report	Exception Report
Issue Report	Project Issue/Request for Change/Off Spec
Lessons Report	Lessons Learned Report
Product Status Account	Product Status Account
End Stage Report	End Stage Report
End Project Report	End Project Report Follow-on Action Recommenda-

4. Postojeća Rešenja

"Softveri za upravljanjem projektima pomažu zaposlenima, administratorima i timovima da rukovode ciljeve, dugotrajne projekte i vrše koordinaciju zasebnih poslova. Programi dizajnirani za ovu svrhu nude mnoge alate da upravljaju poslovima, nadgledaju produktivnost i da dodeljuju resurse." [28]

"Zavisnost poslova je bitna komponenta softvera za upravljanje projektima, jer se može koristiti za koordinaciju dodeljenih poslova koji su vezani za zadatke unutar projekta. Korisnici koriste programi za upravljanje projektima da prate više projekata, prate napredak timova ili individualaca i analiziraju produktivnost tima." [28]

"Ovi softveri se mogu koristiti u bilo kojoj industriji koja zahteva poslove unutar projekta i ciljeve za nedeljne, mesečne ili godišnje projekte. Korisnicima je omogućeno da razbiju projekte u zasebne poslove, stvaraju vremenske linije i analiziraju productivnost kao i da komuniciraju sa drugim članovima tima." [28]

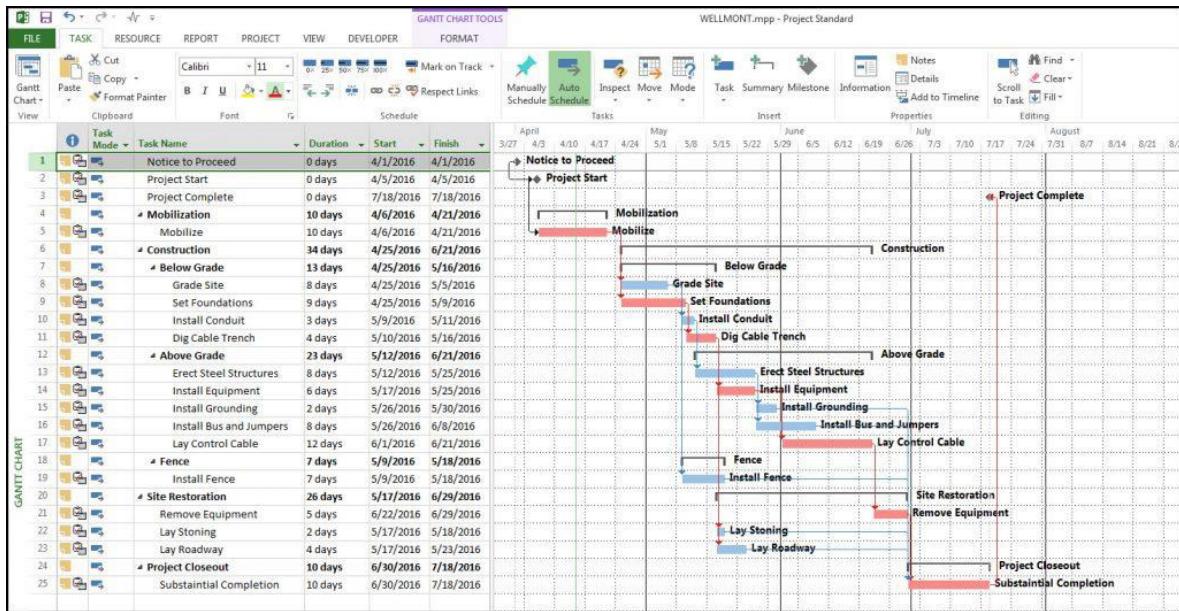
Rešenja za upravljanje projektima mogu da budu desktop rešenja, web rešenja ili čak softver za ličnu upotrebu u običnom životu za organizaciju i planiranje života. [29] Neke od tih softvera je nabrojano ispod:

Desktop softveri za upravljanje projektima

Microsoft Project

Microsoft Project je program dizajnirana od strane firme Microsoft i nudi mogućnost upravljanja projektima. Program ima mnoge alate koji omogućavaju bolje razumevanje i kontrola projekata, rasporeda i finansija, komunikacija među članovima tima i organizacija poslova i ljudi kako bi svi projekti bili završeni na vreme. [30] Vremenska linija koristi gantogram za prikaz aktivnosti (Slika 14). [32]

Program poseduje šablone prilikom kreiranja novih projekata sa mnogim opcijama za podešavanje. Novije verzije su integrisane sa Microsoft Office paketom i sa serverima tako da to omogućava deljenje informacija unutar projekta preko interneta, poput dodela dužnosti, deljenje rokova i planova. [31][32][33]



(Slika 14. - Izgled korisničkog interfejsa Microsoft Project-a [50])

Web rešenja za upravljanje projektima

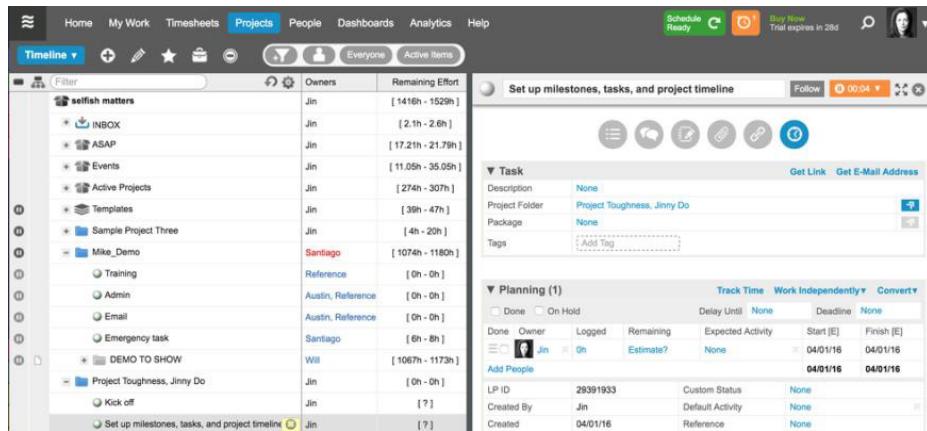
Wrike

Wrike je web aplikacija za upravljanje projektima dizajnirana da poveća brzinu i efikasnost posla. Program omogućava saradnju multi funkcionalnih grupa i usluga dozvoljava raspored, prioritizaciju i diskusiju kao i praćenje dostignuća projekta u realnom vremenu. [34]

Program poseduje korisnički-prijateljski dizajni koji olakšava interakciju sa aplikacijom. Wrike ujedno sadrži mnoge bezbednosne mere da osigura lične podatke. [34] Aplikacija je laka za učenje i ne zahteva mnogo obuke da bi korisnik bio spreman da je koristi. [35]

LiquidPlanner

LiquidPlanner je najobimniji program za upravljanje projektima koji je trenutno dostupan na tržištu. [36] Veoma je složen, pa zahteva duže vreme da korisnik savlada sve funkcije koje su proširne i moćne. [37]

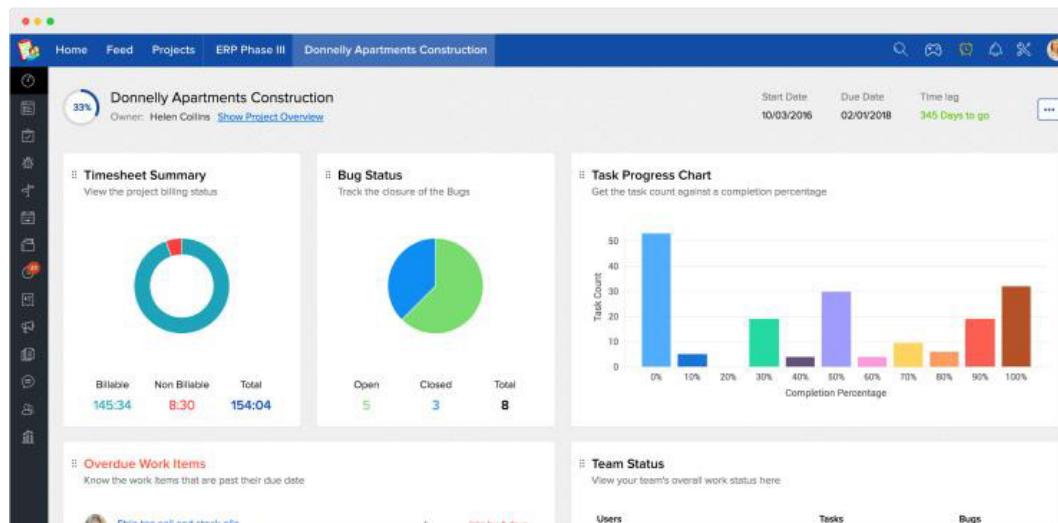


(Slika 15.- Interfejs Liquid Planner-a [37])

Program poseduje vremensku liniju koja se automatski osvežuje prilikom promene priorititeta. To ne samo što olakšava korisniku, nego mu štedi mnogo i na vremenu. [36] Pošto LiquidPlanner može da prati bilo kakav tip projekta, projekat ne mora da sadrži nikakve rokove (Slika 15). [37]

Zoho Projects

Zoho Projects je program za upravljanje projektima koji nudi neograničenu besplatnu uslugu, ali sa ograničenim funkcionalnošću. Interfejs programa je čist, jasan i lak za korišćenje. Interfejs se može dodatno podesiti prema želji korisnika (Slika 16). [38]



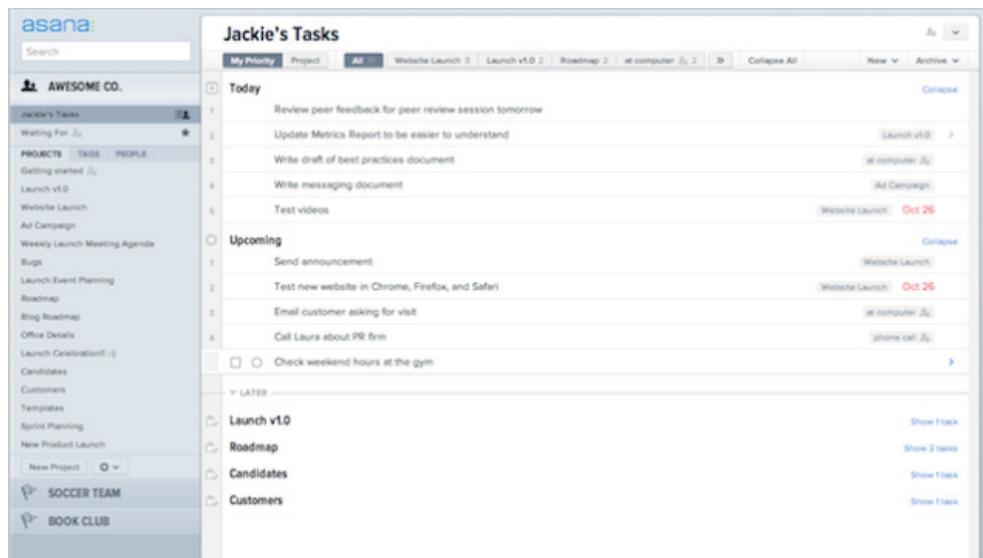
(Slika 16. - Korisnički interfejs programa Zoho Projects [38])

Program sadrži veoma dobar sistem komunikacije. Funkcija za čakanje (eng. chat) poseduje sistem za obaveštenje kada se dobije nova poruka i sama funkcija je stabilna i brza. Zoho Projects ujedno omogućava laku integraciju sa drugim internet servisima poput Dropbox ili Microsoft OneDrive. [38]

Softver za upravljanje ličnih projektima

Asana

Asana je hibridni tip programa za upravljanje poslova i projekata. Program je dostupan na mnogim platformama, poput iOS i Android telefonima. Korišćenje je lako i ima ugrađen fokus na saradnju sa više korisnika. [39]



(Slika 17. - Interfejs programa Asana [39])

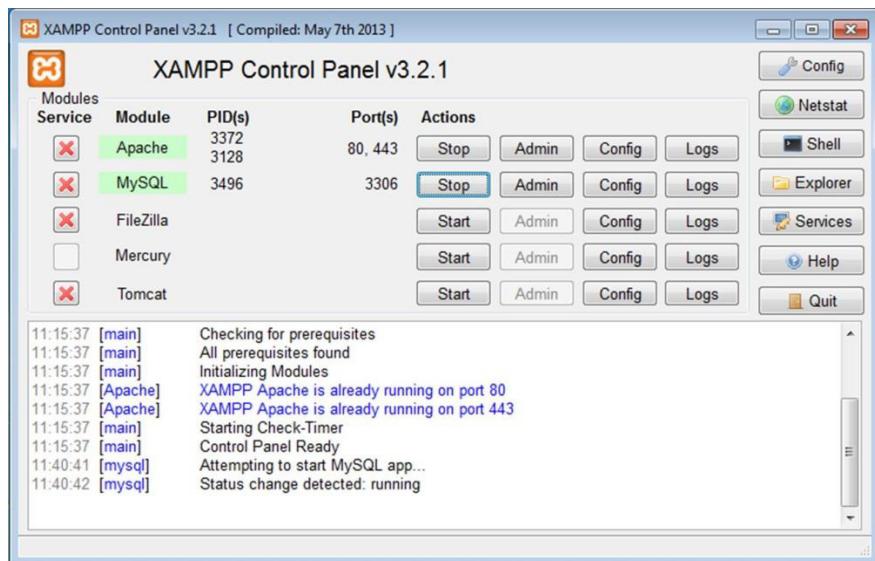
Asana omogućava lako strukturiranje ciljeva koja je prikazana kao lista poslova za strikliranje. Listu je moguće sortirati po datumu, i moguće je staviti zahteve da novi zadatak nemože da počne dok prethodni nije završen. Program omogućava i dodavanje detalja bilo kom poslu, poput kratkih poruka, web linkova ili komentara, što olakšava organizaciju (Slika 17). [39]

5. Opis korišćenih softverskih alata u realizaciji primera

Prilikom izrade ovog rada korišćen je programski jezik PHP. Internet stranice aplikacije se pokreću preko Apache web servera, a za bazu podataka se koristi MySQL sistem. Apache server i MySQL se može pokrenuti preko XAMPP aplikacije.

5.1.XAMPP

XAMPP je besplatna aplikacija koja sadrži paket više tehnologija poput Apache, MySQL, PHP i Perl. Pošto su te tehnologije već integrisane, nije potrebno zasebno ih instalirati. Instalacija je laka i nije čak ni potrebno imati sve service aktivne, nego svaki servis se može zasebno podešiti da se aktivira pri pokretanju aplikacije (Slika 18). [40]

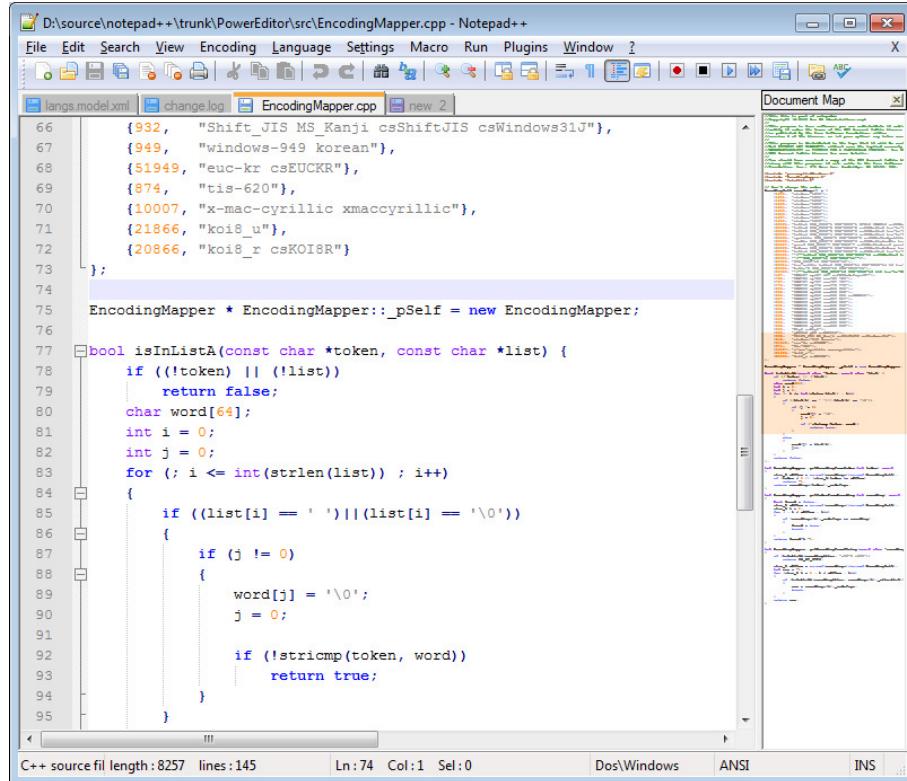


(Slika 18. - Izgled interfejsa XAMPP-a [51])

XAMPP je skraćenica od sledećih reči; Cross platform (X), Apache server (A), MariaDB (M), PHP (P) i Perl (P). [41]

5.2. Notepad ++

Notepad++ je besplatna aplikacija za obradu teksta koja je dizajnirana u cilju da bude zamena za Notepad program unutar Windows operativnog sistema (Slika 19). Program je napisan u C++, i veoma je kompaktan, efikasan i ne zauzima mnogo prostora na hard disku korisnika. [42]



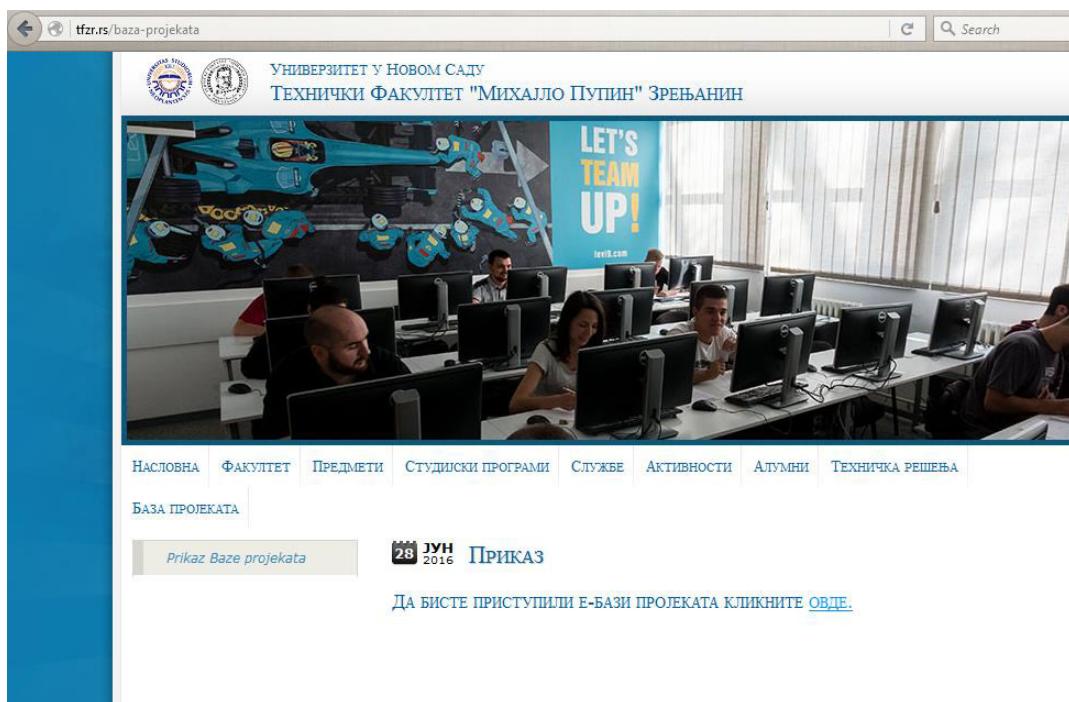
(Slika 19. - Korisnički interfejs Notepad++ [42])

Program sadrži mnoge alate koji su veoma korisni prilikom kucanja teksta. Glavna prednost ovog programa jeste što se može koristiti za obradu koda. Podržani su mnogi programski jezici poput HTML, PHP, C, C++, Java ili Pascal. Notepad++ sadrži opciju za dodavanje dodataka sa interneta koji omogućavaju dalje proširanje funkcionalnosti programa. [43]

6. Realizovani primer

6.1. Specifikacija zahteva

U okviru ovog rada kao potencijalni korisnik rešenja uzet je Tehnički fakultet "Mihajlo Pupin" Zrenjanin. U okviru specifikacije zahteva korisnika, realizovan je snimak stanja i utvđeno je da se trenutno na Tehničkom fakultetu evidencija projekata realizuje u formi MS Excel fajla koji je publikovan na web sajtu fakulteta. Slika 20. prikazuje sajt fakulteta u delu gde su predstavljeni projekti fakulteta.



(Slika 20. - Izgled postojaćeg dela sajta fakulteta za evidenciju projekata [44])

Taj deo sajta daje link ka MS Excel fajlu, koji sadrži sve podatke o projektima. U okviru ovog Excel fajla se vidi da su svi projekti opisani sličnim karakteristikama, samo su kategorisani prema vrstama: IPA projekti, Republički projekti, Pokrajinski projekti, Interni projekti. Takođe, razlikuju se i projekti koji su odobreni i u toku su realizacije, a posebno se vode projekti koji su samo prijavljeni (Slika 21).

(Slika 21. – Sadrzaj Excel fajla sa podacima o projektima [44])

Na osnovu datog rešenja izdvojeni su važni podaci koji treba da se evidentiraju u okviru budućeg rešenja web aplikacije.

Specifikacija zahteva korisnika obuhvata:

- Realizaciju web aplikacije za omogućavanje evidencije projekata – unos, brisanje, izmena, tabelarni prikaz, štampa spiska projekata
- Evidentiranje svih podataka koji su bili uključeni u evidenciju u okviru MS Excel fajla.

Podaci koji se evidentiraju u MS Excel fajlu su:

- Naziv projekta
- Trajanje projekta
- Partneri
- Rukovodilac
- Vrsta projekta
- Učesnici
- Sredstva projekta
- Oprema
- Inventarski broj
- Dokumentacija

Specifikacija poslovnih aktivnosti administratora:

(Tabela 3. - Specifikacija mogućnosti Administratora)

POSLOVNA AKTIVNOST	SOFTVERSKA PODRŠKA I DETALJI POSLOVNE AKTIVNOSTI
Pregled,izmena,brisanje i Dodavanje Vesti	Administrator ima mogućnost i obavezu da omogući objavljivanje najnovijih vesti na stranici kao i izmenu i brisanje već objavljenih
Pregled,izmena,brisanje i Dodavanje Konkursa	Mogućnost pregleda, izmene, brisanje i dodavanja novih konkursa koristeći resurse iz baze podataka
Pregled,izmena,brisanje i Dodavanje Projekata	Pregled,izmena,brisanje i dodavanje projekata na osnovu definisanih konkursa
Pregled,izmena,brisanje i Dodavanje Zaposlenih	Mogućnost pregleda i brisanja već postojačih članova, izmena njihovih podataka kao i mogućnost dodavanja novih
Prikaz Izveštaja	Prikaz spiska konkursa i projekata

Specifikacija poslovnih aktivnosti korisnika (posetilac sajta - bez prijavljivanja):

(Tabela 4. - Specifikacija mogućnosti posetiocioca)

POSLOVNA AKTIVNOST	SOFTVERSKA PODRŠKA I DETALJI POSLOVNE AKTIVNOSTI
Pregled Vesti	Mogućnost prikaza naj novijih vesti o dešavanjima na fakultetu vezanim za projekte
Pregled Konkursa	Korisnik ima mogućnost da vidi prikaz svih dostupnih i ne dostupnih konkursa
Pregled Projekata	Pregled svih projekata koji su u toku, sa datumom početka i kraja, kao i svim ostalim karakteristikama
Pregled Članova	Prikaz javno dostupnih podataka o korisnicima koji učestvuju na aktivnostima vezanim za projekte
Prikaz Izveštaja	Korisniku je prikazan generisan izveštaj na osnovu podataka iz baze podataka. Izveštaj je formatiran na način koji je prikidan za štampu

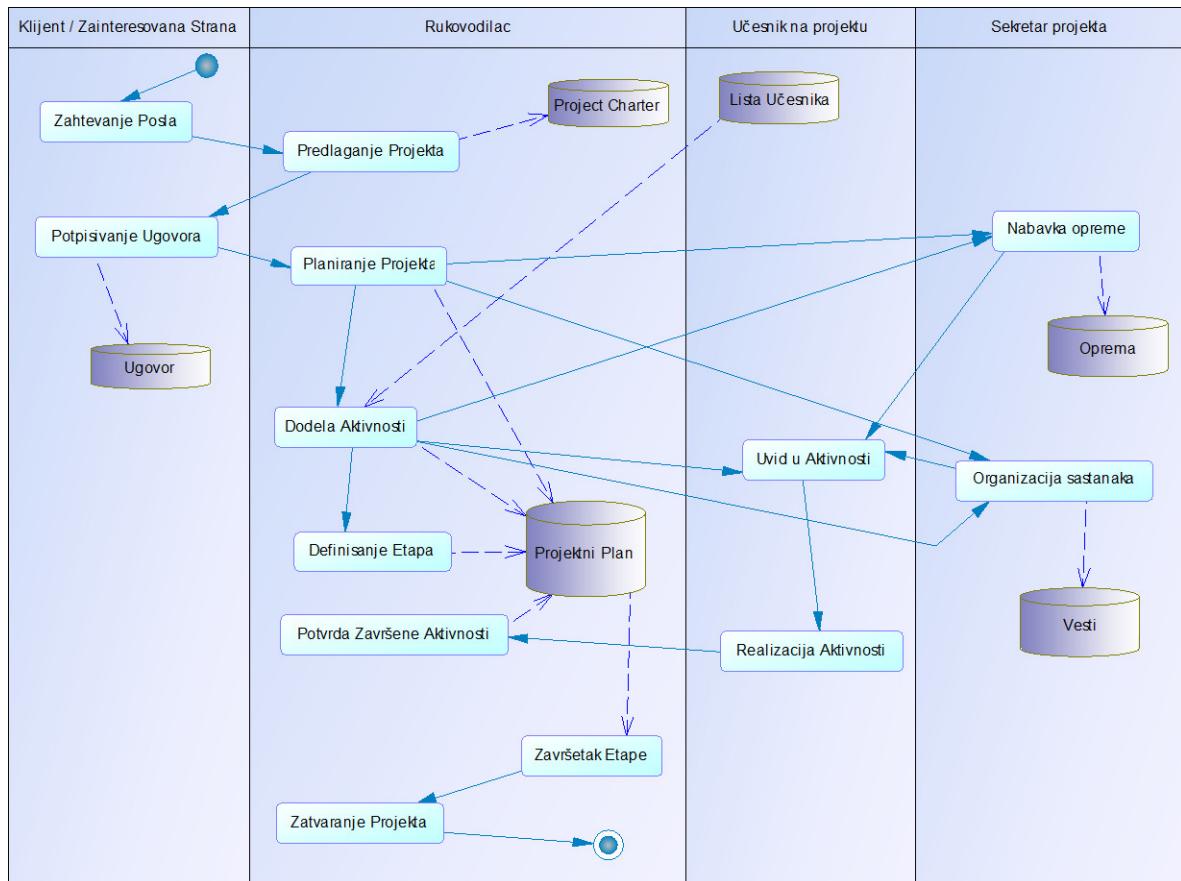
6.2. Opis posla

Opis posla predstavlja tekst opisa životnog ciklusa osnovnog objekta obrade u sistemu. U ovom kontekstu, osnovni ciklus se odnosi na životni ciklus projekta opšteg tipa. Realizacija projekta započinje fazom okvirnog koncipiranja u kom učestvuje zainteresovana strana i rukovodilac projekta. Nakon toga sledi faza planiranja, realizacije projekta i finalne aktivnosti projekta, nakon čega sledi zatvaranje projekta.

Detaljni prikaz i opis svih faza opšteg upravljanja projektima dat je u ranijem odeljku, odnosno u okviru standarda PMBOK.

6.3. Model poslovnog procesa

U nastavku je prikazan model poslovnih procesa koji se odvijaju u okviru pripreme, realizacije i završetka projekta opšteg tipa.



(Slika 22. - Izgled BPM upravljanje projektom opšteg tipa)

6.4. Modeli opšteg dizajna softvera

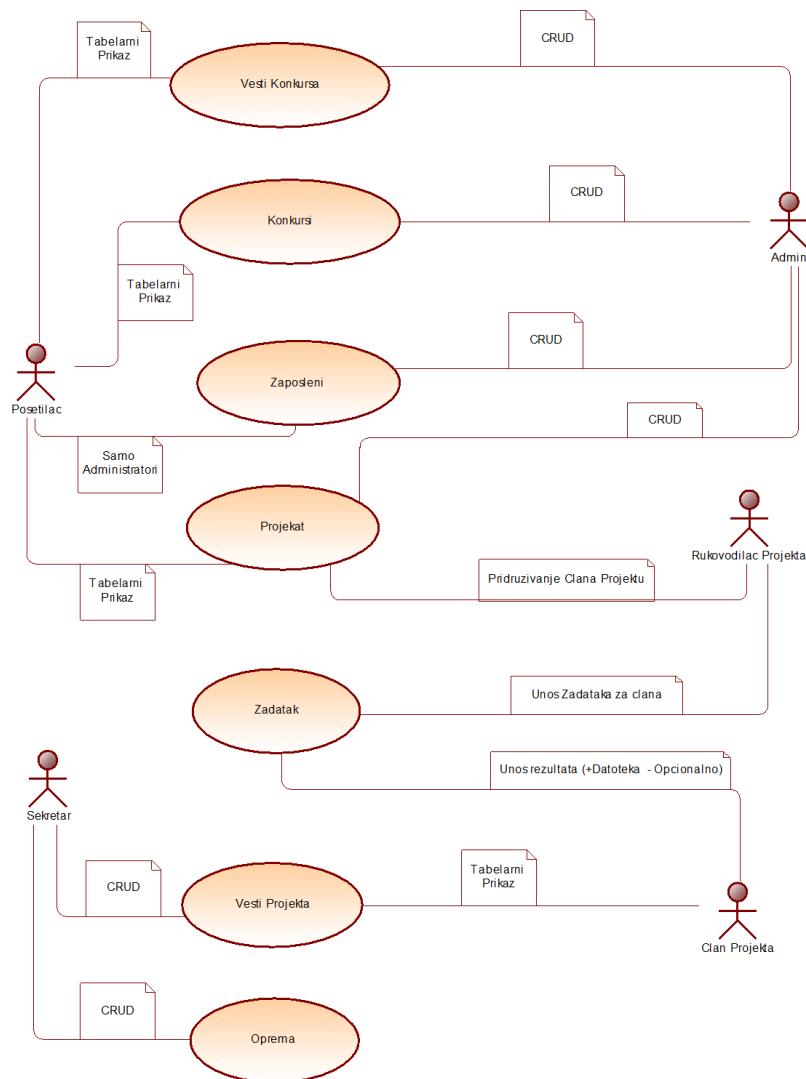
U ovom paragrafu će biti predstavljana tabela (ispod) u kojoj se nalaze primitivni procesi na osnovu BPM (Business Process Model) i softverske funkcije vezane za te procese. Funkcije su sortirane prema važnosti i prema redosledu dešavanja.

1. Pod primitivnim poslovnim procesima, smatraju se procesi administratora i korisnika
2. Radnu ulogu predstavlja izvršilac samog primitivnog poslovog procesa
3. Softverske funkcije su podeljene u dve kategorije prema prioritetu:
 - 3.1. Uloge prvog prioriteta direktno podržavaju proces
 - 3.2. Uloge drugog prioriteta proširuju osnovnu funkcionalnost
4. Lista prikazuje preduslove koje moraju biti ispunjeni da bi mogle osnovne funkcije da se primenjuju
5. U tabeli Aktor se nalaze vrednosti uloge korisnika za dati proces
6. Pod tipom softvera odnosi se platformu na kojoj će se odigravati proces

(Tabela 5. - BPM primitivnih procesa)

Primitivni poslovni proces	Radna uloga izvršioca primitivnog poslovnog procesa	Softverska Funkcija			Aktor (Profil korisnika softverske funkcije)	Tip softvera
		I prioriteta	II prioriteta	Preduslov za uspešno izvršavanje funkcija I prioriteta		
Započinjanje projekta	Rukovodilac projekta	Unos podataka o novom projektu i dodela rukovodioca projekta	Tabelarni prikaz podataka o ranije unetim projektima	Unos podataka o članovima projekata Unos podataka o konkursu	Administrator	Web aplikacija
	Rukovodilac projekta	Unos podataka o pridruživanju zaposlenog projektu	Tabelarni prikaz podataka o pridruženim zaposlenim projektu	Unos podataka o novom projektu i dodela rukovodioca projekta Unos podataka o zaposlenima		
Usmeravanje projekta	Rukovodilac projekta	Unos podataka o zadacima na projektu	Tabelarni prikaz zadatih zadataka	Unos podataka o novom projektu	Rukovodilac projekta	Web aplikacija
Realizacija zadatka na projektu	Ucesnik projekta	Unos podataka o realizovanom rezultatu povodom zadatog zadatka	Tabelarni prikaz unetih rezultata	Unos podataka o zadacima na projektu Tabelarni prikaz zadatih zadataka za clana projekta	Ucesnik projekta	Web aplikacija

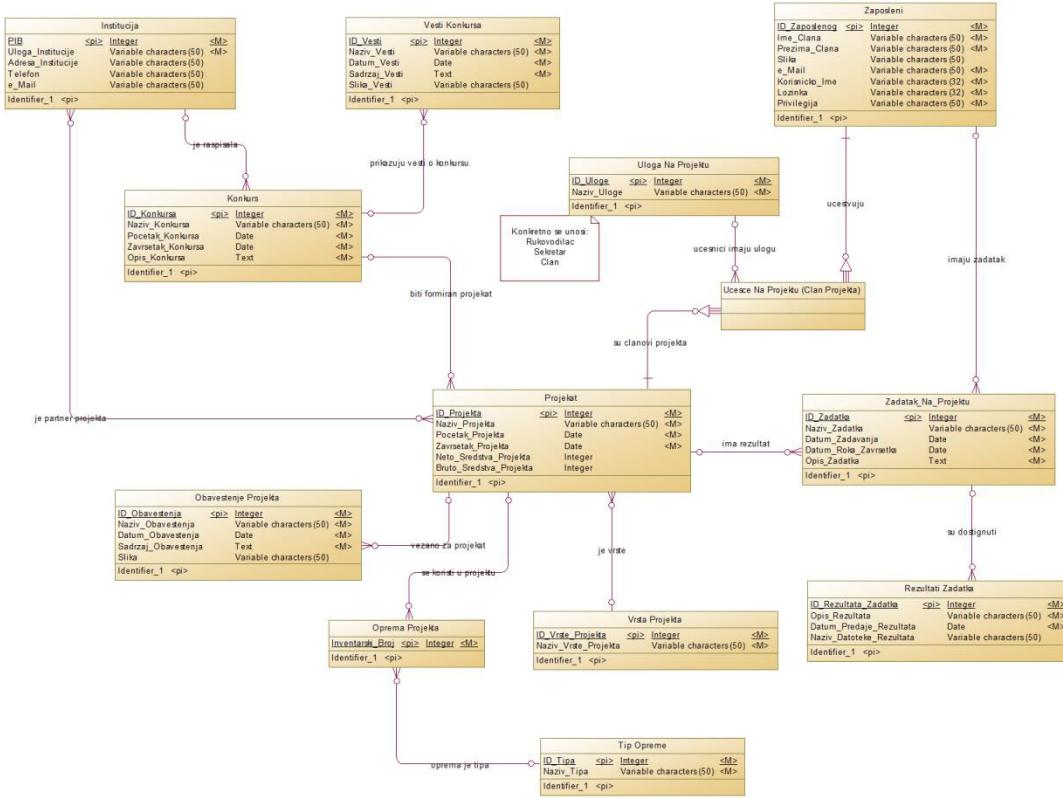
Nabavka opreme za projekat	Sekretar projekta	Unos podataka o novoj opremi	Tabelarni prikaz nabavljene opreme za projekat	Unos podataka o novom projektu (podaci o fondu)	Sekretar projekta	
Organizacija sastanaka u vezi projekta	Sekretar projekta	Unos vesti o projektu	Tabelarni prikaz vesti u vezi projekta	Unos podataka o novom projektu	Sekretar projekta	
Kontrola etapa	Rukovodilac projekta	Unos podataka o verifikaciji rezultata zadatka	Tabelarni prikaz urađenih zadataka Tabelarni prikaz ne urađenih zadataka	Unos podataka o zadacima na projektu Unos podataka o realizovanom rezultatu	Rukovodilac projekta	Web aplikacija
Upravljanje granica etapa	Rukovodilac projekta	Unos podataka o završetku etape projekata	Tabelarni prikaz završenih etapa Tabelarni prikaz ne završenih etapa	Tabelarni prikaz urađenih zadataka	Rukovodilac projekta	Web aplikacija
Upravljanje isporuke rezultata	Rukovodilac projekta	Štampanje izveštaja o rezultatima projekata	Štampanje izveštaja rezultata etape	Unos podataka o završetku etape projekata	Rukovodilac projekta	Web
Zatvaranje projekta	Rukovodilac projekta	Unos podataka o završetku projekata	Tabelarni prikaz završenih projekata	Unos podataka o završetku etape projekata	Rukovodilac projekta	Web aplikacija



(Slika 23. - USE CASE dijagram - Pregled softverskih funkcija i uloga)

6.5. Konceptualni model podataka

U nastavku je prikazan konceptualni model podataka koji odgovara specifikaciji zahteva, koja je izražena u formi Excel fajla. U prethodno prikazanom MS Excel fajlu data je struktura podataka koja je bitna za evidentiranje podataka o projektima na Tehničkom fakultetu "Mihajlo Pupin" Zrenjanin i može se smatrati opštom strukturu za evidentiranje bilokojih projekata. Ipak, postoje specifičnosti s obzirom na način finansiranja i neke druge detalje. Na osnovu konceptualnog modela podataka, koji je dat u nastavku, je realizovan sql upit i baza podataka koja se koristi u softveru.



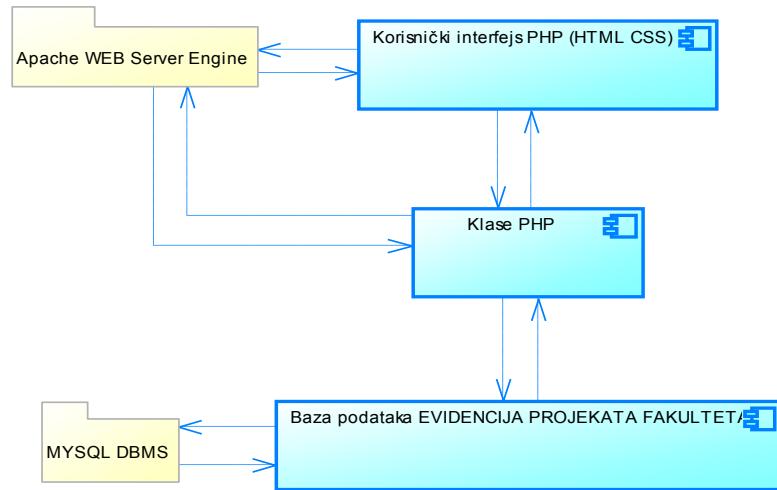
(Slika 24. - Prikaz konceptualnog modela)

6.6. Modeli implementiranog rešenja

U nastavku su prikazanadva najvažnija UML modela kojima se opisuje način implementacije softverskog rešenja - dijagram komponenti i dijagram razmeštaja.

6.6.1. Dijagramkomponenti

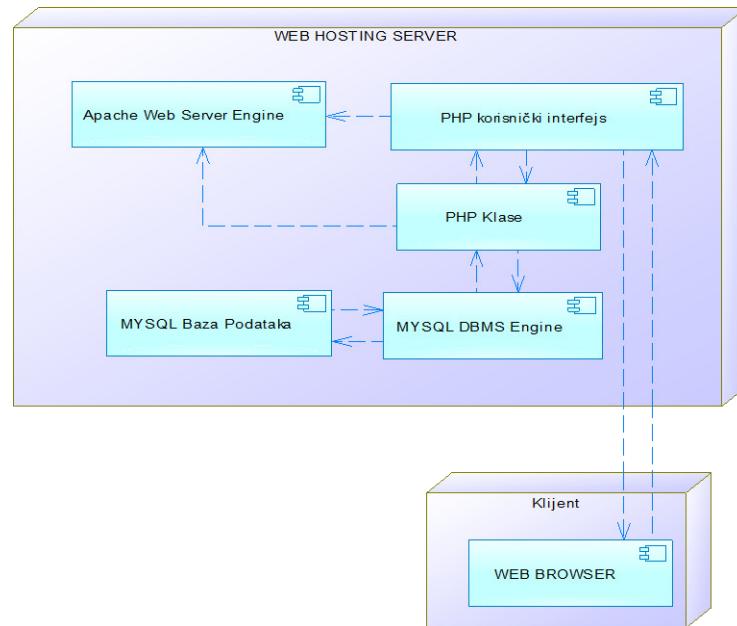
U nastavku je prikazan dijagram predstavlja opšti prikaz međusobnih odnosa softverskih komponenti, koji predstavlja osnovu implementacije koda same aplikacije.



(Slika 25. - Grafički prikaz dijagrama komponenti)

6.6.2. Dijagram razmeštaja

Dijagram razmeštaja se korisniti za prikaz fizičkih komponenti sistema (hardver) kao i softvera koji je instaliran na sistemu.



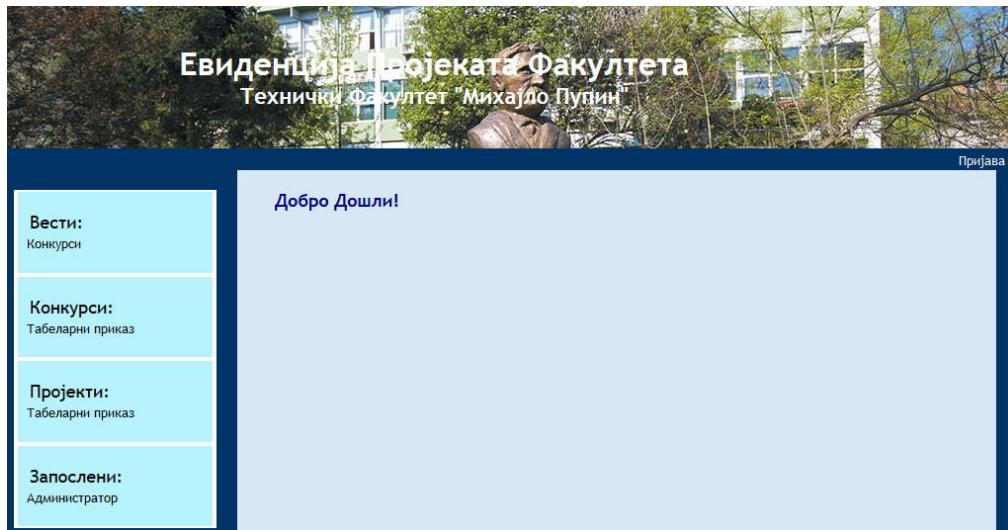
(Slika 26. - Prikaz dijagrama razmeštaja)

6.7. Korisničko uputstvo

Aplikacija koja je prikazana u ovom radu , nije u potpunosti realizovana. Delovi koji su realizovani su pridruživanje zaposlenih nekom projektu od strane rukovodioca. Ta funkcionalnost prikazuje stečeno znanje kreiranje više slojnosti, gde se osoba iz zaposlenih ne može pridružiti nekom projektu ako je deo više od prethodno definisanog broja projekata u eksternom XML dokumentu.

Aplikacija ujedno omogućava logovanje (prijavljivanje) i dva tipa privilegije, administrator i zaposleni. Prilikom logovanja osoba koji imaju privilegiju "zaposleni", aplikacija proverava da li taj zaposleni je deo nekih projekata.

Ako jeste osoba vrši biranje projekata za osobe koji su rukovodioci, sekretari ili članovi na nekim od projekata. Meni sa leve strane je personalizovan u zavisnosti od uloge i privilegije. Realizovani rad nudi i prikaz lista konkursa i projekata.



(Slika 27. - Prikaz "index.php" stranice koju vidi posetilac)

Prilikom posete na prezentaciju, korisniku se prikaže stranica "index.php" koja samo poseduje poruku "Dobro Došli". Sa leve strane se nalazi meni koji je isti za sve ne učlanjene korisnike, kao i opcija Prijava na gornjem delu korisničkog interfejsa za osobe koje imaju nalog na prezentaciji (Slika 27).

Prilikom odlaska na link pod Vesti – Konkursi, prikazuje se lista vesti vezana za sve konkurse (Slika 28).

The screenshot shows a web page titled "Евиденција на пројектата Факултета Технички Факултет 'Михајло Пупин'" (Evidence of the Faculty Project Technical Faculty "Mihajlo Pupin"). On the right, there is a "Prijava" button. On the left, there is a sidebar with four categories: "Вести" (News) with "Конкурси" (Competitions), "Конкурси" (Competitions) with "Табеларни приказ" (Table-based display), "Пројекти" (Projects) with "Табеларни приказ" (Table-based display), and "Запослени" (Employees) with "Администратор" (Administrator). The main content area displays five news items in boxes:

- 11.2.2018.**
Запоћео је конкурс "Меренje киселости"
Данас је raspisan konkurs za projekt Merenje Kiselosti
- 8.2.2018.**
Жељезница raspisala konkurs
Konkurs je raspisala Fabrika Sirčeta
- 6.2.2018.**
Соларне Плоће
Данас је raspisan konkurs за пројекат "Меренje светlosti" на кроу Zrenjaninske Gimnazije
- 4.2.2018.**
Planirano merenje влагности
Raspisan konkurs za projekt "Меренje Vlažnosti"
- 2.2.2018.**
Запоћео је конкурс "ArduinoZR"
Данас је raspisan konkurs за пројекат ArduinoZR. Конкурс је raspisao Tehnički fakultet "Mihajlo Pupin"

At the bottom, it says "Развијено за примену на Техничком Факултету "Михајло Пупин" 2018 *** Бранислав Ђекић".

(Slika 28. - Prikaz stranice "Spisak_Vesti_Konkursi_Index.php", prikazuje vesti vezane za konkurse)

Prezentacija nudi i tabelarni prikaz svih trenutnih konkursa koji su detaljno opisani na stranici "Spisak_Konkursa_Index.php". Tabela sadrži informacije poput, naziva, institucija koje se odnose na realizaciju konkursa kao i početak i završetak konkursa (Slika 29).

The screenshot shows a web page titled "Евиденција Пројектата Факултета" (Evidence of the Faculty Project) with a sub-header "Технички Факултет "Михајло Пупин"" (Technical Faculty "Mihajlo Pupin"). The page has a sidebar on the left with links: "Вести: Конкурси" (News: Competitions), "Конкурси: Табеларни приказ" (Competitions: Tabular presentation), "Пројекти: Табеларни приказ" (Projects: Tabular presentation), and "Запослени: Администратор" (Employees: Administrator). The main content area is titled "Конкурси:" (Competitions:) and contains a table with the following data:

Назив Конкурса	Институција	Почетак Конкурса	Завршетак Конкурса	Опис
Меренje влагности	Farma "Zlatni Orao"	1.2.2018.	25.2.2018.	Меренje влагности земљишта.
Меренje светlosti	Pekara "Golupčić"	1.2.2018.	26.2.2018.	Меренje светlosti krova.
Меренje киселости	Fabrika Sirćeta	1.2.2018.	21.2.2018.	Меренje киселости sirćeta.
Instalacija senzora	Picerija "Rim"	1.2.2018.	27.2.2018.	Postavljanje senzora na pruge Železnice.
Beleženje temperature i vlagnosti vazduha	Tehnički Fakultet "Mihajlo Pupin"	1.2.2018.	11.2.2018.	Beleženje temperature i vlagnosti vazduha preko senzora, postavljenih na Arduino ploče i povezane preko wi-fi mreže. Jedinice se napajaju preko solarnih ploča.

At the bottom of the page, it says "Развијено за примену на Техничком Факултету "Михајло Пупин" 2018 *** Бранислав Ђекић".

(Slika 29. - Prikaz stranice "Spisak_Konkursa_Index.php", prikazuje listu konkursa)

Ne prijavljeni korisnici mogu da vide osnovne informacije o trenutnim projektima na stranici "Spisak_Projekata_Index.php". Stranica nudi elementarne informacije o projektu poput naziva, tipa projekta, početak, završetak i ko je osoba zadužena za realizaciju projekta (Slika 30).

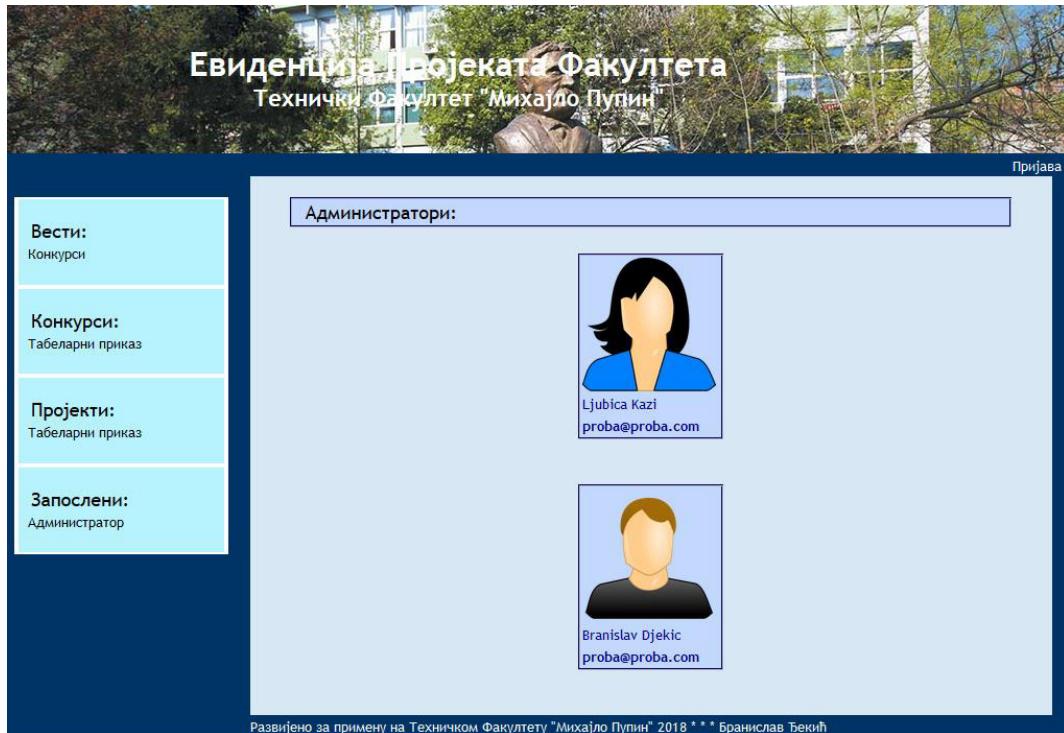
The screenshot shows a web page titled "Евиденција Пројектата Факултета" (Evidence of the Faculty Project) with a sub-header "Технички Факултет "Михајло Пупин"". The sidebar on the left includes the same links as Slika 29. The main content area is titled "Пројекти:" (Projects:) and contains a table with the following data:

Име Пројекта	Тип Пројекта	Почетак Пројекта	Завршетак Пројекта	Руководилац Пројекта
ArduinoZR	Republički	1.3.2018.	31.3.2018.	Zoran Zorić
Меренje киселости	Republički	2.3.2018.	21.4.2018.	Ana Anić
Меренje влагности	Republički	3.3.2018.	22.4.2018.	Dejan Dejanović
Меренje светlosti	Pokrajinski	4.3.2018.	23.4.2018.	Milos Milosic

At the bottom of the page, it says "Развијено за примену на Техничком Факултету "Михајло Пупин" 2018 *** Бранислав Ђекић".

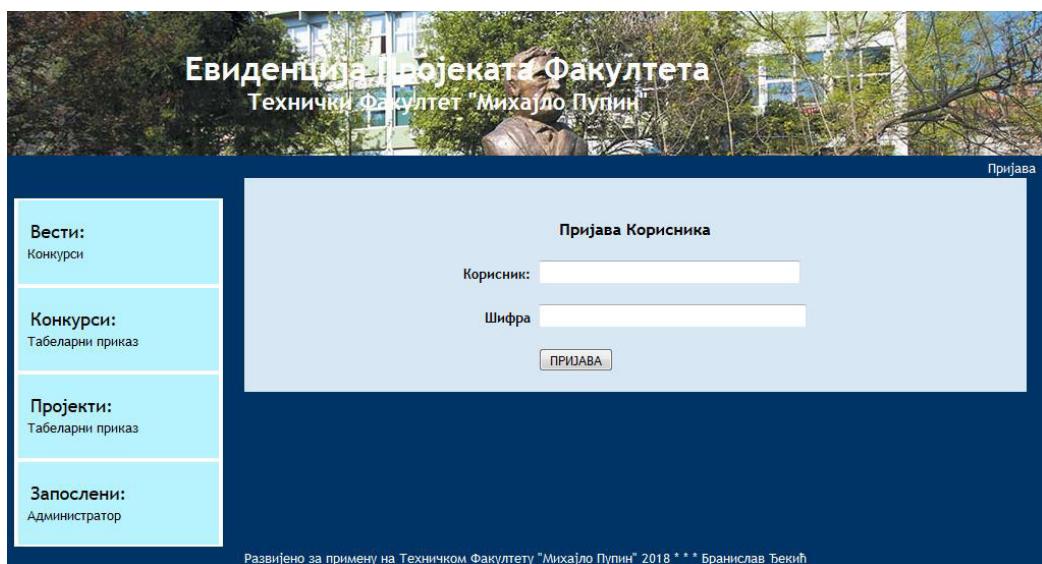
(Slika 30. - Prikaz stranice "Spisak_Projekata_Index.php", prikazuje listu projekata)

Administratori određeni za održavanje aplikacije mogu se videti na stranici "Administratori_Lista_Index.php" (Slika 31).



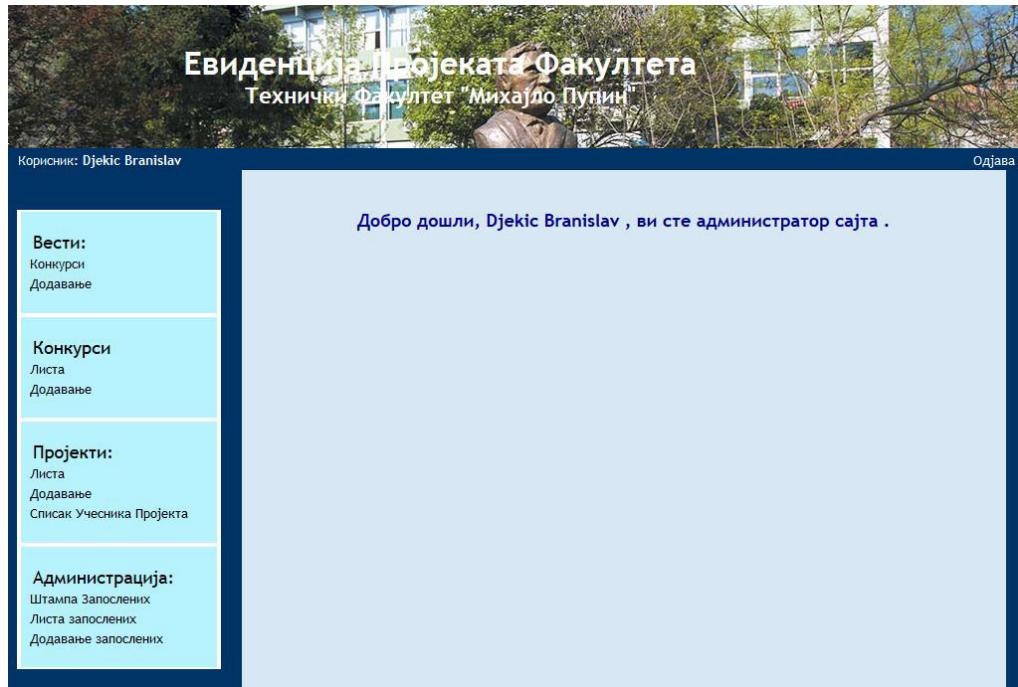
(Slika 31. - Prikaz stranice "Administratori_Lista_Index.php", prikazuje listu administratora aplikacije)

Prijavljivanje korisnika se vrši na stranici "Prijava_Korisnika.php". Stranici se može pristupiti preko linka reči "Prijava" u gornje-desnom uglu korisničkog interfejsa (Slika 32).



(Slika 32. - Prikaz stranice "Prijava_Korisnika.php", stranica za prijavljivanje korisnika)

Meni opcije sa leve strane su drugačije kao i početna strana u zavisnosti od uloge. Administrator nije deo ni jednog projekta. (Slika 33)



(Slika 33. Prikaz stranice "index_Prijavljen.php" za administratora)

Osobe koju su dodeljene projektu će posle uspešnog prijavljivanja biti prvo preusmerene na stranicu za izabir projekat (Slika 34), pa onda posle izabira će biti preusmereni ka personalizovanoj stranici.



Име Пројекта	Тип Пројекта	Почетак Пројекта	Завршетак Пројекта	Руководилац Пројекта	Изаберите
Merenje kiselosti	Republički	2.3.2018.	21.4.2018.	Rukovodilac	<input type="button" value="Изаберите"/>
Merenje vlastnosti	Republički	3.3.2018.	22.4.2018.	Sekretar	<input type="button" value="Изаберите"/>

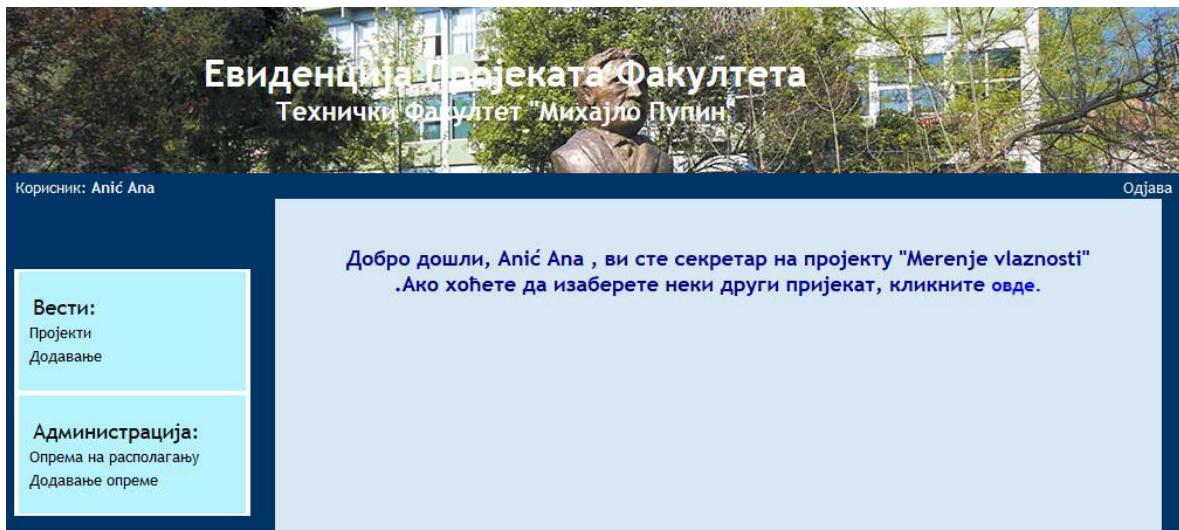
Развијено за примену на Техничком Факултету "Михајло Пупин" 2018 *** Бранислав Ђекић

(Slika 34. - Prikaz stranice "Izabir_Projekta.php" gde prijavljeni zaposleni može da izabere projekat u kome su oni član i ulogu koju imaju)

Rukovodioci, Članovi projekta i Sekretari, imaju mogućnost da se vrate na stranicu izabira projekta preko hiperlinka "ovde" na početnoj stranici. (Slike 35,36 i 37)



(Slika 35. Prikaz stranice "index_Prijavljen.php" za člana projekta, ovde se može videti hiperlink na reči "овде")

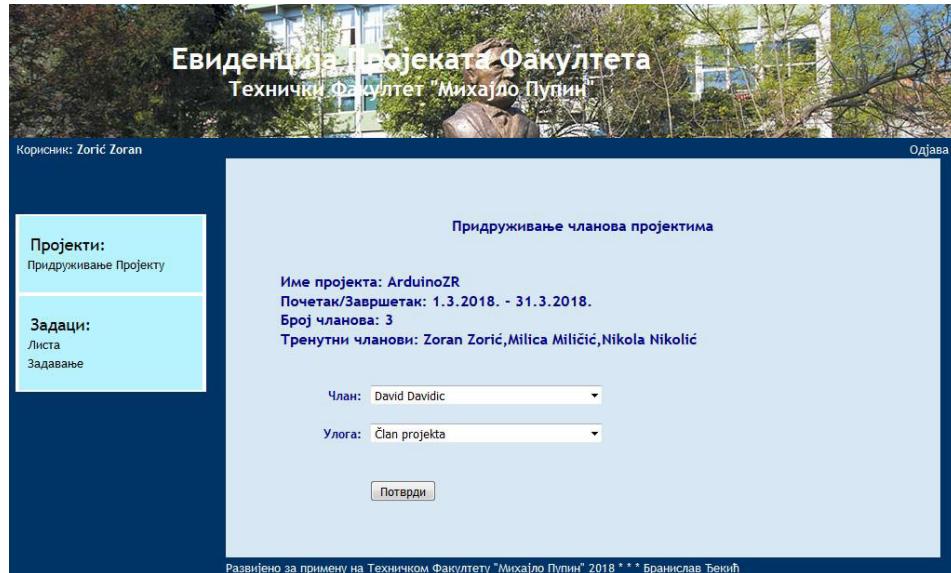


(Slika 36. Prikaz stranice "index_Prijavljen.php" za sekretara)

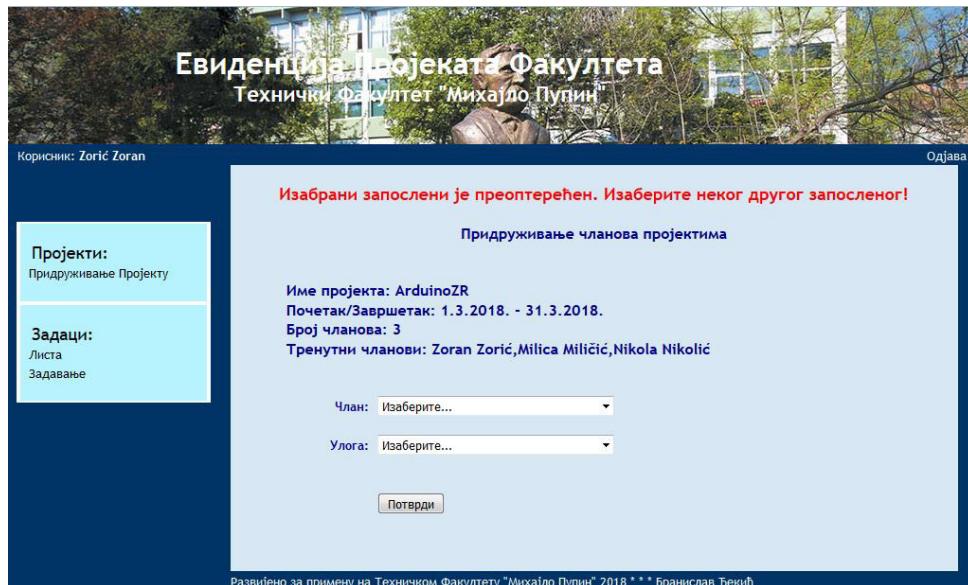


(Slika 37. Prikaz stranice "index_Prijavljen.php" za rukovodiooca)

Stranica za pridruživanje članovima projekta omogućava da rukovodilac projekta dodeli nove članove projektu za koje je rukovodilac odgovoran. Izabrana osoba nesme da bude na više od predefinisanog broja projekata. Broj projekata je definisano u eksternom dokumentu čija vrednost se može promeniti. (Slika 38) Detaljnije objašnjenje rada ove stranice će biti dato u poglavlju 6.8.5 "Poslovna Pravila".



(Slika 38. - Prikaz stranice "Pridruzivanje_Projektu_Index.php")



(Slika 39. - Prikaz poruke koja obaveštava korisnika da izabere nekog drugog zaposlenog, jer se izabrani već nalazi na velikom broju projekata)

Administrator ima mogućnost da prikaže detaljan prikaz liste projekata na stranici "Spisak_Projekata_Prijavljen_Index.php" i preko dugmeta „Prikaži“ prikazuje se sva oprema vezana za taj projekat. Projekti se mogu sortirati po njihovom tipu preko filter combo-boxa i pritiskom na "Filtriraj" dugme. Pritiskom na dugme "Svi" prikazuju se svi projekti. (Slika 40)

Евиденција Пројекта Факултета Технички Факултет "Михајло Пупин"									
Корисник: Djekic Branislav									
Одјава									
Пројекти:									
Тип пројекта: Изаберите тип... <input type="button" value="ФИЛТРИЈАЈ"/> <input type="button" value="СВИ"/>									
Пројекта	Име	Тип	Почетак	Завршетак	Руководилац	Нето	Бруто	Институција	Опрема
Пројекта	Пројекта	Пројекта	Пројекта	Пројекта	Пројекта	Средства	Средства	пројекта	
ArduinoZR	Republički	1.3.2018.	31.3.2018.	Zoran Zorić	20000	15000		Tehnički Fakultet "Mihajlo Pupin"	<input type="button" value="Прикажи"/>
Merenje kiselosti	Republički	2.3.2018.	21.4.2018.	Ana Anić	10000	5000		Fabrika Sirčeta	<input type="button" value="Прикажи"/>
Merenje vlažnosti	Republički	3.3.2018.	22.4.2018.	Dejana Dejanović	25000	6000		Farma "Zlatni Orao"	<input type="button" value="Прикажи"/>
Merenje svetlosti	Pokrajinski	4.3.2018.	23.4.2018.	Milos Milosic	23000	17000	N/A		<input type="button" value="Прикажи"/>
Merenje svetlosti	Pokrajinski	4.3.2018.	23.4.2018.	Milos Milosic	23000	17000	N/A		<input type="button" value="Прикажи"/>

(Slika 40. – Stranica "Spisak_Projekata_Prijavljen_Index.php")

Oprema koja je trenutno alocirana za neki projekat može se prikazati pritiskom na dugme "Prikaži" sa spiska projekata. Pritiskom se otvara stranica "Stampanje_Opreme_Index.php" koja prikazuje tabelarni prikaz sve opreme alocirane za izabrani projekat. Stranica je formatirana na način koji je prikladan štampanju. (Slika 41)

Инвентарски Број	Тип Опреме
11114	Телефони
33344	Laptop Računari

(Slika 41. – Stranica “Stampanje_Opreme_Index.php”)

Administratori mogu da vrše dodavanje novih projekata preko stranice "Kreiranje_Projekta_Index.php". Projekat se dodaje pritiskom na dugme "Potvrdi". (Slika 42)

(Slika 42. -Prikaz stranice za dodavanje projekata "Kreiranje_Projekta_Index.php")

Sekretar može obaviti menjanje,brisanje i dodavanje vesti vezane za projekat. Izmena i brisanje se poziva preko stranice "Vesti_Projekti_Lista_Index.php". (Slika 43)

The screenshot shows a web application interface titled "Евиденција Пројекта Факултета" (Evidence of Project Faculty) under the heading "Технички Факултет "Михајло Пупин"" (Technical Faculty "Mihajlo Pupin"). The user is logged in as "Корисник: Anić Ana". On the left, there are two sections: "Вести: Пројекти Додавање" (News: Projects Adding) and "Администрација: Опрема на располагању Додавање опреме" (Administration: Equipment on hand Adding). The main content area displays a table of news items:

Вест	Датум	Пројекат	Садржај	Слика	Измени/Обриши
Запоčeo је пројекат "Мerenje vlažnosti"	10.3.2018.	Меренje влажности	Рођено је пројекат "Меренje влажности".	Nema Slike	Измени Обриши
Учесници пројекта посетили фарму	12.3.2018.	Меренje влажности	Учесници пројекта "Меренje влажности" посетили фарму "Златни Орао".	Nema Slike	Измени Обриши
Опрема доделена до фарме	14.3.2018.	Меренje влажности	Опрема за меренje влажности доделена на фарму.	Nema Slike	Измени Обриши

At the bottom, a footer note reads: "Развијено за примену на Техничком Факултету "Михајло Пупин" 2018 *** Бранислав Ђекић".

(Slika 43. -Stranica "Vesti_Projekti_Lista_Index.php")

Dodavanje vesti se obavlja preko stranice "Vesti_Projekti_Dodavanje_Index.php". Posle popunjavanja forme, snimanje se potvrđuje pritiskom na dugme "Potvrdi".(Slika 44)

The screenshot shows the same web application interface as Slika 43. The left sidebar remains the same. The main content area is titled "Унос вести:" (Adding news:). It contains fields for "Назив Вести:" (Title News:) with a text input field, "Опис:" (Description:) with a large text area, and "Слика:" (Image:) with a file selection button ("Browse...") and a message "No file selected.". At the bottom is a "Потврди" (Confirm) button. A footer note at the bottom is identical to Slika 43.

(Slika 44. - Stranica "Vesti_Projekti_Dodavanje_Index.php")

Na stranici "Lista_Zaposlenih_Index.php" se može izvršiti parametarska štampa zaposlenih. Korisnik izabere koje grupe zaposlenih želi da prikaže, tako što otkači checkbox na stranici i posle toga pritisne dugme "Prikaži". (Slika 45)

The screenshot shows a web application interface titled 'Евиденција Пројектата Факултета' (Evidence of the Faculty Project) for the 'Технички Факултет "Михајло Пупин"' (Technical Faculty "Mihajlo Pupin").

On the left, there is a sidebar with the following menu items:

- Вести:** Конкурси, Додавање
- Конкурси:** Листа, Додавање
- Пројекти:** Листа, Додавање, Списак Учесника Пројекта
- Администрација:** Штампа Запослених, Листа запослених, Додавање запослених

The main content area has sections for 'Листа Запослених:' (List of Staff) and 'Прикази:' (Display). Under 'Прикази:', there are checkboxes for selecting staff roles:

- Администратори:
- Руководиоци: Пројекта:
- Секретари: Пројекта:

A note at the bottom states: 'Ако не изаберете ни један филтер, приказаће се сви запослени у бази' (If no filter is selected, all staff in the database will be displayed).

At the bottom of the page, it says 'Развијено за примену на Техничком Факултету "Михајло Пупин" 2018 * * * Бранислав Ђекић' (Developed for use at the Technical Faculty "Mihajlo Pupin" 2018 * * * Branislav Djekic).

(Slika 45. - Stranica "Lista_Zaposlenih_Index.php")

Na slici ispod je prikazana stranica prikladna za štampanje na kome su prikazani samo rukovodioци i osobe koje nisu deo ni jednog projekta. (Slika 46)

The screenshot shows a page titled 'Технички Факултет "Михајло Пупин"' (Technical Faculty "Mihajlo Pupin") with a circular logo on the left.

The main content area features a title 'ПРИКАЗ ИЗВЕШТАЈА' (Display Report) and a section titled 'ЛИСТА ЗАПОСЛЕНИХ' (List of Staff).

Below this, there is a table titled 'Руководиоци:' (Supervisors) with the following data:

Назив Запосленог	е-Мејл	Пројекат Учешћа
Zoran Zoric	proba@proba.com	ArduinoZR
Ana Anic	proba@proba.com	Merenje kiselosti
Dejana Dejanic	proba@proba.com	Merenje vlastnosti
Milos Milosic	proba@proba.com	Merenje svetlosti

There is also a section titled 'Запослени који нису на пројектима:' (Staff who are not on projects) with the following data:

Назив Запосленог	е-Мејл
Milan Milanic	proba@proba.com
Mila Milic	proba@proba.com

(Slika 46. - Stranica "Stampanje_Zaposlenih_Index.php")

Prikaz svih osoba koji učestvuju u realizaciji nekog projekta može se obaviti na stranici "Spisak_Ucesnika_Projekta_Index.php". (Slika 47)

(Slika 47. - Stranica "Spisak_Ucesnika_Projekta_Index.php")

Ispod je prikazana stranica prikladna za štampu na kojoj se nalaze svi članovi koji su deo izabranog projekta (Slika 48)

Учесник	Улога
Zoran Zorić	Rukovodilac
Milica Miličić	Sekretar
Nikola Nikolić	Član projekta

(Slika 48. - Prikaz stranice "Stampanje_Ucesnika_Index.php")

6.8. Opis implementacije

Ova aplikacija je realizovana na osnovu elemenata PHP programskog koda, koji je realizovanu okviru nastavnog predmeta "Informacioni Sistemi 2" (link ka resursu dat u Literaturi u poglavlju 8.4.), ali je realizovan na višeslojan način, tako da odgovara nastavnom predmetu "Softversko inženjerstvo 2". Drugi izvori koji su pomogli pri izradi rada su internet resursi koji su navedeni u poglavlju "Literatura / Resursi korišćeni pri izradi aplikacije".

6.8.1. Tabela sa prikazom cele arhitekture po slojevima

Ispod će biti prikazana opšta arhitektura višeslojne aplikacije u tabelarnom obliku. Ujedno će biti navedeno koji od ovih slojeva je podržano od strane ovog rada.

(Tabela 6. -Tabelarni prikaz višeslojne arhitekture)

Glavni Sloj	Podslojevi	Implementacija u radu
Prezentacioni sloj	Korisnički Interfejs	Aplikaciona forma koja se prikazuje na ekranu (HTML)
	Prezentaciona Logika	Kod za pripremu podataka i slanje do drugih slojeva koja se izvršava prilikom akcije "Form Action"
Sloj servisa	Web Servis	Nije realizovano
	Servisni Sloj kao Biblioteka Klasa	Nije realizovano
Sloj poslovne logike	Poslovni Objekti	Klase "cls_Clanovi_Projekta" i "cls_Opterecenje" i su poslovni objekti u ovom radu
	Poslovna Pravila	Klase cls_Opterecenje se poziva posle php koda Pridruzivanje_Projektu_Snimi.php i ona proverava da li zaposleni koji smo mi izabrali da bude deo izabranog projekta se nalazi na suviše projekata. Promenjiva preko koje se ustanavljava opterećenost se nalazi u eksternom XML dokumentu nazvan "Parametar_Opterecenja_Zaposlenog.xml" koji može biti menjan
	Radni Tokovi	Nije realizovano

Sloj za rad sa podacima	Klasa Modela (slicica sa dijagramom klasa – klasa: atributi, metode)	U ovom radu postoje dve tipa klase, klase tabele (klase specifične za preuzimanje prethodno unetih podataka iz baze podataka) kao i klase tipa DB, koje su predviđene za unos podataka iz aplikacije u trajnu bazu podataka
	Klasa Konekcija i Parametri	Ovaj rad koristi generični kod za uspostavljanje konekcije sa bazom podataka i DBMS. Taj kod se nalazi u klasi Konekcija, dok se parametri nalaze u XML dokumentu
	Baza Podataka	U ovom radu su realizovane tabele i njihove relacije

6.8.2. Sloj podataka - SQL Skript i šema baze podataka

Baza podataka koja se koristi u ovom radu se generiše preko više SQL upita, i unosi se na SQL opciji phpMyAdmin-a. To se izvršava na virtualnom serveru koji se pokreće preko XAMPP aplikacije. Ispod je prikazan kod koji je korišćen za generisanje baze "Evidencija_Projekata_Fakulteta" iz dokumenta MYSQL_Skript.sql. Dokument MYSQL_Skript.sql je dat uz projekat i nalazi su folderu "root/Baza Podataka (Skript)".

(Listing 1. - SQL Upit za generisanje baze, tabela i relacija između tabela (strani ključevi))

```
/* Kreiranje Baze */
CREATE DATABASE `Evidencija_Projekata_Fakulteta` CHARACTER SET utf8 COLLATE utf8_general_ci;

/* Kreiranje Tabela */
create table `Evidencija_Projekata_Fakulteta`.`INSTITUCIJA`
(
    PIB      int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ULOGA_INSTITUCIJE  varchar(50) not null,
    ADRESA_INSTITUCIJE  varchar(50),
    TELEFON      varchar(50),
    E_MAIL      varchar(50)
);

create table `Evidencija_Projekata_Fakulteta`.`JE_PARTNER_PROJEKTA`
(
    PIB      int NOT NULL,
    ID_PROJEKTA      int NOT NULL
);

alter table `Evidencija_Projekata_Fakulteta`.`JE_PARTNER_PROJEKTA`
add primary key (PIB, ID_PROJEKTA);
```

```

create table `Evidencija_Projekata_Fakulteta`.`KONKURS`
(
    ID_KONKURSA      int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    PIB              int,
    NAZIV_KONKURSA   varchar(50) not null,
    POCEТАK_KONKURSA date not null,
    ZAVRSETAK_KONKURSA date not null,
    OPIS_KONKURSA    text not null
);

create table `Evidencija_Projekata_Fakulteta`.`OBAVESTENJE_PROJEKTA`
(
    ID_OBAVESTENJA   int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ID_PROJEKTA     int,
    NAZIV_OBAVESTENJA  varchar(50) not null,
    DATUM_OBAVESTENJA date not null,
    SADRZAJ_OBAVESTENJA text not null,
    SLIKA           varchar(50)
);

create table `Evidencija_Projekata_Fakulteta`.`OPREMA_PROJEKTA`
(
    INVENTARSKI_BROJ  int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ID_PROJEKTA      int,
    ID_TIPA          int
);

create table `Evidencija_Projekata_Fakulteta`.`PROJEKAT`
(
    ID_PROJEKTA      int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ID_KONKURSA      int,
    ID_VRSTE_PROJEKTA int,
    NAZIV_PROJEKTA   varchar(50) not null,
    POCEТАK_PROJEKTA date not null,
    ZAVRSETAK_PROJEKTA date not null,
    NETO_SREDSTVA_PROJEKTA int,
    BRUTO_SREDSTVA_PROJEKTA int
);

create table `Evidencija_Projekata_Fakulteta`.`REZULTATI_ZADATKA`
(
    ID_REZULTATA_ZADATKA int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ID_ZADATKA          int,
    OPIS_REZULTATA     varchar(50) not null,
    DATUM_PREDAJE_REZULTATA date not null,
    NAZIV_DATOTEKE_REZULTATA varchar(50)
);

create table `Evidencija_Projekata_Fakulteta`.`TIP_OPREME`
(
    ID_TIPA          int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    NAZIV_TIPA       varchar(50) not null
);

create table `Evidencija_Projekata_Fakulteta`.`UCESCE_NA_PROJEKTU`
(
    ID_ZAPOSLENOG    int NOT NULL,
    ID_PROJEKTA     int NOT NULL,
    ID_ULOGE        int
);

alter table `Evidencija_Projekata_Fakulteta`.`UCESCE_NA_PROJEKTU`
add primary key (ID_ZAPOSLENOG, ID_PROJEKTA);

```

```

create table `Evidencija_Projekata_Fakulteta`.`ULOGA_NA_PROJEKTU`
(
    ID_ULOGE      int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    NAZIV_ULOGE   varchar(50) not null
);

create table `Evidencija_Projekata_Fakulteta`.`VESTI_KONKURSA`
(
    ID_VESTI      int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ID_KONKURSA   int,
    NAZIV_VESTI   varchar(50) not null,
    DATUM_VESTI   date not null,
    SADRZAJ_VESTI text not null,
    SLIKA_VESTI   varchar(50)
);

create table `Evidencija_Projekata_Fakulteta`.`VRSTA_PROJEKTA`
(
    ID_VRSTE_PROJEKTA  int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    NAZIV_VRSTE_PROJEKTA varchar(50) not null
);

create table `Evidencija_Projekata_Fakulteta`.`ZADATAK_NA_PROJEKTU`
(
    ID_ZADATKA      int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ID_PROJEKTA     int,
    ID_ZAPOSLENOG   int,
    NAZIV_ZADATKA   varchar(50) not null,
    DATUM_ZADAVANJA date not null,
    DATUM_ROKA_ZAVRSETKA date not null,
    OPIS_ZADATKA    text not null
);

create table `Evidencija_Projekata_Fakulteta`.`ZAPOSLENI`
(
    ID_ZAPOSLENOG      int NOT NULL AUTO_INCREMENT PRIMARY KEY,
    IME_ZAPOSLENOG     varchar(50) not null,
    PREZIME_ZAPOSLENOG varchar(50) not null,
    URL_SLIKE          varchar(50),
    E_MAIL             varchar(50)      not null,
    KORISNICKO_IME     varchar(32) not null,
    LOZINKA            varchar(32) not null,
    PRIVILEGIJA        varchar(50)
);

/* Strani Kljucevi */

alter table `Evidencija_Projekata_Fakulteta`.`JE_PARTNER_PROJEKTA` add constraint FK_JE_PARTNER_PROJEKTA foreign key (PIB)
    references `Evidencija_Projekata_Fakulteta`.`INSTITUCIJA` (PIB) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`JE_PARTNER_PROJEKTA` add constraint FK_JE_PARTNER_PROJEKTA2 foreign key (ID_PROJEKTA)
    references `Evidencija_Projekata_Fakulteta`.`PROJEKAT` (ID_PROJEKTA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`KONKURS` add constraint FK_JE_RASPISALA foreign key (PIB)
    references `Evidencija_Projekata_Fakulteta`.`INSTITUCIJA` (PIB) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`OBAVESTENJE_PROJEKTA` add constraint FK_VEZANO_ZA_PROJEKAT foreign key (ID_PROJEKTA)
    references `Evidencija_Projekata_Fakulteta`.`PROJEKAT` (ID_PROJEKTA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`OPREMA_PROJEKTA` add constraint FK_OPREMA_JE_TIPA foreign key (ID_TIPA)
    references `Evidencija_Projekata_Fakulteta`.`TIP_OPREME` (ID_TIPA) on delete restrict on update restrict;

```

```

alter table `Evidencija_Projekata_Fakulteta`.`OPREMA_PROJEKTA` add constraint FK_SE_KORISTI_U_PROJEKTU foreign key
(ID_PROJEKTA)
    references `Evidencija_Projekata_Fakulteta`.`PROJEKAT` (ID_PROJEKTA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`PROJEKAT` add constraint FK_BITI_FORMIRAN_PROJEKAT foreign key
(ID_KONKURSA)
    references `Evidencija_Projekata_Fakulteta`.`KONKURS` (ID_KONKURSA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`PROJEKAT` add constraint FK_JE_VRSTE foreign key (ID_VRSTE_PROJEKTA)
    references `Evidencija_Projekata_Fakulteta`.`VRSTA_PROJEKTA` (ID_VRSTE_PROJEKTA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`REZULTATI_ZADATKA` add constraint FK_SU_DOSTIGNUTI foreign key
(ID_ZADATKA)
    references `Evidencija_Projekata_Fakulteta`.`ZADATAK_NA_PROJEKTU` (ID_ZADATKA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`UCESCE_NA_PROJEKTU` add constraint FK_SU_CLANOVI_PROJEKTA foreign key
(ID_PROJEKTA)
    references `Evidencija_Projekata_Fakulteta`.`PROJEKAT` (ID_PROJEKTA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`UCESCE_NA_PROJEKTU` add constraint FK_UCESNICI_IMAJU_ULOGLU foreign key
(ID_ULOGE)
    references `Evidencija_Projekata_Fakulteta`.`ULOGLA_NA_PROJEKTU` (ID_ULOGE) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`UCESCE_NA_PROJEKTU` add constraint FK_UCESTVUJU foreign key
(ID_ZAPOSLENOG)
    references `Evidencija_Projekata_Fakulteta`.`ZAPOSLENI` (ID_ZAPOSLENOG) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`VESTI_KONKURSA` add constraint FK_PRIKAZUJU_VESTI_O_KONKURSU foreign key
(ID_KONKURSA)
    references `Evidencija_Projekata_Fakulteta`.`KONKURS` (ID_KONKURSA) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`ZADATAK_NA_PROJEKTU` add constraint FK_IMAJU_ZADATAK foreign key
(ID_ZAPOSLENOG)
    references `Evidencija_Projekata_Fakulteta`.`ZAPOSLENI` (ID_ZAPOSLENOG) on delete restrict on update restrict;

alter table `Evidencija_Projekata_Fakulteta`.`ZADATAK_NA_PROJEKTU` add constraint FK_IMA_REZULTAT foreign key
(ID_PROJEKTA)
    references `Evidencija_Projekata_Fakulteta`.`PROJEKAT` (ID_PROJEKTA) on delete restrict on update restrict;

```

Prethodni SQL skript kreira bazu podataka sa tabelama i relacijama. Ispod je prikazan kod za punjenje baze podataka sa probnim podacima.

(Listing 2. - SQL Upit za punjenje baze sa probnim podacima)

```

INSERT      INTO      `evidencija_projekata_fakulteta`.`zaposleni`      (ID_ZAPOSLENOG,      `IME_ZAPOSLENOG`,
`PREZIME_ZAPOSLENOG`, `URL_SLIKE`, `E_MAIL`, `KORISNICKO_IME`, `LOZINKA`, `PRIVILEGIJA`)
VALUES
('1', 'Ljubica', 'Kazi', 'FM_default_Icon.png', 'proba@proba.com', 'lkazi', 'lk', 'administrator'),
('2', 'Branislav', 'Djekic', 'M_default_Icon.png', 'proba@proba.com', 'branislav', 'branislav', 'administrator'),
('3', 'Zoran', 'Zoric', 'M_default_Icon.png', 'proba@proba.com', 'Zoran', 'Zoran', 'zaposleni'),
('4', 'Milica', 'Milicic', 'FM_default_Icon.png', 'proba@proba.com', 'Milica', 'Milica', 'zaposleni'),
('5', 'Nikola', 'Nikolic', 'M_default_Icon.png', 'proba@proba.com', 'Nikola', 'Nikola', 'zaposleni'),
('6', 'Ana', 'Anic', 'FM_default_Icon.png', 'proba@proba.com', 'Ana', 'Ana', 'zaposleni'),
('7', 'Petar', 'Peric', 'M_default_Icon.png', 'proba@proba.com', 'Petar', 'Petar', 'zaposleni'),
('8', 'Milan', 'Milanic', 'M_default_Icon.png', 'proba@proba.com', 'Milan', 'Milan', 'zaposleni'),
('9', 'Dejana', 'Dejanic', 'FM_default_Icon.png', 'proba@proba.com', 'Dejana', 'Dejana', 'zaposleni'),
('10', 'Milos', 'Milosic', 'M_default_Icon.png', 'proba@proba.com', 'Milos', 'Milos', 'zaposleni'),
('11', 'Mila', 'Milic', 'FM_default_Icon.png', 'proba@proba.com', 'Mila', 'Mila', 'zaposleni'),
('12', 'David', 'Davidic', 'M_default_Icon.png', 'proba@proba.com', 'David', 'David', 'zaposleni');

INSERT INTO `evidencija_projekata_fakulteta`.`uloga_na_projektu` (ID_ULOGE, `NAZIV_ULOGE`)
VALUES
('1', 'Rukovodilac'),
('2', 'Sekretar'),
('3', 'Clan projekta');

```

```

INSERT INTO `evidencija_projekata_fakulteta`(`tip_opreme`(`ID_TIPA`, `NAZIV_TIPA`)
VALUES
(1, 'Sredstva za pisanje'),
(2, 'Papiri'),
(3, 'Projektori'),
(4, 'Telefoni'),
(5, 'Racunari'),
(6, 'Laptop Racunari');

INSERT INTO `evidencija_projekata_fakulteta`.`vrsta_projekta`(`ID_VRSTE_PROJEKTA`, `NAZIV_VRSTE_PROJEKTA`)
VALUES
(1, 'Republicki'),
(2, 'Pokrajinski'),
(3, 'Interni');

INSERT INTO `evidencija_projekata_fakulteta`.`institucija`(`PIB`, `ULOGA_INSTITUCIJE`, `ADRESA_INSTITUCIJE`,
`TELEFON`, `E_MAIL`)
VALUES
(1, 'Tehnicki Fakultet "Mihajlo Pupin"', 'Đure Đakovica bb', '023 550 515', 'dekant@proba.com'),
(2, 'Fabrika Sirceta', 'Nikola Nikolicica 33', '023 000 000', 'proba@proba.com'),
(3, 'Farme "Zlatni Orao"', 'Milosha Mišica 22', '023 000 000', 'proba@proba.com'),
(4, 'Pekara "Golupcic"', 'Ana Anica 156', '023 000 000', 'proba@proba.com'),
(5, 'Picerija "Rim"', 'Petra Petrica 11', '023 000 000', 'proba@proba.com'),
(6, 'Srednja Škola "Gimnazija"', 'Gimnazijска BB', '023 000 000', 'proba@proba.com'),
(7, 'Železnica Srbije', 'Đorda Đorđevica 1', '023 000 000', 'proba@proba.com');

INSERT INTO `evidencija_projekata_fakulteta`.`konkurs`(`ID_KONKURSA`, `PIB`, `NAZIV_KONKURSA`,
`POCETAK_KONKURSA`, `ZAVRSETAK_KONKURSA`, `OPIS_KONKURSA`)
VALUES
(1, '1', 'Beleženje temperature i vlažnosti vazduha', '2018-02-01', '2018-02-11', 'Beleženje temperature i vlažnosti vazduha preko senzora, postavljenih na Arduino ploce i povezane preko wi-fi mreže. Jedinice se napajaju preko solarnih ploca.'),
(2, '2', 'Merenje kiselosti', '2018-02-01', '2018-02-21', 'Merenje kiselosti sirceta.'),
(3, '3', 'Merenje vlažnosti', '2018-02-01', '2018-02-25', 'Merenje vlažnosti zemljišta.'),
(4, '6', 'Merenje svetlosti', '2018-02-01', '2018-02-26', 'Merenje svetlosti krova.'),
(5, '7', 'Instalacija senzora', '2018-02-01', '2018-02-27', 'Postavljanje senzora na pruge železnice.');

INSERT INTO `evidencija_projekata_fakulteta`.`vesti_konkursa`(`ID_VESTI`, `ID_KONKURSA`, `NAZIV_VESTI`,
`DATUM_VESTI`, `SADRZAJ_VESTI`, `SLIKA_VESTI`)
VALUES
(1, '1', 'Zapoceo je konkurs "ArduinoZR"', '2018-02-02', 'Danas je raspisano konkurs za projekt ArduinoZR. Konkurs je raspisao Tehnicki fakultet "Mihajlo Pupin".'),
(2, '2', 'Zapoceo je konkurs "Merenje kiselosti"', '2018-02-11', 'Danas je raspisano konkurs za projekt Merenje Kiselosti', ''),
(3, '3', 'Planirano merenje vlažnosti', '2018-02-04', 'Raspisano konkurs za projekt "Merenje Vlažnosti",'),
(4, '4', 'Solarne Ploce', '2018-02-06', 'Danas je raspisano konkurs za projekt "Merenje svetlosti" na krovu Zrenjaninske Gimnazije', ''),
(5, '5', 'Železnica raspisala konkurs', '2018-02-08', 'Konkurs je raspisala Fabrika Sirceta', '');

INSERT INTO `evidencija_projekata_fakulteta`.`projekat`(`ID_PROJEKTA`, `ID_KONKURSA`, `ID_VRSTE_PROJEKTA`,
`NAZIV_PROJEKTA`, `POCETAK_PROJEKTA`, `ZAVRSETAK_PROJEKTA`, `NETO_SREDSTVA_PROJEKTA`,
`BRUTO_SREDSTVA_PROJEKTA`)
VALUES
(1, '1', '1', 'ArduinoZR', '2018-03-01', '2018-03-31', '20000', '15000'),
(2, '2', '1', 'Merenje kiselosti', '2018-03-02', '2018-04-21', '10000', '5000'),
(3, '3', '1', 'Merenje vlažnosti', '2018-03-03', '2018-04-22', '25000', '6000'),
(4, '4', '2', 'Merenje svetlosti', '2018-03-04', '2018-04-23', '23000', '17000');

```

```

INSERT INTO `evidencija_projekata_fakulteta`.`oprema_projekta` (`INVENTARSKI_BROJ`, `ID_PROJEKTA`, `ID_TIPA`)
VALUES
('12345', NULL, '1'),
('22334', '1', '5'),
('45454', '2', '2'),
('12367', '3', '3'),
('61515', NULL, '5'),
('11114', '4', '4'),
('33344', '4', '6');

INSERT INTO `evidencija_projekata_fakulteta`.`obavestenje_projekta` (`ID_OBAVESTENJA`, `ID_PROJEKTA`, `NAZIV_OBAVESTENJA`, `DATUM_OBAVESTENJA`, `SADRZAJ_OBAVESTENJA`, `SLIKA`)
VALUES
(1, '1', 'Pocetak', '2018-03-06', 'Poceo je projekat ArdinoZR.', ''),
(2, '1', 'Uspešno izvršeno merenje', '2018-03-07', 'Uspešno izvršeno merenje na projektu "Arduino ZR".', ''),
(3, '2', 'Pocetak projekta "Merenja Kiselosti"', '2018-03-08', 'Poceo je projekat ciji je zadatak da izmeri kiselost sirceta.', ''),
(4, '3', 'Zapocoje projekat "Merenje Vlažnosti"', '2018-03-10', 'Poceo je projekat "Merenje Vlažnosti".', ''),
(5, '3', 'Ucesnici projekta posetili farmu', '2018-03-12', 'Ucesnici projekta "Merenje Vlažnosti" posetili farmu "Zlatni Orao".', ''),
(6, '3', 'Oprema dopremljena do farme', '2018-03-14', 'Oprema za merenje vlažnosti dopremljena na farmu.', ''),
(7, '4', 'Pocetak projekta "Merenje svjetlosti"', '2018-03-04', 'Poceo je projekat ArdinoZR.', ''),
(8, '4', 'Isporucena oprema', '2018-03-07', 'Isporucena oprema za merenje svjetlosti.', ''),
(9, '4', 'Izvršeno merenje', '2018-03-09', 'Izvršeno merenje sunceve svjetlosti na krovu zrejaninske gimnazije.', ''),
(10, '4', 'Projekat uspješno zaršen', '2018-03-15', 'Završen projekat merenja kolicine svjetlosti na krovu gimnazije');

INSERT INTO `evidencija_projekata_fakulteta`.`ucesce_na_projektu` (`ID_ZAPOSLENOG`, `ID_PROJEKTA`, `ID_ULOGE`)
VALUES
('3', '1', '1'),
('4', '1', '2'),
('5', '1', '3'),
('6', '2', '1'),
('7', '2', '2'),
('5', '2', '3'),
('9', '3', '1'),
('6', '3', '2'),
('7', '3', '3'),
('10', '4', '1'),
('3', '4', '3');

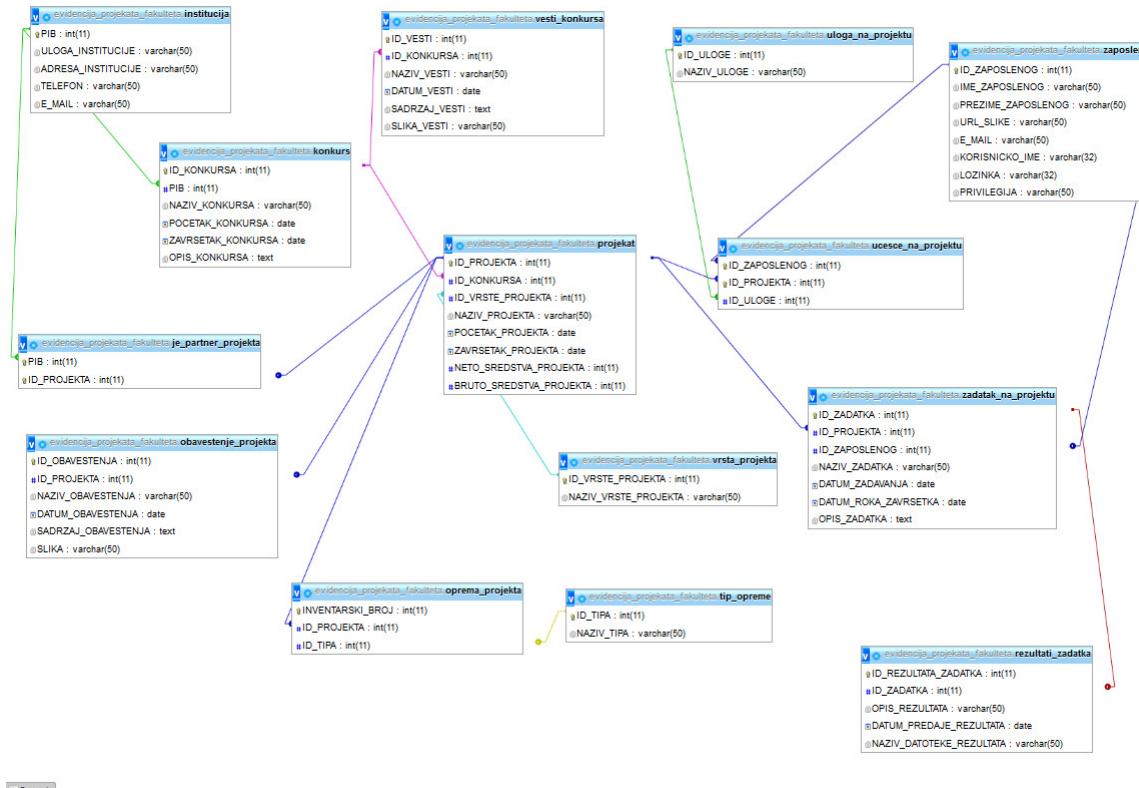
INSERT INTO `evidencija_projekata_fakulteta`.`zadatak_na_projektu` (`ID_ZADATKA`, `ID_PROJEKTA`, `ID_ZAPOSLENOG`, `NAZIV_ZADATKA`, `DATUM_ZADAVANJA`, `DATUM_ROKA_ZAVRSETKA`, `OPIS_ZADATKA`)
VALUES
(1, '1', '3', 'Pronaci kod za Arduino senzor', '2018-03-12', '2018-03-21', 'Pronaci kod za senzor merenje temperature.'),
(2, '1', '3', 'Izvršiti test merenje', '2018-03-07', '2018-03-23', 'Izvršiti test merenje sa senzorima'),
(3, '2', '6', 'Probno izmeriti kiselost', '2018-03-12', '2018-03-21', 'Izvršiti probu');

INSERT INTO `evidencija_projekata_fakulteta`.`rezultati_zadatka` (`ID_REZULTATA_ZADATKA`, `ID_ZADATKA`, `OPIS_REZULTATA`, `DATUM_PREDAJE_REZULTATA`, `NAZIV_DATOTEKE_REZULTATA`)
VALUES
(1, '1', 'Završeno preuzimanje koda za senzore', '2018-03-27', ''),
(2, '2', 'Završeno test merenje', '2018-03-27', '');

INSERT INTO `evidencija_projekata_fakulteta`.`je_partner_projekta` (`PIB`, `ID_PROJEKTA`)
VALUES
('1', '1'),
('2', '2'),
('3', '3');

```

Posle kreiranja baze podataka izvršavanjem ovog koda, phpMyAdmin, omogućava visualni prikaz tabela i njihovih odnosa preko ugrađenog dizajnera baze podataka. Ispod je prikazan grafički prikaz baze korišćene u ovom radu. (Slika 45)



(Slika 48. - Grafički prikaz baze podataka u phpMyAdmin dizajneru)

6.8.3. Klasa Konekcija i Parametri

Da bi se izvršilo povezivanje na bazu podataka, za to je potrebna klasa za postavljanje parametara i uspostavljanje konekcije. Klasa koja se koristi za uspostavljanje konekcije se zove "Konekcija.php", dok su parametari o portu, ime baze, šifre i ime korisnika se nalaze u eksternom XML dokumentu "Parametri_Konekcije.xml".

(Listing 3. –Kod klase “Konekcija.php”)

```
<?php
class Konekcija
{
    // atributi
    public $konekcijaMySQL; // $db_handle;
```

```

public $konekcijaDB; // $db_selected;
public $nazivbaze;

// metode

// ----- konstruktor
public function __construct(){
//
}

public function disconnect()
{
    mysql_close($this->konekcijaMYSQL);
}

public function connect()
{
    // preuzimanje parametara konekcije iz konfiguracionog xml fajla
    $xml=simplexml_load_file("Parametri_Konekcije.xml") or die("Error: Cannot create object");
    $host=$xml->host;
    $korisnik=$xml->korisnik;
    $sifra=$xml->sifra;
    $this->nazivbaze=$xml->nazivbaze;

    // ostvarivanje konekcije ka DBMS-u MYSQL
    $this->konekcijaMYSQL = mysql_connect($host, $korisnik, $sifra);

    // ostvarivanje konekcije ka bazi podataka
    $this->konekcijaDB = mysql_select_db($this->nazivbaze, $this->konekcijaMYSQL);

    if ($this->konekcijaDB)
    {
        // dodatak da moze da radi sa UTF8
        mysql_query('SET NAMES "utf8"', $this->konekcijaMYSQL);
    } // zatvaranje if
}

} // zatvaranje klase

?>

```

(Listing 4. –Kod “Parametri_Konekcije.xml”)

```

<?xml version="1.0" encoding="utf-8"?>
<konekcija>
<host>localhost:3306</host>
<korisnik>root</korisnik>
<sifra></sifra>
<nazivbaze>Evidencija_Projekata_Fakulteta</nazivbaze>
</konekcija>

```

6.8.4. Klase sloja podataka

Ovaj rad koristi dve vrste klase. Prva vrsta su klase koje iščitavaju podatke iz baze podataka preko SQL upita, i druga koje upisuju u bazu podataka preko SQL upita i naziv takvih klasa se završava sa "_DB". Spisak klasa koje su zadužene za rad sa bazom podataka dat je u nastavku. (Slika 49)

 cls_Institucija	3/15/2018 10:31 PM	PHP File
 cls_Konkurs	3/16/2018 3:12 PM	PHP File
 cls_Obavestenje_Projekta	3/16/2018 8:51 PM	PHP File
 cls_Obavestenje_Projekta_DB	3/21/2018 10:41 PM	PHP File
 cls_Oprema	3/15/2018 10:32 PM	PHP File
 cls_Oprema_Projekta	3/22/2018 9:42 PM	PHP File
 cls_Projekat	3/25/2018 11:16 PM	PHP File
 cls_Projekat_DB	3/24/2018 8:04 PM	PHP File
 cls_Tip_Projekta	3/16/2018 3:12 PM	PHP File
 cls_Ucesce_Na_Projektu	3/24/2018 7:15 PM	PHP File
 cls_Ucesce_Na_Projektu_DB	3/14/2018 10:27 PM	PHP File
 cls_Zaposleni	3/24/2018 7:37 PM	PHP File

(Slika 49. –Lista klase sloja podataka)

U nastavku će biti prikazani primer programskog koda klase sloja podataka koji koristi SQL upite . Konkretan primer se odnosi na priključivanja zaposlenog za izabrani projekat (rad sa tabelom Ucesce_Na_Projektu iz baze podataka).

(Listing 5. –Kod klase “cls_Ucesce_Na_Projektu.php”)

```
<?php  
  
class cls_Ucesce_Na_Projektu  
{  
  
    // atributi  
    private $Konekcija;  
    private $bazapodataka;  
  
    public $Broj_Redova_Rezultata;  
  
    public $Dostupni_Zaposleni;  
  
    public $Naziv_Zaposlenog;  
  
    // metode  
  
    // ----- konstruktor  
    public function __construct($NovaKonekcija, $NovaBaza)  
    {  
        // podrazumevamo da je otvorena konekcija, a zatvara se spolja  
        $this->Konekcija = $NovaKonekcija;  
        $this->bazapodataka = $NovaBaza;  
    }  
}
```

```

public function DostupniRukovodioci()
{
    $SQL = "select DISTINCT
    ID_ZAPOSLENOG,
    IME_ZAPOSLENOG,
    PREZIME_ZAPOSLENOG
    from
    `".$this->bazapodataka."`.`zaposleni`
    WHERE zaposleni.PRIVILEGIJA != 'administrator';

    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);

    $this->Dostupni_Zaposleni=$result;
    $this->Broj_Redova_Rezultata=$num_rows;
}

public function SlobodniZaposleni()
{
    $SQL = "select DISTINCT
    ID_ZAPOSLENOG,
    IME_ZAPOSLENOG,
    PREZIME_ZAPOSLENOG,
    E_MAIL
    from
    `".$this->bazapodataka."`.`zaposleni`
    WHERE zaposleni.PRIVILEGIJA != 'administrator' AND zaposleni.ID_ZAPOSLENOG NOT IN
        (
        SELECT
        zaposleni.ID_ZAPOSLENOG
        from
        `".$this->bazapodataka."`.`ucesce_na_projektu` LEFT JOIN `".$this->bazapodataka."`.`zaposleni` ON
        ucesce_na_projektu.ID_ZAPOSLENOG = zaposleni.ID_ZAPOSLENOG)";

    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);

    $this->Dostupni_Zaposleni=$result;
    $this->Broj_Redova_Rezultata=$num_rows;
}

public function ZaposleniSaUlogom($Uloga_Na_Projektu)
{
    $SQL = "select DISTINCT
    zaposleni.ID_ZAPOSLENOG,
    zaposleni.IME_ZAPOSLENOG,
    zaposleni.PREZIME_ZAPOSLENOG,
    zaposleni.E_MAIL,
    projekat.NAZIV_PROJEKTA
    from
    `".$this->bazapodataka."`.`ucesce_na_projektu` LEFT JOIN `".$this->bazapodataka."`.`zaposleni` ON
    ucesce_na_projektu.ID_ZAPOSLENOG = zaposleni.ID_ZAPOSLENOG
    JOIN `".$this->bazapodataka."`.`uloga_na_projektu` ON ucesce_na_projektu.ID_ULOGE = uloga_na_projektu.ID_ULOGE
    LEFT JOIN `".$this->bazapodataka."`.`projekat` ON ucesce_na_projektu.ID_PROJEKTA = projekat.ID_PROJEKTA
    WHERE zaposleni.PRIVILEGIJA != 'administrator' AND uloga_na_projektu.NAZIV_ULOGE = ".$Uloga_Na_Projektu."";

    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);
}

```

```

        $this->Dostupni_Zaposleni=$result;
        $this->Broj_Redova_Rezultata=$num_rows;

    }

?>
```

(Listing 6. – Kod klase “cls_Ucesce_Na_Projektu_DB.php”)

```

<?php

class cls_Ucesce_Na_Projektu_DB
{

    // atributi
    private $Konekcija;
    private $bazapodataka;

    public $ID_Projekta;

    public $greska;
    public $result;

    // metode

    // ----- konstruktor
    public function __construct($NovaKonekcija, $NovaBaza)
    {
        // podrazumevamo da je otvorena konekcija, a zatvara se spolja
        $this->Konekcija = $NovaKonekcija;
        $this->bazapodataka = $NovaBaza;
    }

    public function DodavanjeClana($id_Clana,$id_Projekta,$id_Uloge)
    {

        $SQL_Ucesce = "SELECT * FROM `".$this->bazapodataka."`.`zaposleni` WHERE ID_ZAPOSLENOG='".$id_Clana;
        $result_Ucesce = mysql_query($SQL_Ucesce);
        $num_rows_Ucesce = mysql_num_rows($result_Ucesce);
        if ($num_rows_Ucesce>0)
        {
            $SQL = "INSERT INTO `".$this->bazapodataka."`.`ucesce_na_projektu` (ID_ZAPOSLENOG,ID_PROJEKTA,ID_ULOGE)
                VALUES ($id_Clana,$id_Projekta,$id_Uloge)";
            $this->result=mysql_query( $SQL );
            $this->greska=mysql_error();
        }
    }

}
```

6.8.5. Poslovna Pravila

Poslovno pravilo u ovom radu je definisano kao limitacija pri dodeljivanju zaposlenog već postojaćeg projektu ako je ta osoba koju smo izabrali već alocirana na više projekata (čiji maksimalno dozvoljeni broj projekata smo definisali kao parametar u eksternom XML datoteci). Ovo poslovno pravilo se primenjuje u okviru dve stranice:

- Stranica za dodeljivanje rukovodioca projektu prilikom dodavanja novog projekta.
- Stranica za dodeljivanje zaposlenog projektu.

U nastavku je detaljno opisana implementacija stranice za dodeljivanje zaposlenog projektu - stranica "Pridruzivanje_Projektu_Index.php", koja je namenjena za pridruživanje člana (zaposlenog) projektu, funkcioniše tako što se pri učitavanju stranice izvrši PHP kod (Slika 53) koji priprema podatke za inicijalni prikaz (detaljni programski kod ovog dela dat je u sekciji sa prezentacionom logikom) - preuzima informacije o trenutnom projektu (Početak,završetak i uloge) preko klase "cls_Projekat.php", i preuzima informacije poput broja članova koji su trenutno vezani za projekat, njihova imena, kao i osobe koje su dostupne (osobe koje nisu deo projekta) iz klase "cls_Clanovi_Projekta.php".

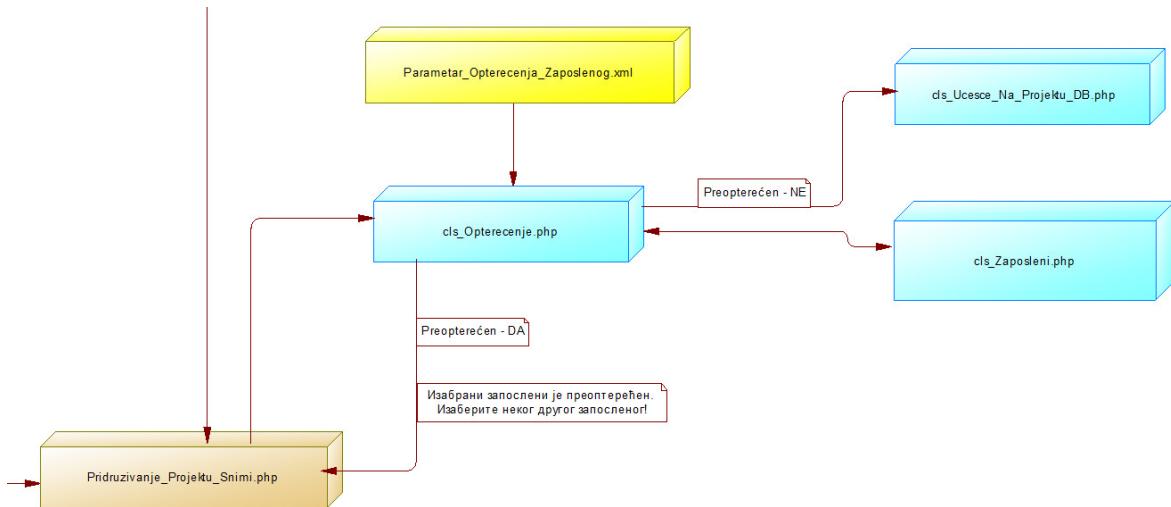
Posebno je značajna funkcija "DostupniZaposleni" za proveru dostupnosti zaposlenih za pridruživanje projektu, gde se izdvajaju zaposleni koji nisu već pridruženi trenutno izabranom projektu i da nisu administratori. (Slika 51)

```
public function DostupniZaposleni($id_Projekta,$ime,$prezime)
{
    $SQL = "select DISTINCT
    zaposleni.ID_ZAPOSLENOG,
    zaposleni.IME_ZAPOSLENOG,
    zaposleni.PREZIME_ZAPOSLENOG
    from
    (((".$this->bazapodataka."`.`projekat` INNER JOIN `".$this->bazapodataka."`.`vrsta_projekta` ON projekat.ID_VRSTE_PROJEKTA =
    vrsta_projekta.ID_VRSTE_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`.`ucesce_na_projektu` ON projekat.ID_PROJEKTA = ucesce_na_projektu.ID_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`.`uloga_na_projektu` ON ucesce_na_projektu.ID_ULOGE = uloga_na_projektu.ID_ULOGE
    RIGHT JOIN `".$this->bazapodataka."`.`zaposleni` ON ucesce_na_projektu.ID_ZAPOSLENOG = zaposleni.ID_ZAPOSLENOG
    WHERE zaposleni.ID_ZAPOSLENOG NOT IN
    (
        select
        ucesce_na_projektu.ID_ZAPOSLENOG
        from
        `".$this->bazapodataka."`.`ucesce_na_projektu` JOIN `".$this->bazapodataka."`.`zaposleni` ON ucesce_na_projektu.ID_ZAPOSLENOG =
        zaposleni.ID_ZAPOSLENOG
        WHERE ucesce_na_projektu.ID_PROJEKTA = '".$id_Projekta."' AND (zaposleni.IME_ZAPOSLENOG!='".$ime."' AND
        zaposleni.PREZIME_ZAPOSLENOG != '".$prezime."')
        AND (zaposleni.PRIVILEGIJA != 'administrator')";
    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);

    $this->Dostupni_Zaposleni=$result;
    $this->Broj_Dostupnih_Zaposlenih=$num_rows;
}
```

(Slika 50. -Prikaz funkcije DostupniZaposleni iz klase cls_Clanovi_Projekta.php koji proverava koji zaposleni mogu da budu pridruženi projektu)

Nakon prikaza stranice za unos (ranije prikazan ekran na slici 38), korisnik unosi podatke o izboru zaposlenog za projekat i dodeljuje ulogu na projektu. Prilikom pritiska na dugme "Potvrdi" prosleđuje se putem POST metode ID izabranog zaposlenog kao i ID dodeljene uloge radi preuzimanja i procesiranja u okviru "Pridruživanje_Projektu_Snimi.php" (ovaj programski kod je dat u sekciji sa prezentacionom logikom).



(Slika 51. - Grafički prikaz realizacije Poslovne Logike preko klase *cls_Opterecenje.php*)

Posle toga, ustanavljava se da li je taj zaposleni opterećen tako što se šalje ID_Zaposlenog (\$id_Clana) ka klasi poslovnog objekta "cls_Opterecenje.php" (programski kod ove klase je dat u narednoj sekciji sa poslovnim objektima). U ovoj klasi se prebroji na koliko se projekata ta osoba nalazi i onda se uporedi sa prethodno definisanim ograničenjem (maksimalno dozvoljenim brojem projekata) u XML dokumentu (Listing 7). Dakle, sam algoritam odlučivanja o opterećenosti na osnovu poslovnog pravila se realizuje u klasi poslovnog objekta *cls_Opterecenje.php*.

(Listing 7. - Sadržaj datoteke "Parametar_Opterecenja_Zaposlenog.xml")

```

<?xml version="1.0" encoding="utf-8"?>
<opterecenje>
<MaksimalnoProjekata>2</MaksimalnoProjekata>
</opterecenje>

```

Ako je osoba preopterećena, dolazi do odgovora koji se šalje preko sesije na stranicu ponovnog pridruživanja projektu i ispisuje se na korisničkoj formi. Ako nije opterećen, ID izabranog zaposlenog, ID uloge i ID projekta se šalju u klasu "cls_Ucesce_Na_Projektu_DB.php" (klasa sloja podataka zadužena za rad sa bazom podataka) gde se upisuju u odgovarajuću tabelu baze podataka.

```

public function DaLiJeZaposleniPreopterecen($id_Clana)
{
    $SQL = "select
        ID_PROJEKTA
        FROM `".$this->bazapodataka."`.`ucesce_na_projektu`
        WHERE ucesce_na_projektu.ID_ZAPOSLENOG = '".$id_Clana."'";
    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);

    $xml=simplexml_load_file("Parametar_Opterecenja_Zaposlenog.xml") or die("Error: Cannot create object");
    $Maksimalno_Projekata=$xml->MaksimalnoProjekata;

    $Broj_Ucestvovanja=$num_rows;

    if ($Broj_Ucestvovanja < $Maksimalno_Projekata)
    {
        $this->Preopterecenje = "false";
    }
    else
    {
        $this->Preopterecenje = "true";
    }
}

```

(Slika 52. - Kod funkcije "DaLiJeZaposleniPreopterecen" iz klase "cls_Opterecenje.php" predviđen za proveru opterećenosti izabranog zaposlenog)

6.8.6. Poslovni Objekti

Primer poslovnih objekata u ovom radu predstavljaju dve klase, "cls_Opterecenje" i "cls_Clanovi_Projekta". Klasa cls_Opterecenje je prethodno vec prikazana svojom jedinom funkcijom. U nastavku će biti prikazan kod klase cls_Clanovi_Projekta.

(Listing 8. – Kod iz klase “cls_Clanovi_Projekta”)

```

<?php

class cls_Clanovi_Projekta
{
    // atributi
    private $Konekcija;
    private $bazapodataka;

    public $ID_Projekta;

    public $Broj_Clanova_Na_Projektu;
    public $Clanovi;
}

```

```

public $Broj_Dostupnih_Zaposlenih;
public $Dostupni_Zaposleni;

public $Broj_Redova_Rezultata;
public $Lista_Ucesnika;

// metode

// ----- konstruktor
public function __construct($NovaKonekcija, $NovaBaza)
{
    // podrazumevamo da je otvorena konekcija, a zatvara se spolja
    $this->Konekcija = $NovaKonekcija;
    $this->bazapodataka = $NovaBaza;
}

public function BrojClanovaProjekta($id_Projekta)
{
    $SQL = "select
    zaposleni.ID_ZAPOSLENOG
    FROM
    (((".$this->bazapodataka."`projekat` LEFT JOIN `".$this->bazapodataka."`vrsta_projekta` ON
    projekat.ID_VRSTE_PROJEKTA =
    vrsta_projekta.ID_VRSTE_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`ucesce_na_projektu` ON projekat.ID_PROJEKTA =
    ucesce_na_projektu.ID_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`uloga_na_projektu` ON ucesce_na_projektu.ID_ULOGE =
    uloga_na_projektu.ID_ULOGE)
    INNER JOIN `".$this->bazapodataka."`zaposleni` ON ucesce_na_projektu.ID_ZAPOSLENOG =
    zaposleni.ID_ZAPOSLENOG
    WHERE projekat.ID_PROJEKTA = ".$id_Projekta.":";
}

$result = mysql_query($SQL);
$this->Broj_Clanova_Na_Projektu= mysql_num_rows($result);

}

public function ClanoviNaProjektu($id_Projekta)
{
    $SQL = "select
    zaposleni.IME_ZAPOSLENOG,
    zaposleni.PREZIME_ZAPOSLENOG,
    uloga_na_projektu.NAZIV_ULOGE
    FROM
    (((".$this->bazapodataka."`projekat` LEFT JOIN `".$this->bazapodataka."`vrsta_projekta` ON
    projekat.ID_VRSTE_PROJEKTA =
    vrsta_projekta.ID_VRSTE_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`ucesce_na_projektu` ON projekat.ID_PROJEKTA =
    ucesce_na_projektu.ID_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`uloga_na_projektu` ON ucesce_na_projektu.ID_ULOGE =
    uloga_na_projektu.ID_ULOGE)
    INNER JOIN `".$this->bazapodataka."`zaposleni` ON ucesce_na_projektu.ID_ZAPOSLENOG =
    zaposleni.ID_ZAPOSLENOG
    WHERE projekat.ID_PROJEKTA = ".$id_Projekta.":";

    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);
    $Nazivi_Clanova = "";

    if ($num_rows>0)
    {
}

```

```

$row=0;

for ($row = 0; $row < $num_rows; $row++)
{
    $Ime_Zaposlenog=mysql_result($result,$row,"IME_ZAPOSLENOG");
    $Prezime_Zaposlenog=mysql_result($result,$row,"PREZIME_ZAPOSLENOG");

    $Naziv_Zaposlenog=$Ime_Zaposlenog.''$.Prezime_Zaposlenog;

    if ($Nazivi_Clanova != "")
    {
        $Nazivi_Clanova = $Nazivi_Clanova.','.$Naziv_Zaposlenog;
    }
    else
    {
        $Nazivi_Clanova = $Naziv_Zaposlenog;
    }

}

$this->Clanovi=$Nazivi_Clanova;
$this->Broj_Redova_Rezultata=$num_rows;
$this->Lista_Ucesnika=$result;
}

public function DostupniZaposleni($id_Projekta,$ime,$prezime)
{
    $SQL = "select DISTINCT
    zaposleni.ID_ZAPOSLENOG,
    zaposleni.IME_ZAPOSLENOG,
    zaposleni.PREZIME_ZAPOSLENOG
    from
    (".$this->bazapodataka.`projekat` INNER JOIN ".$this->bazapodataka.`vrsta_projekta` ON
    projekat.ID_VRSTE_PROJEKTA =
    vrsta_projekta.ID_VRSTE_PROJEKTA)
    INNER JOIN ".$this->bazapodataka.`ucesce_na_projektu` ON projekat.ID_PROJEKTA =
    ucesce_na_projektu.ID_PROJEKTA
    INNER JOIN ".$this->bazapodataka.`uloga_na_projektu` ON ucesce_na_projektu.ID_ULOGE =
    uloga_na_projektu.ID_ULOGO
    RIGHT JOIN ".$this->bazapodataka.`zaposleni` ON ucesce_na_projektu.ID_ZAPOSLENOG =
    zaposleni.ID_ZAPOSLENOG
    WHERE zaposleni.ID_ZAPOSLENOG NOT IN
    (
    select
    ucesce_na_projektu.ID_ZAPOSLENOG
    from
    ".$this->bazapodataka.`ucesce_na_projektu` JOIN ".$this->bazapodataka.`zaposleni` ON
    ucesce_na_projektu.ID_ZAPOSLENOG =
    zaposleni.ID_ZAPOSLENOG
    WHERE ucesce_na_projektu.ID_PROJEKTA = ".$id_Projekta.") AND (zaposleni.IME_ZAPOSLENOG!=".$ime." AND
    zaposleni.PREZIME_ZAPOSLENOG!=".$prezime.")
    AND (zaposleni.PRIVILEGIJA != 'administrator');

    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);

    $this->Dostupni_Zaposleni=$result;
    $this->Broj_Dostupnih_Zaposlenih=$num_rows;

}

```

```

public function NeZaposleni()
{
    $SQL = "select DISTINCT
    zaposleni.ID_ZAPOSLENOG,
    zaposleni.IME_ZAPOSLENOG,
    zaposleni.PREZIME_ZAPOSLENOG
    from
    (((".$this->bazapodataka."`.`projekat` INNER JOIN `".$this->bazapodataka."`.`vrsta_projekta` ON
    projekat.ID_VRSTE_PROJEKTA =
    vrsta_projekta.ID_VRSTE_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`.`ucesce_na_projektu` ON projekat.ID_PROJEKTA =
    ucesce_na_projektu.ID_PROJEKTA)
    INNER JOIN `".$this->bazapodataka."`.`uloga_na_projektu` ON ucesce_na_projektu.ID_ULOGE =
    uloga_na_projektu.ID_ULOGE)
    RIGHT JOIN `".$this->bazapodataka."`.`zaposleni` ON ucesce_na_projektu.ID_ZAPOSLENOG =
    zaposleni.ID_ZAPOSLENOG
    WHERE zaposleni.ID_ZAPOSLENOG NOT IN
    (
    select
    ucesce_na_projektu.ID_ZAPOSLENOG
    from
    `".$this->bazapodataka."`.`ucesce_na_projektu` JOIN `".$this->bazapodataka."`.`zaposleni` ON
    ucesce_na_projektu.ID_ZAPOSLENOG =
    zaposleni.ID_ZAPOSLENOG
    WHERE ucesce_na_projektu.ID_PROJEKTA = ".$id_Projekta.") AND (zaposleni.IME_ZAPOSLENOG!=".$ime.") AND
    zaposleni.PREZIME_ZAPOSLENOG != ".$prezime.")
    AND (zaposleni.PRIVILEGIJA != 'administrator');

    $result = mysql_query($SQL);
    $num_rows = mysql_num_rows($result);

    $this->Dostupni_Zaposleni=$result;
    $this->Broj_Dostupnih_Zaposlenih=$num_rows;

}
?>

```

6.8.7. Prezentaciona Logika

Prezentaciona logika realizuje pripremu podataka za prikaz na formi, kao i preuzimanje podataka i reagovanje na korisničke akcije radi snimanja podataka u bazu podataka i slično.

Prva namena je pripremanje forme za prikaz podataka i opisana je primerom (Slika 38), koja se odnosi na učitavanje podataka radi punjenja combo box-ova i prikupljanje ostalih podataka potrebnih za prikaz i data je za primer stranice za dodeljivanje zaposlenih projektu (Pridruzivanje_Projektu_Index.php), kao što je dato na sledećoj slici.

```

<?php

require 'Delovi/Klase/Konekcija.php';
$novaKonekcija = new Konekcija();
$novaKonekcija->connect();

if ($novaKonekcija->konekcijaDB)
{
    $id_Projekta=$_SESSION["ID_Projekta_Ucesca"];
    $bazapodataka = $novaKonekcija->nazivbaze;

    require 'Delovi/Klase/cls_Projekat.php';
    $Projekat = new cls_Projekat($novaKonekcija, $bazapodataka);
    $Projekat->PocetakZavrsetak($id_Projekta);
    $Projekat->Uloge_Na_Projektu($id_Projekta);

    require 'Delovi/Klase/cls_Clanovi_Projekta.php';
    $Ucesce_Na_Projektu = new cls_Clanovi_Projekta($novaKonekcija, $bazapodataka);
    $Ucesce_Na_Projektu->BrojClanovaProjekta($id_Projekta);
    $Ucesce_Na_Projektu->ClanoviNaProjektu($id_Projekta);
    $Ucesce_Na_Projektu->DostupniZaposleni($id_Projekta,$ime,$ prezime);
}

$novaKonekcija->disconnect();

$Pocetak_Zavrsetak = $Projekat->Pocetak_Zavrsetak;
$Broj_Clanova_Na_Projektu = $Ucesce_Na_Projektu->Broj_Clanova_Na_Projektu;
$Clanovi = $Ucesce_Na_Projektu->Clanovi;

$Broj_Dostupnih_Zaposlenih = $Ucesce_Na_Projektu->Broj_Dostupnih_Zaposlenih;
$Lista_Dostupnih_Zaposlenih = $Ucesce_Na_Projektu->Dostupni_Zaposleni;

$Uloge_Projekta = $Projekat->Uloge_Projekta;
$Broj_Uloga_Projekta = $Projekat->Broj_Uloga_Projekta;

?>

```

(Slika 53. - PHP kod na početku stranice "Pridruživanje_Projektu_Index.php")

Drugi aspekt prezentacione logike se odnosi na preuzimanje podataka sa korisničkog interfejsa i reagovanje na korisničke akcije, čime se prosleđuje poziv odgovarajuće metode klase iz sloja podataka. Primer koda za ovaj aspekt dat je u okviru php koda "Pridruživanje_Projektu_Snimi.php" i on se poziva preko "Form Action" posle pritiska na dugme "Potvrdi" na stranici za unos, odnosno dodeljivanje zaposlenog projektu.

(Listing 9. - Isečak koda iz stranice "Pridruživanje_Projektu_Snimi.php")

```

<?php

session_start();
// citanje vrednosti iz sesije - da bismo uvek proverili da li je to prijavljeni korisnik
$ID_Projekta_Ucesca=$_SESSION["ID_Projekta_Ucesca"];

// preuzimanje vrednosti sa forme
$id_Clan=$_POST['Izabran_Clan'];
$id_Uloge=$_POST['Izabrana_Uloga'];

require 'Delovi/Klase/Konekcija.php';
$novaKonekcija = new Konekcija();
$novaKonekcija->connect();

if ($novaKonekcija->konekcijaDB)
{
    $bazapodataka = $novaKonekcija->nazivbaze;
    require 'Delovi/Klase/cls_Opterecenje.php';
    require 'Delovi/Klase/cls_Ucesce_Na_Projektu_DB.php';
    require 'Delovi/Klase/cls_Zaposleni.php';
}

```

```

$Opterecenje_Zaposlenog = new cls_Opterecenje($novaKonekcija, $bazapodataka);
$Opterecenje_Zaposlenog->DaLiJeZaposleniPreopterecen($id_Clana);
$Naziv_Zaposlenog = new cls_Zaposleni($novaKonekcija, $bazapodataka);
$Naziv_Zaposlenog->NazivZaposlenog($id_Clana);

$Preopterecen = $Opterecenje_Zaposlenog->Preopterecenje;
$Naziv_Izabranog_Zaposlenog = $Naziv_Zaposlenog->Naziv_Zaposlenog;

if ($Preopterecen == "true")
{
    $_SESSION['PROBLEM']="Изабрани запослени $Naziv_Izabranog_Zaposlenog је преоптерећен. Изаберите неког другог запосленог!";
}

else
{
    $_SESSION['PROBLEM']= "";
    $Ucesce_Na_Projektu_DB = new cls_Ucesce_Na_Projektu_DB($novaKonekcija, $bazapodataka);
    $Ucesce_Na_Projektu_DB->DodavanjeClana($id_Clana,$ID_Projekta_Ucesca,$id_Uloge);

    $Greska_Na_Uputu = $Ucesce_Na_Projektu_DB->greska;
    $Rezultat_Uputa = $Ucesce_Na_Projektu_DB->result;

        echo "Greska $Greska_Na_Uputu";
        echo "Укупно процесирano $Rezultat_Uputa zapisa";
    }

}

$novaKonekcija->disconnect();

header ('Location:Pridruzivanje_Projektu_Index.php');

?>

```

6.8.8.Korisnički Interfejs

Korisnički Interfejs ove aplikacije koristi HTML, i CSS za stilizaciju. Stranice ujedno sadrže, u kombinaciji sa HTML osnovom, PHP kod koji se izvršava radi prikazivanja nestatičkih podataka preuzetih iz baze podataka. Ispod je dat primer koda za stranicu na kojoj rukovodilac projekta dodeljuje člana postojećem projektu.

(Listing 10. - Kod stranice Pridruživanje_Projektu_Index.php)

```

<?php
    session_start();

    // citanje vrednosti iz sesije
    $ime=$_SESSION["ime"];
    $prezime=$_SESSION["prezime"];
    $korisnik=$_SESSION["korisnik"];
    $status=$_SESSION["status"];
    $uloga=$_SESSION["uloga"];
    $ID_Projekta_Ucesca=$_SESSION["ID_Projekta_Ucesca"];
    $Naziv_Projekta_Ucesca=$_SESSION["Naziv_Projekta_Ucesca"];
    $GRESKA_PRI_UNOSU=$_SESSION["PROBLEM"];

?>

```

```

<!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>Evidencija Projekata Fakulteta</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="css/style.css" media="screen">
</head>
<body>

    <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/Zaglavlje/Zaglavlje_Index_Prijavljen.php';?>

        <!-- SREDNJI DEO - POCETAK -->
        <tr style="padding:0px;">
            <td style="width:10%;"></td>

        <!-- SADRZAJ - POCETAK -->
        <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>
                    <!-- MENI -->
                    <?php
                        echo "<td style=\"width:20%;padding:0\" cellspacing=\\\"0\\\" cellpadding=\\\"0\\\" border=\\\"0\\\" valign=\\\"top\\\">";
                        if ($korisnik != "") {
                            include 'Delovi/Meni/Meni_Index_Prijavljen.php';
                        } else {
                            echo "<br>";
                        }
                        echo "</td>";
                    ?>
                    <td style="width:2%;"></td>

                    <!-- GLAVNI SADRZAJ -->
                    <?php
                        echo "<td style=\"padding:0\" cellspacing=\\\"0\\\" cellpadding=\\\"0\\\" border=\\\"0\\\" valign=\\\"top\\\">";
                        if ($korisnik != "") {
                            include 'Delovi/Sadrzaj/Pridruzivanje_Projektu_Sadrzaj.php';
                        } else {
                            echo "<br>";
                        }
                        echo "</td>";
                    ?>
                    <td style="width:2%;">
                        </td>
                    </tr>
                </table>
            </td>
        <!-- SADRZAJ - KRAJ -->
    </tr>
</table>

```

```

        <td style="width:10%;"></td>
    </tr>
    <!------- SREDNJI DEO - KRAJ ----->

    <tr style="padding:0px;">
        <td style="width:10%;"></td>
        <td align="center" valign="middle"></td>
        <td style="width:10%;"></td>
    </tr>

    <!------- FOOTER ----->
    <?php include 'Delovi/Footer/Footer_Index.php';?>

</table>

</body>
</html>

```

Listing naveden u nastavku sadrži kod za pripremanje podataka za prikaz na formi stranice kao i HTML kod za prikaz same stranice.

(Listing 11. - Kod stranice Pridruživanje_Projektu_Sadržaj.php)

```

<?php

require 'Delovi/Klase/Konekcija.php';
$novaKonekcija = new Konekcija();
$novaKonekcija->connect();

if ($novaKonekcija->konekcijaDB)
{
    $id_Projekta=$_SESSION["ID_Projekta_Ucesca"];
    $bazapodataka = $novaKonekcija->nazivbaze;

    require 'Delovi/Klase/cls_Projekat.php';

    $Projekat = new cls_Projekat($novaKonekcija, $bazapodataka);
    $Projekat->PocetakZavrsetak($id_Projekta);
    $Projekat->Uloge_Na_Projektu($id_Projekta);

    require 'Delovi/Klase/cls_Clanovi_Projekta.php';

    $Ucesce_Na_Projektu = new cls_Clanovi_Projekta($novaKonekcija, $bazapodataka);
    $Ucesce_Na_Projektu->BrojClanovaProjekta($id_Projekta);
    $Ucesce_Na_Projektu->ClanoviNaProjektu($id_Projekta);
    $Ucesce_Na_Projektu->DostupniZaposleni($id_Projekta,$ime,$prezime);
}

$novaKonekcija->disconnect();

$Pocetak_Zavrsetak = $Projekat->Pocetak_Zavrsetak;
$Broj_Clanova_Na_Projektu = $Ucesce_Na_Projektu->Broj_Clanova_Na_Projektu;
$Clanovi = $Ucesce_Na_Projektu->Clanovi;

$Broj_Dostupnih_Zaposlenih = $Ucesce_Na_Projektu->Broj_Dostupnih_Zaposlenih;
$Lista_Dostupnih_Zaposlenih = $Ucesce_Na_Projektu->Dostupni_Zaposleni;

$Uloge_Projekta = $Projekat->Uloge_Projekta;
$Broj_Uloga_Projekta = $Projekat->Broj_Uloga_Projekta;

?>

```

```

<meta charset="UTF-8">
<!--===== SADRZAJ STRANICE DESNO pocinje ovde ----->


<table style="width:100%;style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0"
bgcolor="#D8E7F4">
<tr>
<td style="width:5%;">
</td>

<td align="center">
<br/>
<b><font face="Trebuchet MS" color="darkblue" size="4px"></font></b>
<table style="width:100%;" bgcolor="#D8E7F4" padding:0" align="center" cellspacing="0" cellpadding="0" border="0">

<tr>
<td style="width:3%;">
</td>
<td align="center">
<font color="#D8E7F4" size="1px">.;</font>
</td>
<td style="width:3%;">
</td>
</tr>

<tr>
<td style="width:3%;">
</td>
<td align="center">
<b><font face="Trebuchet MS" color="darkblue" size="3px">Придруживање чланова пројектима</font></b><br>
</td>
<td style="width:3%;">
</td>
</tr>

<tr>
<b><font face="Trebuchet MS" color="red" size="4px"><?php echo "$GRESKA_PRI_UNOSU"; ?></font></b><br>
</tr>

<tr>
<td align="center">
<font color="#D8E7F4" size="1px">.;</font>
</td>

</tr>

<td style="width:3%;">
</td>

<td align="left">

<!-----FORMA ZA UNOS ----->
<table style="width:90%;" bgcolor="#D8E7F4" padding:0" align="left" cellspacing="0" cellpadding="0" border="0">
<form ACTION="Pridruzivanje_Projektu_Snimi.php" METHOD="POST" enctype="multipart/form-data">

<tr>

<tr>
<br>
</tr>

<tr>
<b><font face="Trebuchet MS" color="darkblue" size="3px">Име пројекта: <?php echo "$Naziv_Projekta_Ucesca"; ?></font></b><br>
</tr>

<tr>
<b><font face="Trebuchet MS" color="darkblue" size="3px">Почетак/Завршетак: <?php echo "$Pocetak_Zavrsetak"; ?></font></b>
</tr>

```

```

?></font></b></br>
</tr>

<tr>
<b><font face="Trebuchet MS" color="darkblue" size="3px">Број чланова: <?php echo "$Broj_Clanova_Na_Projektu";<br>
?></font></b></br>
</tr>

<tr>
<b><font face="Trebuchet MS" color="darkblue" size="3px">Тренутни чланови: <?php echo "$Clanovi";<br>
?></font></b></br>
</tr>

<tr>
<br>
</tr>

</tr>

<tr>
<br>
</tr>

<tr>
<td align="right" valign="top">
<b><font face="Trebuchet MS" color="darkblue" size="2px">Члан:&nbsp;&nbsp;</font><br/></b>
</td>
<td align="left" valign="top">
<select name="Izabran_Clan" required TABINDEX=8 width="250" style="width: 250px">
<option value="izaberite">Изаберите...</option>
<?php

if ($Broj_Dostupnih_Zaposlenih>0)
{
    $row=0;

    for ($row = 0; $row < $Broj_Dostupnih_Zaposlenih; $row++)
    {

        $Ime_Zaposlenog=mysql_result($Lista_Dostupnih_Zaposlenih,$row,"IME_ZAPOSLENOG");
        $Prezime_Zaposlenog=mysql_result($Lista_Dostupnih_Zaposlenih,$row,"PREZIME_ZAPOSLENOG");
        $ID_Zaposlenih=mysql_result($Lista_Dostupnih_Zaposlenih,$row,"ID_ZAPOSLENOG");

        $Naziv_Zaposlenog=$Ime_Zaposlenog.''.$Prezime_Zaposlenog;

        echo "<option value=\"$ID_Zaposlenih\">$Naziv_Zaposlenog</option>";
    }
?>
</select>
</td>
</tr>

<tr>
<td align="right" valign="top">
<font face="Trebuchet MS" color="#D8E7F4" size="2px">. </font><br/>
</td>
<td align="left" valign="top">
</td>
</tr>

<tr>
<td align="right" valign="top">

```

```

<b><font face="Trebuchet MS" color="darkblue" size="2px">Улога:&nbsp;&nbsp;</font><br/></b>
</td>

<td align="left" valign="top">
<select name="Izabrania_Uloga" required TABINDEX=8 width="250" style="width: 250px">
option value="izaberite">Изаберите...</option>
<?php

if ($Broj_Uloga_Projekta>0)
{
    $row=0;

    for ($row = 0; $row < $Broj_Uloga_Projekta; $row++)
    {

        $Uloga_Na_Projektu=mysql_result($Uloge_Projekta,$row,"NAZIV_ULOGE");

        $ID_Uloge=mysql_result($Uloge_Projekta,$row,"ID_ULOGE");

        echo "<option value=\"$ID_Uloge\">$Uloga_Na_Projektu</option>";

    }
}
?>
</select>
</td>
</tr>

<tr>
<td align="right" valign="top">
font face="Trebuchet MS" color="#D8E7F4" size="2px">. </font><br/>
</td>
<td align="left" valign="top">
</td>
</tr>

<tr>
<td align="left">
<br>
</td>
</tr>

<tr>
<td align="center" valign="top">
</td>
<td><input TYPE="submit" name="submit" value="Потврди" TABINDEX=3/>
</td>
</tr>

<tr>
<td align="right" valign="top">
font face="Trebuchet MS" color="#D8E7F4" size="2px">. </font><br/>
</td>
<td align="left" valign="top">
</td>
</tr>

<!-- prazan red ----->
<tr>
<td align="right" valign="top">
<font face="Trebuchet MS" color="#D8E7F4" size="2px">. </font><br/>
</td>
<td align="left" valign="top">
<font face="Trebuchet MS" color="#D8E7F4" size="2px">. </font><br/>
</td>
</tr>

```

```

</form>
</table>

</td>
<td style="width:3%;">
</td>

<tr>
<td style="width:3%;">
</td>
<td align="center">
<font color="#D8E7F4" size="1px">.</font>
</td>
<td style="width:3%;">
</td>
</tr>

<tr>
<td style="width:3%;">
</td>
<td align="center">
<font color="#D8E7F4" size="1px">
</font>
</td>
<td style="width:3%;">
</td>
</tr>

</table>
</td>

<td style="width:5%;">
</td>

</tr>
</table>

```

Kodovi za zaglavljne koje se pojavljuje na vrhu stranice su navedeno u nastavku. Zaglavljne se sastoje od dva dela, "Zaglavljne_Baner.php" i "Zaglavljne_Prijava" (ili "Zaglavljne_Odjava" ako je korisnik prijavljen). Slike banera se nalaze u folderu "root/Images/Baner/".

(Listing 12. - "Zaglavljne_Index.php" za ne prijavljenog korisnika)

```

<meta charset="UTF-8">
<tr>
    <td style="width:10%;"></td>
    <td align="center" valign="middle">
        <?php include 'Delovi/Zaglavljne/Zaglavljne_Baner.php';?>
    </td>
    <td style="width:10%;"></td>
</tr>

<tr>
    <td style="width:10%;"></td>
    <td>
        <?php include 'Delovi/Zaglavljne/Zaglavljne_Prijava.php';?>
    </td>
    <td style="width:10%;"></td>
</tr>

```

(Listing 13. - "Zaglavje_Prijava.php" za ne prijavljenog korisnika)

```
<meta charset="UTF-8">
<table style="width:100%;" bgcolor="#003366">
    <tr>
        <td style="width:1%; "></td>
        <td align="left" valign="middle"></td>
        <td style="width:50%; "></td>
        <td align="right" valign="middle">
            <font face="Trebuchet MS" color="darkblue" size="2px"><a href="Prijava_Korisnika.php">Пријава</a></font>
        </td>
        <td style="width:1%; "></td>
    </tr>
</table>
```

(Listing 14. - "Zaglavje_Baner.php" za ne prijavljenog korisnika)

```
<meta charset="UTF-8">
<table style="width:100%; height:100% padding:0" align="center" cellspacing="0" cellpadding="0" border="0" background="Images/Baner/Zaglavje_Baner_Slika.jpg" no-repeat>
    <tr>
        <td>
            <font color="white" size="1px">.</font>
        </td>
        <td></td>
        <td></td>
        <td></td>
        </tr>

        <tr>
        <td>
            <font color="white" size="1px">.</font>
        </td>
        <td></td>
        <td></td>
        </tr>

        <tr>
        <td style="width:5%; "></td>
        <td align="center">
            <div class="flt1 topblock"><a href="index.php" class="flt1_tp_txtpay">Евиденција Пројектета</a><br>
            <span class="flt1_tp_txtcss">Технички Факултет "Михајло Пупин"</span>
            </div>
        <td style="width:20%; ">
        </td>
        </tr>

        <tr>
        <td>
            <font color="white" size="1px">.</font>
        </td>
        <td></td>
        <td></td>
        </tr>

        <tr>
        <td>
            <font color="white" size="1px">.</font>
        </td>
        <td></td>
        <td></td>
        </tr>
    </table>
```

Kod za Footer je naveden u nastavku.

(Listing 15. - "Footer_Index.php")

```
<meta charset="UTF-8">
<tr style="padding:0px;">
    <td style="width:10%;"></td>
    <td align="center" valign="middle" bgcolor="#003366"><font face="Trebuchet MS" color="white" size="2px">Развијено за
        примену на Техничком Факултету "Михајло Пупин" 2018 * * *
        Бранислав Ђекић </font></td>
    <td style="width:10%;"></td>
</tr>
```

Meni funkcioniše tako što se preko sesije prenosi uloga koju ima dati prijavljeni korisnik i na usonovu te uloge , preko if i case grananja, prikazu je mu se opcije koje odgovaraju njihovoj ulozi.

(Listing 16. - Isečak iz HTML koda za meni opcije)

```
<!----- VESTI pocinje ovde ----->
<?php
if ($uloga == "Administrator Sajta" || $uloga == "Sekretar" || $uloga == "Član projekta")
{
echo <<<VESTI_BLOCK
<tr>
<td valign="top" align="center">
<table style="width:100%; bgcolor="#B7F3FE" padding:0" align="center" cellspacing="0" cellpadding="0" border="0">

<tr bgcolor="#FFFFFF">
<td style="width:2px; border: 1px solid white;">
</td>
<td style="border: 1px solid white;">
</td>
<td style="width:2px; border: 1px solid white;">
</td>
</tr>

<tr>
<td style="width:2px; border: 1px solid white; background-color: white;">
</td>
<td style="border: 1px solid white;">

<table style="width:100%;" bgcolor="#B7F3FE" padding:0" align="center" cellspacing="0" cellpadding="0" border="0">

<tr>
<td style="width:1px;">
</td>
<td style="align:center">
&ampnbsp &ampnbsp &ampnbsp&ampnbsp &ampnbsp &ampnbsp &ampnbsp &br>
</td>
</td>
<td style="width:1px;">
</td>
</tr>

<tr>
<td style="width:1px;">
</td>
<td style="align:center">
<font face="Trebuchet MS" color="black" size="4px">&ampnbsp&ampnbspВести:</font></b><br>
```

```

</td>
<td>
</td>
<td style="width:1px;">
</td>
</tr>
VESTI_BLOCK;
}

switch($uloga)
{
case "Administrator Sajta":
{
echo <<<VESTI_ADMINISTRATOR_BLOCK
<tr>
<td style="width:1px;">
</td>
<td style="align:center">
<a href=""><font face="Trebuchet MS" color="black" size="2px">&ampnbsp&ampnbspКонкурси</font></a>
</td>
<td>
<td style="width:1px;">
</td>
</tr>

<tr>
<td style="width:1px;">
</td>
<td style="align:center">
<a href=""><font face="Trebuchet MS" color="black" size="2px">&ampnbsp&ampnbspДодавање</font></a>
</td>
<td>
<td style="width:1px;">
</td>
</tr>
VESTI_ADMINISTRATOR_BLOCK;
break;
}

case "Sekretar":
{
echo <<<VESTI_SEKRETAR_BLOCK
<tr>
<td style="width:1px;">
</td>
<td style="align:center">
<a href="Vesti_Projekti_Lista_Index.php" ><font face="Trebuchet MS" color="black"
size="2px">&ampnbsp&ampnbspПројекти</font></a>
</td>
<td>
<td style="width:1px;">
</td>
</tr>

<tr>
<td style="width:1px;">
</td>
<td style="align:center">
<a href="Vesti_Projekti_Dodavanje_Index.php" ><font face="Trebuchet MS" color="black"
size="2px">&ampnbsp&ampnbspДодавање</font></a>
</td>
<td>
<td style="width:1px;">
</td>
</tr>
VESTI_SEKRETAR_BLOCK;
}

```

```

        break;
    }

    case "Član projekta":
    {
        echo <<<VESTI_CLAN_PROJEKTA_BLOCK
<tr>
<td style="width:1px;">
</td>
<td style="align:center">
<a href=""><font face="Trebuchet MS" color="black" size="2px">&ampnbsp&ampnbspПројекти</font></a>
</td>
<td>
</td>
<td style="width:1px;">
</td>
</tr>
VESTI_CLAN_PROJEKTA_BLOCK;
        break;
    }
    default:
    {
        echo "<br>";
    }
}

if ($uloga == "Administrator Sajta" || $uloga == "Sekretar" || $uloga == "Član projekta")
{
    echo <<<VESTI_BLOCK
<tr>
<td style="width:1px;">
</td>
<td style="align:center">
&ampnbsp &nbs; &nbs; &nbs; &nbs; &nbs; &nbs; &nbs; <br>
</td>
<td>
</td>
<td style="width:1px;">
</td>
</tr>

</table>
</td>

<td style="width:2px; border: 1px solid white; background-color: white;">
</td>
</tr>

<tr bgcolor="#FFFFFF">
<td style="width:2px; border: 1px solid white;">
</td>
<td style="border: 1px solid white;">
</td>
<td style="width:2px; border: 1px solid white;">
</td>
</tr>

</table>
</td>
</tr>
VESTI_BLOCK;
}
?>
<!----- VESTI zavrsava ovde ----->

```

7. Zaključak

Cilj ovog diplomskog rada bio je da se na primeru razvoja web aplikacije za podršku realizaciji projekata fakulteta prikaže primer razvoja softvera sa višeslojnom arhitekturom.

U okviru ovog rada su prikazane teoretske osnove koje se odnose na razvoj web aplikacija, višeslojnu arhitekturu, upravljanje projektima, kao i opis tehnologije PHP, MySQL i alata Notepad++ i XAMPP. Takođe, realizovan je primer softverske podrške upravljanju specifičnim projektima koji se realizuju na fakultetu, primenom navedenih tehnologija.

Ova aplikacija se može dalje usavršiti implementacijom ostalih funkcija koje nisu pokrivene ovim radom, kao i funkcija koje se tiče validacije podataka pri unosu, funkcije koje osiguravaju bezbednost pristupa aplikaciji, kao i implementacija java skripta za dinamički korisnički interfejs.

Savremeni trendovi industrijskog razvoja softvera orjentisani su na primenu frameworka i višeslojnog razvoja, čime se postiže veća brzina razvoja, unapređuje kvalitet razvijenog koda, kao i omogućava timski rad, posebno na većim projektima. Savremene tehnologije PHP programiranja uključuju framework-e kao što su Laravel, Symfony, ZEND itd.

8. Literatura

8.1. Fizička literatura

- [16] Wiley Publishing Inc, PHP & MySQL for Dummies 4th Edition, Janet Valade, 2010.
- [20] Project Management Institute Inc. , A guide to the project management body of knowledge - 5th edition , Jorge Diego Fuentes Sanchez, 2013.
- [27] Project Smart , An Introduction to PRINCE2 , Frank Turley, 2009.

8.2. Internet resursi

- [1] <https://www.lifewire.com/what-is-a-web-application-3486637>
- [2] <http://www.jguru.com/faq/view.jsp?EID=129328>
- [3] https://techterms.com/definition/web_application
- [4] <https://www.seguetech.com/website-vs-web-application-whats-the-difference/>
- [5] <http://courses.haigarmen.com/intd416/client-server-architecture/>
- [6] <http://predmet.singidunum.ac.rs/mod/folder/view.php?id=10679>
- [7] <https://www.cdnetworks.com/en/news/6-critical-web-application-performance-metrics-to-consider/4257>
- [8] Softversko Inzenjerstvo II Praktikum - dr. Zdravko Ivankovic i dr. Dejan Lacmanovic
<http://tfzr.rs/Predmet/softversko-inzenjerstvo-2/download>
- [9] <http://php.net/manual/en/intro-whatis.php>
- [10] http://download.tutoriali.org/Tutorials/PHP/php_i_mysql.pdf
- [11] <https://www.guru99.com/what-is-php-first-php-program.html>
- [12] <http://www.tizag.com/phpT/>
- [13] https://www.tutorialspoint.com/php/php_introduction.htm
- [14] <http://searchoracle.techtarget.com/definition/MySQL>

- [15] MySQL 5.7 Reference Manual
<https://downloads.mysql.com/docs/refman-5.7-en.a4.pdf>
- [17] <http://searchcio.techtarget.com/definition/project-management>
- [18] <https://www.ittoolkit.com/articles/project-definition>
- [19] <https://www.usability.gov/what-and-why/project-management.html>
- [21] <https://www.techopedia.com/definition/677/project-manager-pm>
- [22] <https://toughnickel.com/business/The-Iron-Triangle-Of-Management>
- [23] <http://cwstrategies.staffingindustry.com/sow-management-cost-speed-and-quality-a-delicate-balance/>
- [24] <https://www.axelos.com/best-practice-solutions/prince2/what-is-prince2>
- [25] <https://www.wrike.com/blog/project-management-basics-prince2-explained/>
- [26] <https://www.projectsmart.co.uk/what-is-prince2.php>
- [28] <https://www.g2crowd.com/categories/project-management>
- [29] <https://zapier.com/learn/project-management/personal-project-management-app/>
- [30] <https://www.mpug.com/education/what-is-microsoft-project/>
- [31] <http://whatis.techtarget.com/definition/Microsoft-Project-Microsoft-Office-Project>
- [32] <https://www.projectsmart.co.uk/introduction-to-microsoft-project.php>
- [33] <https://project-management.com/microsoft-project-online-software-review/>
- [34] <https://reviews.financesonline.com/p/wrike/>
- [35] <https://www.pcmag.com/article2/0,2817,2484389,00.asp>
- [36] <https://www.thesimpledollar.com/best-project-management-software/>
- [37] <https://www.pcmag.com/article2/0,2817,2390482,00.asp>
- [38] <https://www.pcmag.com/article2/0,2817,2373566,00.asp>
- [39] <https://lifehacker.com/five-best-personal-project-management-tools-1441334694>
- [40] <https://www.wpblogx.com/what-is-xampp/>
- [41] <http://a5theory.com/xampp-server-what-is-xampp-server/>

- [42] <https://notepad-plus-plus.org/>
- [43] <https://turbofuture.com/computers/A-Complete-Beginners-Guide-to-Notepad>
- [44] <http://tfzr.rs/baza-projekata>
- [45] Department for Business innovation and skills: *PRINCE 2 templates*,
<http://www.berr.gov.uk/aboutus/corporate/projectcentre/pm-templates/page12526.html>

8.3. Slike preuzete sa interneta korišćene pri izradi rada

- [46] <https://www.guru99.com/what-is-php-first-php-program.html>
- [47] <http://www.pmwares.com/blog/project-management-knowledge-areas/>
- [48] <https://boagworld.com/digital-strategy/think-outside-of-the-project-management-triangle/>
- [49] <https://mplaza.pm/prince2-diagrams/>
- [50] <https://tensix.com/2016/06/displaying-two-gantt-chart-baselines-in-microsoft-project/>
- [51] <http://www.malavida.com/en/soft/xampp/>

8.4. Resursi korišćeni pri izradi aplikacije

Početni kod, sa klasom "Konekcija.php": <http://tfzr.rs/Predmet/informacioni-sistemi-2/download>

Logo Fakulteta, Univerziteta i baner preuzeti sa: <http://tfzr.rs/>

PHP: <http://php.net/manual/en/index.php>

SQL: <https://dev.mysql.com/doc/refman/5.7/en/tutorial.html>

HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>

Ikonice korišćene na projektu:

http://www.clipartpanda.com/clipart_images/clipart-male-user-icon-17951048

<http://www.clker.com/clipart-female-user-icon-bright-blue-.html>